Creating Believable, Emergent Behaviour in Virtual Agents, Using a ‘Synthetic Psychology’ Approach

Micah Marlon Rosenkind

A thesis submitted in partial fulfilment of the requirements of the University of Brighton for the degree of Master of Philosophy

December 2015

The University of Brighton

I declare that the research contained in this thesis, unless otherwise formally indicated within the text, is the original work of the author. The thesis has not been previously submitted to this or any other university for a degree, and does not incorporate any material already submitted for a degree.

18/12/2015

ABSTRACT

Drawing on the fields of robotics, computer games and usability testing, this thesis contextualises the work of cyberneticist Valentino Braitenberg within the current field of game AI and demonstrates how there remain aspects of his propositions that cannot yet be adequately captured theoretically, or tested empirically or conclusively, in simulations of complex biologically inspired behaviours.

Justification for this stems from the view that believable, simulated characters are increasingly prevalent and exposed to mass-market scrutiny in visual entertainment media, such as visual effects in films and in particular the growing video games business, but will also have a growing, significant role in serious games and educational applications.

This thesis combines analytical, synthetic and empirical methods and discusses the contributions and limitations of each to the evaluation of believability in the context of human computer interaction scenarios. It first analyses Braitenberg's more intuitive architectures and demonstrates how these can be adapted, using a synthetic approach, to function as controllers for believable virtual agents, in a series of typical gaming scenarios. It then presents a set of empirical studies that were performed to develop a method for evaluating agents for their ability to elicit suspension of disbelief in the user. The resulting method, which is user centred and combines a qualitative content analysis approach with believability metrics for virtual agents sourced from the literature, was used to evaluate a series of increasingly complex agent models in a simple game scenario. The data was then compared to the results of the analysis of the underlying agent architecture to determine the correlation between AI design intent and the resulting user reactions and observations gathered from the empirical study.

This study concludes that a theoretical and analytical approach to Braitenberg's architectures can give us insight into the range and limits of the behaviour dynamics that are possible, but that an effective and useful evaluation and categorisation (labelling) of their impact on user perception, and their ability to elicit suspension of disbelief and trigger a desired response in the user, needs to also draw on empirical methods that record and analyse the observations and responses of human users.

Further work is presented that explores how the Braitenberg architectures featured in these experiments can be utilised in a number of gaming scenarios and how they could be extended with connectionist components that add associative and predictive capabilities.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

Table of Terminology as Used in this Thesis

Agent	An autonomous character with a computer simulation.
Animation	The act of making an inanimate object appear lifelike to observers.
Base Architecture	Braitenberg's 4 Wire combinations: Crossed Excitatory (CE), Parallel Excitatory (PE), Crossed Inhibitive (CI) and Parallel Inhibitive (PI).
Base Behaviour	An 'atomic' agent behaviour that is not a mixture of other behaviours.
Believable	The property of a subject to convince its observer of its own agency and integrity.
Classic Animation	Animation that is defined by a set of frames that are displayed in sequence to create the illusion of movement.
Cooperative Behaviour	A behaviour formed by combining base behaviours.
Complex Cooperative Behaviour	A behaviour formed by combining cooperative behaviours.
CTRNN	Continuous Time Recurrent Neural Network
Emergent Behaviour	Behaviour that is formed by the cooperation between agent behaviour and the interaction with other agents and the environment.

NPC	Non-Player Character. A character in a video-games that is not directly controlled by the player, but is in some indirect way interacted with (i.e. quest-giver, enemy).
Passive Motor Action	Forward (F,) Backward (B) and Neutral (N) motor drive caused independently from the architecture being investigated.
Procedural Animation	Animation that is defined by a set of algorithms (procedures) instead of individual frames.
Simple Cooperative behaviour	A behaviour formed by combining base behaviours with the passive motor actions "forward" or "backward".
STDP	Spike Timing Dependent Plasticity
Vehicle	A physical robot or simulated agent based on the architecture described in Valentino Braitenberg's thought experiment "Vehicles: Experiments in Synthetic Psychology"

Acknowledgements

I would like to thank my supervisors Prof. David Arnold and Dr. Graham Winstanley for their guidance and support during these years. I also want to thank all those who ever took the time to review my work and give friendly advice. Special thanks go to my loving parents Manuela and Harry Rosenkind, my brother Benjamin Rosenkind, my inspiring grandparents Mutti, Opa, Oma and OpaLala and my lovely lady Victoria. My thanks go to Dr. Andrew Blake, Dr. Heloisa Candello and Dr. Graham McAllister for their advice on user research, Dr. Paul Chorley, Dr. Dominik Daisenberger, Dr. Aidan Delaney, Dr. Wolfgang Hanrieder, Dr. Stelios Kapetanakis, Prof. Miltos Petridis and Dr. William Wilkinson for their advice on methodology, mathematics and thesis structure. Special thanks go to my friends James Barron, Scott Leicester Boyce, Owen Dine, Sam Freed, Oliver Hensel, Justin Hunting, Claudia, Thomas and Sofia Löwe, Freddie Mack, Yanni Papaioannou, Simon Radford, Samuel Rodman, Neil Shortland, Stefan Speer, Meghann Smith, Cassandra Treasure and Spyridoula Trivizaki for their patience, love and enduring moral support. Thank you for guiding, sustaining and inspiring me. Finally, I would like to thank those in the fields of believable agents, robotics, game AI, computational neuroscience, animation and the character arts for providing the basis for my contribution.

Thanks to Valentino Braitenberg for being the inspiration for this thesis.

1 INTRODUCTION

Creating believable, lifelike characters is at the core of storytelling itself. Inspired by the work in performance media such as theatre, audio-visual mediums such as painting, radio or television and written literature; we are seeing the emergence of virtual, interactive worlds as the next great storytelling medium. Creating believable characters for these rich, new, virtual worlds and finding new tools that can aid in the design and development of believable characters is the central motivation of this thesis.

With the video and computer games industry becoming one of the largest entertainment industries (ESA 2014) a novel demand for autonomous lifelike characters has emerged. With the increasing complexity of the worlds presented in video games it is becoming difficult to artistically and financially justify the established approaches to populating these worlds with pre-defined 'hand-made' non-player characters (NPC). This has already led to an increased use of procedural solutions for creating visual assets as well as animation, and a resurgent interest in procedural solutions for artificial intelligence systems that offer the promise of emergent and complex behaviour, without increasing the workload of behaviour designers.

Among these techniques is an approach to artificial intelligence that takes inspiration from biological mechanisms known as artificial life, or Alife for short. Previous projects have successfully used Alife architectures to create virtual characters, but their focus remained chiefly on recreating specific animal behaviour (Reynolds 1987; Tu and Terzopoulos 1994; Isla and Burke et al. 2001) or replacing traditional animation approaches by creating procedural motion generators (Naturalmotion 2005).

However, characters within games and computer animations were, at that time, a special effect. Just like adding realistic physics to cloth simulations, they serve the purpose of making characters look more realistic, dynamic and perhaps more alive. But while they affect how the character animates, moves and feels, they do not define its decisions, or create unique, emergent narratives.

Instead of focusing on emulating animal behaviour, the research presented in this thesis draws from a parallel field of research that approaches the notion of believability from a character-driven perspective (Maes and Darrell 1995; Loyall 1997; Mateas 1997; Perlin & Goldberg 1996; Gorniak and Blumberg 2005; Perlin 2009).

With autonomous agent behaviour finding its way back into the mainstream, the simple act of trying to identify what an agent is doing, or is thinking, has become a commonplace effort. As autonomous characters, both robotic/physical and virtual find their way to more consumers, the identification of behaviours and the communication of their intentions to the users, especially in the context of safety-critical scenarios, is becoming a critical feature. Contributions in the domain of virtual characters could also find application in the field of robotics. Expressive and believable robots are at the centre of research into using robots in medical and psychological treatment (Chang 2013, Lee 2012) as well as pedagogical applications (Leyzberg 2011).

The first part of this thesis tests how a small set of underlying impulses based on basic emotions can operate in tandem to create intuitively and consistently identifiable "personal" characteristics. The aim is to investigate the effects that using a biologically inspired ALife architecture to control behaviour has on the believability of synthetic characters. The second part is to elicit human factors that affect how users interacting with virtual agents identify their behaviour and what behaviour characteristics are responsible for breaking suspension of disbelief. This part serves to inform the ongoing debate on using metrics to evaluate virtual agents.

1.1 Personal Motivation

This work aims to advance games technology by exploring alternative development models that follow the current trend toward procedurally generated content. In addition, it is motivated by the realisation that industry standards are difficult to break away from while in the industry and seeing academic research as a tool that allows for blue-sky thinking, testing of novel, unproven approaches, in an effort to serve as a proof of concept for the developers of the future.

1.2 Justification for Research

After graphics and physics, artificial intelligence will be the next major domain of development and investment in games development. As the gaming audience grows, widens and matures and game supporting technology becomes more sophisticated, the expectations toward what game experiences can deliver are becoming higher.

There is an on-going debate in the gaming industry on where the development of game AI should go and there seems to be a rift between the development community and the gaming audience regarding the quality and sophistication of AI characters in games.

While players and reviewers are regularly found criticising the behaviour of agents in games for being too simple or simply “stupid”, developers have recently become more vocal with the counter-argument that players actually don’t want smart AI (Hernandez 2014). Peschke (2013) interviewed industry veterans who argue that sophisticated AI simply does not make it past initial internal user testing. So while players claim that they want more “realistic” AI, developers argue that this would simply not be “fun” to play against. Instead, so the developers argue, the game AI design should focus on making characters more believable and increase the ways in which they communicate what they are thinking to the player. Narayek also argues that game Ai should be focused “not on optimizing an NPC’s behaviour, but on the player’s fun and experience in general” (Narayek 2004).

Currently developers resort to a few well-known “stylistic-elements” when designing AI for games that players enjoy. An example of such a stylistic element is having enemies shout out their “thoughts” so that the player can anticipate their actions. This not only gives the player an information advantage that allows them to react sooner and plan ahead, but was also found to make the players believe that the agents were more coordinated and smarter (Peschke 2013).

While there are many well-known stylistic elements in game AI, there is as yet no underlying framework that connects player experience with AI design choices in the context of games. This research aims to contribute to the field of game AI by informing this debate and development practices.

The second component of the thesis uses a development approach inspired by cyberneticist Valentino Braitenberg’s “Vehicles” thought experiment (Braitenberg 1984). Although the field of Artificial Life was only formally defined in 1990 by Langton (1990), it could be argued that Braitenberg was one of the earliest proponents of this constructivist, bottom-up and biologically inspired approach to AI. Braitenberg’s ‘Synthetic Psychology’ approach to AI is defined by his theory of ‘uphill analysis and downhill synthesis’, which is also an example of the constructionist theory of learning that was popularised by Papert (1980).

This choice is motivated by an increasingly apparent asset/cost problem found in game development, which may be addressed by finding new approaches to designing and implementing behaviour controllers for virtual agents.

Current techniques in game AI rely heavily on scripting and most AI systems use a combination of finite state machines (FSM) for decision making and standard AI search algorithms such as A* for navigation. Bottom up approaches and Alife architectures are currently mainly used as steering behaviour components and for swarm animations (Reynolds 1987; 1999). Approaches that may offer adaptive agents and more emergent behaviour are largely ignored. While there are certainly practical production issues to be addressed, the lack of proof of concepts demonstrating the merits of these architectures applied to the context of games is hindering them being adopted or even considered.

As the interviews conducted by Peschke (2013) show, it can prove extremely challenging for developers to introduce novel AI concepts from academia into the game development process. However, there are several examples of game developers looking toward academic research to inform their work and breaking open the established and often entrenched opinions on their feasibility as Game AI components.

A prime example of a developer who managed to do so is Damian Isla, whose work on the AI in "Halo 2" (Bungie Studios 2004) was highly publicized (Isla, 2005; Rabin, 2006) and popularised the use of hierarchical finite state machines (HFSM), or behaviour trees in games development. Using behaviour trees instead of the then-standard non-hierarchical finite state machines (FSM), allowed Isla to design and maintain a set of complex multi-purpose enemy behaviours.

The HALO2 AI architecture was designed around the concepts of scalability, modularity and transparency. Where scalability and modularity are common requirements in the design of complex software systems, Isla defines a lack of "transparency" as a problem often caused by complex AI behaviour:

> "It is not enough that the AI be able to do it a lot of things, it is equally important that they do all those things right, at the right times, and in a way that does not break the illusion of life, or threaten the player's understanding of the AI's intentions or motivations" (Isla 2005)

Isla defines "transparency" as a design goal of successful game AI development, a metric for believable, complex agent behaviour. He stresses that "it must be possible for the untrained observer to make reasonable guesses as to the AI's internal state as well as explain and predict the AI's actions" (Isla, 2005). The use of HFSM allowed Isla to compartmentalise the complexity of the AI system, making it easier for him to communicate efficiently when working with the non-programmers on the animation and level design teams. Working closely with the AI developer, animators ensured that the enemy agents in the game telegraphed their intentions to the player through additional non-combat animations (shout-outs to squad members, body language). Their improved understanding of the AI systems also allowed the level designers to create architecture that specifically exploited and showcased the full range of AI behaviour states. According to Isla, this merging of AI development, animation and level design played the greatest part in achieving AI "transparency". The success of the Halo franchise and the developer's efforts to make their AI development approach public have since led to HFSM becoming a new standard game AI development approach across genres, in favour of the then-established combination of state transition tables and FSM. The Halo series acted as a proof of concept, setting a new standard for dynamic enemy combat behaviour and promoting the importance of AI in action games to the public. This lead to a trend of other developers following suit and introducing novel AI approaches into their games. Jeff Orkin used a goal oriented action planner (GOAP) based on STRIPS to control action selection and a blackboard model for squad communication in the first person shooter "F.E.A.R." (Monolith Productions 2005). This approach was in turn built on by Alex Champandard who developed a multi-layered hierarchical task network (HTN) planner to control the strategies, squad tactics and individual combat behaviour of the enemies in "Killzone 2" (Guerrilla 2009). HFSM, GOAP and HTN have all since been adapted to other genres and become established standard practice game AI development models.

Other approaches to game AI garnered equal praise at the time, but found it harder to become established in standard practice. In contrast to the above examples, where new AI systems were introduced to simplify the design and team-management process of AI development, the following examples introduced their novel AI systems as a new gameplay feature. This relates to the role that the AI component plays in the context of the game design. For example, Demis Hassabis developed a complex reinforcement learning algorithm to drive the behaviour of the virtual pet creatures in "Black & White" (Lionhead 2001). Though "Black & White" was not a critical or commercial failure, reinforcement algorithms have seldom been used in games since. One reason for this is that "Black & White" failed to address the issue of "transparency" that Isla later identified. The creature AI in "Black & White" demonstrated how unpredictable the character behaviour generated by the reinforcement learning algorithm can appear and how this can lead to player frustration. Legendary game designer Sid Meyer

("Civilization" Meyer 1991) is often quoted for saying that good gameplay is "a series of interesting choices" (Rollings and Morris 2000). This mirrors Isla's definition of "transparency". Games need to make an effort to clarify the cause and effect relationship between player choices and their consequences in the game. The reinforcement learning algorithm in "Black & White" made it too difficult for players to trace back an undesired AI behaviour to the choices that they made when "training" their pet creature, leaving it unclear to them how to "correct" the behaviour. This lack in clarity between cause (training) and effect (learnt behaviour) results in the interactions with the creature feeling inconsequential and ultimately arbitrary.

Even though the AI system developed for this thesis aims to introduce novel features to game AI the relationship between design intent and player perception is a focus. This is done in order to ensure that it falls into the former category of AI systems, which augmented and simplified the AI development process and allowed developers to enhance the transparency and believability of virtual character behaviour.

When it comes to advanced techniques such as adaptive and evolving agents, a major component of agent behaviour is the fitness criteria, or behaviour goals that guide automated development or tuning approaches. Interactive video games provide a unique challenge here, since guidelines forming the fitness criteria must take into account the human factors that determine whether an agent is "successful" or not.

This differentiates this research from other typical research in robotics, Alife and even game AI, for example the work of Karl Sims (1994) and Jordan Pollack's research group at Brandeis (Sevan 2007; Harrington 2014) on applying evolutionary algorithms to tune behaviour and create adaptive opponents using co-evolution in games. However, the performance metrics used in this research did not consider the perspective of the user and their interpretation of behaviour. The focus was typically on improving objective performance or enhancing the agents' ability to perform specific actions. This type of research also includes projects that measure how closely a virtual or robotic model resembles a biological counterpart. In the game AI field, it is also possible to find projects that aim to optimise specific agent behaviours and find efficient solutions for performing a specific task. For example, achieving human-like behaviour remains one of the most popular research topics in games AI (Magerko 2004; Thurau 2004; Livingstone 2006; Wang 2009; Laird 2012).

However, the research in this thesis approaches the notion of believability from a character-driven perspective (Maes and Darrell 1995; Loyall 1997; Mateas 1997;

Perlin & Goldberg 1996; Gorniak and Blumberg 2005; Perlin 2009) and aims to contribute to this field by establishing criteria that incorporate subjective notions (e.g. of fun and suspension of disbelief).

Underlying this work is the belief that investigating the relationship between design choices made during the development of a virtual agent, and the interpretation of users, to refine existing metrics and find potential pitfalls, will directly aid and inform future research and development.

1.3 Aims and Objectives

The aim of this research is to propose an alternative game AI design approach that introduces a new way of thinking about non-player character behaviour. This is grounded in the constructionist philosophy and the idea that new approaches to design can lead to novel solutions to existing problems.

The objective is to present a series of proof of concept prototypes that demonstrate the use of the "Synthetic Psychology" design approach in the context of video game AI development. A comparison between design intent and user perception is made by combining formal system design with an empirical user study. This will provide feedback on the viability and veracity of this approach as a development technique. Undertaking these two steps to developing a game character and comparing the perspectives on the system gained during each of them is intended to mirror and further inform the state of the art agent development approach found in industry.

Furthermore, the potential affordances and limitations of this design approach will be explored by experimenting with simple extensions to the architecture originally proposed in Italian-Austrian cyberneticist Valentino Braitenberg's thought experiment "Vehicles: Experiments in Synthetic Psychology" (Braitenberg 1984).

This thought experiment presented 14 agent architectures that were each given anthropomorphising names drawn from the field of psychology that suggested certain behavioural characteristics. It demonstrated how simple components interact to produce emergent behaviours that an observer may interpret as complex, and seemingly intelligent. This work is summarised and discussed in section 2.2.2.

1.4 CONTRIBUTIONS

The experiment presented in this work evaluates "Synthetic Psychology" as an approach to developing agents for games. The prototypes and the feedback gained from the user study stand as a proof of concept for using this approach in the context of games development. The extensions to Braitenberg's "Vehicles" architecture presented in Chapter 6 "Extended Architectures" represents a look forward into what might be possible when incorporating adaptive systems into agents based on bottom-up biologically inspired control systems.

This work also aims to contribute to the study of key human factors that affect the perception of believability of AI characters in games. The review of a survey instrument using believability criteria in the context of user-centred design of gaming agents contributes to the ongoing discussion on using metrics to measure subjective, aesthetic qualities and the performance of agents designed for entertainment.

1.5 RESEARCH QUESTIONS

Can "Synthetic Psychology" as a bottom-up, constructionist AI design philosophy inform, improve and augment established top-down game AI development approaches? What new perspectives on agent architecture design can be gained from adopting this approach?

How can we create behaviourally complex, believable and tractable software agents in virtual worlds designed for entertainment using a "Synthetic Psychology" design approach?

What are the human factors that need to be taken into account when making design decisions during the development of believable agents for games?

1.6 RESEARCH METHOD

The bottom-up development of agent architectures was guided by Braitenberg's iterative "Synthetic Psychology" approach. This resulted in a series of simple agents that exhibit complex emergent behaviour that were tested in the context of a game scenario, to elicit behavioural factors that contribute or interfere with the sense of "believability" of the virtual agent.

The user tests were conducted using a lab-based user experience study. Prototypes were tested in a gaming context by a sample user group consisting of non-gamers and typical end users from diverse demographics backgrounds.

Qualitative content analysis was used to evaluate the data gathered in the lab study to elicit emergent themes and categories in the user responses that help to identify the key factors that affect the interaction and relationship between users/players and the behaviour of the virtual agent.

1.7 Scope of the Research

The thesis is situated in the field of game AI and user experience in games research. The scope is restricted to analysing agents developed using Braitenberg's Synthetic Psychology approach to test its viability as a development method for gaming agents and elicit factors that affect how "believable" an agent's behaviour appears to users interacting with it in the context of an adversarial game.

This thesis aims to find ways in which bottom-up development techniques may be employed to contribute to existing Game AI systems, by encouraging the use of architectures that allow for more emergent behaviours and interactions.

Anthropomorphising agent behaviour is a central notion of "Synthetic Psychology". However, the architectures presented in this thesis do not seek to emulate or model animal behaviour, though analogies will be drawn.

1.8 Outline of the Thesis

The introduction has discussed the motivation, questions and objectives that guide this research. It briefly summarised the methods used to address these questions and the limits and scope of its contribution.

Chapter 2 is a literature review of related work in the believable agents and robotics research field and the video games industry. It reviews relevant research in designing believable characters, evaluation metrics and user testing.

Chapter 3 discusses the methods used and simulation tools and models that were created to explore Braitenberg's architectures and development approach and to perform the user experiments in Chapter 5.

Chapter 4 documents the adaption of Braitenberg "Vehicle" architectures for game AI and details the models used for user testing in Chapter 5.

Chapter 5 presents the results of the empirical analysis of the agent models in the context of an adversarial game.

Chapter 6 explores the limits of Braitenberg's approach using different simulation environments and connectionist components. It also discusses why bottom-up architectures like Braitenberg's are set to become more relevant as connectionist architectures become more mainstream and applicable in a consumer context.

Chapter 7 concludes, reflecting on the findings, summarising the contributions of the thesis, while suggesting further work.

2 LITERATURE REVIEW

The following literature review provides the foundation for the research in this thesis. It highlights significant work done in the study of the notion of believability, suspension of disbelief in media and what in particular, makes virtual characters believable. It then discusses examples of believable agent development in both research and industry, highlighting agent models and development approaches that significantly advanced the field in their time. The final section discusses projects from robotics that, while not focusing on virtual agents, adopted methods from that field to enhance social interaction and create an emotional bond between robots and humans.

2.1 CRITERIA FOR BELIEVABLE AGENTS

To test whether an agent will be perceived as "believable", it is important to explore the main approaches and identify the key factors that make an agent seem believable to an observer. This chapter reviews existing definitions of and criteria for believability in various domains and describes how these may be adapted to be used to evaluate existing agent architectures and contribute to the development of a new design approach.

This chapter then reviews examples of believable agents in related media and discusses the work of research groups from the virtual agents and robotics field which followed similar objectives to this thesis.

2.1.1 THE DISTINCTION BETWEEN BELIEVABLE AND LIFELIKE

The first issue at hand is to define what "believable" means and to differentiate this property from other terms, such as "lifelike" and "realistic".

An early decision was that this project would not aim at creating an agent that is based on any *real*, living being. The term *realistic* is therefore not applicable to the agent architecture developed in this thesis, but can be used to describe some of the properties of the simulation environment that was used, which features *realistic* physics. The model of the brain that is used to control the agent's behaviour does, however, not attempt to emulate the properties of *real* brains (even though it is certainly inspired by them) and should therefore not be measured against them.

The distinction between *lifelike* and *believable* is more difficult. Loyall provides a good differentiation between the two properties in his thesis (Loyall 1997):

> "*Lifelike* and *believable* are both terms borrowed from the character-based arts. I use the term *believable* throughout this thesis because its meaning is less ambiguous. Lifelike characters in the arts and lifelike computer characters in computer science are sometimes used to connote believability, but at other times these terms are used to denote work that focuses on realism, cognitive plausibility, or other concepts not necessarily central to believability."
>
> (Loyall 1997 p. 10, emphasis in the original)

Loyall states that the criteria for *lifelike* and *believable* characters can be quite different. Taking an example from acting can make this distinction clear: An actor playing his character badly will certainly be lifelike, but might not be believable to the audience.

Because of its less ambiguous nature and stronger differentiation from *realistic*, the term *believable* will be used as the central term in this thesis. The following sections describe how the criteria for believability and the corresponding art of suspending disbelief were affected, as the representation of characters moved through increasingly complex media.

2.1.2 Believability vs Human-Like Behaviour

Another distinction that needs to be made is between *believable* and *human-like* behaviour. Setting human-like behaviour as a goal has recently gained a lot of traction in the research and game development community, mostly spurred by the success of systems that employ player analytics in their models.

The 2k BotPrize Competition

There has been a lot of press and even mainstream media attention surrounding the success of two teams passing what the media tends to call the Turing test of gaming, the "Botprize" in 2012 (Oppenheimer 2012; Rossignol 2012; Rundle 2012). This competition, which was first held in Perth, Western Australia in 2008, sees teams of AI developers competing in designing adversarial agents for the first person shooter (FPS) game of "Unreal Tournament 2006". As with similar Turing inspired AI tests, the goal is to fool human players into thinking they are playing against another human, when in fact they are playing against a non-player character (NPC). Two agents passed the 50% threshold rating for "humanness" and shared the prize. The UT^2 agent developed by a University of Texas at Austin team lead by Risto Miikkulainen and the

MirrorBot created by Romanian computer scientist Mihai Polceanu, both used a similar approach to emulate human behaviour that involved imitating the human players in the game.

According to Miikulainen UT^2 uses a neuro-evolution process to tune the neural network controlling the agent's movement, aiming and decision making, and the key to achieving human-like behaviour lies in the imperfections and occasional incoherence of human behaviour. Using this method, the aiming of the agents will become less accurate in chaotic situations and while moving fast. The agent is also capable of holding "grudges" against particular opponents and chases them, even if that causes it harm. Miikulainen points out that it was the latter property especially that convinced the judges.

Player Modelling in Racing Game AI

Recent successful examples in the games industry include player-model driven AI in the racing games "FORZA 5" (Turn 10 Studios, Microsoft Studios 2013) and Firemint's "Real Racing" (Firemint 2013). The winners of the 2012 "BotPrize" also used a similar player-imitation approach. Both Firemint's approach and that of the BotPrize winners record player data that is then used to set the constraints of a neural network. Through neuroevolution the neural network controlling the agents is optimised within the restraints to prevent it from becoming too accurate.

The "Drivatar" feature has been in development at Microsoft Research in Cambridge since 2002 and lets the players "train" an AI system to drive like them (Microsoft Research 2013). The statistical data captures the player's racing strategy and ability to follow the ideal racing line. This data is then used inform the driving style of the "Drivatar", an AI agent driver that can compete in races for the player. The Drivatar system also affects the AI opponent drivers. The performance data of players around the world is shared across Microsoft's cloud network. The game downloads driver profiles from other players with similar skill to control the opponents in the game. This creates a form of dynamic difficulty based on the actual performance and habits of the player community. A very similar system is used in Firemint's "Real Racing". This system uses a similar approach to the winners of the 2012 BotPrize in that it evolves a neural network controller using genetic algorithms informed and constrained by player data derived from human players. It combined Microsoft's approach with the UT^2 and MirrorBot technical approach in that it sources its player data through the cloud, collecting data from players around the world and modelling agents based on their behaviour.

Significance to this Research

According to Livingstone (2006), measuring the success of AI characters with regard to their ability to convince players interacting with them that they are another human player is a useful and popular type of metric for judging AI behaviour in games. The key difference to the believability metrics discussed in the rest of this chapter, is that the agents designed to win the BotPrize or play like human racers are specifically engineered to imitate human behaviour. This metric is most suited for gaming contexts that usually involve player vs. player (pvp) scenarios and thus the illusion and suspension of disbelief is grounded in that notion.

Believability metrics for virtual characters do not have this focus. They are not exclusively concerned with imitating the behaviour of a player *controlling* a virtual character. Rather they are concerned with what generates the *perception* of character – in a similar vein to how principles of acting (Stanislavsky and Popper 1961; Stanislavski 1968) and animation (Thomas and Johnston 1981; Lasseter 1987) inform actors and animators on how to act believably or make their drawings come to life. Livingstone (2006) differentiates between the two notions of believability using a question: "should AI be indistinguishable from a human player or should it try to be a better role-player". The definition for believability used in this thesis, focuses on the latter.

A central idea that is shared by research in believable characters and projects that pursue human-like behaviour is the notion of suspension of disbelief. This idea will be discussed in the following section.

2.1.3 Suspension of Disbelief

Suspension of disbelief is the willingness and desire of the viewers to believe in a fictional world. Earning and keeping this "poetic faith" as Samuel Taylor Coleridge calls it, lies at the heart of the craftsmanship of the character arts.

> "…to transfer from our inward nature a human interest and a semblance of truth sufficient to procure for these shadows of imagination that willing suspension of disbelief for the moment, which constitutes poetic faith" (Coleridge 1817)

The mastery of this craft, in a given medium, requires acute attention to detail. However, it is not the absolute amount of detail and adherence to realism that are the key to coercing the audience, but upholding a contract between audience and author. The presentation of an internally consistent, fictional world and the audience giving its *belief* in return, are the reciprocal terms of this contract. This means that if a story is presented consistently and is interesting to the audience, it does not necessarily have to be realistic.

What Coleridge calls “a semblance of truth” can be regarded as the internal consistency of the fictional story told and “our inward nature” as the source of this knowledge. While this notion can be applied to character arts that are under the control of a human artist, author, actor or animator, the following section defines the properties of believable characters in general. This is necessary for the purpose of creating characters that are autonomous and therefore detached from direct human influence after they have been initialised.

The fact remains that even these autonomous characters must adhere to the contract between what they represent and their observer, in order to convince them to willingly suspend their disbelief.

2.1.4 The Fundamental Principles of Classical Animation

Rules that were initially developed for classical hand-drawn animation also apply to computer-generated characters. The leading innovators of this craft were the animators at Walt Disney studios, who compiled their experiences in a collection of techniques and principles in the seminal work “The Illusion of Life” (Thomas and Johnston 1981). Chuck Jones from Warner Brothers presents a similar approach in “Chuck Amuck” (Jones 1989). These works are in turn strongly inspired by previous work in the character arts of stage and film acting (Stanislavsky and Popper 1961; Stanislavski 1968) and the art of dramatic writing (Egri 2004).

John Lasseter of Pixar later expanded on these principles and applied them to the new work environment and tools for digital computer animation. According to Lasseter (1987), the fundamental principles of traditional animation that he adapted from Disney (Thomas and Johnston 1981) are:

1. **Squash and Stretch**: Lasseter states that perhaps the most important principle is squash and stretch. This “defines the rigidity and mass of an object by

distorting its shape during an action" (Lasseter 1987 p. 36). He describes how this technique is often required to alleviate the unnatural-looking strobe effects that occur when an object moves very fast.

2. **Timing and Motion**: Timing is the speed of action. It not only reflects the weight and size of the animated object, but can also be used to carry emotional meaning.
3. The **anticipation** of an action includes a character preparing for an action e.g. by adjusting poise or flexing its muscles. Its main use is to direct the viewers' attention to the action.
4. The **staging** of the action itself. This is the animator telling the audience what object to look at. The object of interest needs to contrast the rest of the scene. According to Lasseter, if a scene is busy, a still object will stand out, while in a still scene, movement will attract attention. He also cites Disney (Thomas and Johnston 1981) who found that performing actions in silhouette also enhances their clarity.
5. The *reaction* to the action. Lasseter describes how **Follow through and overlapping** shows the relationship between actions. Some actions may *lead* others and appendages or loose objects will drag behind the leading action. In a series of actions, each action should flow into the next, as this will give the impression of the character having *planned* the sequence of actions in its *mind*.
6. **Straight ahead action and Pose-to-Pose action** describes two different approaches to animation. The first sees an animator drawing the animation frame-by-frame. Lasseter states that this is best for wild and spontaneous actions. The second approach starts by establishing the key poses of an animation before filling in the frames in between. This approach is best suited when accurate timing and a believably acted performance is required.
7. **Slow in and Out** refers to what is now commonly known as *easing* animation. It determines when an animation features more in-between frames surrounding an important key-frame, thus slowing down perceived movement and emphasising that pose.
8. **Exaggeration** is used to accentuate the above principles in order to make the animation both more *realistic* and entertaining. In his example, Lasseter describes that the Jr. Luxor lamp portrayed in "Light and Heavy" was given exaggerated child-like proportions and that its movements had to be exaggerated accordingly to match this appearance. Important to note is the fact that Lasseter states "The animator must go to the heart of anything or any idea

and develop its essence, understand the reason for it, so that the audience will also understand it" (Lasseter 1987 p. 41). This notion of the character-animator relationship, which again resembles Coleridge's "semblance of truth" (Coleridge 1817), is key to the difficulty of automating believable agents.

9. **Secondary actions** could be referred to as the *details* in a scene. This could be the movement of long hair, the swinging of a tail or the bobbing of a hat. In some cases, however, Lasseter explains, a secondary action can also be the facial expression on the character that *follows the primary action* of the characters' body movements.
10. **Appeal** is the *charisma* of an actor or scene. In animated terms this would be the graphical design of the character, the quality of the computer graphics or artistic style. Scenes and characters should neither be too simple, nor too complex.

Principles 4, 8 and 10 are concerned with the appearance of the agent and govern the visual impact of the character within the context of a scene. When making virtual agents more believable, emphasizing certain behaviour traits becomes an important since, according to Lasseter and Thomas and Johnston, they lie at the core of the personality.

Principles 2, 5, 7 and 9 all consider the physical properties of an animated object, which could be handled by using physics simulations in the animation. At the time when Thomas, Johnston and Lasseter defined these requirements, realistic physics simulations for animations did not exist, especially not in real-time (in Lasseter's case). Animators had to therefore imagine the effects of the physical properties of the animated object. Even the first principle *squash and stretch* has lost some of its importance in light of recent developments in computer graphics image processing. Object-motion blur can now be applied to fast moving objects and fulfils a similar purpose to *squash and stretch* (although the latter is still used for non-realistic rendering).

Principle 3, *anticipation*, is significant in that it represents the only principle from classical animation that directly informs the development of behaviour controllers. While it can be seen in realistic simulations of animals, such as the physically actuated body simulations found in Terzopoulos realistic fish (Tu and Terzopoulos 1994) and natural motions "dynamic motion synthesis" system (Naturalmotion 2005), it is not something that is inherent to all virtual agents. For agents that do not use simulated muscle systems, where early fluctuations and flexing can signify and "announce" an impending action, anticipation has to be integrated deliberately.

Lasseter describes how *straight ahead* and *pose-to-pose* animation techniques have been replaced by hierarchical animation techniques in the transition to computer graphics (Lasseter 1987 p. 40). Similar approaches are still used for handcrafted animations today and are often combined with techniques such as motion and performance-capture, which uses performances of real actors as the basis for body and facial animations.

Thus, while they still hold true today, adhering to the fundamental principles of computer animation is not the sole responsibility of the animator anymore. Physical properties of objects can be simulated in real-time, image processing effects can support the impact of the appearance of motion and modern hierarchical animation and motion capture techniques provide the building blocks for entire performances. This has allowed the role of the animator to shift toward a focus on the composition and staging of animated scenes, letting them concentrate on adding detail, appeal and exaggeration where appropriate.

2.1.5 Criteria for Believable Autonomous Interactive Agents

In his PhD thesis Michael Mateas defines a set of believability metrics adapted from the principles for believable characters from classical animation and the performing arts. He justifies this, stating that "creators of non−interactive characters have written extensively on what makes a character believable" (Mateas 1997). In Chapter 2 of his thesis, Loyall (1997) compiles a set of criteria for believable agents that he later used to evaluate virtual agents.

Based on his findings, Mateas (1997) presents a list of guidelines formed of six requirements for believability. The following list is quoted from Mateas (1997). However, note that numbers instead of bullets are used for reader convenience and later reference:

1. **Personality** – Rich personality should infuse everything that a character does, from the way they talk and move to the way they think. What makes characters interesting are their unique ways of doing things. Personality is about the *unique* and *specific*, not the *general*.

2. **Emotion** – Characters exhibit their own emotions and respond to the emotions of others in personality–specific ways.

3. **Self–motivation** – Characters don't just react to the activity of others. They have their own internal drives and desires, which they pursue whether or not others are interacting with them.

4. **Change** – Characters grow and change with time, in a manner consistent with their personality.

5. **Social relationships** – Characters engage in detailed interactions with others in a manner consistent with their relationship. In turn, these relationships change as a result of the interaction.

6. **Illusion of life** – this is a collection of requirements such as: pursuing multiple, simultaneous goals and actions, having broad capabilities (e.g. movement, perception, memory, language), and reacting quickly to stimuli in the environment. Traditional character artists do not mention these requirements explicitly, because they often get them for free (from a human actor, or as a deep assumption in animation). But builders of interactive characters must concern themselves explicitly with building agent architectures that support these requirements.

   (Mateas 1997, p. 6, emphasis in the original)

The first four criteria stem from the central notion of internal consistency, which was explored in the previous section. The fifth is optional, as a character should be able to convince on its own, without interacting with others.

Of specific interest is point 6, which refers to a collection of requirements for creating the "illusion of life". Mateas states that these are taken for granted by the character artists (Mateas 1997), a statement consistent with Coleridge's notion that the "inward nature" of the author provides "a semblance of truth" to the work (Coleridge 1817). The importance of these requirements is further emphasized by Mateas and Loyall, who proceed to define a specific subset of criteria that focus solely on the illusion of life (Mateas 1997; Loyall 1997). These criteria are discussed in the following section.

2.1.6 *Believable Agents and the Illusion of Life*

The requirements for creating the "illusion of life" is a set of detailed criteria that focus on the physical nature of the character being portrayed. While the first five criteria for believability focus on what components the "mind" of a believable agent should have, the "illusion of life" focuses on how these components should act and interact.

In his thesis, Loyall (1997) breaks down these elements into the following list, which is summarized from (Loyall 1997 pp. 23-27):

1. **Appearance of Goals** – Characters must appear to have goals. Loyall's method of implementing these is to "use explicit representations of goals in the architecture" (Loyall 1997 p. 24). An author would then create behaviours to express these goals.
2. **Concurrent pursuit of Goals and Parallel Action** – An agent must be able to perform multiple actions simultaneously.
3. **Reactive and Responsive** – Characters must be reactive to events occurring around them. Loyall states, "It is not enough to have this ability to be reactive. These reactions must be at speeds that are reasonable" (Loyall 1997 p. 24). The reaction time should be dependent on the character's current disposition (nervous, tired etc.) and situation it is in.
4. **Situated** – Described as a basic need for characters. It indicates the requirement that a believable agent has to choose appropriate actions according to its situation. In other words, instead of just taking orders, the agent must interpret them and demonstrate and awareness of its situation.
5. **Resource Bounded (body and mind)** – Physical needs and resources, such as food or stamina, often govern the behaviour of living beings. A believable character will also have to appear to be "embodied" in this sense.
6. **Exist in a Social context** – The social context of a character can be the back-story or setting of the scene in which the character takes part.
7. **Broadly Capable** – Loyall states that believable agents must be broadly capable in that they need to be able to perform a wide range of different internal and external actions similar to those of real living beings.
8. **Well integrated (capabilities and behaviours)** – With this criterion Loyall states that all agent actions should be performed in real-time and that any deliberation that takes longer has to be masked by other concurrent actions, while not causing any unrelated action to stop (he cites the example of a robot

stopping to process visual information). Loyall also mentions the more subtle discrepancy that comes from converting symbols from one knowledge system to another. A classical AI robot might know more in the context of speech than it does in the context of navigating an environment (e.g. it might be able to talk about what a "door" is, but cannot detect one if it sees it).

These requirements extend the fundamental principles for animation, which were covered in Section 2.1.4 to include the notions of real-time responsiveness and the properties of embodied agents.

In their current form, these criteria for believability are useful as a guideline, but they do not provide any indication of measurable metrics that could be used to evaluate a set of believable agent architectures against each other, or to establish whether an agent that fulfilled a subset of them, was still believable as a whole.

The next section discusses how this set of criteria for believability was developed into a format that made them suitable for evaluating and comparing agents.

2.1.7 MEASURING BELIEVABILITY

The criteria for believability presented in the previous chapter have proven to be good guidelines for animators and virtual agent developers, but they do not constitute a set of reliable metrics for comparing existing agents. While Loyall and Mateas state that a prototype showcasing their agents was presented to a large number of users (Loyall 1997 p. 162), the believability criteria were not used to evaluate the agents. Loyall provides only one measure of success in that he states that users were engaged with the prototypes and spent up to ten minutes playing with the agents.

Gomes et al formalised the believability criteria into a believability metric that can be used to measure the believability of computer controlled characters (Gomes 2013). They propose a series of questions and specify the format in which they should be presented on a questionnaire. The questions are designed to be understood by an audience unfamiliar with the abstract concept of believability, such that empirical studies with a wide diversity in the demographic spectrum are possible (Gomes 2011).

The following believability dimensions were defined by Gomes (2013). They suggest that each of these should be presented to the user in the form of a question to be answered on a Likert scale, with its boundaries ranging from "totally disagree" to "totally agree":

1. **Awareness** – *The Agent perceives the world around him/her.*

   A way to show that the agent is aware of its surroundings is important to make sure that it reacts to changes in the environment around it. Gomes (2013) draws upon Loyall's 3rd believability requirement for the illusions of life, namely that an agent must exhibit "reactive and responsive behaviour" (Loyall 1997 p. 24).

2. **Behaviour understandability** – *It is easy to understand what the agent is thinking about.*

   This question is intended to elicit whether the participants are able to create an internal mental model of the agent's thought process. To enable the participants to do this, Gomes cites Bates (1997) who stated that the agent's actions must reflect both its thoughts as well as its emotions.

3. **Personality** – *The agent has a personality*

   Gomes adopts this metric from Loyall's 1st requirement for believability. This defines that the agent has unique and specific ways of acting that differentiate it from other agents that may be performing the same type of action. Gomes states that the participants should be able to clearly identify what these personality traits are.

4. **Visual Impact** – *The agent's behaviour draws my attention*

   This dimension rates the degree to which important actions and emotions of the agent are emphasized by the animation of the agent. Gomes refers to Lester and Stone (1997) who state that enhancing visual impact requires a collaborative effort between the behaviour designer and animator to make sure that animations display behaviours with a varying degree of visual impact and should be in accordance with the personality originally defined. This seems to map directly onto Lasseter's computer animation principles 3,4 and 10.

5. **Predictability** – *The agent's behaviour is predictable*

   This is the only metric where higher scores can have a negative impact on perceived believability. Behaviour patterns should not repeat too much and animations should not be too recognisable. However, the behaviour should not be entirely unpredictable either, since this can negatively affect the behaviour coherence dimension. Gomes suggests that the best scores in this dimensions are ratings that are not close to the boundary values of the scale.

6. **Behaviour coherence** – *The agent's behaviour is coherent*

This question asks the user to interpret the agent's internal state. Is the agent acting on its own behalf, or do its actions seem random, irrational and incoherent?

7. **Change with experience** – *The agent's behaviour changes according to experience*
   Gomes defines this as a significant, permanent change in the agents due to "story" event. In an interactive narrative, these major events would define plot arc.
8. **Social Expressiveness** – *The agent interacts socially with other characters*
   Gomes uses Loyall's definition of social relationships for this metric. This is an interesting metric since it is the only metric that requires the presence of another agent to make sense. In cases where the only other character to interact socially with is the observer, different dynamics should be considered. Neither Gomes nor Loyall seem to address this issue specifically.
9. **Emotional Expressiveness** – *Which emotions best describe the agent's behaviour*
   The test for emotional expressiveness was not presented as a dimension on a Likert scale. Instead, the users were asked what basic emotion they believed the agent was exhibiting in significant situations during the observed scene. The number of matches between the participants' perceptions and the emotion intended to be displayed formed the score for this metric.

These metrics form the framework for a user study conducted to test a series of prototype agents in this thesis. During this user study, the metrics themselves will be evaluated with regards to their effectiveness in allowing for the comparison of agents and the measurement of the overall "believability" of an agent.

The next section reviews a selection of user centred testing approaches that could be used to evaluate the agents using Gomes' metrics as the central testing tool.

2.1.8 User Experience Testing

To inform the debate on measuring believability for characters that don't attempt to act in a human-like fashion, the evaluation of agents will be based on traditional user experience research techniques from the HCI field, combined with the metrics for believability from the literature discussed in the previous sections.

Types of Research

There are parallel ongoing discussions on how to best evaluate the experience of users playing games and an equal amount of research on the effects that virtual agents have on observers. Both are relevant to this research, but it was found that there is little work that combines the fields i.e. there are no examples of research in games testing that focused specifically on the discrepancy between agent design intent and perceived behaviour and also no believable agents research that focused on context of games.

Within the field of testing user experience in games there are several examples of work that discusses the use of metrics. Tychsen (2008) describes how game metrics, which were derived from general productivity software testing (Kuniavsky 2003) are used to evaluate the user's interactions within the game. Interaction with "entities" is considered as one of these metrics, but only evaluated within the context of the overall game experience. Other examples of user-centred research were reviewed (Nareyek 2004; Bernhaupt 2007; Chen 2007; Kim 2008; Nacke 2008), but none utilised or suggested specific (believability) metrics that take into account the interactions between players and non-player characters.

In research that does consider the believability of agents, testing for human-like believability remains the most common approach (Magerko 2004; Thurau 2004; Livingstone 2006; Wang 2009; Laird 2012). As discussed in Chapter 2.1.2, this interpretation of believability is different from the character-driven stance that Loyall (1997), Mateas (1997), Gomes (2013) and Perlin (Perlin & Goldberg 1996; Perlin 2009) take, which approaches the issue from a direction more related to the character arts than AI.

User Sampling Approaches

Since the focus of this study is to elicit the details of the interactions between users and the agent architectures, rather than the quantification of agent architecture performance, the empirical experiments will take the form of a qualitative study. For this purpose a small group of potential users will be identified and their feedback analysed in depth. Bryman (2008) suggests different kinds of user sampling approaches for observational studies. The following were considered for this study:

Purposive sampling: Is a non-probabilistic sampling approach. It does not allow for the generalization to a population. Subjects are selected due to their relevance to the research question.

Snowball sampling: A form of purposive sampling, snowball sampling can extend other sampling methods by encouraging initially selected participants to recruit others. More specific request can be made to access specific demographics i.e. ask your partner, friend, family etc.

Theoretical sampling: Also a form of purposive sampling, theoretical sampling is an important element in grounded theory (Glaser and Strauss 1967; Strauss and Corbin 1998), theoretical sampling sees the selection of 'objects' (since it can include more than just people) to be investigated as an ongoing process and not a single stage of research. The selection is guided by the emergent themes and evolving theory on the subject being studied.

Convenience sampling: The researcher selects participants that are easily accessible i.e. available locally, workplace colleagues, students.

Representative sampling: Aims to select participants such that the sampled group contains representatives of a wide range of characteristics within the population that is the focus of the study.

Gathering User Feedback

Tullis and Albert (2013) suggest a variety of user experience metrics. They broadly differentiate between three types of metrics that can be evaluated:

Performance Metrics: Measure the performance of the users with regards to a given goal. This could the time to complete a given task, or reaching a certain high score in a game.

Self-Reported Metrics: Are subjective feedback on the experience given by the users themselves. This feedback is filtered through the expectations of the users. This means that responses can be analysed both at face value and at a deeper level, where the research investigates the causes for the expectations.

Behavioural/Physiological Metrics: These metrics are focused on the physical responses of the user. This can include involuntary responses such as facial expressions, as well as physical performance metrics.

In the game development industry approaches that combine gameplay data (performance metrics) with qualitative analysis of (self-reported) user responses have become popular since they were publically stated to be behind the success of Halo 3's map design (Thomson 2007; Bungie 2007). Novel approaches such as "Biometric Storyboards" (McAllister 2011; Mirza-Babaei 2012) also seek to incorporate physiological user data into the analysis process.

Data Analysis

Bryman (2008) refers to the term *ethnographic content analysis* (ECA) as a process of analysing documents where the role of the investigator in the construction of meaning in the text is emphasised. The term was coined by Altheide (1996) and is sometimes called *qualitative content analysis*. The term *ethnographic* is used to indicate that there is an emphasis on "allowing categories to emerge from the data and on recognising the significance for understanding meaning in the context in which an item is being analysed" (Bryman 2008). Altheide (1996:16) defines ECA as a "recursive and reflexive movement between concept development-sampling-data, collection-data, coding-data and analysis interpretation".

This approach suits the objective of this thesis well, since it emulates a typical development scenario in games development. Here game developers often have "daily build" play sessions at the end of the day to evaluate the recent changes made to the game. The developers will sit with other colleagues and review the impact of newly implemented features in a semi-formal manner.

Bryman (2008) details a procedure for *ethnographic content analysis* that can be followed to elicit the central themes and issues from a dataset combining of textual and multimedia data. To our knowledge there is no *specific* content analysis technique that focuses solely on evaluating games. However, qualitative usability evaluation techniques from HCI have been successfully employed as part of evaluation procedures in the industry and in game-related research such as (Mirza-Babaei et al 2012; McAllister 2011).

In addition to this game-specific data will be captured and correlated to the feedback from the participants. This approach has gained wide-spread popularity since Microsoft used it to aid the development of the multiplayer scenarios in Halo 3 (Bungie 2007; Thompson 2007). The approach, which sees the developers capturing gameplay data and visualising it in ways that can inform design decisions was also described by Tychsen (2008). One of the popular techniques that Bungie used was to create heat-maps of the places where players died in a level. This allowed the developers to see “hot spots” where players tended to die, allowing them to change aspects of the level, to address the issue. The central benefit of this approach over pure qualitative analysis of user feedback is that the quantitative data is used to find and support significant points of interest in user feedback which would otherwise have been easily missed.

2.1.9 The Relationship Between the Artist and the Character

Although the previous two sections 2.1.5 and 2.1.6 defined the criteria for believable agents, the question remains what *personality*, what *emotions* and what *motivations* a specific agent should have. This section discusses some of the techniques that traditional character artists use to fulfil the criteria listed in Section 1.1.5, making them into additional requirements for autonomous believable agents.

Apart from technical guidelines and techniques (e.g. squash and stretch, anticipation) regarding the way animated characters can be brought to life, one of the key criteria mentioned by Disney (Thomas and Johnston 1981) and elaborated upon by Loyall (1997) and Mateas (1997) was concerned with the relationship between animator and the character portrayed.

The criterion in question is that an observer *must be able to perceive the thought process a character goes through* in its behaviour and in its movement. For example, thinking could be expressed via a short pause, a moment of deliberation before the character performs an action. During this pause its internal thought process would be expressed by a squinting of its eyes, staring blankly at a point in space or rubbing its chin until it grasps an idea and proceeds to perform the appropriate action. This would be done in a similar way when animating expressions of anger, exertion or other emotions. In all cases it is traditionally the animator who must consider and incorporate the subtle expressions and time delays that internal thought process causes.

This is a key aspect of the classical animation approach as detailed by the Disney animators (Thomas and Johnston 1981); all of this information – the thoughts of the character, the flow of ideas - comes from the author, in this case the animator of the character. This is perhaps the main reason why Disney often had a single animator focus on animating (doing the key-framing for) a specific character in a given motion picture. This author/character pairing helped to keep the portrayed character consistent throughout the picture. Author/character consistency is also one aspect in which animation studios had to re-adapt after early industrialization (having many animators draw every character) and take cues from the art of creating believable characters in dramatic writing, where author/character integration is usually a given.

In terms of creating different *personalities* for characters, there are many theories about different personality types such as the character archetypes described in "A Hero with a thousand faces" (Campbell 1993). Cognitive psychology also has theories about how emotions are processed, expressed and formed from basic primitives (Ortony, Collins and Clore 1988; Ekman and Rosenberg 1998). While these do not provide "blueprints" for character types, they can act as guidelines that a human animator can relate to. It is still up to the animators to hone their skills of transferring their own personality, moulding it, augmenting it with different personality traits (not everyone is a tyrant, lover, hero, and villain after all) and applying it to the animated character.

This is why the workplaces of character animators at studios often have a mirror for reference – the animator mimics the thought process that his character should portray and bases the character animation on their own expressions.

As was discussed in Section 2.1.4 of this thesis, most of the technical principles regarding the physical properties of animated objects have been implemented in computer software in the form of simulations, the essential component that remains unsolved is the need for the personality of the artist to define and understand what "goes on" inside the character being portrayed. This knowledge is still required to inform the animation process. As Oscar Wilde cleverly puts it in his novel "The Picture of Dorian Gray" from 1891:

> "Every portrait that is painted with feeling is a portrait of the artist, not of the sitter."
>
> (Oscar Wilde "A Picture of Dorian Gray" - 1891)

There seem to be two main approaches to this problem stemming from the artist/character relationship in the context of believable, autonomous characters. The first approach gives the designer of the autonomous agent a large amount of control over the internal attributes of the agent, thus making the designer the *character artist*. This artist-based approach is common in current media such as video games and is one that Loyall (1997), Mateas (1997), Massive (Regelous 2009) and Perlin (2009) support with their believable agent projects. The second approach is to use a bottom-up approach to character design. In this approach, the agent's personality is an emergent property of the internal mechanisms that drive the agent interacting with the environment. The agents in games such as Creatures (Grand and Cyberlife Technology 1997) and the Sims (Wright and Maxis Software 2000) have demonstrated the use of emergent techniques to create characters, which is also the approach taken for the architecture developed in this thesis.

2.1.10 Observer Psychology and Agent Behaviour Interpretation

During her time as a member of Bruce Blumberg's "Synthetic Characters Group" and the A.L.I.V.E project at MIT, Pattie Maes confronted questions regarding the requirements of creating lifelike, believable agents. According to her, to build a lifelike entertaining agent, "the researcher is forced to address the psychology of the user" (Maes 1995 p.111). She addresses this concern with the following questions:

- How will a typical user perceive the virtual characters?
- What behaviour will she or he engage in?
- What misconceptions and confusing situations can arise?

Like Braitenberg (Braitenberg 1984 p.31) she is referring to the interpretations of the agent's behaviour by a typical user. Braitenberg pursues this notion with his "critical philosopher" personas, which he uses to represent a set of varied and often opposing opinions and interpretations that an observer might have regarding the agent behaviours that he describes.

The term "Synthetic Psychology" stems from this relationship between agent behaviour and observer. As Braitenberg states: "when we analyse a mechanism, we tend to overestimate its complexity" (Braitenberg 1984 p.20). An observer might believe an agent "likes" or "fears" another object in the environment, while in truth there is merely a simple internal wiring between sensors and motors. A threshold waiting to be met might lead some observers to think the agent is going through a deliberation process. In short, some observers will attempt to describe the reasons for a given behaviour using terms and processes that they are familiar with, which often results in them overestimating the actual complexity of the underlying mechanism. This is what Braitenberg calls "the law of uphill analysis and downhill synthesis" (Braitenberg 1984 p.58) and is the essence of "Synthetic Psychology", which is carried through the entirety of Braitenberg's book.

A similar phenomenon called the "Eliza effect" was observed earlier by Joseph Weizenbaum (1966). When he presented his ELIZA chatterbot (a computer program that imitates human conversation) to unsuspecting users, they consistently over-interpret the machine's complexity. He found that as long as the system did not actively destroy the illusion of this complexity, people tended to continue to see complex thinking mechanisms, where they didn't exist. They imbued the system they did not understand with human properties they supposedly understood. They tried to make sense of its behaviour by anthropomorphising it. Taking this into account, it suddenly seems less surprising that users of today's computers are capable of "getting angry" at a machine.

This leads to another aspect of Braitenberg's Synthetic Psychology approach. While it can be interpreted in several ways (Boden (2006) considers it tongue in cheek), the use of the term "Psychology" in the title of Braitenberg's book refers not only to the incremental process of building a brain, but also to Braitenberg's liberal use of psychological terminology throughout his thought experiments. His use of words such as *love* and *fear* to describe the behaviours of his vehicles is significant in that it provides the reader with an anthropomorphic interpretation of what would otherwise be cold descriptions of a series of behaviours. In the context of using Braitenberg's architectures as behaviour controllers for believable agents, this anthropomorphic terminology provides a comprehensible set of parameters that behaviour designers could use. This thesis follows in his footsteps in that it adopts Braitenberg's terminology and expands upon it, referring to the process of building biologically inspired architectures, while using anthropomorphic terms from cognitive psychology to describe emergent behaviours, as the *Synthetic Psychology Approach.*

2.1.11 Perceived Emergence

The study of perceived emergence by Ronald, Sipper and Capcarrère (Ronald et al. 1999a; 1999b; Ronald and Sipper 2001) found that a lower proficiency of an observer in fields relating to the internal mechanisms generating the behaviour (robotics, AI, biology, neuroscience, artificial life etc.), will cause an increase in the level of perceived emergence when observing agent behaviour. This means an observer who is less familiar with the system, is more likely to overestimate the complexity of its internal mechanisms. This finding mirrors Braitenberg's "law of uphill analysis and downhill synthesis" (Braitenberg 1984 p.58), which states that complex behaviour leads observers to assume complex internal processes.

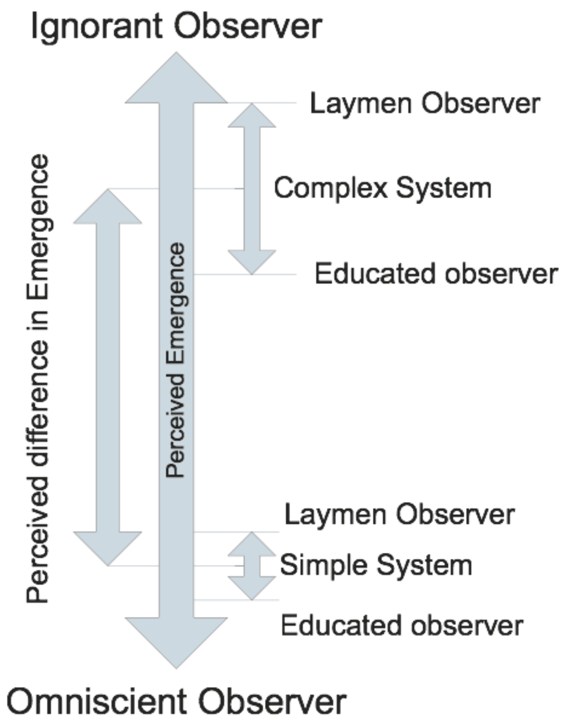


Figure 1 Perceived emergence hypothesis

Figure 1 illustrates that the discrepancy between prior knowledge of the observer in respect to the inner-workings of an observed subject determines the perceived level of emergence. It shows how the subjective observations of a mixed group of educated and laymen observers can be used to compare the perceived emergence of two systems.

Based on this notion, a correlation between perceived emergence and suspension of disbelief could also be suggested. In this case, the more emergent an agent's behaviour appears to observers, the more likely they are to be convinced to believe that the agent is an autonomous, living being with its own agenda. Therefore the goal is to create diverse and complex behaviour and avoid repetitive, *robotic* behaviour.

2.1.12 The Uncanny Valley Effect

For creators of believable human-like characters and humanoid robots the Uncanny Valley concept has been a problematic issue. Originally the term "uncanny" or the German "unheimlich" was coined by psychologist Ernst Jentsch in 1906 (Jentsch 1997) and Sigmund Freud (Freud 1919). In 1970 roboticist Masahiro Mori developed the theory that states that if a realistic human-like figure comes close to looking like a real human being, there is a theoretical region during which an observer will suddenly switch from an empathetic, to a repulsed response (Mori 1970). This effect can be observed for static or moving images, figurines and robots and does not just apply to the visual impression (looking like a human), but also to movement (moving like a human) (Mori 1970; White et al. 2007; MacDorman et al. 2009).

The movie "Final Fantasy: The Spirits Within" (Sakaguchi and Sakakibara 2001) is a good example cited by Pollick (2009) in his review of recent examples of the uncanny valley in media. Final Fantasy featured characters that look very realistic when static. The artists considered almost all the criteria for physical photo-realism, such as light reflections and refractions in the skin and eyes, texture, moisture and natural colours. Yet when seen in motion, the animated characters often elicit a feeling of discomfort, a feeling that somehow these characters look less like living breathing people and more like walking, talking corpses.

A way to avoid the uncanny valley altogether is by avoiding realism. Non-realistic anthropomorphic characters, such as Disney's Donald Duck do not run the risk of seeming too human-like, yet can be used to transport believable human traits – a method similar to the use of animal characters in Aesop's Fables (Aesop 2003).

2.2 Believable Agent Models

This section reviews the existing models for believable agents that inform this thesis. Being from several disparate fields, the relevance of some of these models might not be initially apparent, but the reviews will cite their relation to the model presented in this thesis.

After an introduction to the work that inspired Braitenberg's thought experiment, this section will then focus on work that is either directly inspired by him, or bears relevant similarities to it. Other research that could inform the specification and development process in this thesis are also reviewed.

The first section is a review of agents in Virtual environments, as these are closest to the prototypes presented in this work. Next follows a review of robotics architectures, as virtual agent models take a majority of their techniques from this field and classical AI.

The following sections review other research initiatives in the believable agents domain and how they compare to the work in this thesis. The review extends to agent models employed in industry, namely character animation tools and agents in interactive games.

2.2.1 *From Ducks and Tortoises to Vehicles*

In the 1940s William Grey Walter built a series of mobile, autonomous robot "tortoises". These were some of the first mobile autonomous robots and displayed complex emergent behaviour while having only a small set of simple components. The robot platform had two wheels and a front coaster and carried a set of light sensors and a transparent plastic bump sensor in the shape of a dome surrounding the innards. The shell-like dome is what gave these robots their description.

The behaviour of the robots was complex and life-like, even though the mechanisms controlling behaviour were extremely simple. The robots followed light sources in unpredictable paths and were able to move around obstacles using the bump sensor. Walter provides a set of criteria that he set for his "Machina Speculatrix" and "Machina Docilis" to meet:

> Not in looks but in action, the model must resemble an animal. Therefore, it must have these or some measure of these attributes: exploration, curiosity, free-will in the sense of unpredictability, goal-seeking, self-regulation, avoidance of dilemmas, foresight, memory, learning, forgetting, association of ideas, form recognition, and the elements of accommodation.
>
> (Walter 1953, Ch. 5)

These criteria would later be picked up by the emerging field of artificial life (Alife). Aside from Langton's definition of Alife "life as it could be" (Langton 1990) there have been attempts at defining a set of criteria for 'life', or rather a set of properties that all living things must meet. Although they state that the following list is likely incomplete and imprecise, Farmer and Belin (Farmer and Belin 1992) compiled the following list of properties of life:

1. *Life is a pattern in space-time*, rather than a specific material object. For example, most of our cells are replaced many times during our lifetime. It is the pattern and set of relationships that are important, rather than the specific identity of the atoms.

2. *Self-reproduction,* if not in the organism itself, at least in some related organisms. (Mules are alive, but cannot reproduce.)

3. *Information storage of a self-representation.* For example, contemporary natural organisms store a description of themselves in

the DNA molecules, which is interpreted in the context of the protein/RNA machinery.

4. *A metabolism,* which converts matter and energy from the environment into the pattern and activities of the organism. Note that some organisms, such as viruses, do not have a metabolism of their own, but make use of the metabolisms of other organisms.

5. *Function interaction with the environment.* A living organism can respond to or anticipate changes in its environment. Organisms create and control their own local (internal) environments.

6. *Interdependence of parts.* The components of living systems depend on one another to preserve the identity of the organism. One manifestation of this is the ability to die. If we break a rock in two, we are left with two smaller rocks; if we break an organism in two we often kill it.

7. *Stability under perturbations* and insensitivity to small changes, allowing the organism to preserve its form and continue to function in a noisy environment.

8. *The ability to evolve.* This is not a property of an individual organism, but rather of its lineage. Indeed, the possession of a lineage is an important feature of living systems.

(Farmer and Belin 1992 p.818 emphasis as in original)

Following on from the earliest automata by Vaucanson (e.g. Mechanical Duck, 1735), Walters simple architectures provided the inspiration to the field of Alife and Braitenberg's Vehicle 2 (Braitenberg 1984) who also used a very similar differential drive platform and light-following behaviour in his first designs.

2.2.2 VEHICLES – VALENTINO BRAITENBERG

This section provides a detailed description of Braitenberg's thought experiment, which presents an agent architecture through a series of incremental steps. With each chapter Braitenberg increases the complexity of the agents he calls "Vehicles", culminating in a final chapter to create a *complete* (yet purely theoretical) agent model.

Being a neuroscientist and cyberneticist, Braitenberg's main inspiration comes from his study of natural organisms. Prior to his seminal book "Vehicles: Experiments in Synthetic Psychology" (Braitenberg 1984) he published a paper titled "Taxis, Kinesis and Decussation" (Braitenberg 1965), which introduced many of the principles he would elaborate on in his thought experiment. *Taxis* is reflex-oriented movement in relation to a source of stimulus, *Kinesis* is movement that depends on the intensity of a stimulus and *Decussation* refers to the prevalence of cross-connecting fibres in animal nervous systems and particularly brains. The paper also introduced the idea of "Vehicles" as little, wheeled robots with sensors and simple inter-connecting wire networks as brains, to illustrate the effects that the different controller architectures may have on behaviour.

The book (Braitenberg 1984) also includes chapters of biological notes that provide justification for the models he presents in the body of his work. Yet even with these, Braitenberg's descriptions were sometimes too vague or incomplete, so alternative bio-inspired models had to be considered. These are also discussed in the following sections, but more are referenced in the prototype implementation chapters where required.

Braitenberg's architecture relies heavily on analogue properties in the connectionist networks he describes. Although digital logic components are introduced to the architecture, these components are still connected by links of varying and continuous strength. This approach is heavily inspired by biology, where many examples of Analogue networks can be found, for example:

1. Genetic regulatory networks
2. Metabolic networks
3. Neural networks

In line with the terminology of cybernetics, Braitenberg's thought experiment uses the metaphor of electric circuits to portray its design. For some of the experiments presented in this thesis, the analogue circuits described in the thought experiment had to be turned into an "analogue design", a digital model that considers the continuous physical and temporal properties of components and the connections between them.

The 'Synthetic Psychology' Approach

The term "Synthetic Psychology", which Braitenberg uses to describe his methodology refers to an insight that he calls the "law of uphill analysis and downhill invention" (Braitenberg 1984 p.58). His claim is that it is much easier to invent machines that exhibit complex behaviour than it is to guess the internal structure of such a machine from observing its behaviour.

Braitenberg's law of uphill analysis and downhill synthesis "learning by building" is a constructionist approach to cognitive science. Around the same time as Braitenberg presented his work, this methodology was also being explored by Seymour Papert (Papert 1980) who created his seminal educational tool and programming language LOGO. The LOGO software inspired by Jean Piaget's experiential approach to learning (Piaget 1929) and the "turtle" robot models originally designed by William Grey Walter in the 1940s. Presented as a drawing tool, LOGO provided a simple set of instructions that allowed students to learn programming and robot control.

Summary of Braitenberg's Architectures

Braitenberg starts his thought experiment with a minimalist design, using the fewest number of components possible. This imaginary machine, which he names "Vehicle 1", has a single temperature sensor and a motorized wheel attached to opposite ends of a small platform. The temperature sensor is connected to the motor by a wire. When the sensor becomes active, the wire transmits a signal proportionate to the temperature measured to the motor of the wheel. The resulting behaviour of the robot illustrates the principle of Taxis, since the robot will only start to move in response to external stimuli received by its on board sensor. The robot also exhibits Kinesis, since its speed of movement varies with the intensity of the measured stimulus. Even though this simple machine does not possess any steering capabilities, Braitenberg argues that interactions with a dynamic environment can affect its movement trajectory and result in surprisingly complex, emergent behaviour.

The second chapter combines the components of two "Vehicle 1" robots, creating a "Vehicle 2" type with two sensors and two wheels. The wheels can be driven independently, providing these robots with a steering mechanism in the form of a bilaterally symmetric differential drive. Using this platform, the thought experiment explores the biological concept of decussation. The wires that connect the sensors to the motors can either be attached in a parallel (uncrossed) or crossed configuration.

When connected in parallel, each sensor controls the speed of the wheel on the same side, while the crossed configuration makes each sensor control the wheel on the opposite side of the robot body. This results in two drastically different reactions toward a perceived stimulus, which Braitenberg compares to the instinctual "fight" (crossed) or "flee" (parallel) behaviour of animals, naming them "aggression" and "fear".

The following chapter explores two further variants of these base configurations. Instead of having an excitatory effect, the sensors now inhibit the activity of the motors they are connected to. The stronger the signal from the sensor, the more the corresponding motor slows down. The behaviour of these parallel and crossed inhibitory configurations is again described using anthropomorphic psychological terms and compared to animal behaviour. The inhibitory-parallel configuration is likened to "love", since the robot will seek out and permanently remain near the nearest source of stimulus. The robot with crossed wire connections is called "explorer", since it will also slow down when near a source of stimulus, but will actively steers away from it.

While the architectural features presented in first three chapters serve to illustrate the principles Braitenberg had discussed in "Taxis, Kinesis and Decussation" (Braitenberg 1965), chapters four and five extend his basic "Vehicle" platform. In chapter four he explores the effect of non-linear dependencies between sensory stimulus and motor output. With it, the concept of thresholds is introduced, which he uses in chapter five to build logical units called "threshold devices". As he states in the biological notes that accompany his thought experiment, these "threshold devices" are in fact Braitenberg's adaption of the McCulloch Pitts artificial neuron model (McCulloch and Pitts 1943).

The first five chapters of the book are perhaps the most well-known as they describe very simple robot architectures that are easy to build in reality. These architectures have been referred to and used by other scientists so often that they are now commonly called ''Braitenberg Architectures" (Koball and Wharton 1988; Hogg, Martin and Resnick 1991; Mobus and Fisher 1994; Lambrinos and Scheier 1995; Seth 1998; Kowall 2005). One reason for this popularity is perhaps Braitenberg's choice of names for his architectures. Chapter Two is called "Fear and Aggression", while the third *Vehicle* is called "Love". The juxtaposition of the simple architecture he proposes (literally consisting of only two wires) with terms that have such complex psychological connotation raises philosophical questions about complexity and emergence in the reader.

It is these simple architectures that are most suited for traditional applications of Games AI and interactive animation, where control over the agent's behaviour by the author is required. After establishing the base principles of his architecture, Braitenberg introduces the elements of evolution, memory and prediction. While the initial models are simple enough to design by hand, these additional dynamic elements introduce the risk of making it more difficult for a developer to predict the behaviour resulting from choices made during agent design process.

Chapter Six takes a break from the practical work in order to describe the process by which the engineer should construct new *Vehicles*. Akin to Darwinian evolution, his methodology introduces natural selection and mutations into the creation process. Braitenberg calls these "a source of intelligence that is much more powerful than any engineering mind" (Braitenberg 1984 p.26).

It is Chapter Seven, "Concepts" that introduces the first fictional component, the Mnemotrix wire. The name suggests a memory capability and indeed Mnemotrix wires are subsequently used to construct the first *Vehicle* with an associative mind. This is the first of the more controversial chapters in the book. While the first five *Vehicle* designs have been implemented and tested by several scientists, *Vehicles* seven to fourteen rely heavily on the idea that "association is the most important principle by which information about the environment is incorporated into the brain" (Braitenberg 1984 p.114). This connectionist idea was inspired by the work of D.O. Hebb (Hebb 1949). In his biological notes for the latter *Vehicles*, Braitenberg admits that the neuron model he employs is a simplification and indeed over the past years neuroscientists (even Braitenberg himself) have actually proven that the way information is represented in our brain is a rather more complex process than pure association, being actually more akin to the growth of a plant (Lucic, Kossel et al. 2007). Nevertheless, Braitenberg's proposed model is still functionally sound.

The prior critique has led to the latter chapters of the book to be less well known and referenced than the first five *Vehicle* designs. This is a pity, because they contain some of the most interesting ideas in the book. The following paragraphs summarize these key ideas.

In chapters eight and nine, Braitenberg describes the construction of an artificial visual cortex using different neural (or Mnemotrix) networks for different types of properties found in perceived objects, such as symmetry and movement in space.

Chapter Ten discusses the emergent property of "Getting Ideas", which focuses on the notion that sensory perceptions and concepts can form themselves into groups to form new, overarching concepts. He explains this by describing how the concept a two-faced coin can be formed from the separate perceptions of the "heads" and "tails" face and the perceived action of "flipping the coin", which combines the two.

The next three chapters make use of the second fictional component, the Ergotrix wire. Chapter Eleven "Rules and Regularities" introduces this wire, describing how it enables the *Vehicles* to remember sequences of sensory events. In Chapter Twelve "Trains of Thought", Braitenberg notes that all the *Vehicles* containing Mnemotrix wires are in danger of a pathological condition similar to epilepsy (Braitenberg 1984 p.63) caused by the reflexive activation of interconnected neuron elements (or Threshold devices as he calls them). He goes on to describe the cure for this condition, a controlling element that subdues activation across the whole brain if too much activation of neural elements is detected. The interesting thing is that he suggests that this will give rise to the emergence of trains of thought, the ability to move from one thought to the next without the need for an external "trigger".

Chapter Thirteen "Foresight" introduces a system called the "Predictor". This system causes the *Vehicle* to act upon expected events, rather than currently perceived events. This means that a *Vehicle* that repeatedly sees a ball rolling across a table and falling off the edge will (after sufficient training) "see" the ball fall off before it actually happens. In addition to this, Braitenberg also introduces a mechanism to deal with unexpected deviations from the expected event. Similar to psychological trauma, this mechanism reinforces the memory of those exceptions from the rule.

The final chapter "Egotism and Optimism" refines the "Predictor" architecture described in Chapter Thirteen by adding the ability to judge predicted events, categorizing them into good and bad. Incorporating an affinity for positive events (following the "pleasure principle"), Braitenberg states that such a *Vehicle* will now act with anticipation towards a desired event, imbuing this final creation with an air of personality.

It is possible to summarize this series of fourteen stages that Braitenberg presents into five incremental steps involved in constructing a "Vehicle" based agent:

1. INSTINCTUAL BRAIN: Evolve discrete sensor/actor connections. This is a process of evolution that intrinsically embeds an "understanding" of the relationship between agent and environment within the architecture that is analogous to the evolutionary development of instincts in higher animals. Braitenberg refers to this as the "Darwinian" part of the agent's brain (Braitenberg 1984 p.77)

2. CONCEPTS: Forming Concepts via a connectionist approach. Inspired by neural models, utilizing weighted connections with a variety of learning rules.

3. TEMPORAL PATTERNS: Enhance the prior neural model to include time-delays that enable Sequences (temporal patterns) to be stored and reproduced.

4. ACTION/PREDICTION: Split Instinctual brain / predictive brain
    a. Follow instinctual brain if positive/negative trauma arises
    b. Teach trauma to predictive brain (internally repeating the event)

5. TENDENCY: Select from multiple possible predictions the most:
    a. Positive (optimism)
    b. Negative (pessimism)
    c. Flow (challenge: between boredom & anxiety)

Vehicle Property Table

Below is a table summarizing the descriptions of the agent designs in Braitenberg's "Vehicles" (Braitenberg 1984), which were amended with terminology from the notes by Lafave (2000). This table will form the basis of the specification of components and the experiment design of this thesis.

Vehicle	**Components**	**Concepts explored**	**Behaviour**
1	1 Sensor 1 Motor Single Wire	ALIVE	Kinesis, Moves in proportion to stimuli

Vehicle	Components	Concepts explored	Behaviour
2a	2 Sensors 2 Motors Uncrossed excitatory connection	COWARD	Turns away from source Speeds up when near source → Flees
2b	2 Sensors 2 Motors Crossed excitatory connection	AGGRESSIVE	Turns towards source Speeds up when near source → Attacks
2c	2 Sensors 2 Motors Uncrossed & Crossed excitatory connection	ALIVE	Like Vehicle 1, Moves in proportion to stimuli
3a	2 Sensors 2 Motors Uncrossed inhibitory connection	LOVE	Turns toward source Slows down near source
3b	2 Sensors 2 Motors Crossed inhibitory connection	EXPLORER	Turns away from source Slows down near source
3c	Multiple Sensors Cooperating Monotonic Dependences 4 Sensors 2 Motors Example: 1. Uncrossed/excitatory: heat 2. Crossed/excitatory: light 3. Uncrossed/inhibitory: smell 4. Crossed/inhibitory: oxygen	VALUES	Shows COWARD, AGGRESSIVE, LOVE and EXPLORER behaviour towards different stimuli Example: 1. Cowardly toward areas of high temperature 2. Aggressive toward light sources 3. Loves smell sources 4. Explores for oxygen
4a	3c -> with Smooth Non-monotonic dependences	KNOWING INSTINCTS	May circle sources, run between them, approach them to a certain point and turn around Same as 3c, but less predictable

Vehicle	Components	Concepts explored	Behaviour
4b	4a -> Non-monotonic dependencies with Thresholds	DESCISIONS WILL (free?)	Vehicles seem to ponder before acting abruptly
5	4b -> with Threshold Devices Some of them networked (counters)	NAMES LOGIC MEMORY	Reacts to specific situations. Counting Elementary (binary) Memory Externalisation of memory through action
6	5 -> with Evolved connections	EVOLUTION CREATIONISM	Exact wiring cannot be determined Evolutionary adaption of wiring to environment
7	6 -> Mnemotrix wires connect all threshold devices	ASSOCIATION CONCEPTS ABSTRACTION	May associate things that occur at the same time If multiple things that are associated happen to belong to a group (e.g. colours), abstraction may occur
8	7 -> Object Detector Movement/ Directionality Detector Delay Element Lateral inhibition Internal representation (maps) of space	Reality of Objects Edges Movement 2D & 3D Space	Detects Objects, their movement Can determine distances between points in 2d and 3d space (pathing).
9	8 -> Shape Detection Bilateral symmetry Radial symmetry Periodicity & cross-correlation (using Fourier analysis)	"Having me in mind": personal relation Community Singularities (sources)	Detect symmetrical shapes Form reaction to other vehicles heading towards one (confrontation)
10	Trained Mnemotrix connections-> Emergent originality of ideas and conceptual images	HAVING IDEAS Thinking Foresight	Sees the same string of stimuli many times and learns to associate its elements by association as in vehicle 7
11	10 -> with Ergotrix Wires	Temporal Causality	Associates temporal events that are active in succession

Vehicle	Components	Concepts explored	Behaviour
12	11 -> with Epilepsy inhibitor	Trains of thought	Epilepsy caused by reciprocal activation is counteracted by: 1. Measuring the rate of change of the number of active elements in the whole brain. 2. Raising all the thresholds by an appropriate amount if the danger of reciprocal activation arises 3. Lowering all the thresholds to encourage circulation of activity if overall activation level is low.
13	12 -> splitting the brain into Predictor and Sensor parts. Short Term memory Darwinian Evaluator	Prediction Short term memory	Compares expectations with sensory input. If comparison yields strong difference, predictor is turned off to believe/act on the sensors. Darwinian Evaluator is used to trigger trauma/feedback loop to "give special weight to the rare but decisive" experiences and train these important sequences of events in the Ergotrix-powered predictive brain.
14	13 -> Imposes selection on multiple possible predictions	EGOTISM OPTIMISM	In the case where the Predictor points toward multiple equally likely states, the Darwinian evaluator is used to determine the most positive (optimism) state and propagates it into activity

Table 1 Summary of agent designs in Braitenberg's thought experiment

2.2.3 *The Rodney Brooks Subsumption Architecture*

Around the same time that Braitenberg published his thought experiments, Rodney Brooks presented his "Layered Control System for a Mobile Robot" (Brooks, 1986). This reactive agent architecture combines a collection of simple behaviours into a layered architecture. The lower layers would for instance deal with avoiding obstacles or backing away from the edge of a table. Higher layers would deal with more abstract behaviours, such as "exploring the world" or "find food". This created robots that showed similar behaviour patterns to those described by Braitenberg.

The subsumption architecture was able to create a variety of seemingly complex behaviours, such as hiding in shadows or seeking out a recharging station. The animal-like, quick response time of the robots was impressive at the time, when deliberative systems were still far from being able to keep up with real-time events. The main issue with this architecture was that it was difficult to create complex systems, due to interference between behaviour components when many layers were active. Also, action/behaviour selection was difficult to implement using only inhibition between behaviour components. Brooks' research group eventually moved on from his subsumption architecture when he found that it was not sufficient on its own to enable autonomous robots to adapt, learn and perform sequences of actions or discern between several viable behaviour options.

2.2.4 *LEGO Bricks – From Braitenberg to Mindstorms*

This research project ultimately led to the development of the LEGO *Mindstorms* robotics toolkit. Indeed the name "Mindstorms" comes from the analogy Hogg, Martin and Resnick (Hogg et al. 1991) made between Braitenberg's "Synthetic Psychology" approach and Seymour Papert's constructionist approach to learning, which was detailed in his book "Mindstorms: Children, Computers, and Powerful Ideas" (Papert 1980).

Presenting a series of modified LEGO bricks, the group constructed a series of robot designs based on Braitenberg's early Vehicle architectures:

Timid is a Vehicle 1 architecture that moves in bright areas and stops in shadow.

Indecisive is also a Vehicle 1 design, but drives forward in light and backward in shadow. *Paranoid* is similar to Vehicle 2, in that it has two motors, yet only has one light sensor (instead of Braitenberg's 2 sensors) connected to an arm protruding from the front of the robot's body. In light, the sensor causes the two wheels to spin in the same direction, moving the robot forward, but when the sensor hit a shadow one is reversed, thereby spinning the robot around until the sensor is out of the shadow and the robot can continue to move forward. The behaviour caused is that of avoiding shadows.

The *Dogged* design is most similar to Braitenberg's Vehicle 5 in that is uses a flip-flop gate, which in Braitenberg's terms would have been a simple memory network of Threshold devices. The flip-flop gate acts as a forward/backward toggle switch for a single motor and is connected to bumper sensors on either end of the robot's body. The behaviour this creates is that the robot will reverse after every collision, thereby avoiding getting stuck at obstacles. The *Insecure* robot uses a whisker-like touch sensor that is connected to one motor directly and the other motor via an "inverter". Similar to Vehicle 3's inhibitory connections, the inverter sends the inverse of the activation signal from the whisker sensor to the other motor. This causes the robot to edge along walls as the motor couple switches between turning toward (when the whisker is not bent) and away (when the whisker is bent) from the wall.

The *Driven* design has the same architecture as Vehicle 2b in that it uses two crossed connections between two light sensors and two motors to make the robot turn towards and approach a light source. The *Persistent* robot combines both light following and collision detection behaviour in a similar manner to how multiple sensors are coupled in Vehicle 3c. Here we see multiple sensory-motor control couplings competing to form the overall behaviour. The *Persistent* robot uses its front bumper to trigger a timer, which is set to a given period during which it causes the motors to reverse their direction and back up from the obstacle. After the timer stops the motors resume to drive in their usual direction and the robot proceeds to move forward.

Two robot designs are paired to show a simple interactive scenario. The *Repulsive* robot has a set of bright lights attached to its front and drives continuously forward. The *Attractive* robot has a light sensor attached to its rear, which causes its motor to move it forward when light is sensed. The example scenario put the two robots in a line, the Repulsive robot behind the Attractive robot. Once the Repulsive robot is near enough to shine light on the Attractive's light sensor, the Attractive robot drives off. It is unclear why the *Attractive* robot has been given that name. It has no property that actually attracts the other robot and the experiment only works (and displays the desired “avoidance” behaviour) when the robots are set up so the forward moving *Repulsive* robot will approach the *Attractive* robot from behind.

The *Consistent* robot uses a sound sensor connected to a sequence of two flip-flop gates, which connect the two motors. The flip-flop gate sequence has the effect of a counter that cycles through the four states on-on, off-on, on-off and off-off. Because each flip-glop gate is connected to only one motor these states correspond to the four different types of motion forward, left, right and backward respectively. The resulting behaviour is that when the sensor registers a loud noise, the robot switches to the next motion state in the sequence. This architecture is a Braitenberg Vehicle of type 5, which uses threshold devices to create counters.

The final design re-purposes the components used for the robots to build a mousetrap. A light shining at a light sensor placed in front of the bait is used to trigger the trap. When the beam is broken a timer activates a motor for as long as it takes to close the door of the trap. A flip-flop gate is used to make sure the timer is only triggered the first time the light beam is broken. This final experiment has almost nothing in common with Braitenberg Vehicles or believable characters, but shows that components and mechanism developed for robotics have wider application.

A key difference between the architectures proposed in the MIT project and Braitenberg’s thought experiment, is that the motor speed is not dependent on the intensity of the stimulation of the sensors. Braitenberg refers to this dependency as “kinesis” – movement that depends on the intensity of stimulation – and cites it as one of three key factors affecting behaviour (Braitenberg 1965). In the prototypes developed during this project all activation is either completely on or off, which would be equivalent to using Braitenberg-style threshold devices between every connection. This causes somewhat more erratic and less smooth-looking behaviour, especially with the robots that alternate between the activation of two motors in a differential drive, such as the *Paranoid*, *Insecure*, *Driven* and *Persistent* designs. These will “wiggle” towards their destination, rather than smoothly adjusting their trajectory towards it, as is the case with Braitenberg Vehicles and their direct sensory-motor dependencies.

From the paper presented it is not clear why kinesis was not considered in the preliminary designs, but what is known is that the resulting product, LEGO *Mindstorms* do allow designers to define these kind of direct sensory-motor dependencies.

2.2.5 *EXTENDING BRAITENBERG VEHICLES*

Braitenberg's work inspired several projects since its release, which all provided different extensions to his work. Among this work is a set of projects from the field of Evolutionary Robotics, which sets its primary focus on the automation of design process behind the generation of agent control structures and morphology (Harvey et al 1997, Nolfi and Floreano 2000). While this is not the approach taken in this thesis, evolutionary robotics projects have utilized Braitenberg's thought experiment as a resource for architectural inspiration.

Arbib's "Rana Computatrix" (Arbib 2003) focused on visio-motor coordination inspired behind the biological mechanisms in frogs. Beer's (Beer 1990) computational cockroach approached it from a neuroethological perspective while providing the basis for his later work in dynamical systems theory (Beer 1994). Later studies by Cliff and Miller (Cliff and Miller 1995;1996) investigated the co-evolution of pursuit and evasion behaviours using a pair of differential-drive based simulated robots. Their studies used Beer's descriptions of continuous time recurrent neural networks (CTRNN) as a model to implement the agent's behaviour controller and memory. During this study, the authors found that the "random genotypes often created simple but very effective Braitenberg –*Vehicle*–like controller architectures" (Cliff and Miller 1995 p.3).

Floreano and Mondana also studied evolved Braitenberg agents, but instead of using simulated agents, they used Khepera robots in a real environment (Floreano and Mondana 1994). Their experimental setup was initially simple, featuring neither traps nor obstacles (except the walls of the environment) and only a single resource. The agent was evolved to seek a light source using 8 light sensors attached to its body. These sensors were not distributed equally and the robot had 6 in the front and 2 in the back. The light source in the arena was attached above a battery recharging plate.

The evolutionary goal of the agent was to move straight, while avoiding obstacles. The results were interesting in several unexpected ways. For example, evolution displayed a clear adaption to the body shape of the agents, favouring those who move in the direction that had more light-sensors attached. This allowed these agents to see and avoid obstacles better than those who had evolved to move in the other direction. Furthermore the emergence of neurons that signified certain locations in the arena was observed. The activity of these 'place' neurons depended solely on the orientation and position of the robot, but wasn't affected by its battery level.

The group continued their studies with Braitenberg's Vehicle 6 architectures and investigated competitive co-evolution (hunter/prey) and cooperative agents, demonstrating the versatility of the base architecture and powerful adaption through evolution.

In a later study, the group investigated an evolved learning model (Floreano and Urzelai 2000). This system incorporated ideas from Braitenberg's Vehicle 10 architecture, featuring weighted connections between sensor and motor neurons. They allowed the evolutionary model to choose between 4 different learning rules for each synaptic connection. All of these were based on Hebbian learning and were differentiated mainly in the way weight decay was used. The following four learning rules are paraphrased from (Floreano and Urzelai 2000):

1. Plain Hebb: strengthen the synaptic weight in proportion to correlated activity of the pre and post-synaptic neurons.
2. Postsynaptic rule: Like Hebb, but the connection is weakened when only postsynaptic neuron is active
3. Presynaptic rule: Like Hebb, but the connection is weakened when only the presynaptic neuron is active
4. Covariance rule: Synapse is strengthened when activity levels of both connected neurons are similar. If the difference between the two activity levels is less than half their maximum activity, the weight increases. If the difference is larger, the weight decreases.

This system successfully evolved a controller that was capable of learning simple associations and was even capable of executing sequential events. Overall the performance of evolved Vehicles that incorporated adaptive associations was better than that of Vehicles with only fixed sensory-motor connections.

A similar study by Seth on action selection in genetically evolved reactive agents (Seth 1998) incorporated ideas from several of Braitenberg's simpler Vehicle architectures. The virtual agent model comprises a collection of distance sensors measuring the proximity of food, water and trap objects placed in its environment. The model also includes two types of internal batteries, one for food and one for water that can be recharged by approaching food and water objects in the environment. The sensors were directly connected to the wheels, but a genetic encoding scheme determined the non-linear relationship between sensor input and motor output. The level of the batteries acted as a form of fitness function, favouring those agents that kept both batteries filled, while avoiding traps. These relationships were evolved through a series of experiments for 430 generations.

The outcome was that the agents developed sophisticated and efficient reactive *instinctual* behaviour. The evolved non-linear relationships between the sensors and motors were sufficient to produce traits of *action selection*, such as prioritising needs, exhibiting opportunism, persistence in actions versus action dithering all the while being able retain the agent's ability to react to sudden changes in the environment.

One of the key extensions cited in his work is the addition of learning mechanisms. Seth's work shows that Braitenberg's architecture is robust up until Chapter Six, providing additional justification for further study of his more complex architectures.

2.2.6 MAVRIC Extended Braitenberg Architecture

MAVRIC or the "Mobile, Autonomous Vehicle for Research in Intelligent Control" was developed by Mobus and Fisher self-described as "roughly a Type 11 vehicle in Braitenberg's menagerie" (Mobus and Fisher 1994, p.2). For it to satisfy this claim, the architecture needs to be capable of associating and reproducing temporal activation patterns. This is however not presented in the literature. Instead Mobus presents a system that observes changes in synaptic weights at different timescales and uses this statistical information to grant stronger and more persistent associations to stimulus that was repeated over longer time periods. This does not satisfy Braitenberg's Ergotrix wire criteria "to reproduce sequences of activation at the same pace as the original occurrence".

The architecture is comprised of a set of reactive behaviours ('instincts') and BAN, the "Basic Associative Network" (Mobus 1994). BAN features a neural component named "Adaptrons", a type of synapse that does not store a single weight defining the association between elements, but instead a vector of weights that has 'levels' that denote the causality of the association at a given timescale. Each successive timescale is based on the previous shorter one. This way long-term memory is based on an average of short-term memory, which again is an average of real-time stimulus recordings. The number of timescale levels is flexible and can be adjusted.

For association, the weight at each level is "pulled up" if activity in the level below it is raised above the weight of the current level. The difference between the two is calculated, multiplied by a learning rate constant and added to the current weight.

Learning at each level thus takes the shape of a logarithmic increase approaching the upper bound of that level, which is the lower bound of the level below it. In the case of the lowest level, where no lower level exists, the bound is the maximum of the scale.

Decay is handled in a similar manner. The weight is 'pulled down' by the difference to the weight of the next higher level. Thus the weight will approach the upper bound of the next higher level at a logarithmic rate.

The overall effect is that as each successive level is trained, the response curve at the lowest lever (the one closest to real-time) is raised, while the decay is flattened. This means that if a stimulus is persistent over several timescales, subsequent spikes will see their initial weight increase over time and their weight decay reduced.

Mobus' Adaptron model brings the temporal representation from the level of a single neuron using self-excitation, to the level of the synapse. The architecture presented in this thesis instead takes the temporal representation from the level of a single synapse, to a network of synapses. Another key difference to BAN, is that it intends to keep the Mnemotrix (purely associative) and Ergotrix (temporal causality) separate, while Mobus' BAN architecture integrates both notions into a single learning unit. This latter separation might be possible to implement using BAN Adaptrons, but this is not considered or further explored in Mobus' work.

2.2.7 BEAM ROBOTS

Initially inspired by a Rodney Brooks lecture (Brooks 1986) BEAM Robots are a group of reactive robots based on an artificial neural network architecture invented by Mark W. Tilden (Beam 2008). They use a series of pulse delay circuits to mimic the function of real biological neurons.

The BEAM robots follow a similar approach to the early Braitenberg Vehicle designs (Braitenberg 1984) in that they use simple, interlinked behaviours and mostly direct connections between sensors and actuators. Tilden's methodology also seems inspired by Braitenberg in that it states that a robot designer should:

1. Use the lowest number of electronic elements
2. Re-use scrap pieces from other electronics/robots
3. Use radiant energy, such as solar power

Perhaps more interesting is Tilden's stance towards the internal complexity of a robot architecture. (Beam 2008) quotes Tilden "If, for a linear increase in ability, an exponential increase in complexity is required, then start over from scratch". This stance is reflected in one of the early observations made by Braitenberg about his Vehicle 3, about the incorporation of learning into Vehicle 7 versus the predefined rules that are built into that Vehicle: "once knowledge is incorporated, the resulting vehicle may look and behave quite like our Vehicle 3c" (Braitenberg 1984 p.14) .

The BEAM approach does not entail architectures for creating higher level reasoning however, which might be down to the limiting effect of the first principle and the fact that all BEAM robots are implemented in Hard AL (hardware based Artificial life) and would therefore be very difficult to construct if a more complex Neural net were to be designed.

2.3 BELIEVABLE AGENTS RESEARCH GROUPS

Throughout the 90s and early 2000's there were several research groups focusing on the creation of "believable characters". In particular, Bruce Blumberg's Synthetic Characters group at MIT brought forth a wide range of research and publications that influenced the work in the field. Notable alumni from this group include Bruce Blumberg himself who among other projects now works with video game companies on Zoo animal games; Damian Isla, who created the award-winning AI for Halo 3; Jeff Orkin, who introduced AI planners to first person shooters in F.E.A.R's AI. These and other MIT alumni such as Rodney Brooks and Karl Sims before them practically defined the field and associated techniques (subsumption architecture, swarm behaviour etc.). However, more recently the drive of this research area seems to have subsided, with most of the researchers mentioned above having moved into industry. Unfortunately this means that the number of publications relating to "believable characters" has somewhat subsided, even though their role in society, particularly in the games and movie industry is steadily growing.

The following sections describe some of the significant work that was done by other research groups to provide a context for the work set out in this thesis and the role of Braitenberg's "Vehicle" agents in the field of believable characters.

2.3.1 MIT A.L.I.V.E PROJECT

The ALIVE system developed by Pattie Maes (Maes and Darrell et al. 1995) and her colleagues from the Synthetic Characters group at MIT is an immersive virtual reality interface that acts as a "magic mirror" and enabled the user to see himself imposed into a virtual environment that is inhabited by virtual objects and creatures. This allows the user to interact with autonomous artificial life agents via simple gestures and without the need for peripheral control equipment.

The contribution of the A.L.I.V.E project to believable characters was in the interface and not in the characters' behaviour. The software that ran the artificial life characters that were showcased using the immersive "magic mirror" interface was called "Hamsterdam" (Blumberg 1994).

The autonomous agents that inhabit this world use an agent architecture developed by Bruce Blumberg for his PhD thesis (Blumberg 1997). This architecture draws heavily on Ethology (the study of animal behaviour) and focuses on controlling the temporal aspects of behaviour, implementing a hierarchy of behaviours similar to Rodney Brook's subsumption architecture (Brooks 1986) and modelling internal factors such as their needs and motivations. Later implementations of this architecture (Gorniak and Blumberg 2005) also added the ability to train agent behaviours by online sequence learning that incorporates a notion of temporal causality of perceived events, including their timing and rate (Burke and Blumberg 2002). Later work by Fujita on the Sony AIBO project (Fujita 2001) was inspired by this architecture, albeit in the context of robotics and not virtual characters.

2.3.2 DUNCAN SHEEPDOG

Blumberg's temporal prediction algorithm was expanded upon in the "Duncan Sheepdog" project (Isla and Burke et al. 2001). This project successfully implements a system for predicting near future events through the processing of visual sensory data acquired via an artificial retina.

This advanced retinal imaging system is similar to that used by Tu for the virtual fish simulation (Tu and Terzopoulos 1994). The data from the visual system is processed by a "predictor" module, which maps information gathered from visual data to the locations of objects on a two-dimensional topographical map called the "probabilistic occupancy map".

Isla uses this as a discrete representation of the probabilities of the location of objects in the world. This representation comes in the form of a hexagonal grid that is overlaid over a pre-defined model of the environment. Temporal reasoning and prediction is done by calculating the probability of an object occurring in a specific field on the grid.

This work demonstrated that, incorporating a prediction mechanism into a navigation system, can increase the believability of an agent (Isla and Blumberg 2002). The ability to perceive object persistence and to predict movement paths and intercept moving objects is also a robust capability useful for any autonomous agent.

In contrast to our architecture, the "predictor" mechanism described by Braitenberg (1984 p.73) uses a neural approach to representation over the abstract approach employed by Isla and Blumberg . Braitenberg's predictor generates no discrete maps. Being a general temporal pattern predictor, it instead perceives and predicts sequences of sensual data, which train a connectionist network much in the manner that single percepts/concepts are usually trained. This also allows for other forms of prediction other than spatial; as the predictor "does not care" what type of sense provides the sequential/temporal information.

2.3.3 REALISTIC FISH

The artificial life fish created by Demetri Terzopoulos and Xiaoyuan Tu (Terzopoulos and Tu 1994) were designed as "self-animating graphical characters", which aim to realistically simulate the behaviour of various piscine species interacting with one another in a physically simulated, virtual oceanic environment. Terzopoulos' fish had ALife architectures to control the movement of the fish. They used a physics simulation for the muscle-actuated body paired with simulated retinal imaging vision to locomote and detect situations (Terzopoulos and Rabie 1997). The biologically inspired approach is not applied to the decision making process, which is instead controlled by a hierarchical behaviour AI system. Given the limited behaviours required to make a fish seem believable, this is a good choice (since real fish are not generally known for exhibiting complex emotions).

Further work by Terzopoulos (Shao and Terzopoulos 2005) applied a similar architecture to control the locomotion and behaviour of pedestrian crowds in various social settings. The behaviour of the pedestrians is defined by goals to achieve within the environment (e.g. buying a ticket at the train station) and personal needs (tiredness, hunger). Systems for navigation and simple planning allow the pedestrians to move through the environment intelligently, achieving their goals, all the while avoiding obstacles and each other. Although this creates overall quite believable crowd movements, on closer inspection repetition becomes apparent. The limited number of behaviours that each character is able to perform is insufficient for the large number of characters and the variety of behaviours we expect from humans. However, this may not be an issue if the crowd is not the centre of attention and the observer is focused on more complex characters within the environment- either agents or player avatars.

2.3.4 *Oz Project*

The Oz Project run by Joseph Bates attempted to simulate believable characters based on principles from classical animation. The so-called "Woggles" that inhabit the simulation use a goal-directed behaviour-based architecture (Bates 1997) based on those developed at M.I.T by Pattie Maes (Maes and Darrell et al. 1995) and Rodney Brooks (1986).

For deciding which emotion to portray given a certain event, they used the work of Ortony, Collins and Clore (1988). This system provides a very rigid correlation between the current event and a specific emotional reaction, as it does not incorporate a world model or planning (foresight) of any kind. The emotional reactions are also predefined by the designer of the character.

These rigid correlations between events and emotions were later disputed within the project group. During tests with observers, seemingly illogical behaviour caused by an error in the code was perceived as more interesting, and caused more detailed interpretations of the creatures emotional state, than most "correct" behaviours (Bates 1997). In his evaluation of the Oz project Bates observes that, while they did fulfil the first key point (a clearly defined emotional state) they did not incorporate systems for accentuating emotions (providing correct timing and exaggerating the dominant emotions) or revealing the thought process of the Woggles (which would require a system for planning or foresight) (Bates 1997) .

As Loyall states, (Loyall 1997 p.162) subjective judgment by critical observers seems to be the only way to obtain evidence for how believable an agent is. Loyall's team exhibited the 'Woggles' project at the Boston Computer Museum for about a year, having around 30.000 visitors use it in the time. According to Loyall the exhibition was a great success, with users who got involved being highly engaged and spending up to 10 minutes with the simulation.

2.4 Believable Agents in The Entertainment Industry

Since the shift of focus of the MIT research groups from interactive believable characters to user interfaces and more general AI, the major developments in the field are currently largely found in the movie and video games industry. Unfortunately this has resulted in a decrease in publications in the field, since the technology in current games is largely under non-disclosure.

The first projects discussed in this section give an overview of some of the animation tools available or currently in development that aid animators in creating believable agents. Interviews and post-mortems with selected developers have provided some useful insight into the architectures driving modern interactive game characters. This section discusses several contributors that set the current standard for believable interactive characters in video games.

2.4.1 Massive

An example of a tool that addresses the problem of animation diversity in crowds successfully in the film domain is the software “Massive” by Stephen Regelous (2009). “Massive” was used to create the Oscar winning effects of battling armies in the “Lord of the Rings” trilogy and was innovative in that it gave each individual character its own perception and behaviour controller. Each soldier would thus “fight for himself” and depending on the “perceived” situation could flee, attack or defend. The result was a dynamic battlefield, with individuals and groups of soldiers fighting, shifting positions and defending each other, creating chaotic and believable mayhem.

2.4.2 Natural Motion

Natural Motion as a company has created a collection of software tools utilizing a biomimemic approach to body simulation similar to the muscle-actuated body found in the artificial fish (Tu and Terzopoulos 1994) described above. They refer to this technique as Dynamic Motion Synthesis (Naturalmotion 2005). Using a biology-inspired muscle-actuated physical model coupled with a neural network controller they provide a series of behaviours such as jumping, walking or grabbing hold of objects. Simulated characters can perform these actions dynamically and blend between them, all the while reacting to physical interference and user control.

2.4.3 ACTOR MACHINE

Ken Perlin's procedural character animation toolkit (Perlin 2009) is focused on providing a flexible interface for artists to animate characters that express emotional states and animate in a believable manner. The approach is similar to the one natural motion applied (motion synthesis), but goes further in that it also aims to include facial expressions and not just locomotion. It is also related to the work by Ortony, Collins and Clore (1988) in that it aims to define a set of emotion "primitives" - modifiers that can be mixed to form any emotion. In terms providing an interface for these emotion modifiers it is similar to the facial expression creator that supports Half-Life 2 (Valve 2004), which was itself based on Paul Ekman's FACS and his Action Units (Ekman and Rosenberg 1998).

Just like (Loyall 1997), Perlin provides a toolkit for animators and does not aim to create autonomous agents in the sense that they act and choose their behaviour autonomously.

2.4.4 CREATURES

The artificial life simulation *Creatures* (Grand and Cyberlife Technology 1997) was critically acclaimed as a big advance in artificial life simulation. The game-play was based on the breeding and nurturing of artificial agents called "Norns". These Norns had a complex sensory system and employed neural network training as a learning device. The Norns are able to learn through the interaction with the player. Right and wrong behaviours could be taught using punishment and reward. A natural language system provides the ability for the Norns to communicate with the player. During their "childhood" the Norns will often point at objects for the player to define for them. Over time, the Norn will use these definitions to communicate its needs to the player.

The Norns are sophisticated and believable artificial life agents, but the simple animation system used at the time that was incapable of displaying more subtle changes in their internal state. The Norns can express raw emotions such as fear, love and hate via their animations and the language system, but less definite emotions and thought processes are lost in translation.

2.4.5 THE SIMS

"The Sims" (Wright and Maxis Software 2000) is a game based around the simulation of human social interaction, The Sims allows players to create and manage a virtual dolls house and its inhabitants, the so-called Sims. These semi-autonomous agents (the player can give orders to the Sims to satisfy specific actions) keep an internal model of their needs and a list of actions needed to perform those needs. The Sims agents have the ability to interact with every object in their environment. This is achieved by using *smart objects*, a system whereby the object that is interacted with and not the agent itself carries the knowledge of how the interaction between them will play out. To simulate communication between separate agents and to express their emotions the Sims were given their own simplified language called "Simlish". This, in combination with animations that changed according to the Sim's emotional and physical state created some of the most believable virtual characters to date. Apart from statistics about their "life" such as their age or their affinity to particular other Sim characters the Sims did not incorporate any world model or knowledge of events in their past. They were also not able to "learn" as such, as their capabilities were measured only by their efficiency of receiving satisfaction from using given smart objects.

2.4.6 FAÇADE

Façade is a game that developed from Michael Mateas' experiences with the Oz project (Mateas 1997) and a collaboration with Andrew Stern, who had previously worked on the virtual pets Catz and Dogz (Stern and P. F. Magic 1995). The game takes the form of an interactive drama that lets users interact with two characters using a natural language parsing system and a few simple actions such as hugging, comforting or kissing. The narrative plays out as a drama involving a young married couple, Trip and Grace, and an old friend visiting them- the player. The game centres around the conversations that play out between the characters and the player. The non-player characters respond to everything the player does or says with an appropriate emotional reaction. This reaction is expressed through a facial animation system similar to FACS (Ekman and Rosenberg 1998) and body language (such as crossing their arms in defence, or turning their back in dismay).

The display of emotions through facial expressions and body-language puts façade above any purely text-based conversational agent interface such as ELIZA (Weizenbaum 1966) and SHRDLU[1] (Winograd 1980). It subtly encourages the player to "play along" with the conversation, making it hard for him to feel indifferent to a

[1] SHRDLU comes from the arrangement of letters on linotype typesetting machines, where letters were arranged in columns according to their frequency in the English language. ETAOIN was the first column, SHRDLU the second.

failed response. For example failing to recognize a player's input will cause Trip and Grace to "feel puzzled", which can be easily seen in their expression and stance. Depending on the context of the situation this can cause them to visibly darken or lighten their mood- making the player feel responsible for his action, even if it was misinterpreted. The interactive narrative also spans a wide range of socially familiar topics, such as love, sex, art, career and interior design –making it easy for the player to feel involved.

Apart from the dynamic facial expressions, Façade does not use procedural animation to animate the characters. It also does not use any form of sensing, other than keeping an awareness of the current actions of the player and non-player characters. It is essentially a conversation agent with an added body-language interface. The narrative takes the form of a pre-defined branching storyline that is affected by the emotional states of its main characters. An AI director uses the agent data to determine the flow of the scene and one of several outcomes to the narrative. All the content in the game, including voice acting performances, writing and basic animation is predefined. Therefore, while the structure is emergent, the narratives' content is not. The agents do keep track of their emotional internal states, but are not autonomous in the sense that they act autonomously and sense their surroundings. Nevertheless the model of a branching narrative, while being highly work-intensive for a developer, is an intriguing proposition for the implementation of believable characters in the context of interactive animation.

2.4.7 Grand Theft Auto IV

Grand Theft Auto 4 (Rockstar Games 2008) was the first commercial blockbuster to use Natural motions' (Naturalmotion 2005; Champandard 2008) procedural animation system on a large scale. Every agent inhabiting its simulated city was animated using this system. This allowed for an unprecedented level of interaction with the player. Previously shooting a character or hitting them with a car would trigger a canned animation regardless of the force behind the action. Now a slight bump with the car will see a character merely stumbling, while slightly faster collision will see them landing on the bonnet and holding on for their dear life. A full speed collision will see them flying and even trying to cover their face upon landing. It's gruesome, yes, but highly interactive. The same goes for almost every other physical interaction with characters in the game.

GTA4 is a great example for procedural animation applied to locomotion and physical effects. But the animation system required much tweaking by artists to avoid characters performing unnaturally. While the animation uses procedural algorithms, the behaviour controller of the characters uses a standard behaviour tree to determine activities. Grand Theft Auto's characters may individually not be very believable but the developer has taken great care in designing interactions between them that make the city seem "alive". These interactions include: driving badly, causing a collision with another character's car and subsequently getting into a fight. If someone is injured the ambulance will be alerted. Some characters may commit crimes, causing the police to chase after them. Although all these agent interactions are based around a simple state machine and randomized events – the dynamics that emerge form a coherent image of a modern, egotistic and gritty city – making the city, together with its population, one of the most believable characters in a game to date.

2.4.8 SPORE

Spore (Wright 2008) allows the player to build his own alien species out of large collection of predefined body parts (which all give the alien specific "abilities") and a mouldable Play-Doh like body structure. Depending on what type body-parts (sensors and actuators) the player chooses, the creature will dynamically animate, while taking the moulded body shape into account. Usually this ends up with an endearing wobbly gait and while literally hundreds of variations are possible, the results are not as creative as those displayed by the likes of evolved morphologies by Karl Sims (Sims 1994). The algorithm that is used to animate the behaviour of the creatures does not evolve over time, nor is it trained specifically to each "species" the player creates. This makes the result less believable than for example the muscle actuated biomechanical model used by (Tu and Terzopoulos 1994). The result is saved by abstraction though, in that the body parts and body shapes the player can create have a very comic-like appeal to them, making the springy, wobbly animation consistent within the context of the virtual environment.

2.4.9 HALF-LIFE 2

Half-Life 2 (Valve 2004) uses mostly motion-capture data, inverse kinematics and animation blending techniques to animate its characters (Eldawy 2006) . Yet for facial expressions an interesting approach directly based on FACS (Ekman and Rosenberg 1998) was employed. Presented as a tool to developers for the "Source" engine, which form the technical foundation of the game, the "Face Poser" allows artists to lip-sync characters to recorded speech and apply matching emotional phonemes, also described as action units by Ekman, to the animation. Emotional expressions can also be applied to other in-game animations, such as being shot, or firing a gun.

This tool allowed the designers of the game to reach a hitherto unprecedented quality of emotional expression in its non-player characters. The AI system that controls the character's behaviour however, does not feature any emotional modelling- meaning that every expression of emotion is pre-scripted and non-autonomous.

2.4.10 Total War Series

According to Richard Bull (Hardwidge 2009), an AI developer for Creative Assembly, the AI in *Empire: Total War* (Creative Assembly 2009) uses a planner that draws from a repository of tactics and objectives to form strategies for the battling armies it controls. On the single-agent (combatant) level it uses an agent perception model similar to the system employed by Massive (Regelous 2009). Each individual agent uses several factors to decide which of several pre-defined actions to perform. Among others, these factors include the proximity of nearby friends and enemies and their current health.

This system results in some believable actions performed by the fighting characters – with combatants seemingly aware of their surroundings and attacking/fleeing when appropriate. The game uses pre-defined animation sequences to display the characters' actions. Yet the result of using an agent perception model ensures that not all agents trigger their animations at the same time. Another system is used to introduce delays into idle animation cycles, such as that a standing army is not swaying in sync or reloading in sync.

While this ensures a seemingly realistic chaotic battlefield experience, on closer inspection the pre-canned animations are apparent. In addition to that the character models of each unit type are not highly characterized. This does not affect the experience when viewed from a birds-eye perspective like the one usually seen by the player, but it does hurt the suspension of disbelief when scrutinized from up-close.

2.4.11 The Companion Trilogy

Fumito Ueda's companion trilogy started with the adventure game *ICO* (Ueda 2001). Its game-play is based on exploring an obstacle-laden environment while protecting a second, AI controlled character. Central to the experience was the intricacy of the handcrafted animation that permeated every action performed by the player-controlled protagonist Ico and his non-player character companion Yorda.

Ueda started his career as an animator and not a programmer and he was thus very familiar with the guidelines for believability from the character arts (Thomas and Johnston 1981; Jones 1989; Loyall 1997; Egri 2004). *ICO* was inspired by a number of successful games that prioritised expressive animation over responsive controls, such as "Prince of Persia", "Another World" and "Flashback" (Mechner 1989; Delphine 1991; Delphine 1992). The game was released in 2001 in North America. Critically well received (ICO has an average critic's score of 90 as of 01.12.2010 according to Metacritic), it was initially not highly commercially successful. However, the game garnered more attention over the years since its release and was re-released to strong sales in 2006. Sony recently released a remake of the game in the form of a compilation, together with its sequel "Shadow of the Colossus" (Ueda 2005), for their PlayStation 3 video game console.

The animation includes nuances such as visible anticipation before swinging a wooden sword or jumping across a ledge. Momentum is carried through each animation as Ico jumps and bounds across platforms and swings of ropes. Yet the single most effective animation mechanic is displayed when Ico holds Yorda's hand. Her gait displays her detachedness from the situation, making her seem almost ethereal. Ico has to literally tug her along when she is scared and she will refuse to jump with him over gaps - instead requiring an extra run-up, re-assuring words and a catch by Ico. These interactions are displayed with such vivid attention to detail that they allow Ueda to inspire an entire story in the viewer's mind, even without the characters being able to communicate verbally (in the story, they don't speak the same language). The second game in the Companion trilogy, *Shadow of the Colossus* (Ueda 2005) featured the same attention to detail in the animation as before (and the same sluggish controls).

Reviewers often referred to the power of the animation in *ICO* and *Shadow of the Colossus (Herold 2001).* The animations in Ueda's games were all hand-crafted, no motion capture performances were used for the main characters (GIA and Ueda 2001). In *Shadow of the Colossus* these animations were coupled to an advance dynamic collision detection system that would handle the myriad of positions that character could get into while clinging to a Colossi's body.

The tight focus on only a few interaction mechanics and a low level of autonomy for non-player characters allowed the artist to pay close attention to every detail in the animation and to evoke a strong sense of personality and believability. Therefore I would count this series as a benchmark for what creators of believable and autonomous interactive characters should aim to achieve.

2.4.12 Assassins' Creed

Using an advanced set of motion blending techniques similar to the collision engine used in *Shadow of the Colossus, Assassins' Creed* protagonist Altair was able to bound over rooftops and climb up even the most complex surfaces realistically. Unlike animation engines that came before, this engine ensured accurate collision detection – that hands and feet really met the surfaces the character was climbing up on (Autodesk 2008) .

The system is highly responsive and dynamic, yet it does not convey a sense of emotion. Altair seems strong and super-humanly agile, but his movements do not convey a sense of personality or plot as the animation in *ICO* or *Shadow of the Colossus* did (Herold 2001).

The game also featured a large amount of non-player characters (NPC) that inhabited the simulated medieval cities. Different groups of inhabitants, such as beggars, salesmen, guards, noblemen and preachers create a bustling atmosphere. They also react to the actions of the protagonist. For example sprinting through a crowd and climbing up onto a roof might cause the surprised and angry exclamations of nearby bystanders, a fight will cause panic in the crowd. The system was impressive in terms of scale; the city streets are filled, with often more than 20 NPC present on screen at a time. The illusion of life was broken however, by repeated assets. While the comments of onlookers seem spontaneous and impressive at first, they are repeated very often and only use a handful voice recordings. The character models and animations are also re-used. In a crowd one is likely to spot an unnerving amount of identical twins.

Assassins Creed was ambitious for its time, but clearly showed the limitations of an asset-based approach to believability. No matter how many character models and voice samples are prepared, in the 10+ hours of playtime that are commonly spent in these environments, they are sure to repeat, thus breaking the illusion.

2.4.13 *INTERACTIVE ANIMATION*

Non-interactive believable characters have been largely realized by the likes of Disney, Pixar and DreamWorks animation films such as “Finding Nemo”, “Wall-E” (Pixar 2003; 2006) and “How to Train your Dragon” (Dreamworks 2010). While many of the techniques used to animate these films are applicable to characters in interactive scenarios, such as video games, the technique used most commonly in interactive simulations is creating pre-set non-interactive animations that are triggered by an interactive program.

The difference between a *pre-set* and a *procedural* approach to animation is analogous to the difference between *top-down* and *bottom-up* agent development. Just like the pre-set approach to animation composes behaviour from a given set of unchangeable, hand-made animation sequences, top-down agent development usually sees the agent pick from a list of pre-defined atomic behaviours. In the past this approach had the shortcoming of being unable to adapt to sudden changes. Once a particular animation/behaviour had been started, it could not be interrupted. The procedural approach to animation on the other hand uses a set of cooperating simulations to create the final animation. This final animation is not changed directly, but can be influenced by adjusting a set of parameters. For example, natural motion’s “Euphoria” engine (Naturalmotion 2005) uses physics, sensory system and muscle control simulations to animate its characters. While the benefit of procedural/bottom-up techniques is their reactiveness and adaptability, not being able to modify the final output directly and instead having to find suitable parameter settings for the simulations can be difficult.

However, modern approaches to character animation in games often see a mixture of both pre-set and procedural animation approaches. Popular among these techniques is animation blending, which uses pre-set animation sequences, but is able to seamlessly blend and transition between several of these to produce a wider set of animations. The first commercial video game to use this approach was Assassins Creed (Ubisoft 2007), which used Autodesk’s MotionBuilder and HumanIK middleware (Autodesk 2008).

Procedural Animation and Bottom-up Agent Development

"Procedural animation" as opposed to scripted or "canned" animation is animation that is based on algorithms rather than being purely reliant on an artist's expression. This section gives an overview and presents a timeline of the developments in this field.

As discussed in 2.4.2 and 2.4.3 there have been several advances in the field of procedural animation such as the technology demonstrated by Natural Motion (Naturalmotion 2005) and Ken Perlin (Perlin 2009).

Procedural animation has been demonstrated in several high-budget cinema productions. For example, the software that later became known as "Massive" (Regelous 2009) was created to realize the fighting armies in the "Lord of the Rings" films and solved the problem of having to hand-animate thousands upon thousands of soldiers by using a procedural, AI-based approach.

Massive's approach is to give each virtual agent/actor a fixed set of basic behaviours and animations. Which behaviour should be performed, given a current situation, is decided by an AI system that gives each agent its own senses, such as proximity of friends and foes, or its own "health".

The system is very much inspired by the flocking algorithms used by (Reynolds 1987) and the reactive real-time AI systems found in many current video games. However, in this case the animations themselves, the motion and the performance of each agent is not algorithmically animated, but is instead a pre-recorded animation created using motion-capture of actors.

Briefly introduced in section 2.4.2, Natural Motion advanced algorithmic character animation using a system they call "dynamic motion synthesis" (Naturalmotion 2005). This takes the aspect that animators usually cover – the expression of a given behaviour – and tries to find a set of algorithms that do the same as the animator. In addition, the system allows for simple behaviour goals and desired stances to be set, such as "protect your head" or "stand upright". The virtual character will then react appropriately to interactions i.e. when it falls or is pushed. Unlike hand-made or pre-recorded motion capture animations, these are fully dynamic and can be influenced by physical effects and interrupted at any time. Natural Motion's approach primarily aims to create realistic behaviours. However, realism should not to be confused with expressiveness and believability (see the previously mentioned criteria for believable characters at the beginning of this chapter).

Natural motion has taken some of the work that animators previously did, but instead of having pre-fixed animations they have dynamic, so-called procedural animation. The AI system used by Massive, although well suited to movies, is not yet fast enough for real-time simulations on current generation gaming hardware.

Ken Perlin's "actor machine" (Perlin 2009), which was introduced in section 2.4.3, presents an approach more sensitive to believability criteria, in that it supports mechanisms for over-exaggerating movements and animating non-human characters. However, unlike Massive's and Natural Motion's systems, "actor machine" is currently focused purely on authoring character animations and does not include tools to control the behaviour selection of characters.

Video games such as Grand Theft Auto IV (GTA IV) (Rockstar Games 2008) provide a good example of real-time procedural interactive character animation. GTA IV uses Natural Motion's Euphoria animation engine (Naturalmotion 2005) and has a fully interactive real-time AI for each of the characters that use the animation system. The result is that often surprising and emergent situations can arise. For example, the player might accidentally push another character, who stumbles (euphoria calculating the motion) onto the street, causing an approaching car to evade (AI system) and crash, which in turn alerts the nearby police.

From these examples we see that the greatest benefits in terms of believability of procedural animation over classical animation approaches are that it fulfils points 2 and 3 of Loyall's "Illusion of Life" criteria: Parallel Action and Reactive/Responsiveness. GTA IV also shows how it can lead to emergent scenarios.

Both of these points would require an infeasible amount of artistic work by animators, if procedural animation were not employed. While non-interactive domains such as film might still make a "hand-crafted" approach possible, modern interactive domains such as videogames require the flexibility and emergence possible only with procedural approaches.

2.4.14 Emotions in Game Characters – Future perspective

In his talk "Emotion in Game Characters" at the Paris Game AI Conference 2009, Phil Carlisle from the University of Bolton described how different models of emotions and character such as OCC (Ortony, Collins and Clore 1988), the five personality factor model OCEAN (which stands for Openness, Conscientiousness, Extraversion, Agreeableness and Neuroticism) (Costa, McCrae et al. 1991) and others can be implemented into games using behaviour trees. One of the projects cited was the Oz project (Loyall 1997; Mateas 1997).

A good example of what future emotional characters in games might look like is the "Milo" prototype that was presented by Peter Molyneux and Lionhead Studios at TEDGlobal in August 2010. This prototype simulates a virtual young boy and allows a player to interact with the character via voice commands and a sophisticated 3D camera to track the user's movements. At certain key decision points Milo asks the user to make a (often moral) choice. Molyneux explains how these experiences will shape the personality of Milo and affect his autonomous decisions during later events. The internals of the AI are not discussed, but from the decision points demonstrated in the prototype, such as killing a snail, it seems that the simulation is constructing a psychological profile similar to the ones mentioned by Carlisle. The prototype also shows that Milo can hold conversations with the user about previous key experiences in the game, opening up the possibility for revising the moral opinion Milo has about them. The conversation feature also ties into the aspect of Milo's AI that Molyneux mentions, that Milo's "mind is based in the cloud". This means that as Milo learns new words and meanings from users, these are stored in a shared online database. This way, Molyneux proposes, Milo's AI will learn from its global community of users over time.

As Carlisle stated, the most common approaches to emotion within AI systems in games are rather rigid behaviour trees implementations. While these can be quite powerful when they use a psychological model such as OCC or OCEAN, more dynamic or adaptive systems, such as planners or neural models are currently not used for the implementation of character emotion.

2.5 Robotics

The previous section reviewed the current state of the art for screen-based believability in entertainment. Believability is not just an issue on the screen but also with mechanical robots and toys. Although this is a huge topic in itself, much of the knowledge is transferable between screen-based characters and robots.

First of all robots are embedded in real life – meaning they suffer from mechanical issues such as material flaws. They also suffer from noisy sensors and imprecise actuators, which is not such a big problem in simulated environments. Secondly, reality has perfect fidelity; there can be nothing more realistic than real-life after all. It has real physics and dynamics – It is like the perfect simulation in that sense. Thirdly developers cannot cheat. Most AI used in virtual environments uses shortcuts and cheats for processor intensive tasks such as vision and navigation. Usually they do not use senses and 3Dimensional Perception systems such as the eyes of the character. Navigation is instead often handled using waypoints that are pre-calculated or set by a designer.

Cheating is allowed and desired in these kinds of simulations in order to save on valuable processing resources. This is often called the "smoke and mirrors" approach to AI (Hardwidge 2009). The trade-off in losing "realism" is usually justified by the claim that one can achieve almost the "same" behaviour using an efficient cheat, when compared to a more realistic simulation.

2.5.1 Honda Asimo

The Honda Asimo (**A**dvanced **S**tep in **I**nnovative **Mo**bility) robotics project is famous for having been at the forefront of locomotion and balance control in robotics (Hirose and Takenaka 2001) . The system controlling Asimo's behaviour is a behaviour based state machine (Sakagami, Watanabe et al. 2002). In December 2002 Honda presented a new version of Asimo that was more than a study of biped locomotion. By then Asimo was able to move around sufficiently and so the developers began incorporating "Intelligence technology". The main features they implemented into the 2002 Asimo were:

1. Recognition of moving objects
2. Posture/Gesture recognition
3. Environment Recognition
4. Sound Recognition
5. Face Recognition
6. Network integration (user network and internet access)

While the human-like gait of the robot can seem uncanny, the design of the robot itself was optimised to avoid the uncanny valley. The designers at Honda took cues from sports car design, rather than trying to imitate a human being (Koike and Koshiishi, 2001). Asimo is relatively small (130cm tall) and does not have facial features or other human-like means of expressing its emotion.

The algorithms used for locomotion inspired virtual gait systems such as "natural motion" (Naturalmotion 2005), although the latter worked with a simplified virtual physics model. Asimo is a good example of how advances in Robotics can be applied to the virtual domain and entertainment in a simplified form.

2.5.2 Sony AIBO

Released in 1999, the now discontinued AIBO series (Sony Global 2005) of personal entertainment (toy) robots was one of the first commercial artificial-life inspired robots to be released (Sony Global 1999). Modelled after a puppy dog, AIBO has a recharging station that it is able to autonomously seek out. It features camera eyes and touch sensors that enabled it to perceive its environment and detect objects. Later AIBO models contained a Wi-Fi module that enabled wireless communication between groups of AIBOs (Daman 2008) and a user's computer.

AIBO was very popular as a standard platform for AI competitions such as the Sony-organized "Sony four legged-league" founded in 1998 and held at the yearly international RoboCup competition. In 2007, Sony's league was renamed the "standard platform league", since the AIBO was discontinued in 2005 and new robot models (the bipedal Aldebaran Nao) have since replaced it as the standard platform.

AIBO used a novel tree-structure architecture that at its core resembled a layered multi-agent system with 3 agents cooperating and competing to create complex behaviour (Fujita 2001). The three agents are a *target behaviour generator* an *action sequence generator* and a *motor command generator*. The top layer *target behaviour generator* forms the deliberative part of the system. It uses external events and keeps track of internal states, which it uses to generate behaviours. It then gives commands such as "follow the ball" or "find charging station" to the *action sequence generator*. This second layer then creates a list of actions to be performed by the *motor command generator*. As Fujita describes, the action sequence generator "resolves mechanical constrains or posture transitions, so that the upper layer does not need to consider the mechanical configurations" (Fujita 2001 p.785).

In addition, every component of the robot has a separate controller that keeps track of individual motivations. For example tracking visual or aural stimuli while keeping the head horizontal are handled by the "head" controller. The same system is used for the tail and legs.

The final part to AIBO's behaviour is the artificial emotional and instinctual model. The emotional model was designed to alter current behaviour according to the basic emotions "joy" and "anger". This means that depending on its internal emotional "state", AIBO will perform different actions, even if the sensor inputs stay the same. For the extremes of "joy" and "anger" additional behaviours are triggered such as laughing happily, or stomping the ground in anger. Fujita refers to factors such as hunger, tiredness and curiosity as instincts. These are modelled the same way as emotions; meaning that they affect any currently performed action and possess extremes at which they trigger a given associated behaviour (for example sleeping when too tired, or ignoring orders when too curious about something else).

All these motivations are embedded in a hierarchical tree structure. Some controllers subsume others in a fixed hierarchy of priorities, while others, like the main 3 agents described above, cooperate to form overall behaviour.

Fujita compares this architecture to Blumberg's Hamsterdam (Blumberg 1997) architecture, yet notes that the key difference is that a virtual dog like Blumberg's does not have to cope with misrecognized external stimuli or mechanical control issues. In terms of the behaviour model he cites a difference in the use of tree structures in the architecture. While Blumberg uses this branching hierarchy to organize behaviours into ethologically inspired classes, the Sony AIBO model is a multi-agent system.

The AIBO development team had an interesting approach to finding a measure of success for their robot designs. As Fujita states "there is not a good evaluation method for life-like appearance" (Fujita 2001 p.781). Running subjective evaluations of life-likeness with numerous users under strict controlled observation situations is possible, but for development, this is not practical. Instead, the AIBO development team chose not to "concentrate on the details of motions but rather on the mechanism of their generation" (Fujita 2001 p.781). Thereby the AIBO team was able to "reformulate this problem [of maximizing lifelike appearance] as maximizing the complexity of responses and movements" (Fujita 2001 p.781).

The project presented in this thesis follows a similar rapid prototyping approach to Fujita's and its focus also lies on the mechanisms of behaviour generation, rather than the details of how individual actions are performed, as would be the case in an animation-focused approach.

2.5.3 *Social Robotics*

An area of robotics which has had much exposure in the media in recent years are robots that are used in medical treatment of dementia patients (Chang 2013) and as aids for communicating with autistic children (Lee 2012).

The Paro robot is a robot built to emulate the haptics and behaviour of a baby seal. It moves and reacts to touch, making it look like it "enjoys" being stroked. The trials using this robot showed that the participants would build a sense of affection, ownership and even responsibility for the robot in a similar way that they would to a living pet.

Robot assistants have also proven surprisingly useful in engaging highly autistic children. Their simplified facial expressions proving easier to identify and more consistent than other humans', the children find it easier to engage with them without getting confused or distracted. Lee et al (2012) investigated the features of a robot model ifbot would make the children engage in a number of ways. Their research showed that the children found it easier to make eye contact, responded to more verbal cues and were able to identify the emotional facial expressions better, when engaging with a robot, rather than another human.

Currently the research above is focused on medical and social care applications. As robots find their way into more application scenarios they will be interacting with an even more diverse set of people. Social robotics research will have to extend to incorporate broader notions of socially acceptable or desirable behaviour, to find out how robotic assistants of all kind will integrate with and assist our society and culture.

2.6 Literature Review Conclusion

This literature review started by reviewing the history of the craft of creating believable characters and encouraging suspension of disbelief in an audience. Section 2.1 showed how guidelines had helped actors, authors, animators and eventually programmers to create the illusion of life. It discussed attempts to formalise these guidelines into a set of testable metrics that could be used to measure believability in virtual characters and how these could be incorporated into methods for evaluating the user experience in games.

The metrics by Loyall (1997), Mateas (1997) and Gomes (2013) are used to evaluate the agents models presented in this thesis. The empirical evaluation method is based on Tychsen (2008) for his use of metrics in a user study and Tullis and Albert (2013) for the general experimental design. The data analysis of user feedback uses qualitative content analysis as described in Bryman (2008). In addition, objective gameplay data is correlated to user responses as suggested by Thompson (2007) and Tychsen (2008).

Sections 2.2 - 2.4 present an overview of existing believable agent models. Since the crafting of believable characters is not limited to a specific medium, the projects discussed range from early robotics to modern computer games and films. After an introduction to the early attempts of producing life-like behaviour in robotics, section 2.2 introduces the work of cyberneticist Valtentino Braitenberg (1984), who's "Synthetic Psychology" approach to designing and building machines with believable characteristic behaviour forms the inspiration for the virtual agent development approach presented in this thesis. Subsequent modern projects discussed in sections 2.2 – 2.4 are compared to his approach to highlight why a virtual agent development method based on its principles could provide novel insights and ultimately produce more believable behaviour in interactive virtual agents.

Section 2.3 reviews the models and methods of the most significant academic research groups that contributed to the field of believable agents to date, while section 2.4 highlights work from industry that had a significant impact. These two sections also provide the context within which this thesis should be seen.

The final section 2.4 closes the loop by pointing back at the beginnings of believable autonomous agent research, which began in robotics. It reviews current projects in the field that focus on producing believable behaviour. In particular the Sony AIBO project (Fujita 2001) is highlighted, since it employed many of the methods that researchers working with virtual agents had since developed. It is also included to demonstrate how believable agent research now spans multiple disciplines that co-inform each other with methods and insights.

This review contextualises believable agent research and this thesis as a multi-disciplinary endeavour that requires its researchers to draw from usually disparate fields including theatre, classical animation, robotics, neuroscience, psychology, human-computer interaction and game design.

The following chapter discusses how elements from all of these fields informed the experimental methodology and the virtual agent model presented in this thesis.

3 METHODOLOGY

Braitenberg's thought experiments (Braitenberg 1984) suggest that a small set of underlying impulses operating in tandem can create behaviour that is believable by exhibiting consistent, yet complex and identifiable ("personal") characteristics. This project aims to investigate the effects that using a biologically inspired ALife architecture to control behaviour has on the believability of synthetic characters.

This chapter describes the methods used to adapt Braitenberg's "Synthetic Psychology" approach to design and implement gaming agents based on his "Vehicle" designs. It then details methods for analysing these agents in a gaming context. The process is also illustrated in Figure 2 below:

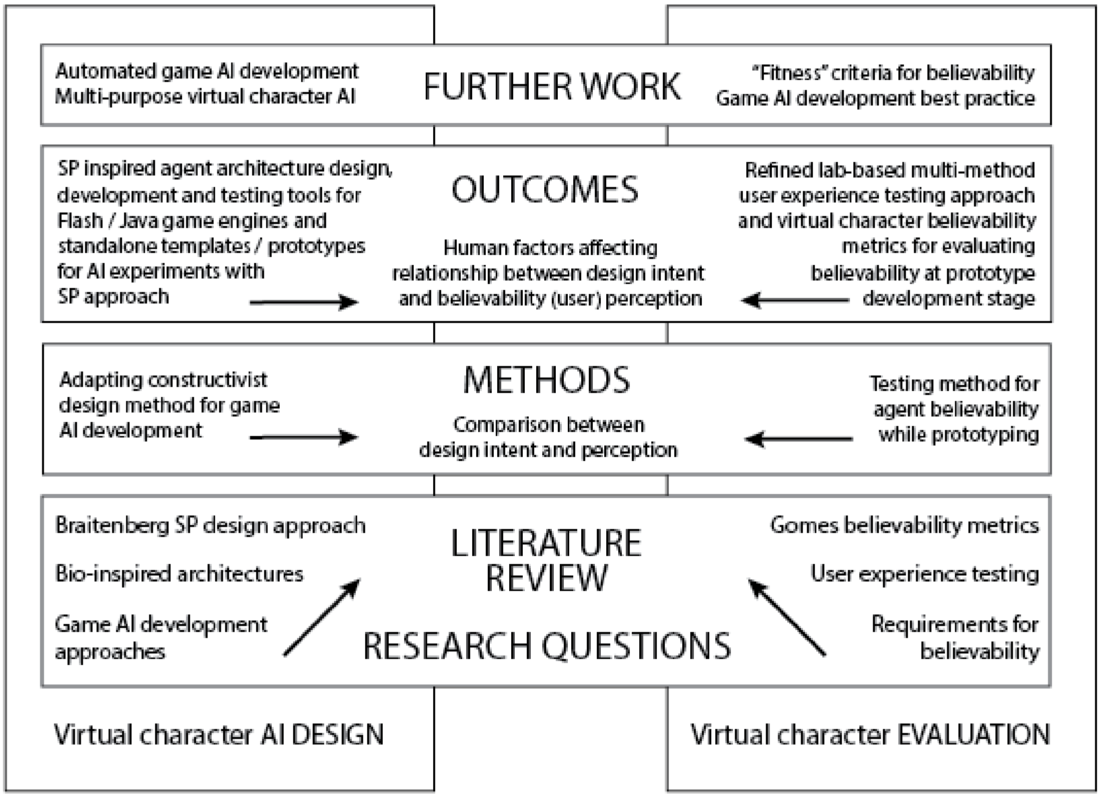


Figure 2 Research Methodology

Figure 1 illustrates that this thesis is a diptych that draws from and combines methods from two research fields in order to address the research questions stated in Chapter 1.5. Starting with the literature review, the "Virtual character AI design" field encompasses the academic research and industry practices concerned with the design and development of autonomous agent architectures. The review of this field informed and inspired the agent architectures and experimental tools developed and tested in this thesis. “Virtual character evaluation” categorizes academic research and creative industry practices concerned with establishing design guides and evaluation methods that aid in the creation *believable characters*. This literature review drew on a wide range of inter-disciplinary sources. This included HCI research, classical animation, performance arts and current research on establishing metrics for testing the believability in virtual characters. While the research questions guided the literature review, they were also informed and refined by its findings.

Drawing from both fields, the method of this research adapts agent architectures from the Alife field and tests them using an empirical lab-based user study for evaluating the believability of virtual agents. At the intersection between both fields the intentions and design decisions made during the development of the agents are compared to the interpretations of agent behaviour made by the participants during the user study.

The outcomes of this research contribute to both virtual character AI design and evaluation practices. The adapted agent architecture and research tools developed for the study allow other researchers or game developers to construct and test Synthetic Psychology inspired virtual agents. Feedback from the participants during the user study was used to extend and update the believability evaluation methods employed, contributing to the virtual character evaluation field. Combining findings from both sides, human factors affecting the relationship between agent design intentions and user interpretations of agent behaviour were identified.

The outcomes also aim to inform further work in both fields. The agent architectures tested in this thesis and the development tools used to design agent behaviour are not intended to be (game) genre specific. Though the majority of prototypes in this thesis are 2D Flash implementations, an additional 3D engine prototype demonstrates the architecture’s viability in a more complex environment with realistic physics simulation. The findings from the user study and the exploration of human factors regarding the conflict between agent behaviour design intent and user perception aim to inform further work in game design best practice, by providing tractable design criteria for believable agents. The current results aim to aid developers in constructing more believable virtual agents. However, they are also a step toward a formal set of user perception metrics that could be used to facilitate automated agent development.

3.1 AGENT ARCHITECTURE DESIGN PROCESS

This research provides insight into the requirements for creating believable agents in virtual environments and additionally provides a tangible evaluation of the theories of Valentino Braitenberg. This first part of the study aims to:

1. Create a behaviour control architecture based on key concepts in Braitenberg's robotic vehicle thought experiments.
2. Synthesize agents suitable to be embedded in a gaming context

The gaming agents designed in step 2 provide the platform for a qualitative user experiment designed to elicit the human factors affecting the "believability" virtual agents as perceived by the users. Part of this analysis uses a questionnaire based on the requirements for interactive believable characters as set out by Loyall, Mateas and Gomes (Loyal 1997; Mateas 1997; Gomes 2013). This will inform the central idea that the design process itself can be enriched by adopting the Synthetic Psychology approach to NPC creation in games.

3.1.1 REPRODUCIBILITY

The models specified in adaption of Braitenberg's architectures are documented as a set of formulae that are platform agnostic and not reliant on any particular implementation. In addition research tools in the form of a set of simulation environments were developed and utilised for the user experiments. The libraries and software used for experimentation are included in the appendix of this thesis.

3.1.2 DESIGN PROCESS WORKFLOW

The research methodology is inspired by two main approaches. A prototype based robotics development approach as detailed in Fujita (Fujita 2001) and the sequential and iterative approach presented by Braitenberg (Braitenberg 1984). A mixture of preliminary structured system analysis and rapid prototyping is used during the development phase. The development phase follows the constructionist prototyping approach suggested by Braitenberg's thought experiment (Braitenberg 1984 p.1).

After designing each model, a simulation is constructed. The simulation is run and log data is stored and visualised prior to analysis. Based on this analysis, the logic of the model was updated as necessary, changing the way components communicate or in some cases, returning to the literature to re-examine core concepts.

A large part of the work involved a cycle of ‘tuning the parameters of the model, such as motor speeds, activation thresholds of devices and learning rates of connectionist components. Each prototype was based on the result from the previous model, while introducing new components to the architecture.

3.2 User-Centred Empirical Study

This section discusses the design of an empirical laboratory study conducted at the University of Brighton to identify the key factors that affect the notion of believability of interactive agents in an adversarial gaming context. An observational approach was used to study a group of participants, ranging from participants who claimed to never play video games themselves, to participants who stated they were avid gamers and students in games development courses. Getting feedback from users that have little experience interacting with virtual characters is important for this study. Prior knowledge of AI development techniques or games may affect the level of behavioural emergence the user perceives and their willingness to suspend disbelief. Sections 2.1.10 “Observer Psychology and Agent Behaviour Interpretation” and 2.1.11 “Perceived Emergence” in the literature review discuss the research that formed the basis for this hypothesis and the choice to include participants with little or no gaming experience in the study.

The study was conducted in two parts; an interactive study in which the participant played against a set of agents of varying complexity and sophistication and an observation study, in which the participant is asked to identify and describe the key behaviours of agents in a series of prototypes. Details on the design of this study are discussed in section 5.1.

3.2.1 Addressing the Research Questions

The design of the study started with a set of questions that expand on the original research questions defined in Chapter 1.5 and complement the theoretical analysis of the virtual agent in Chapter 4 “Adaption of Braitenberg Vehicles for Games”. That chapter discusses the behaviour as it is envisaged by the designer of the agent. It is therefore an *intended* design. However, game AI is a user-centred discipline, which requires that the subjective interpretation of the agents’ behaviour by its intended audience is included in the discussion of the AI architecture.

The study therefore aims to complement the discussion of the designers' intended design of an agent, with the perceptions of users interacting with the agent. A significant part of this thesis contributes to the ongoing discussion on the use of "believability criteria" as a measurable metric for evaluating believability in virtual agents. This empirical study focuses on the criteria for believability from the literature (Mateas 1997; Loyall 1997) presented in the format suggested by research instrument proposed by (Gomes 2013).

Broadly speaking this study investigates how users interpret agent behaviour and whether this is different from how a designer intended it to be. In addition it also tries to establish which factors affect this interpretation.

This study was designed to address the research questions discussed in Chapter 1.5. For this purpose the method aims to answer the following specific questions that guided the experimental design:

Q1 **Designer vs. Users' behaviour interpretation:**
How do users interpret behaviour and is this different from the designers' interpretation?

Q2 **Measuring Believability (using metrics):**
Can we use "criteria for believability" sourced from the literature as a metric for "believability"?

Q3 **Believability Criteria (according to the users):**
What do users think makes a virtual character more believable and fun to interact with?

To address Q1, **Designer vs. Users' behaviour interpretation:**

The designer intended that more complex and sophisticated behaviour of the agent is perceived as more believable by the users.

From the users, we want to know:

- How is the increased complexity and sophistication of an agent perceived by the users?
- What effect do changes in agent behaviour have on their believability?

To address Q2, **Measuring Believability (using metrics):**

As in the field of animation, it would be useful for designers to have "criteria for believability" to guide the development of their agents. Here the use of such metrics from the literature is explored.

With the user, we test:

- How effective are "criteria for believability" metrics (in the form of a questionnaire) at eliciting useful feedback on virtual agent design from the users?
- Are the criteria interpreted unambiguously (i.e. in the same way) in the same way by different users?
- Are the criteria applicable to the agents tested?

To address Q3, **Believability Criteria (according to the users):**

Aside from the believability criteria for virtual agents in interactive narratives (Mateas 1997; Loyall 1997; Gomes 2013), what other criteria might there be?

- What aspects of an agent's presentation and behaviour contribute to the notion of "believability" according to the users?
- What aspects of a game can "break suspension of disbelief"?

3.2.2 *Empirical Methods*

The study consists of three parts: A formal games test, a questionnaire and an informal prototype observation study. The particulars of these are detailed in Chapter 3.2.4. Each part is designed to address one of the overarching research questions stated in Chapter 1.5. These are based on the gameplay evaluation techniques employed in user-orientated development processes in the games industry. These in turn were adapted from the HCI and psychology field and include techniques such as behavioural observation, heuristic and questionnaire evaluation and interviews as well as gameplay data analysis.

Qualitative Research

This research project aims to gather feedback on the interpretations of an existing metric for believability as proposed by Gomes (2013) in order to investigate where potential issues may arise due to misinterpretation or inapplicability of the metrics. The goal is to analyse the feedback from participants who used these metrics to test a series of prototypes for their interpretation of the metrics' questions and suggest refinements to the questions where appropriate. The qualitative research may be a precursor that allows more quantitative research on believability in virtual agents.

Think Aloud Technique

Participants were encouraged to verbalise their thoughts while testing the agent and filling out the questionnaire. In conjunction with the questionnaire data and gameplay telemetry data, the audio recordings of participant feedback formed the main body of data and were processed using content analysis techniques.

The think aloud technique can be difficult to apply. Participants may feel embarrassed to speak to themselves when they are alone in a room. To make participants feel more comfortable and further encourage vocal feedback, the researcher joined the participants in the lab and asked the participants "what are you thinking?" when they seemed particularly affected by events in the game, or puzzled by questionnaire questions.

Content Analysis

Bryman (2008) details content analysis techniques that process the data collected from audio transcripts, video recordings and questionnaire responses gathered during the experiment, by coding data into themes that emerge from the data itself. Processing the same data several times, each time using the newly emerged themes the research can discover trends and common opinions in the user responses.

Bryman (2008) suggests the following four stages of qualitative research:

1. Read Text, broad notes
2. Read again, highlight, find labels for codes, suggest analytic ideas.
3. Systematically Code the text
    a. Eliminate repetition
    b. Group nodes
4. Relate Theoretical ideas to text.
    a. Interpret what was found
    b. Identify relationships / interconnections
    c. Key ideas
    d. Relation to Research question

The objectives of the content analysis were twofold:

1. To identify potential issues with the existing believability metrics in particular when used in the context of early prototyping of agents.

2. To get feedback on the agent architectures in order to gain further insight into what phenomena 'break' suspension of disbelief. To extend/add to existing criteria for believability.

Participants

To gather participants for the user-centred study, a purposive snowball sampling approaches was used. The objective when sampling was to include participants that had an interest in games development i.e. aspiring game developers as well as users with as little experience with games as possible. As such, this study did not require that the participants have to be experienced players themselves. Participants were mostly recruited from the Computer Science division at the University of Brighton.

During the introductory lecture for Computer Science Games students the researcher was introduced by the course leader. After a short introduction, the students were invited to leave their e-mail with the experimenter if they were interested in being contacted when it was time for the study. Other participants were recruited in a similar fashion during lectures in the Artificial Intelligence module and two guest lectures at a Masters Level research poster day and MSc Digital Media Production mini-conference, which featured audiences from a wider academic and demographic background. Nielsen (1994) suggests a group of at least 3 to 5 participants when evaluating a system using a qualitative approach. Of the 14 potential participants whose contact details were gathered, 7 were able to take part in the study. Participants were aged between 18 and 44 years old.

Lab Equipment

The study was carried out in a lab setting using the usability suite at the University of Brighton. This suite features a room for the participant and an adjacent recording room. The room for the participants can be observed via a one-way mirror from the recording room. For this experiment, the experimenter sat in the room with the participant, the recording room was only used to configure the camera equipment and control the recording of the lab cameras.

Three cameras were set up in the room:

1. A side view showing the participants upper body. This perspective was used to observe gestures and the hand movement of the participants while playing the game.

2. A frontal view of the participants face and upper body. This was used to identify facial expressions and body language during gameplay, as well as gestures the participant used to explain their observations to the experimenter in the interview.
3. A ceiling mounted camera filming over the shoulder of the participant. This was to identify the subject of the conversation when the participant pointed at objects on the screen.

In addition to the 3 cameras filming the participants, two screen captures were recorded.

1. A low-quality screen capture, recorded using the usability lab equipment. This was used solely to synchronise with the high quality screen capture.
2. Since the screen capture system in the lab was found to be of too low quality for the analysis, a separate capture tool (Open Broadcast Software) was used to record directly on the test machine. This was later synchronised with the lab recording using the timer visible in the game.

Two audio recordings were made, using two microphones in the lab. As with the screen capture, the audio equipment in the lab was of low quality, so a second microphone was attached to the test computer.

1. A room microphone placed on the wall to the left of the participant.
2. A desk microphone placed between the participant and the experimenter.

Figure 3 below shows the lab and camera set up on the left and the screen capture recorded using “Open Broadcast Software” on the right.

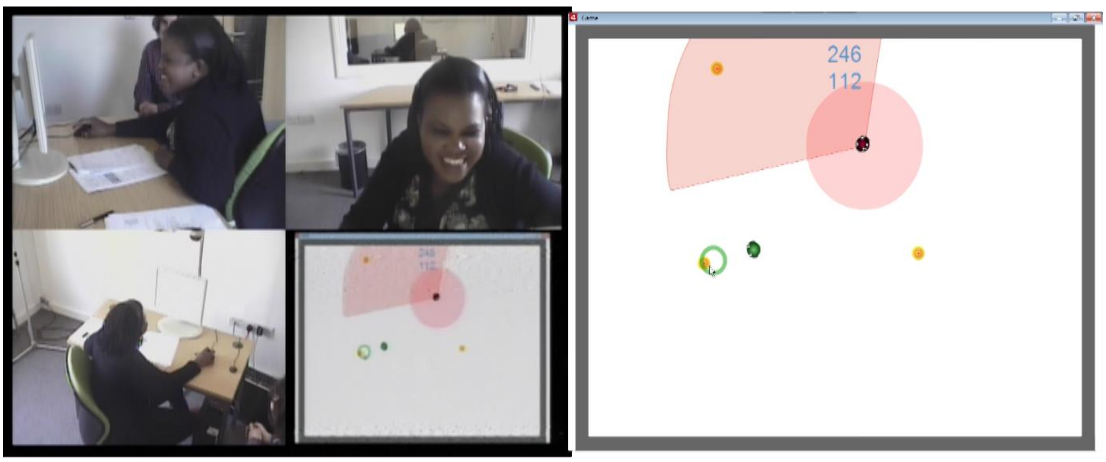


Figure 3 User experiment lab set up (with kind permission of the participant)

The video footage and audio recordings were imported into an Adobe Premier project, where the separate streams were synchronised. Figure 4 shows how each participant's recordings were organised into separate Adobe Premier Sequences.

Transcription

To facilitate transcription, the beginnings and ends of trials against the agents, as well as the separate parts of the study were annotated using markers. Trials were labelled with the trial number and the number of times a particular agent had been played against. So for example, as shown in Figure 4 the marker "4 A3" denotes the 4th trial overall, playing against agent "A" for the 3rd time.

Another tool that was used for transcription was "AutoHotKey". This was used to set up macros that allowed the experimenter to easily switch between Microsoft Word and the Adobe Premier, as well as providing shortcuts for controlling the video playback.

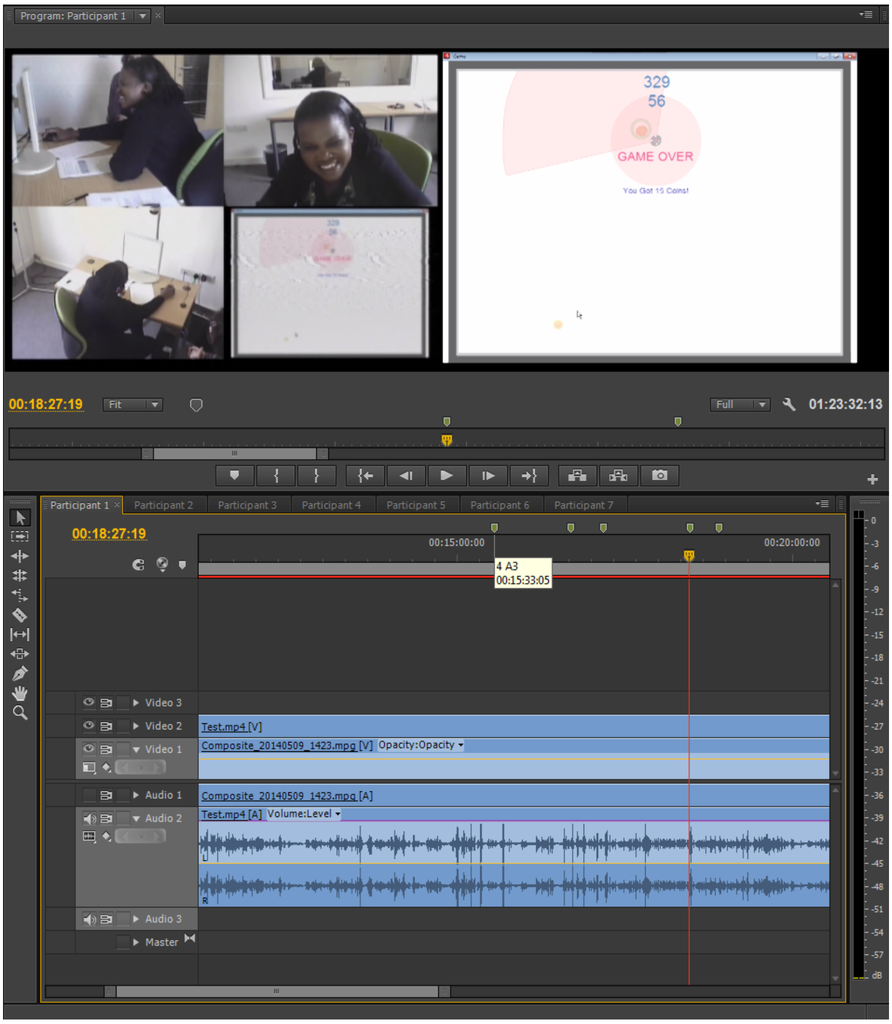


Figure 4 Composing video and audio in Adobe Premier

The experiments were transcribed in Microsoft Word. The transcriptions included everything that could be observed, for both the participant and the experimenter. This included the following:

- What they said
- Significant actions, including gestures
- Facial expressions and emotional reactions.

In addition, the transcription was annotated with contextual and implied information that would not otherwise be apparent from a pure audio transcription. An important example for this are deictic phrases; instances in which participants would often just refer to particular agents as “this one” or “the other one”. This information was added to the transcription.

The format used for the transcript was the following:

00:04:57:07 – Time codes were based on the timeline in the Adobe Premier Project after all the recordings have been synchronised.

The participant speaking was presented in normal font

[Participant actions, including gestures and emotional reactions i.e. laughing are written in bracket]

The experimenter speaking was presented in italic font

[Participant actions, including gestures and emotional reactions are written in italic brackets]

Words that couldn’t be identified from the audio recording are written in red

(Additional annotations and implied meaning such as the subject of a conversation was transcribed in round brackets.)

Agent names are written in capitals i.e. GREEN agent

Figure 5 Transcription format

Example from Participants 4 transcript:

[Participant starts first game against Opponent A. Stays out of sight of the RED agent until]

00:04:57:07

[120 Game Time]

Alright, so I’m thinking that the eyes (distance sensors) are where he’s (A) looking.

[Gets caught for the first time at 160 Game time]

Oooooh, He’s (A) seen me [Manages to turn and evade at 178 Game time] AHhhh [smiles] Alright, he (A) speeds up as well when he sees me, which makes it a bit difficult. I like it. 00:05:16:05

Figure 6 Example transcription (Participant 4)

Annotations that reveal which agent the participant was referring to in a statement were important for later analysis in NVivo 10.

Performance Data

As part of the *performance metrics* (Tullis and Albert 2013) gathered for this experiment, the game prototype (“artefact” in Purchase 2012) recorded the position of every object, coins collected, enemy AI state variables and user interactions at every time step of the simulation. Each record had the following data:

Data Label	**Description**
time	Time stamp. Milliseconds passed since simulation start
seePlayer	V whether the Enemy sees the player
seenTime	Number of milliseconds that V has been 1
sources	Number of sources (Coins) left in the arena
playerX	Player Vehicle X coordinate
playerY	Player Vehicle Y coordinate
MoveTargetX	X coordinate of Mouse cursor when clicking or holding button
MoveTargetY	Y coordinate of Mouse cursor when clicking or holding button
EnemyX	Enemy Vehicle X coordinate
EnemyY	Enemy Vehcile Y coordinate
behaviour	β Behaviour bias value of Enemy agent
behaviourChange	$\Delta\beta$ Change in behaviour β
base	θ Base behaviour value
baseChange	$\Delta\theta$ Change in Base behaviour θ
baseChangeRate	η Rate of change in θ

Table 2 Performance data gathered

In addition to graphing the player and enemy paths, this data also allowed us to calculate other useful data, such as the score and the distance between the player and enemy.

The data was saved in a CSV database file at the end of each trial. The filename consisted of a timestamp, the mapping code (of agents to the labels A, B, C), the agent played against, whether the player had won the round and the score.

e.g. “Date_2014-5-9_Time_17_24_Code_201_training_win_546.csv”

This data was captured in order to support the qualitative data gathered from the participants and was not intended for statistical analysis. It is however useful to have this data available, since it allows for some interesting views of the participants behaviour, without having to review the captured video footage.

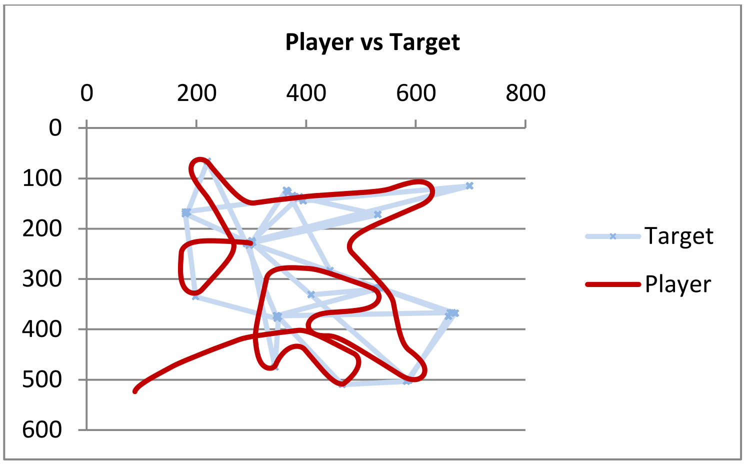


Figure 7 Mouse input (target) vs. agent position

Behavioural Data

In addition to recording audio, video recordings that show the upper body movements and facial expressions of the participants was recorded. This data was primarily used to guide the analysis of the audio transcripts since it allowed for easier identification of the inflection and the meaning of what was said. Since participants were asked to be critical, use of sarcasm and ironic comments was abundant. The video recordings made these easier to identify. One of the cameras also captured when participants pointed at the screen, which helped to identify the subject of comments when the participants were not being explicit in speech.

During the transcription process significant physical reactions to game events were also noted. This revealed frustration and other emotional reactions that would otherwise not have been captured.

3.2.3 *Pilot Study*

Both the game prototype and the empirical methods for this study went through several iterations and changes during a pilot study. The participants during this period included not only friends and research colleagues who tested the experimental procedure, but also game developers and researchers at the Game AI conference in Paris, including Ken Perlin and Bruce Blumberg, who kindly took the time to evaluate the prototype agents. The following list of changes and additions were made based on the feedback from the pilot study.

Game Design

The initial game design was a simple chase game, with multiple agents reacting to the player-controlled agent in different ways. There was no objective to the game and the participant was simply tasked with describing the behaviour of the autonomous agents. Feedback from the testers was that the game was not engaging enough and that it was difficult to focus on the behaviour of multiple agents at once. Feedback from research colleagues also suggested that the game should include objective performance measures i.e. a score of some kind that could be correlated to the purely subjective behaviour interpretations that participants would give.

To address these concerns, the game was re-designed as a competitive game played against a single opponent. The objective was to evade the agent for as long as possible. The score was that the time that the player spent not getting caught. Early responses to this new approach were positive. The players found the game more engaging and spent more time playing the game. However, since the objective was to merely hide from the opponent agent, players quickly converged on a single strategy which involved hiding in a corner of the game space and only moving if the opponent came too close.

The final game design introduced mechanics inspired by classic "heist" or collection games such as the arcade classic "Pac-Man" (Namco 1980) or the more recent independent game "Monaco: What's Yours is Mine" (PocketWatch Games 2013). The player now had to collect a series of "coins" in the environment while evading pursuit of the opponent. The scoring system underwent a few iterations as well. Initially, only the time to collect all coins was taken into account. There was no penalty for the player when they were seen by the opponent or were being chased, the game would only end if the opponent agent came in contact with them. This caused players to simply focus on collecting coins and disregard the enemy chasing them.

The final scoring system now added a time penalty when the player was seen by the opponent agent, thus encouraging players to both evade them, while collecting the coins as quickly as possible. To make this system transparent for the players, the user interface shows two counters at the top of the screen, one for the time passed (in seconds) and a second for the penalty time, which starts counting whenever the player is within the field of view of the agent.

Controls

Initially the player's agent was controlled using the arrow keys on a keyboard. Several users found this extremely difficult to master. A new, direct mouse-driven control scheme was introduce, where the player would simply click on a location in the environment and the agent would drive there and stop. This worked well for most users, but there were issues when players wanted their agent to drive in a curved path instead of a straight line.

The final control scheme addressed this feedback by not only allowing players to click on a location that they want their agent to move to, but also continue to hold the mouse button down and "drag" the target location across the playing field. This offered more direct control over the agent's path and felt natural and intuitive to the testers. As a last amendment, visualizations were added to the game to show a green circle on the target location when then player used the single-click method and an arrow indicating the direction of movement when using the click-and-hold method.

In addition to accommodating different control preferences an additional training scenario was added to the game that let the participants practice the control scheme without an opponent.

Opponent Selection

The game features three different enemy agents to be evaluated. In the first version of the game simply cycled through opponents once the player won the round. There were two issues with this approach. It did not afford the players the opportunity to improve their score against an opponent they had already defeated and it also gave the impression to the players that each successive opponent would be more difficult than the first.

To address the first issue, a menu screen was added that allowed users to select which opponent they wanted to play against. Initially these were numbered 1, 2 and 3. This, however, still gave the impression that the higher numbered opponents were more difficult. The labels were changed to A, B and C, but even then some impression of sequence was affecting the user's selection habits and they played them in order of (what they thought was) difficulty.

To address this issue, a trick was used in the final game. The mapping between the alphabetic labels was made configurable by the experimenter prior to starting the game. Thus the labels A, B and C led to different opponents for each participant. While users still had the impression that these were in order of difficulty, between participants this was not the case anymore. Feedback from the users regarding agent difficulty could thus be separated from the order of presentation, giving insights into both sequence bias and agent difficulty.

In addition to these measures the scenario presented to the participants also encouraged players to face the opponents in any order and let them decide how many times they wished to do so.

Questionnaire – Open Questions

One of the most important aspects of this study was to investigate how users described the behaviour of the opponent agents in the game and compare the terms and phrases they used to the anthropomorphic Synthetic Psychology terminology used to develop the agents.

The initial prototype presented users with a list of terms to choose from an approach similar to Benedek and Miner Product Reaction cards (Benedek and Miner, 2002). However, the pilot study showed that suggesting terms to the participants could induce them perceiving these behaviours. However, simply prompting them to describe the behaviour of the agents in their own terms revealed other issues. Descriptions focused on game events like "He sees me" or "He caught me" and not on the agent's behaviour *during* these situations.

The final questionnaire combines both approaches by prompting the participants to describe the opponent's behaviour in their own words, but suggests a short list of example terms. It also splits the question up into three questions that each focus on a specific situation that can occur in the game, encouraging more focused and detailed responses from the users.

Questionnaire – Game Experience

The study needed to capture whether the participants had prior experience with games since this could significantly affect both their objective game performance and their evaluation of the agents. During the pilot study participants who the experimenter did now know well, were asked what their experience was.

The responses were surprisingly varied. Some participant's online experience with games was second-hard, i.e. through their children or siblings. Others initially stated they did not play games, but when prompted whether they played popular 'casual' titles such as solitaire or games on their Smartphone, they confirmed. Unlike books, music or films, games are a medium that presents itself in a variety of different formats – some of which a not dedicated game playing devices. It seems that to some of these participants "playing games" meant owning a dedicated game playing device.

Based on this feedback the questions regarding games use were changed from simply asking about the number of hours that participants played games, to asking more specific questions about the nature of their relationship to games, including second-hand experience.

In addition, if the participant stated that they did have significant prior experience with games, they were also prompted to list their favourite genres. This was included because experience with games in general does not indicate that the participant's skill is equal across genres. Since the game used in the study is an action game, it was this question that gave insight into whether their prior experience would impact their objective performance in the game.

Agent Difficulty

Once the coin collecting/evasion game design had been settled on several refinements to the agent behaviour had to be made. Since there was now a penalty for being spotted by the opponent, the players got frustrated when it was not apparent to them when this was the case.

Looking to inspiration from commercial games with similar game mechanics a "sight cone" was added to the UI of the game. This showed the opponent agent's field of vision and changed its colour to red when the player was spotted. Initial feedback was positive and players were more adept at avoiding the agent. However it also made the game too easy, since players simply focused on the outline of the cone instead if the opponent agent.

In the final prototype, the display of the cone varies with the behaviour state of the agent. When the opponent hasn't spotted the player, his view cone is invisible to the player, but as the player opponent notices the player, the cone fades into view. This compromise still gives the players enough feedback to avoid evasion, but forces them to remember the range of the field of view and focuses their attention on the body of the opponent agent.

Gameplay Feedback

The early prototypes of the game featured no sound effects or feedback when the player either won or lost a round against an opponent. In the final prototype several effects were added to make failure clearer and success more satisfying.

A sound effect that plays when the player collected coins was included to make each coin collected feel like a little success. To increase the sense of urgency and prevent players from having to look at the timer clock displayed on the screen, a clock-ticking sound effect was added.

In the pilot prototype the game also simply returned to the title screen when a round was won or lost. Some users found this confusing, since it sometimes happened that they were just about to collect the final coin as the opponent caught up to them. They also did not have time to look at their score displayed on the screen before the game returned to the title screen.

To address this, the final game pauses and a "game over" screen displays when the player is either caught or collects the final coin. A crowd cheering sound effect plays when the player won and a theatrical "failure" fanfare plays when they are caught. The screen also displays the score of the player, including the number of coins remaining when they lost or a congratulatory message when they won the game.

3.2.4 *EXPERIMENTAL PROCEDURE*

Participants were asked to play against, compare and evaluate a series of 3 different agents in the context of a simple coin-collection game. They were given the objective to complete the game against each agent. The goal of the game is to collect all the coins (16) in the game arena, while avoiding being seen by the red enemy agent.

At the beginning of the session, participants were handed the following materials:

1. A consent form ensuring the ethical use data collected during the study, a summary of the tasks to be performed and total duration of the study. Before continuing this form had to be signed by the participant and the experimenter.
2. A Demographics questionnaire.
3. A Participant Script detailing the procedure of the experiment
4. A Questionnaire booklet, containing questions about the Game and the agents

After making sure the participants understood and agreed with the use of the gathered data, they were asked to fill out the demographics sheet and then read the introduction to the study in the script.

The lab experiment was conducted in three parts. These are detailed below:

Part 1: Formal Games Test

The first part of the lab experiment consisted of a gameplay observation study that combined formal and informal methods to analyse participant behaviour and vocal feedback while they played an adversarial game against three agents of varying complexity.

Data Gathered:

1. Users' vocal responses are recorded.
2. Users' actions are captured on video and transcribed alongside the audio recording.
3. Data is recorded by the game itself. This produces "telemetry" data of each game users play against the AI, which can be used alongside the data from the video and audio recordings.

Participants were asked to be vocal about any questions they had regarding the game or the study itself. The experimenter would remain in the room during the study.

The script instructed the participants to assume the role of a member of a games development team who has been tasked to review a prototype of a game. This specific scenario was given to encourage the participants to not only respond with positive or negative comments on the game and the agent, but to think deeper and suggest how they would improve the game. Fear of doing badly at the game was a sentiment expressed by several participants before the trial. This scenario, which places the participant in the role of a member of a development team was able to alleviate this fear of being tested themselves, empowering participants and allowing them to focus on the evaluation task.

Before pilot testing the method, it was intended that the participant plays one game against each agent architecture and evaluates their experience using the questionnaire provided after each trial. During the pilot study it became clear that participants required more than just one trial to be able to make any statements about the game and the enemy agent behaviour. While the participants still had to play against the agents in order, they were allowed to repeat trials to compare the agents and improve their score.

Part 2: Questionnaire

Users are asked to fill out a questionnaire, answering questions about each of the agent they play against and some general questions about their preferences regarding the game.

The data gathered consists of each player's answers to the following:

1. A set of questions on a Likert scale based on the "believability criteria" by narratives (Mateas 1997; Loyall 1997) and presented in the format suggested by Gomes (2013).
2. A set of three open questions asking the users to describe the behaviour of the enemy agent in different game situations.
3. General and preferential questions regarding the game and the agents.

The participants filled out the questionnaire sheet while they were evaluating the agents. While participants had to play the agents in order, they were allowed to replay previous agents and amend their answers to the questions on the questionnaire. If they did amend questions, this was transcribed.

Participants were also encouraged to be vocal about any of the questions on the questionnaire. This is referred to as the meta-evaluation of the questionnaire. After the participants had described their own interpretation of the questions, the experimenter would elucidate their meaning, according to the definitions by Gomes (2013). The meta evaluation was done to gather feedback on whether participants thought that the questions were unambiguous and suitable for agent evaluation and comparison.

Part 3: Informal Prototype Observation Study

After the analysis of a set of agents in the context of a game was completed, each user was asked to look at a series of 5 prototypes. Each of these showcases groups of agents in a variety of typical game scenarios.

Several of these prototypes showcase behaviours different from the ones the opponent in the game in test 1 could perform. The objective of this test was to see how participants interpret the behaviours of agents when simply observing them and not playing against them.

This test took a much more informal form than the previous study as it was meant to encourage a more open conversation between the participant and the experimenter, since the previous game study was regarded a high-intensity task by most of the participants.

In this test, participants were simply asked to describe the behaviour of the agents they observed. The questionnaire sheet provided space to write notes. Participants were advised that they could write down notes on their thoughts, but were encouraged to explain these to the experimenter, so that they can be captured in the transcript.

3.2.5 Data Analysis

The data gathered during the user experiment was analysed using qualitative and descriptive statistical methods. Even though statistical methods were applied to the data gathered via the questionnaires, the number of participants that carried out the experiment was too small to ensure statistical significance. However, a rich dataset combining observational and qualitative interview data supported more in depth analysis.

The method used for data analysis is based on Bryman's (2008) 4 stages of qualitative research. Design of the study was informed by the research questions discussed in "3.2.1 Addressing the Research Questions" and the initial expectations regarding the outcome of the study discussed below. With these in mind, an initial review of the data revealed common themes and categories in the data, which were subsequently organised into a preliminary list of codes. The focus during the analysis was on discovering themes pertaining to the participants' definitions of believability, criticisms of the agents tested and their understanding of the questions posed in the questionnaire.

Further review of the data using these codes refined these categories and themes to form a set of classification codes.

The resulting classification scheme and the findings facilitated the formation of theories about the relationship between AI developer design intent, the perception of agent behaviour from the player's perspective and the notion and practical application of believability metrics in virtual agent research.

Data Available

The data sets that were analysed consisted of video transcriptions, questionnaire responses and telemetry/performance data recorded by the game prototype.

In addition to transcribing what the participants said, the video transcriptions noted participants' actions during the experiment, which included gestures and emotional body language and facial expressions.

Multiple video streams included camera perspectives that showed the participants posture, upper body (hand movements), their view of the screen (to identify what they are pointing at) and facial expressions. In addition, a high quality screen capture of the game was recorded, since the quality from the video cameras was quite low.

The videos were synchronised and composited using video editing software (Adobe Premier). Two audio recordings were available; one room microphone mounted to the wall closest to the participant and one table-top microphone placed between the participant and the experimenter. Having two audio recordings to switch between was helpful, since participants often spoke quietly when playing the game.

Figure 8 shows the different types of data that were generated for each of the three parts of the experiment:

PART 1: Gameplay Agent Evaluation

1. Participant Observation While Playing the Game
    a. What they said
    b. What they did
2. Gameplay "Telemetry" data
    a. Trial Time and Penalty (seen) time
    b. Number of Coins, Score
    c. Agent and Enemy Position throughout trial
    d. User Input (Mouse Clicks)

PART 2: Questionnaire Feedback

1. Answers to questions
2. (Meta) Feedback on questions/believability criteria

PART 3: Informal Observation

1. Informal Observation Study Descriptions
2. Interactions with the prototypes

Figure 8 Data available for analysis

Agent Behaviour Description Predictions

An important aspect of this study is to compare how the participants interpret the behaviour of the agents to how the behaviour was intended to be perceived by the designer. Below is a list of the four agent types with a brief description of what behaviour the designer *intended* to implement. These will be referred to during the evaluation of the participants' feedback.

1. **Binary Switch**:
    - The binary switching agent is immediately aggressive when it sees the player, but "forgets" about them just as quickly. It is expected that participants will interpret this as slightly "incoherent" or "unrealistic".
2. **Continuous Switch**:
    - The slower transition to the chasing behaviour is intended to show "suspicion" in the player when they only stay in the field of view for a short time.
    - The slow decay of the behaviour bias back to the patrolling behaviour is meant to convey the presence of short term memory.
3. **Remap Behaviour**:
    - This agent is continuously changing its behaviour mappings during the game and should therefore seem less "predictable" to the participants.
    - The behaviour mapping changes every time the player collects a coin, regardless whether the agent sees the player doing so. It is expected

that participants interpret this as a form of long term memory and an awareness of how many coins there are in the environment.
    - When the primary behaviour is fully "aggressive", the agent will transition toward "exploring" when it sees the player. It is expected that participants perceive this behaviour as "incoherent".
4. **Adaptive Behaviour:**
    - This agent initially ignores the player and actually avoids bumping into them while exploring. This should be perceives as "indifferent" or even "friendly" behaviour.
    - When the player has been spotted stealing a few coins, the agent is "suspicious" of them, but will forget about chasing them relatively quickly when the player is not in their field of view. This should be perceived as "suspicious" or "guarded" behaviour.
    - When the player has collected most of the coins in the environment, the agent should seem "angry" at the player and will not give up chase when it spots the player, even if they manage to evade the field of view.
    - The above behaviours should convey a sense that the player has "hurt" the agent's feelings, since the agent started out being neutral and friendly and only became hostile when the player stole in front of them.

Initial Codes

Preliminary codes were defined at the beginning of content analysis. These would be refined during the process, based on participant actions and responses sourced from the audio transcript and video recordings of their behaviour in the game and in person.

Two perspectives were assumed to guide the coding process. The first is the perspective of the developer, who is interested in aspects that are considered important about the agent and the game design. The second is the perspective of the researcher, who would look for additional data relating to the issues investigated by the questionnaire that were not written down by the participants and also looks for feedback on the questionnaire and experimental design itself.

From the perspective of the AI developer, the following expected codes where initially proposed, which focus on the agent and game design:

Agent

- Behaviour
- Graphics
- Intelligence
- Emotions

Game

- Difficulty

- Controls
- Rules

Codes proposed from the perspective of the researcher included a list of all the questions on the questionnaire sheet that was filled out by the participants during the experiment. These codes are intended to capture additional responses to the questions that the participant didn't write down and also questions or comments that the participants had about the questionnaire questions or the experiment itself. The latter comments and are labelled as "meta" responses (i.e. questions about the questions).

Refined Codes

Far more codes than initially expected emerged from the data during the content analysis process. Data was coded and re-coded in an iterative process as more themes and codes emerged. The codes were organised into two main branches, one for the interactive study (game evaluation) and the other for the observation study. The interactive study codes were organised into feedback given while filling out the questionnaire and the actions and speak-aloud protocol made while the participants were playing the game. A total of 186 themes and subthemes were defined over the course of the analysis. The full list hierarchy of nodes can be seen in appendix 10.2 "User Experiment: Refined Codes".

Apart from finding significant excerpts and categorizing them into emerging themes, all the participants' responses were also sorted by topic, that is, by the type of agent architecture they were describing. These nodes were used in conjunction with the themes during the analysis process.

Initial Predictions

Before starting the thematic coding of the data, a series of predictions were noted that would guide the initial steps of the analysis process. Predictions 1-4 are based on common difficulties involved in games user testing. Especially participant skill and potential pre-existing familiarity with similar games and control methods are factors that can have a significant impact on the player experience. During the pilot study, several balancing/difficulty and control method changes were made to the game prototype to accommodate different users and minimize the effect of these factors. However, they still need to be considered when qualifying user statements and test performance results. Predictions 5-9 pertain to the perception of agent behaviour, where 5-7 are based on the literature review (see references), while 8 and 9 are the author's.

1	It is likely that novice players are initially too distracted by playing the game to notice agent behaviour.
2	We predict that the game may be too difficult for users who do not usually play games
3	We predict that the game may be too easy for experienced players
4	We predict that users will prefer the second control method (holding the mouse button down) over the first (clicking)
5	We predict that users will over-estimate the complexity of the AI controlling the agent - Braitenberg's law of uphill analysis (Braitenberg 1984 p.58)
6	We predict that users will discover new emergent behaviours that the designer did not know about – Perceived emergence (Ronald et al. 1999a)
7	We predict that not all behaviour features that the designer implemented will be noticed by the users. – Game AI developer interviews (Peschke 2013)
8	Agent type 2 "Binary behaviour switching" will be the most predictable
9	Believability Criteria 3 "Personality" is too complex a term to be measured.

Table 3 Initial predictions

3.3 Simulation Tools and Environments

To test the Synthetic Psychology (SP) design approach as a viable approach to game AI development, a set of software tools were needed that fulfilled the requirements that this approach brings. Below is a list of the main features of SP that the tools and environments needed to cater for:

- **Iteration** – Governs the entire design and development process and involves rapid prototyping, incremental improvements and "tuning" variables.
- **Experimentation** – Trying out new architectural ideas based on observation, rather than deliberate top-down design. This can even seem random at first, since the behaviour dynamical systems can be very difficult to intuitively predict.
- **Combination** – Combine behaviours or even entire architectures with each other. This is also a feature used in artificial evolution.
- **Observation** – Having the designer and other observers interpret the behaviour of the agent.

In addition to the requirements of SP listed above, the simulation environment should be using a platform that employs technology typically available in games development. This shifts the focus from using dedicated research or modelling tools such as Matlab, Octave or Scilab toward the use of languages and platforms more commonly associated with games.

From this a set of requirements for the simulation tools was derived:

1. The system must allow for rapid prototyping. The designer should be able to easily implement and test new behaviours, while also allowing for some random experimentation (through combination of agent behaviours for example).
2. The system must visualise agent behaviour, but also allow for the capture of behavioural data to enable other forms of analysis i.e. graphing movement patterns.
3. The language and platform chosen for the system should resemble those used in typical games development.

While this defines the first set of requirements for the system, the features that should be included to enable the construction of agents formed the second set. Similarly to how Braitenberg used his imaginary robots as a vehicle (the pun was surely intended) to illustrate his theories, this thesis adopts the agent architectures presented in Braitenberg's thought experiment as a model for testing the SP approach in the context of games. Braitenberg synthesis his "Vehicles" from simple components that are in turn inspired by the field of robotics and in several cases biological mechanisms. The basic agent components that the simulation environments needed to simulate were initially based on the early experiments in Braitenberg's book, which were then extended to create sensors and actuators that suit the digital realm and the context of video games.

Since these were designed specifically to suit each agent design and game scenario, they will be detailed where appropriate in Chapter 4 "Adaption of Braitenberg Vehicles for Games".

3.3.1 TWO TEST PLATFORMS

Two test platforms were developed to support the experimentation with Braitenberg inspired agents in a gaming context. Each had its own features, affordances and limitations.

- 2D Simulations were implemented in Flash. The toolkit developed on this platform is easy to use visual drag-and-drop features and is suitable for rapid prototyping. The library written for these experiments can be easily included in any Flash project.
- 3D Simulations were implemented in Java Monkey Engine. It uses the Open Dynamics Engine (ODE) for rigid body physics simulation, thus allowing for more realistic simulation of agents. The 3D render also enables the implementation of a "Virtual eye" sensors, which can utilise computer vision algorithms.

For the majority of tests, in particular the empirical user study, a 2D Flash simulator was used. This abstract iconographic representation of Braitenberg's architectures allows users to focus solely on the behaviour of the agent, without distractions from potential glitches in the physics and 3D rendering systems. It was also developed in preparation for further study, where having the option of conducting web-based user testing, such as Amazon's "Mechanical Turk", was kept open.

The 3D simulation was primarily developed to test the limits of Braitenberg's architecture. For example, using a 3D representation allows for the implementation of more sophisticated sensors that utilise an agent's first person perspective of the environment. The next sections describe the features of the simulators in more detail.

3.3.2 2D Flash Simulator

The library that was developed for implementing Braitenberg-like virtual agents in Flash includes a simple kinematic model to handle collisions and the differential-drive of the vehicle platform.

Unlike the 3D simulation where realism and an acceptable level of parity to real world robotics was the goal, the library developed for the 2D Flash simulations were designed to be flexible enough to also allow for the implementation of non-realistic movement models.

This means that the agents should be able to move in ways that are unrealistic for differential-drive platforms, but would enable it to simulate the movement of non-differential drive controlled characters (like humans), while using the same architectural ideas. A good example of this is the addition of the strafing wheel, which allows vehicles to move sideways (laterally to the wheel axis), without using the differential drive.

A full rigid body physics simulation would have been a hindrance when implementing such movement mechanisms (since wheel friction would make it difficult to implement). Unlike Java, Flash is also a single threaded low-performance, closed-source platform with few optimization options available to end-user developers (non Adobe employees). Including rigid body physics would have added significant processing overhead to the library. Since the library is intended to be implemented alongside existing game engines, which might use their own physics implementation, it instead opts for a non-general physics solution that specifically controls only the agent's movements. Collision detection and handling are implemented as an optional feature so that they can easily be replaced by more sophisticated systems specific to project that the library is being used in.

The model underlying the differential drive is implemented using a set of vector calculations. Since the simulation is run within a graphics API and is not meant to realistically approximate a physical environment, the terminology might initially not seem intuitive and the scales of numbers might seem wrong. In the virtual world distance is measured in Cartesian coordinates, which are eventually converted into pixels on the screen. Speed also doesn't run in seconds or milliseconds, but in frames, which are equivalent to time steps in common simulations. Thus a valid speed would be 10 pixels/frame, instead of 10 m/s.

2D Flash Simulator Agent Kinematics

The following formulas are used to determine the position of the agent on the next time step, given the force exerted by the two wheels v_l and v_r and the current velocity of the vehicle V. We consider the vehicle's 2-dimensional coordinate system to be centred at the mid-point between the wheels with the x-axis aligned with the axle and the y-axis orthogonal to it in the plane of motion. In this local coordinate system the centre of the vehicle is always at the coordinate 0, 0 in our virtual models.

Calculation is done in four steps:

1. Calculate the angle of rotational displacement, based on the differential between the wheel's forces and taking into account their position relative to the agent's centre of mass.

2. Apply the rotational displacement
3. Calculate the distance of forward displacement, based on the force from the wheels, the previous velocity and the effects of friction and inertia.
4. Move the agent in the new direction, by the distance of forward displacement.

1. Rotational displacement

To calculate the differential steering caused by the two wheels spinning at different speeds, we could simply subtract one scalar force from the other and define some range of angle values to correspond to a given result. However, to give a more realistic result the model takes the position of the wheels, relative to the centre of mass of the body, into account as well.

Given the force scalars of the wheels v_l and v_r and their x-offsets relative to the centre of mass of the agent (which is always at the coordinate 0,0 in our virtual models) x_l and x_r, we first calculate the magnitude of the force vectors F_l and F_r that result from using the respective x and v scalars as their components.

$$\overrightarrow{F_l} = (x_l, v_l) \qquad \overrightarrow{F_r} = (x_r, v_r)$$

To calculate the angle of rotation we take the unit vectors of the left and right force vectors and add them to create a sum vector $\overrightarrow{S}$.

$$\overrightarrow{S} = \widehat{F_l} + \widehat{F_r}$$

The arctangent of the sum vector then gives us the rotational displacement, which we convert to degrees for compatibility with the Flash display model, which uses degrees for object rotation.

$$\alpha = \frac{\left(\tan^{-1}\frac{S_x}{S_y}\right) * 180}{\pi}$$

Since the resulting angle of this calculation can be rather small (and also depends on the distance between wheels) the steering model allows for additional customisation via a “power steering” value p. This scaling factor simply multiplies the effect of α. Thus the final rotational displacement value is:

$$\alpha = \frac{\left(\tan^{-1}\frac{S_x}{S_y}\right) * 180}{\pi} * p$$

Before calculating the new position of the vehicle body, the rotation displacement is applied to the current orientation θ, giving the new orientation $\dot{\theta}$.

$$\dot{\theta} = \theta + \alpha$$

2. Forward Displacement

The implementation of forward displacement f starts by taking the average of the force scalar v (y-axis) components of the wheels.

$$V = \frac{v_l + v_r}{2}$$

The resulting forward displacement for the current time step $\dot{f}$ is the sum of the forward displacement on the previous time step f and the average force exerted by the motors V.

$$\dot{f} = (f * \mu) + (V * i)$$

Even though it does not model mass (as a Newtonian model would), friction μ and inertia i are implemented as optional factors that are taken into account in the final calculation. In this simplified model they are related constants in the sense that friction acts to slow down the agent exponentially, while inertia scales the force incoming from the wheels.

$$0 \leq \mu \leq 1 \quad default\ 0.9$$

$$i = 1 - \mu \quad default\ 0.1$$

Friction μ is a real number where $0 \leq \mu \leq 1$ such that multiplying the calculated forward displacement by it on every time step, exponentially decays the velocity. A lower threshold of 0.1 (which corresponds to 1/10 of a pixel of movement on screen) is set at which the velocity is set to zero 0.

The inter-relatedness of μ and i mean that when friction is reduced (by increasing the friction value), the force applied from the motors and resulting speed of the agent will increase proportionately.

Just like the power-steering factor is used to adjust turning speed to model movement that would be impossible with a differential drive, friction μ and inertia i can be configured to create a variety of movement models. Virtual recreations of robot platforms may want to use the defaults ($\mu = 0.9, i = 0.1$), which were found to approximate the movement of 2-wheeled robots like the Khepera. However, it is possible to configure these parameters to adjust the sense of weight behind the character's movement, from lightweight characters that can instantly turn and stop immediately, to heavy characters that accelerate and stop slowly and manoeuvre in long turning arches. Figure 9 and Figure 10 illustrate the two different movement models.

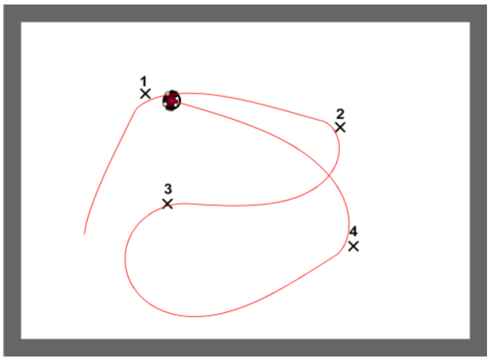


Figure 9 Power steering at 15

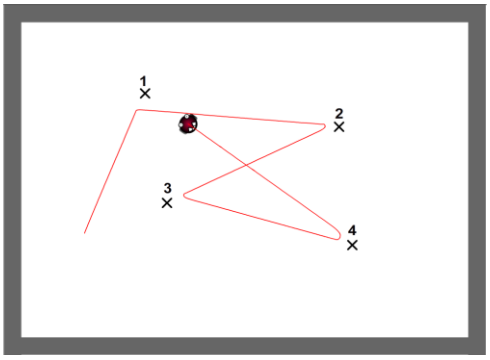


Figure 10 Power steering at 100

3. Combining into Displacement Vector

The previously calculated new orientation $\dot{\theta}$ and the forward displacement f are then combined into the directional displacement vector d.

$$d_x = \sin(\dot{\theta})$$

$$d_y = \cos(\dot{\theta})$$

$$\vec{d} = (d_x, d_y)$$

Which is then added to the current position vector $\vec{\rho}$ of the vehicle to give the new position of the vehicle $\dot{\vec{\rho}}$ in the world coordinate system.

$$\dot{\vec{\rho}} = \vec{\rho} + \vec{d}$$

The basic process (without power steering and friction) is illustrated in Figure 11 below. First angular displacement α is calculated and applied. Then forward displacement V is calculated and applied to the rotated body, giving the new position and rotation for the next time step.

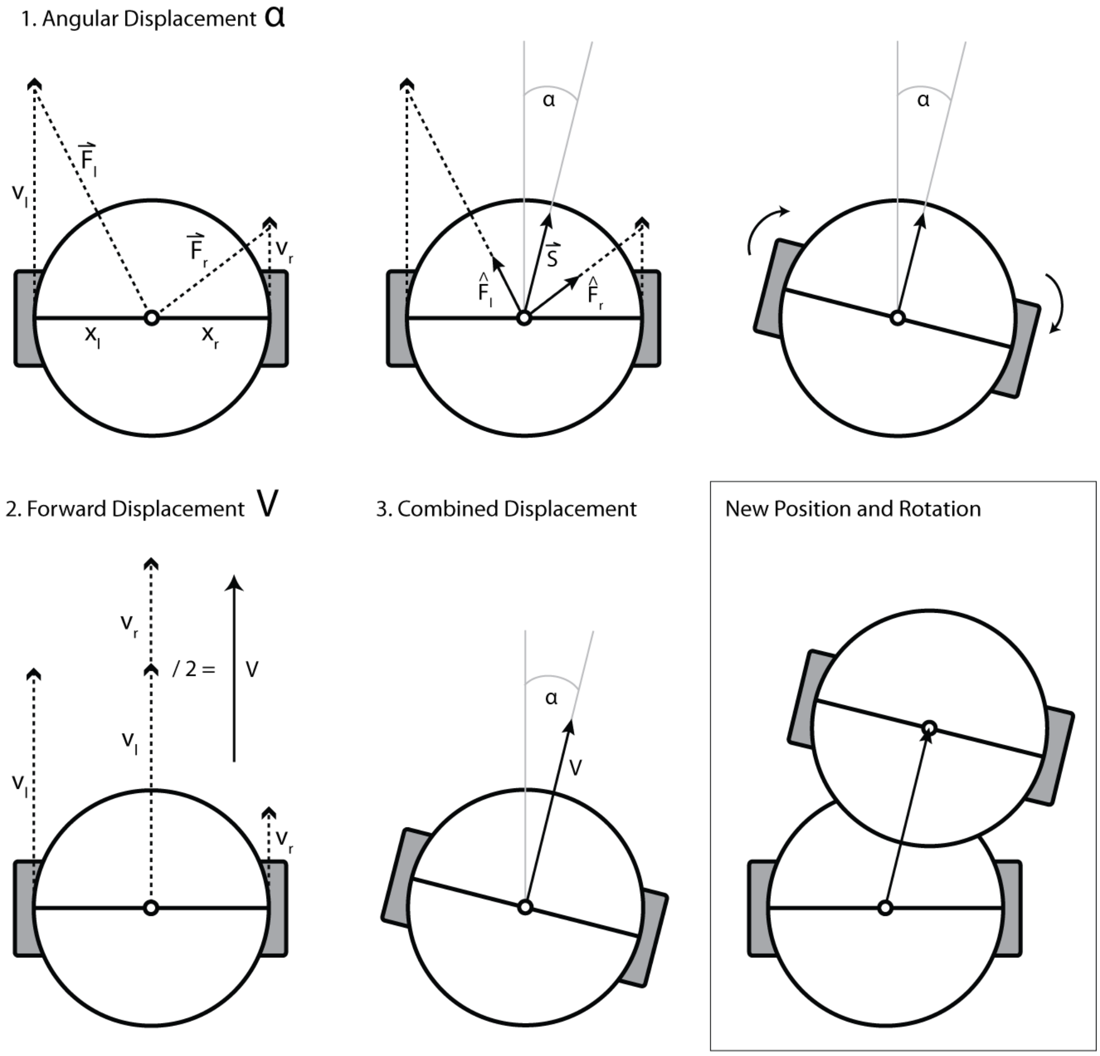


Figure 11 Calculating the new position of the vehicle body

3. Lateral Movement (Strafing Wheel)

The model in its current state is able to simulate a variety of different movement dynamics. However, they are all kinetic models of the same thing, a two-wheeled differential drive robot platform. To make the model useful for simulating behaviour of virtual characters that have more than two degrees of freedom, the concept of a "strafing wheel" was introduced. This wheel is named after the typical strafing behaviour primarily seen in modern first person shooters, where the character moves perpendicular to the direction it is facing in. This is equivalent in our case to movement along the x-axis of the vehicle's local coordinate system.

A realistic dynamical model would have difficulties simulating sideways movement alongside the differential drive, but the simplified kinematic model faces none of these issues.

The virtual strafing wheel creates a strafing vector $\vec{\beta}$ from the components of the current orientation angle of the simulated vehicle body θ. Since the wheel should move the body along an axes perpendicular to the body orientation 90^{o} are added to the orientation θ.

$$\beta_x = \sin(\dot{\theta} + 90)$$

$$\beta_y = \cos(\dot{\theta} + 90)$$

$$\vec{\beta} = (\beta_x, \beta_y)$$

This gives us the perpendicular axes to move along. The component values for $\vec{\beta}$ are then multiplied by the force scalar of the strafing wheel to give the displacement along the axis perpendicular to the orientation of the vehicle body. The resulting vector is then added to the current position vector $\vec{\rho}$ alongside the directional displacement vector $\vec{d}$, which was calculated by the differential drive model previously, to give the final position of the vehicle on the next time step. The new position of the agent $\dot{\vec{\rho}}$ is thus given by:

$$\dot{\vec{\rho}} = \vec{\rho} + \vec{d} + \vec{\beta}$$

3.3.3 3D Java Simulator

Before deciding on a custom solution for the simulation environments used to conduct the experiments in this thesis and initial review of existing solutions was conducted with respect to the requirements listed at the beginning of this section.

The initial review included virtual agent development environments such as MorphEngine (Bongard 2003), Breve (Klein 2002) and Microsoft Robotics Developer Studio (Microsoft 2008) as well as several tools that specifically focused on Braitenberg's vehicles, some of which used 2D environments (Gerken 1998; Lambrinos 1999; Will 1999; Wiseman 1999), while others were in 3D (Komosinski and Ulatowski 1998a; Kohler 1999; Wyeth and Purchase 2001).

Of these Microsoft Robotics Developer Studio (MRDS) initially appeared to be the ideal simulation environment and a significant development effort was applied to designing and implementing initial prototypes. However, bugs and shortcomings in the closed-source development philosophy of the product did not allow work to be continued using this tool.

After the experience with MRDS it was decided that a custom, open-source tool should be developed from scratch.

3D Engines

After learning the caveats of relying on proprietary software with MRDS, it was decided to develop a custom simulation environment using open-source tools. Java Monkey Engine (JME) (jMonkey-Community 2010), Ogre 3D (Streeting 2010) and Open Scene Graph (OSG) (Osfield 2010) are some of the most popular open source scene graph engines currently around. All three can use the cross-platform Open GL graphics API. Projects can thus be run on both Windows and Unix based systems.

All three engines provide roughly the same functionality: a scene graph, lighting and texturing systems and integrated physics solutions. As the features were so similar, the decision ultimately came down to the two factors of development effort and performance. Ogre3D and OSG are written in C++ and would therefore potentially provide better performance than the Java based JME.

Ultimately a background in Java and Flash development, combined with Java's inherent suitability for rapid prototyping approaches, influenced the choice of JME.

3D Simulator Features

A custom, open source solution for the simulation environment was adopted, which required the following list of features:

- **Virtual Eye** - an off-screen renderer computes a separate camera perspective from the agent's point of view.
- **Physics Simulation** – Realistic rigid-body physics were a requirement for the 3D virtual environment. The Java Monkey game engine provides support for both ODE and JBullet physics implementations. ODE is used in the current implementation as it provides better performance and stability.

- **Scripters** – Experiment script classes are helper classes that modify the configuration of the simulation. These are used to set up experiments and determine which data is recorded.
- **Multiple Agents** - The agent model supports instantiation. This means that multiple complex agents can be simulated. Note that the number of agents that employ virtual eyes is currently limited by the graphics hardware since each off-screen renderer requires a separate texture unit.
- **Vehicle Drive System** - The vehicle drive system is a differential drive configured to emulate the physical properties of two-wheeled robots such as the Khepera or iRobot platforms.
- **Data Display** - On-screen data display for sensor data and internal states (Mnemotrix wires etc.) in the form of bar charts is supported.
- **Data output** - Sensor data and internal processing values can be written to external data files that are CSV datasheet formatted. It is possible to access all the sensory data from the simulated agents, as well as the position coordinates of world objects. The latter can be used to draw object paths to an image file, enabling the post-experiment analysis of agent behaviour

Physical Agent Model

The diagrams below show the agent model used in the Java simulation that formed the platform for all virtual agent simulations in this thesis. It is based on the 2-wheeled differential drive based robot platform described by Braitenberg, but is also a virtual implementation of popular 2-wheeled robotics platforms such as the Khepera and iRobot Roomba. Since the simulation includes a semi-realistic rigid-body physics system including weight, inertia and friction properties, control systems built upon the virtual agent platform should be transferable to real robot platforms.

Notable properties of the agent model are:

- The body has a textured surface that makes the agent more visible to other agents that use the virtual eye, since the lateral inhibition it incorporates filters out un-textured, uniform surfaces.
- The entire model is constructed in a physics based simulation engine, with body and wheels having the properties of wood and rubber material respectively.

- The white lines protruding from the agent's front end are visualizations of simulated infrared distance sensors. In-engine ray casts are used to simulate the behaviour of these important sensors, which allow the agent to avoid collisions with walls and other objects in the environment.
- Although not visible on the agent body, a virtual eye can be fixed to the agent body and is typically place at the top front edge of the agent's body.

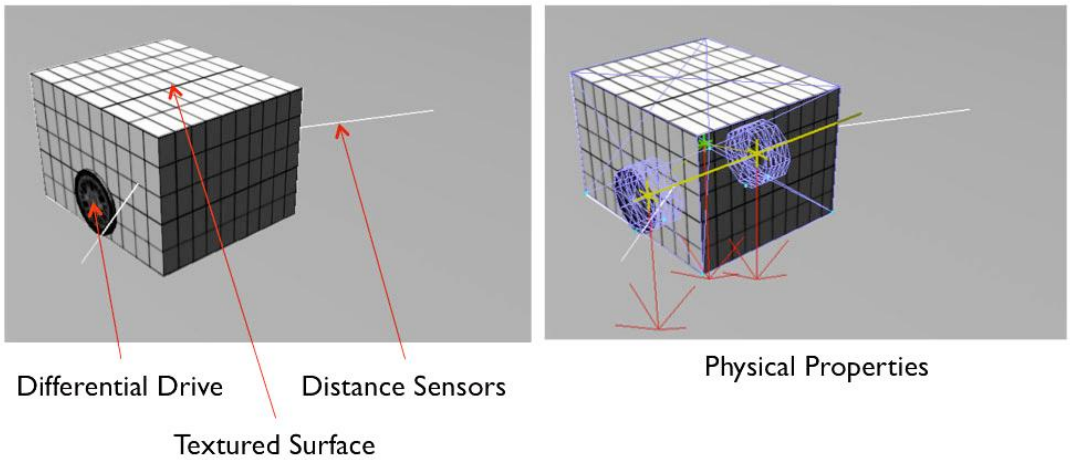


Figure 12 Agent model in 3D simulation and with physics placeholders

Figure 12 shows a screenshot of the virtual agent platform running in the simulation engine. The left image shows the outside of the agent as it appears to other agents in the world. The right image overlays the rigid-body objects that are used for collision and dynamics modelling.

The Virtual Eye

This and the following section describe the tests of the lateral inhibition algorithm adapted from Braitenberg (1984). These were conducted in order to ensure the research equipment could be used to successfully support the simulation of a wide variety of Vehicle models. This section discusses an initial Flash Prototype of the algorithm, whereas the next section shows the results of the test of the 3D Java prototype, which used the lateral inhibition algorithm to control the behaviour of a simulated Vehicle.

Besides realistic physics, the ability to render the world from the point of view of the agent was one of the driving factors behind developing a realistic 3D simulation environment. Several of the Vehicles that Braitenberg describes (1984 Chapter 8 pp.33 onwards) use arrays of photo sensors with aggregates of interspersed filters to perceive and navigate their environment. In addition, modern physical robots often resort to using attached web-cams for navigation. To be able to simulate both Braitenberg's more advanced designs and provide for the simulation of modern physical robot platforms, a virtual eye had to be developed.

Before implementing the virtual eye in the main Java simulation environment, the component was written and tested as a separate application in Flash. The reason for initially using a Flash application for a proof of concept prototype was that it provides an excellent Webcam API and using real image data instead of artificial model data was favourable since the virtual eye that would provide the data did not yet exist in the simulation. In addition, porting the final code from ActionScript 3 to Java was trivial, since both languages have almost identical syntax at this level.

Figure 13 Lateral inhibition prototype realised in Adobe Flash.

Figure 13 shows a screenshot of the Adobe Flash-based proof of concept prototype for the webcam-based vision prototype. The top-left screen shows the source image. The top-right image shows an implementation of a pixel-difference based motion detector. The bottom-left image shows a standard edge detection algorithm. The bottom right image shows the working implementation of Braitenberg's lateral inhibition filter, which enhances edges, while supressing most uniform surfaces and retaining colour information

The same algorithm was ported to the 3D Java monkey engine simulator. By rendering a camera perspective to a hidden texture and accessing the texture via the frame buffer, the RGB colour values for each pixel were directly accessible from the agent's Java code. This allows for the creation of direct sensory-motor connections between the render-to-texture camera viewport and the steering code for the agent.

Feature Extraction through Lateral Inhibition

The Flash prototype successfully demonstrated that Braitenberg's model for lateral inhibition was sound and the resulting image produced by the algorithm matched the description by Braitenberg in that it:

1. Enhanced detail and edges
2. Filtered out "uninteresting" uniform shading
3. Retained colour information

Figure 14 shows the perspective of the virtual agent using its virtual eye. In the simulation environment this small side-window can show the perspective of one single agent at a time. Although only one agent's perspective can be viewed using this window, the total number of internally processed virtual eyes is currently limited by the number of image buffers available on the local graphics hardware.

Figure 14 shows the effect of lateral inhibition, enhancing textured objects and environment boundaries while retaining colour information. Uniformly un-textured surfaces are subdued and therefore not passed to the agent's brain. The purpose of lateral inhibition is to reduce the overall amount of stimulus, while retaining the information that is most important. This notion of *patterns over uniformity* carries through to other aspects of Braitenberg's architecture. For example, competition between elements as seen in lateral inhibition can be used for pattern differentiation in the time-series predicting Ergotrix network (Braitenberg 1984).

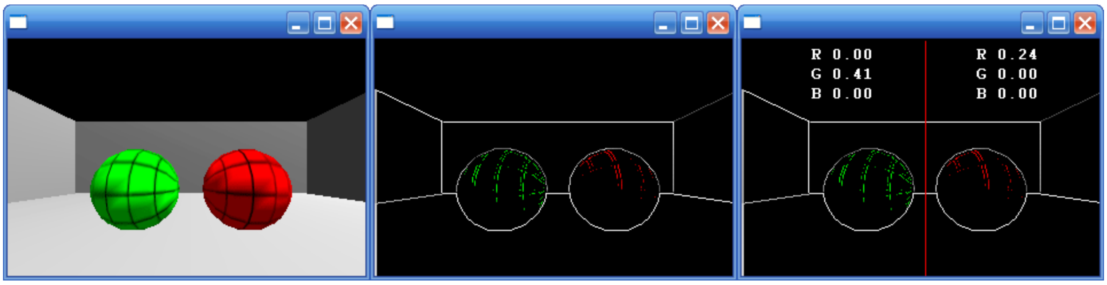


Figure 14 Virtual Eye viewport in the 3D simulator.

Figure 14 shows the perspective through the agent's virtual eye. The first two screens show the raw image and the result of applying the lateral inhibition filter. The third image shows how the single image is split into two halves and individual colour percentages are calculated for each side. Note that the current colour value algorithm is set to ignore equal mixtures such as white and grey by subtracting the colours from each other.

4 ADAPTION OF BRAITENBERG VEHICLES FOR GAMES

This chapter details the adaption of Braitenberg's "Synthetic Psychology" (Braitenberg 1984) into a tractable game AI development approach. It starts with a test of the simulation tools that were created to support Braitenberg-inspired *Vehicle* agent development for games. The second section details the agent models used in the user study, which were developed using the simulation tools and the adapted approach. The final section presents a series of prototypes that demonstrate the use of additional components that extend the capabilities and behaviour repertoire of the agent model.

4.1 TESTING THE SIMULATION ENVIRONMENTS

The first tests conducted using the simulation environments developed for this study established the range of behaviours that are possible with a purely reactive Braitenberg agent architecture. This test also served to evaluate the virtual agent simulation platforms that were created for this project by verifying that Braitenberg vehicles can be simulated faithfully, i.e. a simulated Vehicle should demonstrate the physical properties and the behaviour of the imaginary Vehicle it is based on.

4.1.1 BACKGROUND:

The agent architectures that Braitenberg describes in the early chapters of "Vehicles: Experiments in Synthetic Psychology" (Braitenberg 1984) are constructed from an extremely limited number of components. This simplified approach has the benefit of being well suited for demonstrating the "emergence" of behaviour that takes place when a simple architecture is combined with a complex environment.

The basic set of components consists of wires that transmit "activity" from the agent platform's sensors to the motors and "threshold devices". These threshold devices are simple processing units that are interspersed between the wires to modulate the dataflow between the sensors and the motors. Similar to neurons in a biological brain, this makes threshold devices the basic building blocks for complex non-linear relational mappings between sensory input and actuator output.

In this initial test sensory-motor connections are kept linear, to allow for an easy differentiation between behaviour that these simple direct connections can create. The later architectures that will be presented to users will utilise more complex non-linear and dynamic sensory-motor connections.

4.1.2 *METHODS AND EQUIPMENT:*

This is the first set of experiments that use the 2D Flash and 3D Java simulation environments developed to support this research project. The base architecture that is used for all the experiments is shown below.

A basic Braitenberg Vehicle will have two wheels for locomotion, a left and a right photo-sensitive "eye" and a pair of wires that connect the sensors to the motors of the wheels. Figure 15 below illustrates the simple architecture, both in the form used by Braitenberg (right) and the iconic representation used in the diagrams of this thesis (left):

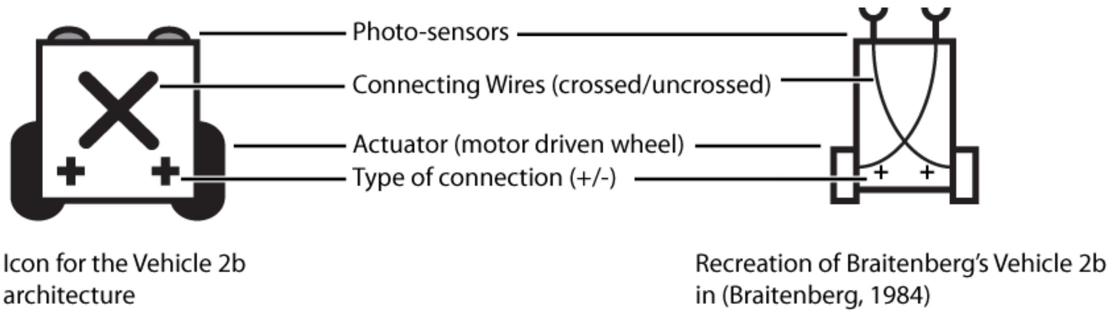


Figure 15 Braitenberg Vehicle (right) and the iconic representation (left)

These connections between sensors and motors can either have an excitatory or an inhibitory effect on the activity of the motors. This, combined with whether the connections are crossed, or parallel in relation to the sensors, determines the type of architecture. The different combinations of wire architecture and effect of the motors means that overall, there are 4 different basic configurations to be tested. These are illustrated in Table 4.

Braitenberg describes wires as transmitters of continuous data. The model he initially proposes in the early chapters (1-5) does not factor in resistances, heat generation or any other physical properties of real wires that could have an effect on the behaviour of the agents if they were implemented as real robots. However, in the later chapters discussing the associative memory components (Braitenberg 1984 Chapter 7 p.29), Braitenberg introduces these physical factors to his model by using the heat-dependent resistance of a wire as a metaphor for memory "decay".

While later agent prototypes may use this idea, the agents used to test the simulation environments only use simple "wires" represented by linear functions.

Table 4 below lists the behaviour spectrum of the agent is based on the four wire architectures and basic "emotions" described by Braitenberg (1984 pp.3-19). The definition of the base behaviours is the first step in defining a "library" of behaviours that an agent designer may combine to create more complex behaviours.

	Configuration	**Code**	**Braitenberg's Labels (1984)**
	1. Crossed-Excitatory	CE	Aggressive
	2. Parallel-Excitatory	PE	Cowardly
	3. Parallel-Inhibitory	PI	Loving
	4. Crossed-Inhibitory	CI	Exploring

Table 4 Braitenberg's four base architectures for a two-wheeled robot

Even with these simplest architectures, Braitenberg's descriptions leave room for interpretation. For example, two of the basic architectures have an inhibitory effect of the motors. While initially Braitenberg refers to their effect on the motors as "slowing down", in later chapters he refers to the agents driving backwards, to escape threats (Braitenberg 1984 p.77). In the interpretation of Braitenberg's vehicle architecture used for this study and the agent architectures that follow, the inhibitory effect of the motors is general interpreted as a negative force. In other words, the "loving" and "exploring" configurations will drive the simulated agents *backwards* if no other force is affecting the motors. To make these configurations create the behaviour that Braitenberg describes a "passive" motor force needs to be applied as well.

As a result, the agent models discussed in the following are slightly more complex than those initially proposed by Braitenberg.

4.1.3 Experimental Design

Figure 16 below is a depiction of the experimental setup and shows one agent facing a single source of stimulus. In this set-up the stimulus is offset to one side. This was done since some of the behaviours to be tested turn the agent toward the source, which would not be visible if the agent is already facing it head on. The offset was also deemed a realistic situation for a dynamic environment where objects are expected to be moving (or seem to be moving due to the agent's own movement) and a perfectly symmetrical start position applies to only a limited set of situations, compared to an offset.

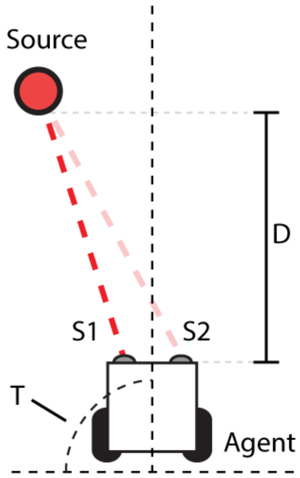


Figure 16 Experimental setup

The changes in the distance to the stimulus D and the angle between the starting orientation and final rotation T were observed while motor activity was recorded.

Each test was then run until the behaviour completed, or until ten seconds passed. For behaviour to be deemed "complete", the agent had to either lose sight of the stimulus and come to a rest, or navigate into a position that it could not get out of. The latter was able to occur due to the boundaries of the environment, so the agent could "get stuck" in corners or be stopped by the wall. In both cases, the data was recorded until a point was reaches where no novel action could be observed.

This recorded data consisted of the velocities of the two wheels of the agent and was used to determine the kind of movement the agent performed in relation to the stimulus and analysed graphically.

2D Simulation Setup

In the 2D simulation the type of sensor used has a significant impact on how the rest of the agent is implemented. The first decision is to choose whether the sensors that sense the source are simple *distance* sensors or whether the sensor should simulate the behaviour of a *photo-sensor* or *thermometer*. While a *distance* sensor simply calculates the distance between its location and a given source, the other two sensors are a bit more involved.

Both thermometers and photo-sensors output low values when far away from the source and increase their output as they get closer. There are several ways of implementing this kind of behaviour in the 2D simulation, but the general principle remains the same, the greater the distance, the lower the value. However, care must be taken with the bounds of the output values since the direct sensory-motor connections can result in erratic agent behaviour. Figure 17 shows the result of using unbounded or wrongly scaled sensor outputs (proving that these agents can be very artistic indeed).

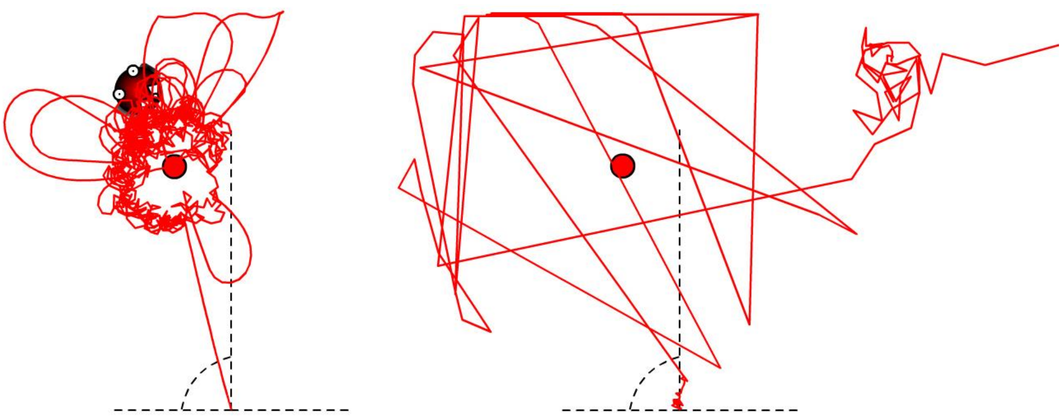


Figure 17 Unbounded sensors output causing erratic agent behaviour

In this experiment there is only a single source, so its "heat" value is calculated using a single fixed "range" value. The distance to the source is subtracted from this range and the resulting output is scaled.

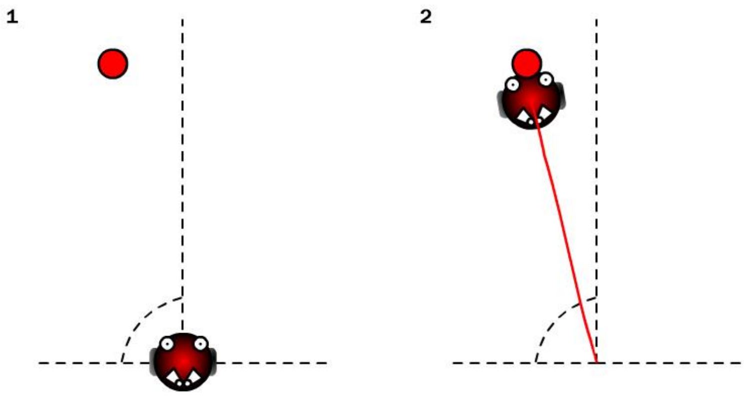


Figure 18 2D Flash simulator test setup

Figure 18 shows the desired test behaviour with bounded and correctly scaled sensor outputs. These early tests also made use of positive side-effect to working in a visual development environment such as Flash. Data visualisation and annotations can easily be incorporated into the simulation for illustration purposes.

3D Simulation Setup

In the 3D simulation, the stimulus is a red ball, textured so that the lateral inhibition filter in the virtual webcam/eye of the agent can see it (a non-textured ball would only appear as an outline, which would provide too little stimulus). The agent is programmed to react only to the colour red in its field of vision. The virtual camera's field of vision is split in half to form two of these virtual photo-sensors, on each for the left and right eye. For each test, the sensor pair is connected to the two motors that drive the left and right wheel, using one of the four base architectures presented by Braitenberg (1984) and shown in Table 4. The agent and the source of stimulus are placed in a distraction free environment (uniformly coloured and un-textured).

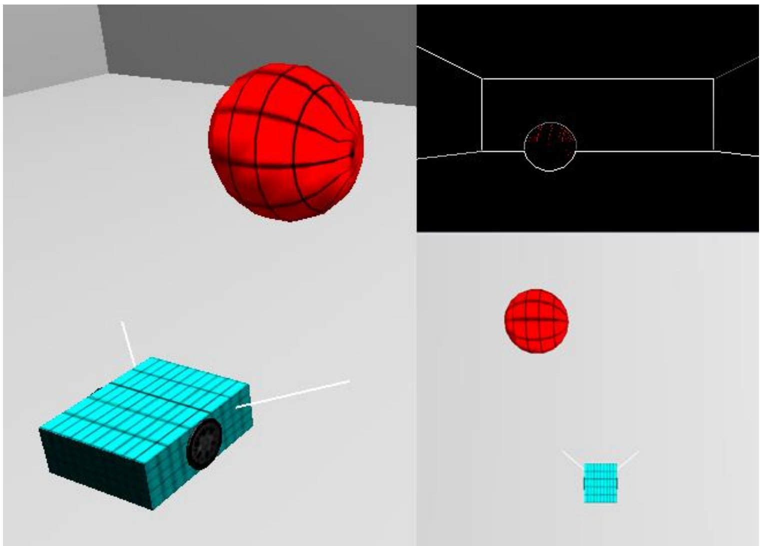


Figure 19 View of the experimental setup in the 3D simulation

Figure 19 above shows the agent model running in the Java based 3D simulation environment. The simulation environment models the agent body and wheels as physical entities in its virtual space. The photo-sensors are implemented as a single virtual webcam pointing forward from the centre of the front of the agent. The view with the lateral inhibition filter applied can be seen in the top right hand corner of Figure 19. The webcam image is split vertically into two halves and the pixel data in each half is summed and averaged to return a single intensity value, simulating the behaviour of a single photocell, while retaining the ability to detect colour in the RGB colour space. While the simulation environment also supports distance sensors (visualised by the white "antennas" protruding from the agent body in Figure 19), these were not used in this experiment to avoid interference with the tested behaviours.

4.1.4 RESULTS: DEFINING A "BEHAVIOUR LIBRARY"

The agents built using the new simulation environments were designed to test the range of different behaviours that the four basic wire configurations can produce. First they were tested on their own (with no other motor signals present). This revealed that the configurations that inhibit the motor signal (crossed-inhibitory CI and parallel inhibitory PI) could be used to create new behaviours that drive the agent backwards, away from the target stimulus. To recreate the expected "loving" and "exploring" behaviours these two configurations require the presence of a "passive" motor signal that initially drives the agent forward and that they could inhibit. Thus, behaviours that only appeared when a basic configuration was paired with another, passive opposite force were called "simple cooperatives".

All of the basic wire configurations and *simple cooperatives* architectures were tested and the resulting behaviour observed. A summary of the observations can be seen in Table 5 below.

	Configuration			Behaviour		
Synthetic Pschology Label	**Crossed (C) Parallel (P)**	**Excitatory (E) Inhibitory (I)**	**Passive Motor Drive**	**Distance to Stimulus (D)**	**Angle to stimulus (T)**	**Easing of movement**
Aggressive	C	E	Neutral	-	-	Speed up
Cowardly	P	E	Neutral	-	+	Speed up
Loving	P	I	Forward	-	-	Slow down
Exploring	C	I	Forward	-	+	Slow down
Suspicious	C	E	Backward	+	-/+	Slow down
Tempting	P	E	Backward	+	+	Slow down
Defensive	P	I	Neutral	+	-/+	Speed up
Disgusted	C	I	Neutral	+	+	Speed up

Table 5 Simple cooperative behaviours

As is the practice in Synthetic Psychology, all of the behaviours have been given "labels". These are basic emotional terms borrowed from psychology and serve to give an intuitive understanding of the behaviour displayed by the agent. The four "new" behaviours that see the agent back away from the stimulus are highlighted in grey.

The "Configuration" column set shows the different combinations possible using the 4 basic configurations. The "Behaviour" column set shows whether the agent moved towards or away from the stimulus and whether it simultaneously turned toward or away from it. The "easing" of the movement is a term borrowed from classical animation, where it traditionally refers to the gradual increase or decrease in animation frames during an animation sequence.

Figure 20 below is an iconographic depiction of the configurations that constitute the range of basic behaviours possible with the differential drive agent platform. The "simple cooperatives" are behaviours that require a "passive drive" on the motors (indicated by a grey arrow on the body of the Vehicle) in addition to the input from the sensory system. For example, the behaviours "Loving" and "Exploring" only work if a passive motor drive is present that has a "stronger" effect on the motors than the sensory input.

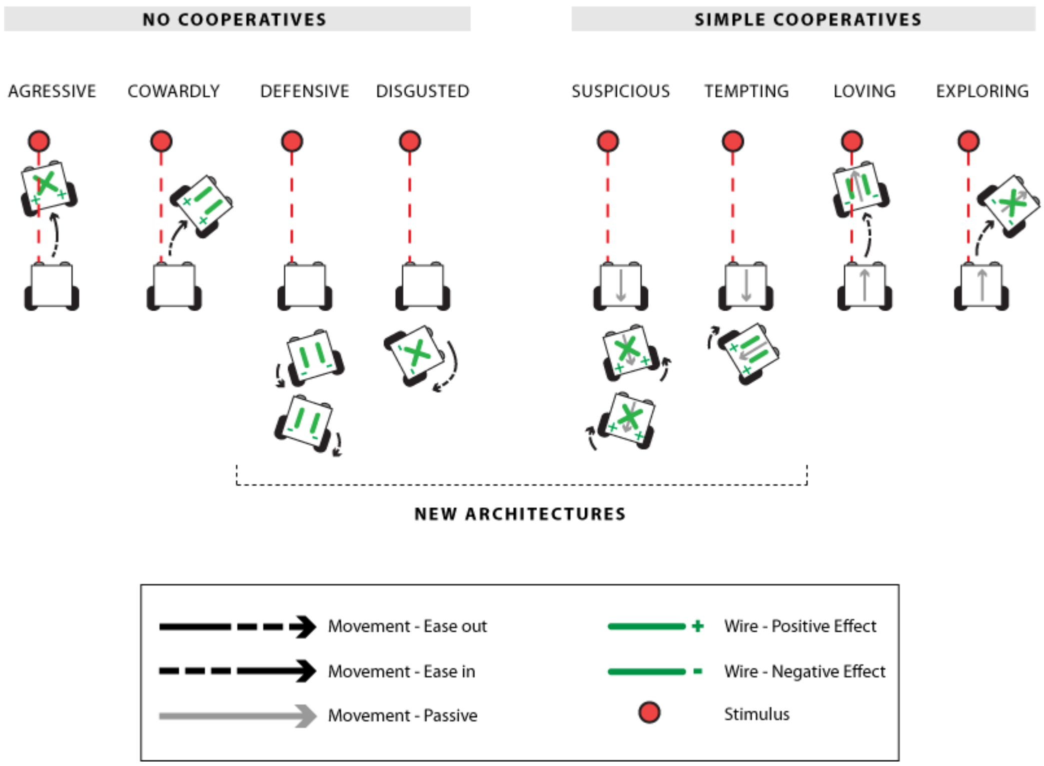


Figure 20 Basic and simple cooperative behaviours of Braitenberg vehicles.

After confirming that the simple agent configurations worked as expected, these "basic" and "simple cooperative" behaviours were also combined and the resulting behaviour observed. As expected, most of the excitatory and inhibitory effects cancelled each other out, but it was interesting to see that combining the "aggressive" with "defensive" behaviour (CE+PI) resulted in the agent simply turning toward the stimulus. In a similar way combining the "disgusted" with the "cowardly" behaviour (CI+PE) resulted in the agent simply turning away.

While some of the behaviours shown in Figure 20 seem similar when viewed in pictures, in motion the differences are more apparent. Table 6 lists the observations made.

These definitions and labels for the base behaviours form the "library" of behaviours that were used to design the more complex behaviours of the gaming agents specified in Chapter 4.2. In chapter 5 an empirical user study investigates how users perceive these behaviours and whether their interpretations match the ones defined here.

Code	**Label**	**Description**
Complex Cooperative Behaviours (combining 2 base architectures)		
CE+PI	Turn toward	Turns toward the stimulus. Will not move even when stimulus gets close.
CI+PE	Turn away	Turns away from the stimulus. Will not move even when stimulus gets close.
Basic and Simple Cooperative Behaviours that drive the agent forward		
CE	Aggressive	Violent. Rushes toward the target and smashes into it at high speed. Drives away, but returns to smash into it again.
PE	Cowardly	Fearful. Immediately turns away from the target and races as far away from it as possible until it hits a wall.
PI+F	Love / Approach	Rushes toward the target, but does not hit at high speed. Instead slows down and slowly circles it.
CI+F	Exploring	Drives toward the target initially, but then turns away and drives off. Comes to a stop before hitting a wall.
Basic and Simple Cooperative Behaviours that Drive the agent backward		
PI	Defensive	Shuffles backwards while facing the target until it backs into a wall.
CI	Disgusted/Revolted Shy/Polite	Difficult to identify. Abruptly turns to face away from the target and moves backwards. Could also be making way, bowing, being polite.
PE+B	Tempting/Taunting	Difficult to identify. The agent drives backwards and always turns to face way from the stimulus. Turns "shoulder" toward the target, but does not move backwards far.
CE+B	Suspicious	Turns toward the target, but backs away. Stops before it hits the wall, keeps its distance if stimulus gets closer.

Table 6 Interpretation of basic agent behaviours

4.2 The Interactive Game Prototype

This section specifies the behaviour of the agents used in the user experiments. The agent controlled by the player and the opponents are all controlled by architectures inspired by the behaviours described in Braitenberg's Vehicles. They use a combination of sensors and simple logic to navigate the environment and react to different situations in the game. The following sections detail how the player controlled and enemy agents were implemented.

This is followed by a documentation of the game prototype itself, the screens presented to the user during the trial and an overview of other prototypes that were also shown to the participants during a purely observational trial.

4.2.1 Game Design

The game itself is designed to be complex enough to be engaging, but simple enough to allow the participants to focus on the behaviour of the agent, without being distracted. Multiple game types were considered which included the "capture the flag" and "combat" game prototypes, which are detailed in section 4.3 of this chapter. In the end it was decided to build a custom prototype for the purpose of evaluating the agents that featured a game scenario that would provide transferable feedback, by focusing on a simple interaction between player and gaming agents that commonly found in a variety of genres.

The design that was chosen is based on an abstract interpretation of the "heist" genre, which has its most notable origin in the game "Pac-Man" (Namco 1980). The goal of a "heist" game is to collect all the pills (or coins) in the playing arena, while avoiding being "caught" by an opponent. The game ends when the player is caught, or when they have collected all the pills.

While the original implementation of this game type in "Pac-Man" featured maze-like arenas, portals and power-ups, the game designed for this study was simplified to make the participants focus solely on the opponent's behaviour. Maze structures, power-ups and multiple agents were removed in favour of making the behaviour of the enemy agent more dynamic. This included a "field of view" mechanic that visualises the area in which the agent can "see" the player and is also used to communicate the agent's internal state by fading in and using red and green colour tints. Use of this mechanic also draws this game design closer to modern representatives of the heist genre commonly referred to as "stealth" games. These have similar goal structures, but typically feature enemies with more advanced sensory systems simulating the aforementioned field of vision and hearing (Kojima 2008; Ubisoft 2012; 2013; Arkane 2012). Also adapted from these modern representatives is the idea of a "patrol route", which sees the enemy agent follow a set path through the environment, challenging players to learn their movement pattern and exploit it.

4.2.2 *PLAYER AGENT AND CONTROLS*

Both the player controlled and enemy agents are implemented as Braitenberg Vehicles. Their movement is based on a differential drive and they use simulated sensors to perceive their environment.

The participant does not control their agent directly. Their mouse inputs control the position of a "target" object, which the agent is programmed to follow. This "target" object is represented as a green indicator, which can take two forms, depending on the type of input control type the player is using. Figure 21 below illustrates the difference between these two control types.

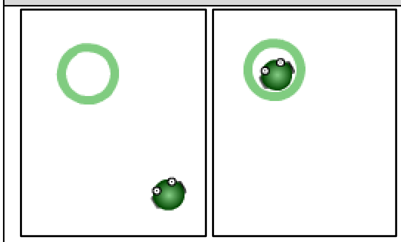
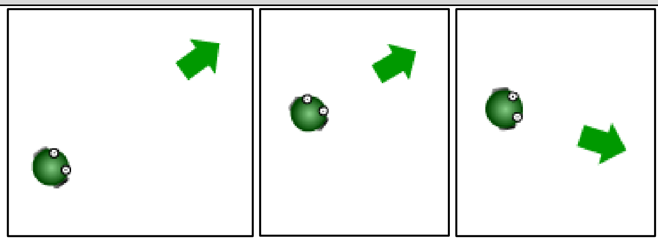
CLICK - Indirect Control Type	HOLD – Direct Control Type
	
When the player clicks on a location in the arena, the target marks that position with a circle. The player's agent will move to that position and stop when it has reached it.	When the player holds the left mouse button down, the target changes into the shape of an arrow to indicate the direction the agent should drive in. The agent will still follow the target as before, but they player now has direct control over its movement and can easily perform evasion manoeuvres.

Figure 21 Player agent: Direct and indirect control types

Player Agent Target Chasing Behaviour

The player agent is programmed to chase the target. This behaviour is implemented using a pair of "distance" sensors and a mixture of Braitenberg's "aggressive" and "love" behaviours, which could also be described as "drive towards and get faster when near" and "drive towards and get slower when near". Note that Reynolds (1999) refers to the “love” behaviour as “approach”- perhaps the more apt term in this scenario. The two behaviours mix and cancel out when near the target to make the agent stop.

$$l = 2 - R + L$$

$$r = 2 - L + R$$

Where l and r are the left and right wheel velocities and R and L are the readings from the distance sensors. Both raw R and L sensor reading were previously scaled by multiplying by 0.01.

During the pilot studies the behaviour was slightly adjusted. A base velocity of 2 was added to make the speed less dependent on the distance to the target (using direct controls near the agent caused it to slow down too much). A failsafe to stop the agent was also added, to prevent the agent from passing over the target and having to turn around. This failsafe simply sets the velocity to 0 when the agent is within 5 pixels of the target.

4.2.3 Enemy Agent Behaviours

The objective of this part of the experiment is to find out how three different agent control mechanisms affect the believability of a character. The three agents presented to the participants vary only in the way that they transition between two basic behaviours. From the perspective of the behaviour designer, the two behaviours are:

1. The default patrolling behaviour, where the agent guards the coins in the environment, “exploring” the environment to seek out clusters of coins, but never stopping.
2. The “aggressive” chasing behaviour, where the agent chases the player’s agent and attempts to catch it.

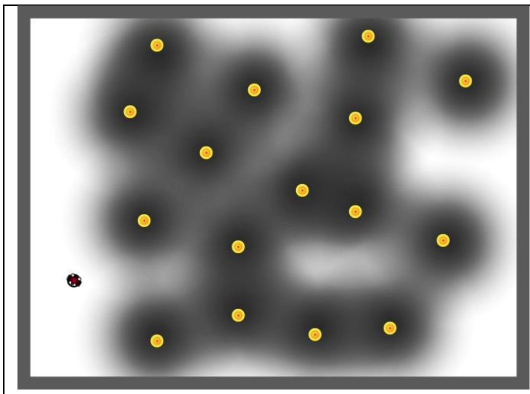
Both of these behaviours use distance sensors. The “exploring” behaviour only targets the coins in the environment, while “aggressive” behaviour only targets the player’s agent. In addition to these two behaviours, the enemy agent also has a pair of simulated infrared sensors, which are used to implement obstacle avoidance. The implementation of these behaviours is detailed below.

Distance Sensors Modes

The coin-guarding and player-chasing behaviours are implemented using a pair of distance sensors used in two different modes.

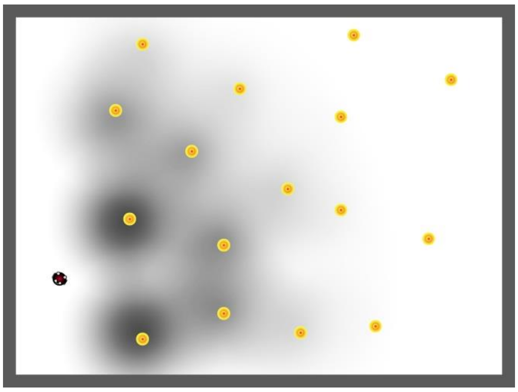
Set to "heat sensor" mode the sensor can target a collection of sources (Coins) in the environment at the same time. The effect that each source has on the output signal is based on its distance to the sensor. A "falloff" variable is used to determine how much the effect of each targeted source decays with distance, effectively setting the range that sources can be sensed in.

The result is that this sensor perceives the environment as a Gaussian distribution or heat map of Coins. The process is illustrated in Figure 22.

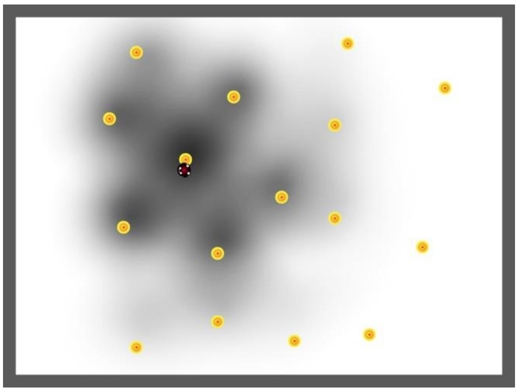


The distance sensors perceive the sources (Coins) in the environment as sources of "heat".

The distance from the sensors is taken into account. This scales individual "heat" readings.



The result is a sensor that can target the nearest individual and clusters of sources.



Using "exploring" behaviour, the agent will slow down near "pockets" of heat and stay near the areas with the most sources.

Figure 22 Agent behaviour: Distances sensor "Heat" sensing

The second sensor setting "distance", also targets all the coins in the environment, but simply takes the average of all the measured distances. In this mode the sensor setting allows the agent to find the "middle" of the space covered by all the available sources.

The key difference to note here is that in "heat sensor" mode the output signal gets stronger as the sensor gets closer to the source, while in "distance sensor" mode the output signal gets stronger the further away the source is.

Guarding Behaviour Implementation

The Braitenberg-inspired behaviours used to implement the coin guarding behaviour of the enemy agent, are the "explore" and "love" behaviours. The "love" behaviour moves toward sources perceived in the distance and slows down when nearby. The "explore" behaviour makes the agent move toward sources, but turn away from them when it gets close.

$$Pl = 2.0 + (\bar{L} \cdot 0.005) - H_R$$

$$Pr = 2.0 + (\bar{R} \cdot 0.005) - H_L$$

Where Pl amd Pr are the "patrol" behaviour's left and right wheel velocities. First a base velocity of 2 on each wheel is set for the agent. For the "love" behaviour, the mean "distance" reading $\bar{L}$ and $\bar{R}$ from the distance sensor on the same side of each wheel is added. Since the sensor output in this mode gets greater with distance the wheel closer to the source will be slower, causing the agent to turn toward the source and eventually stop (when the distance is small enough). The "explore" uses the sensors in "heat" mode, which means that the signal gets stronger the closer the sensor is to the source. Thus the sensor readings H_R and H_L from the sensors on the side opposite the wheel are subtracted, which makes the wheel further away from the source move slower. 0.005 was a suitable scaling for the mean distance value found while prototyping behaviour and during the pilot experiments.

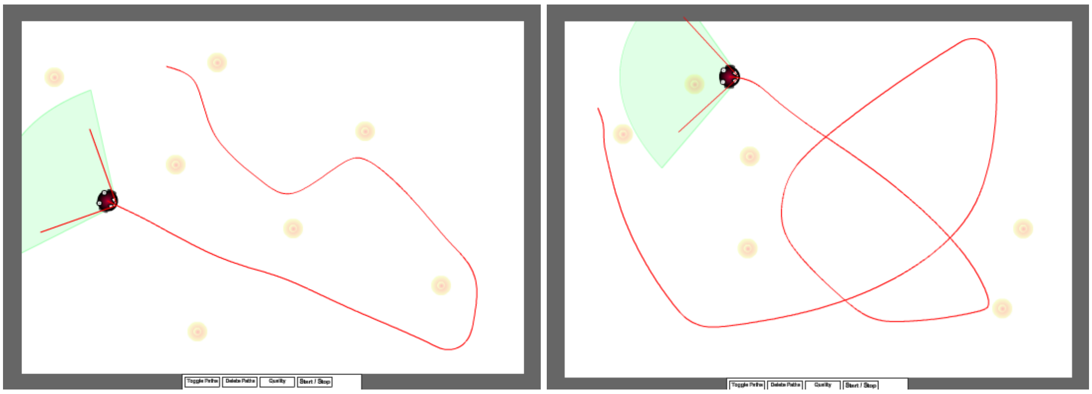


Figure 23 Agent prototype: Adaptive patrol route examples

Figure 23 illustrates the resulting behaviour working in an early prototype of the agent. This prototype was developed to test whether the agent would eventually settle on a patrol route that lets it "see" every source object (Coin) in the arena. This was tested with randomised source positions and a varying numbers of coins. It was found that the agent even adapts to coins being removed from the environment, and can find a new patrol route to cover the remaining coins.

Chasing Behaviour Implementation

The chasing behaviour used the same distance sensor pair as the patrol behaviour. However, instead of tracking multiple targets, this behaviour only tracks the player's agent.

$$Cl = 2.0 + (L \cdot 0.02) + ((900 - R) \cdot 0.003)$$

$$Cr = 2.0 + (R \cdot 0.02) + ((900 - L) \cdot 0.003)$$

Where Cl and Cr are the "chase" behaviours left and right wheel velocities. The behaviour is implemented using Braitenberg's "aggressive" wiring in two ways. The first part controls the velocity of the enemy agent when chasing the player, while the second part was added to increase the turning speed of the enemy when chasing.

The behaviour starts with a base velocity of 2.0 for each wheel. Then the first part, meant to control the speed of the agent is added to the equation. This part simply takes the output of the distance sensor on the same side as the wheel and scales it by 0.02.

The second part of the equation starts with the approximate maximum measurable distance in the arena (based on the arena size of 800x600, 20 pixel wide walls and the agent size) and subtracts the current distance value from the sensor on the other side.

The result is scaled by 0.003. The combined effect is that a low sensor reading on one side will cause a weak input signal to the wheel on the same side, while also adding to the input of the wheel on the opposite side, resulting in the agent performing sharper turns when chasing the player.

Wall Avoidance Behaviour Implementation

This behaviour is always active and prevents the enemy agent from getting stuck on corners, or sliding along walls too often.

It is implemented using both the "fear" and "explore" Braitenberg inspired wirings and uses simulated infrared sensors configured to have a range of 80px. First the distance values are calculated by subtracting the reading from the range of 80. The result was scaled by 0.005 after prototyping.

$$D_L = (80 - IR_L) \cdot 0.005$$

$$D_R = (80 - IR_R) \cdot 0.005$$

Where D_L and D_R are the scaled distance reading from the left and right IR sensor. IR_L and IR_R are the raw sensor readings. 80 is the maximum range of the IR sensors and 0.005 is the scaling factor. The values for D_L and D_R are used twice in the equation to calculate the final wall avoidance motor signals.

$$Wl = D_L - D_R$$

$$Wr = D_R - D_L$$

Where Wl and Wr are the final wall avoidance component that is added to the other active behaviours.

Final Behaviour Implementation with Bias

The final behaviour of the agent is determined by the mix of the above three behaviours *patrolling, chasing* and *wall avoidance.* The impact of the *patrolling* and *chasing* behaviour is controlled by a behaviour bias factor β, which ranges between the two extremes 0 (patrolling) and 1 (chasing). The final input signals for the left and right motor actuators are calculated using the following equations:

$$Vl = 0.1 + (1 - \beta) \cdot Pl + \beta \cdot Cl + Wl$$

$$Vr = 0.1 + (1 - \beta) \cdot Pr + \beta \cdot Cr + Wr$$

Where Vl and Vr are the final input signals to the left and right motor actuator respectively. This implementation allows the agent to perform either patrolling or chasing behaviour, or a seamless mixture of the two, while avoiding collisions.

4.2.4 *Enemy Agent Types*

To recap, the two central connected ideas behind Braitenberg's "Synthetic Psychology" are his "law of uphill analysis and downhill synthesis" and the underlying concept of emergent behavioural complexity. The first suggests that observers will tend to overestimate the complexity of the internal mechanisms controlling a system, if they base their analysis on observations of its behaviour. The second idea is founded in Braitenberg's awareness that the dynamics of the complex interactions between an agent's control systems, the environment and the interpretation of the observer, can result in the identification and labelling of behaviours and control mechanisms that are emergent i.e. are the result of interactions between systems, rather than separately specified components.

This section presents the specifications for a series of agent prototypes. The objective behind developing these agents is or the purpose of conducting a set of user experiments, during which users interact with and observe the agent's behaviour and are subsequently asked to evaluate them using a set of predefined criteria.

The user experiment splits the agents into two groups:

1. To test for emergent behaviour, two simple agent architectures were compared that are identical apart from how they transition between behaviours. The test is to see whether participants identify the transition between the behaviours as a new "emergent" behaviour and assign a new behaviour label to it.
2. Two more complex agent designs were used to test how participants interpret complex systems. Does their interpretation match the specification of the designer? Do they over or underestimate the complexity of the control systems behind the agents?

The specification of the four agents also mirrors the increasing complexity of systems presented in Braitenberg's "Vehicles", as the agent types presented to the participants go from simple, binary behaviour, over mixed behaviours to complex multisensory systems.

The experiment ultimately intends to elicit how user perceptions differ or match the expectations of the AI architect. This is considered a vital part of designing autonomous agents that interact with users from various backgrounds.

The previous section "Enemy Agent Behaviours" showed how a mixture of basic behaviours results in the final movement of the enemy agent. The key component was a behaviour bias factor defined as β. Before the final input signals to the motors Vl and Vr are calculated, the behaviour bias value β is updated. This section describes how the change in behaviour $\Delta\beta$ is implemented in the agent architectures tested by the participants.

The difference between the four agents presented to the participants is in how they transition between the patrolling and chasing behaviour. The default behaviour is referred to as the **primary** behaviour, while the behaviour to which the agent transitioned when it sees the player is called the **secondary** behaviour.

The implementation is detailed in the next section "Agent Behaviour Switching", however the four agent behaviour switching types can be described as the following:

1. **Binary Switch**: The agent immediately switches from "exploring" to "aggressive" when it sees the player. It immediately switches back when it doesn't see the player anymore.
2. **Continuous Switch**: The agent gradually transitions between "exploring" and "aggressive behaviour the longer it sees the player. After losing sight of the player, it gradually transitions back to "exploring" behaviour.
3. **Remap Behaviour**: This agent behaviour is more sophisticated than the previous two in that it uses negative reinforcement to adapt to the player. It generally behaves the same as the "Continuous Switch" agent, smoothly transitioning between the primary and secondary behaviours. However, each time the player collects a coin, the agent shifts its primary behaviour *towards the opposite of what it is currently doing*. So for example, if the agent was "exploring" when the player collects a coin, its primary behaviour will be shifted toward "aggressive". Vice versa, if the agent was "aggressive" toward the player when they collected a coin, it will shift its default toward "exploring". When the primary behaviour is fully "aggressive", the agent will transition toward "exploring" when it sees the player. It is expected that some participants may describe this behaviour as "incoherent".

4. **Adaptive Behaviour:** This agent type was introduced midway through the set of experiments in an attempt to improve the "Remapping" agent based on user feedback. Its design is based on the user feedback from the initial trials and introduces a set of new behaviour features that can be described as "adaptive". Initially this agent will not chase the player if it sees them. Only when it spots the player collecting coins, will it first become "suspicious" of and eventually "angry" at the player.

Architecturally, the binary and continuous switching behaviours are almost identical, except for the way they transition between behaviours. The experiment tests how participants perceive subtle changes to the architecture and whether participants see these two agent behaviours as having different "personalities", or whether they simply point out the difference is in behaviour transition speed.

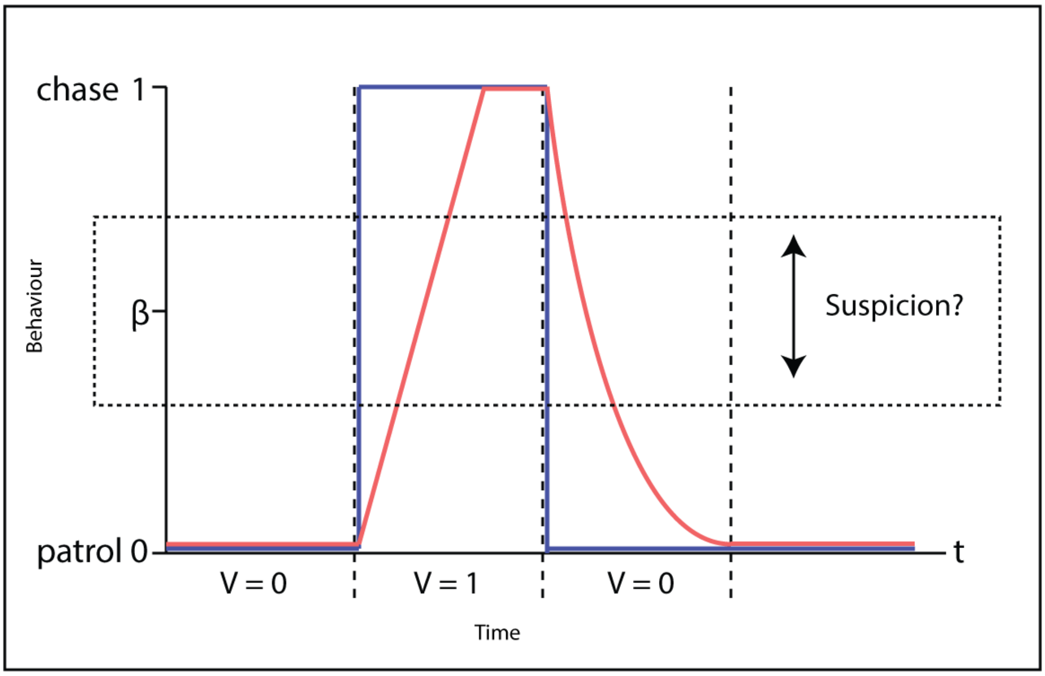


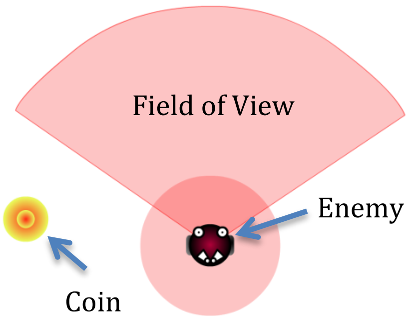
Figure 24 Comparison between agents 1 (blue) and 2 (red)

Figure 24 is a graph comparing the behaviour of agent types 1 and 2 and shows the difference in the response of the agents to spotting the player (V = 1). While agent type 1 switches immediately (binary behaviour states), agent type 2 transitions gradually and gives the potential for the users to observe a third, emergent behaviour state, indicated in the graph by the region labelled "suspicion". The user experiment will test whether participants identify the transition as a separate behaviour and if they do what labels they attribute to it.

The third and fourth behaviours "remap" and "adaptive" are more complex, multisensory systems that were designed to create interesting interactions with the user (gameplay). This part of the experiment was designed to mirror the user-centred design process that is common practice during the prototyping phase of professional games development. An initial design, in our case the "remap" behaviour is presented to users and subsequently re-designed based on their feedback. The "adaptive" behaviour is therefore a refinement of the "remap" behaviour.

Enemy Agent Behaviour Switching

The Enemy agent has two behaviours, a primary behaviour and a secondary behaviour. This is denoted by two architectural configurations that result in individual motor signals that can be sent to the motors. Mixtures of both signals are also possible and define the behaviour of agents 2 and 3. The primary and secondary behaviours tested here are *guard/patrol* (primary) and *aggressive/chase* (secondary).



While the Player Character is outside the Agent's field of view (FOV):

$V = 0$

While the Player Character is within the agent's FOV:

$V = 1$

Two main stimuli can alter the enemy agent's behaviour:

1. **The player enters the enemy agent's field of view (FOV)**, denoted by a visual cone-shaped area in front of the agent. The player's presence in the enemy agent's FOV is denoted by the variable V in the following specifications.
2. **The player collects 'coins'** in the environment to score points. This only has a direct impact on the behaviour of enemy agent 3. Agents 1 and 2 ignore this stimulus. This stimulus is not continuous, meaning it takes the form of atomic events that occur, affect a part of the calculation and disappear within one time step. This event is denoted by the Boolean variable C in the following specifications.

The general form of the equation for this experiment focuses on the change in behaviour$\Delta\beta$. Note that we use $(V-1)$ to denote that the change in β is negative when the agent does not see the player (i.e.$V=0$).

$$\Delta\beta = (V \cdot p) + \big((V-1) \cdot (\lambda \cdot (\beta - \theta))\big)$$

Where:

$\beta := \ Behaviour\ Bias\ (1 = chase, 0 = patrol)$

$\Delta\beta \ := \ Overall\ change\ in\ \beta\ at\ every\ time\ step$

$p^{\beta} \ := \ Behaviour\ Bias\ change. Step\ increase\ in\ \beta, while\ viewed(V = 1)$

$\lambda^{\beta} := \ Decay\ rate\ of\ \beta, while\ not\ viewed\ (V = 0)$

$\theta := \ Baseline\ (bias). \lambda\ always\ returns\ \beta\ to\ \theta.$

$V := Player\ is\ within\ field\ of\ view\ (1 = Yes, 0 = No)$

$C := Coin\ Collected\ (1 = Yes, 0 = No).\ Atomic\ event\ \therefore\ (C_{\ t} = 1) \rightarrow (C_{t+1} = 0)$

The maximum value for β is 1. Behaviour Bias change rate p^{β} is applied as a linear increase. This can cause the value of β to overshoot the maximum of 1 if not prevented. In the Adobe Flash based simulation developed for the user experiment the new value for β is limited to 1 using the $min(a, b)$ function.

$$\beta^{t+1} = \ min\big((\Delta\beta + \beta^{t}), 1\big)$$

Where:

$\beta^{t} := \ The\ Behaviour\ Bias\ at\ the\ current\ time\ step\ t$

$\beta^{t+1} := \ The\ Behaviour\ Bias\ at\ the\ next\ time\ step\ t + 1$

$min(a, b) = \begin{cases} a\ if\ \ \leq b \\ b\ otherwise \end{cases}$

Alternatively the value for $\Delta\beta$ itself can be limited before applying it toβ. The $bound(a, b)$ function is used to either allow $\Delta\beta$ to be set by p^{β} or instead by $(1-\beta)$. Using $(1-\beta)$ instead of p^{β} brings β up to the limit of 1 in instances where$(\beta + p) > 1$.

$$\Delta\beta = V\left(\left(bound\big((\beta + p), 1\big) \cdot p\right) + \left(1 - \left(bound\big((\beta + p), 1\big) \cdot (1 - \beta)\right)\right)\right)$$
$$+ \big((V - 1) \cdot (\lambda \cdot (\beta - \theta))\big)$$

Where:

$$bound(a, b) = \begin{cases} 1\ if\ a \leq b \\ 0\ otherwise \end{cases}$$

Since it was used in the simulation and for the purpose of clarity, the first version of the general equation that uses $min(a, b)$ to limit the value of β will be used to define the following four agent architectures.

Agent 1 Behaviour: Binary Switching

The behaviour tested is a ***Binary switch*** between two behaviours. When the agent sees the player (player within field of view) its behaviour immediately switches from the primary behaviour *patrol* to the secondary behaviour *chase* and stays there until the player leaves the area of visibility V = 0).

$$\Delta\beta = (V \cdot p) + \big((V - 1) \cdot (\lambda \cdot (\beta - \theta))\big)$$

$p = 1$ A constant. Switching from primary to secondary behaviour is instant.

$\lambda = 1$ A constant. Switching from primary to secondary behaviour is instant.

$\theta = 0$ For this agent type θ is constant. The baseline for the primary behaviour never changes

$$\Delta\beta = (V \cdot 1) + \big((V - 1) \cdot (1 \cdot (\beta - 0))\big)$$

Leaving us with the simplified equation for this experiment:

$$\Delta\beta = V + \big((V - 1) \cdot \beta\big)$$

Figure 25 below illustrates the behaviour of agent type 1 during a typical trial. Since the agent only changes its behaviour bias β when it sees the player and does not react to the player collecting coins, only the effect of the variable V changing its state from 0 to 1 and back is illustrated. Note that the effect of behaviour bias change rate p on β is indicated by the rising red line, while the effect of the bias decay rate λ on β is represented by the light green line.

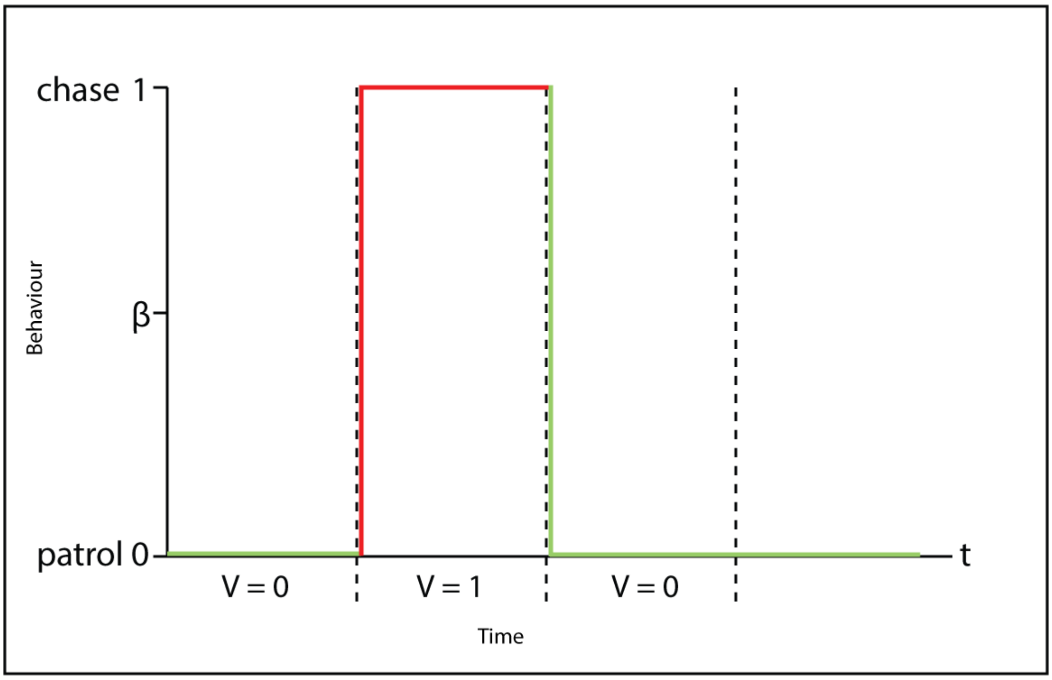


Figure 25 Behaviour illustration of agent type 1 "binary switching"

Agent 2 Behaviour: Continuous Switching

When the agent sees the player, the behaviour starts to slowly change toward the secondary behaviour. When the player leaves the field of view, the behaviour does not immediately switch back to the primary behaviour, but gradually returns at exponentially decaying speed. The result is that the agent is controlled by a mixture of the primary and secondary behaviour.

$$\Delta\beta = (V \cdot p) + \big((V-1) \cdot (\lambda \cdot (\beta - \theta))\big)$$

Value ranges and defaults:

$0 < p \leq 1$ A constant real value above 0 and up to 1, set to 0.003 in our simulation. Bias change rate p controls how quickly the agent switches from the primary to the secondary behaviour.

$0 < \lambda < 1$ — A constant real value between 0 and 1 (exclusive). Set to 0.003 in our simulation. Bias decay rate λ controls how quickly the agent returns to the baseline/primary behaviour.

$\theta = 0$ — Constant value of θ is set to 0. The baseline for the primary behaviour never changes, meaning that the agent will always return to its default *patrol* behaviour such that $\beta = 0$.

$$\Delta\beta = (V \cdot p) + \big((V - 1) \cdot (\lambda \cdot (\beta - 0))\big)$$

Since θ is constant and zero, the formula for this experiment simplifies to:

$$\Delta\beta = (V \cdot p) + \big((V - 1) \cdot \lambda \cdot \beta\big)$$

Figure 26 below illustrates the behaviour of agent type 2 in a typical trial. When the agent sees the player (V = 1) bias change rate p increases the bias until it reaches the secondary behaviour. As soon as the agent loses sight of the player (V = 0) the bias decay λ affect the β, returning it to the primary behaviour at zero. The behaviour bias line β is coloured red when p is affecting it and green when λ is returning it back to the baseline.

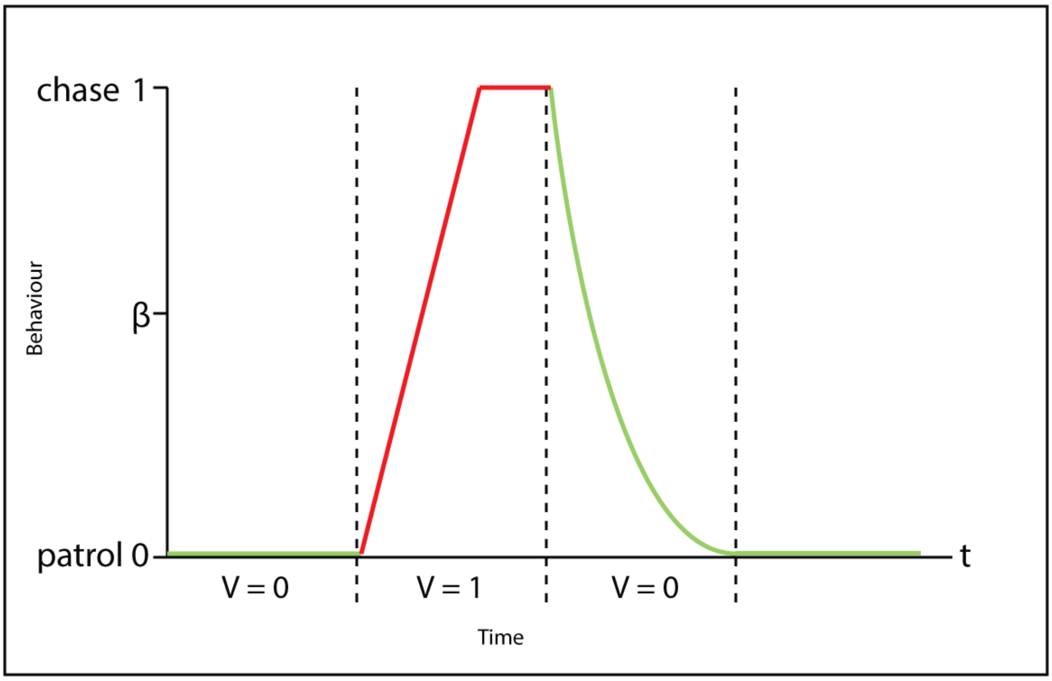


Figure 26 Behaviour illustration of agent type 2 "continuous switching"

Agent 3 Behaviour: Behaviour Remapping

This agent type can switch its baseline behaviour θ from the default primary behaviour value 0 to an arbitrary baseline (between 0 and 1) that the behaviour β will return to when no visual stimulus is present (i.e. $V = 0$).

In this experiment θ is not a constant anymore. Its value is modified each time the player collects a coin in the environment. As a form of negative reinforcement, this signals to the agent that the current behaviour is "bad" and that it would be a good idea to pursue alternative measures. Formally, it shifts the baseline θ of the behaviour away from the current behaviour. We introduce a variable C, which has a default value of 0, but switches to 1 when the player is collecting a coin on the current time step. Under normal circumstances, C will only ever have a value of 1 for one time step (or frame in the Flash animation). This makes $C = 1$ and subsequently the updates to θ an atomic, non-continuous event.

Since the transition point between primary and secondary behaviour is 0.5, if the current behaviour is above 0.5, then θ will be moved down, otherwise θ will be moved up by the constant step size η. The value of $\Delta\theta$ is given by:

$$\Delta\theta = C\left(\eta \cdot \frac{(0.5 - \beta)}{|0.5 - \beta|}\right)$$

Where we introduce new variables and constants:

$\eta :=$ *Constant step size determining the learning rate for baseline shift.*

$C :=$ *Has value* 1 *if player is collecting a Coin during this time step*, *otherwise* 0.

Since θ is now flexible, it needs to be taken into account in the equation for the change in behaviour. The change in behaviour bias p is now controlled by an additional value d that determines whether the behaviour shift will be positive or negative. If base level θ is below 0.5, behaviour change will be positive, if it is above 0.5, change will be negative. If the base level θ is at exactly 0.5, d is still positive. In all other cases The value of d is given by:

$$d = \frac{(0.5 - \theta)}{|0.5 - \theta|}$$

This gives us the final formula for the behaviour of the agent:

$$\Delta\beta = (V \cdot (p \cdot d)) + \big((V - 1) \cdot \lambda \cdot (\beta - \theta)\big)$$

Where:

$\beta :=\ Behaviour\ (1 = chase, 0 = patrol)$

$\Delta\beta\ :=\ Overall\ change\ in\ \beta\ at\ every\ time\ step$

$p^{\beta}\ :=\ Behaviour\ Bias\ change. Step\ increase\ in\ \beta, while\ viewed (V = 1)$

$d^{p} :=\ Direction\ of\ bias\ change\ (1 = Up, -1 = Down)$

$\lambda^{\beta} :=\ Decay\ rate\ of\ \beta, while\ not\ viewed\ (V = 0)$

$\theta :=\ Baseline\ (bias). \lambda\ always\ returns\ \beta\ to\ \theta.$

$V := Player\ is\ within\ field\ of\ view\ (1 = Yes, 0 = No)$

Value ranges and defaults:

$0 \leq \beta \leq 1$	A real value ranging from 0 to 1 (inclusive) e.g. 0.01 that defines the mixture between primary (patrol) and secondary (chasing) behaviour.
$\beta_{\emptyset}\ = 0$	The default at time $t = 0$ for behaviour β is 0.
$0 \leq \Delta\beta \leq 1$	A calculated real value between 0 and 1 (inclusive) e.g. 0.01 that determines the change in behaviour bias β for the next time step.
$0 < p \leq 1$	A constant real value above 0 and up to 1, set to 0.003 in our simulation. Bias change rate p controls how quickly the agent switches from the primary to the secondary behaviour.
$0 < \lambda < 1$	A constant real value between 0 and 1 (exclusive), set to 0.003 in our simulation. Bias decay rate λ controls how quickly the behaviour β decays to the baseline/primary behaviour.
$d \in \{-1,1\}$	A calculated value determining the direction of bias change p. A value of -1 causes p to be negative, while a value of 1 keeps p positive. In both cases the shift in bias caused by p will be away from base behaviour θ.
$0 \leq \theta \leq 1$	A variable real value between 0 and 1 (inclusive) that defines the behaviour baseline. Decay rate λ will return β to θ.
$\theta_{\emptyset}\ = 0$	The baseline for the agent's behaviour starts at 0 at time $t = 0$ but changes when the player collects Coins ($C = 1$).
$0 < \eta \leq 1$	A constant real value between 0 and 1, set to 0.1 in our simulation. Controls by how much the behaviour baseline θ is moved each time the player collects a coin.

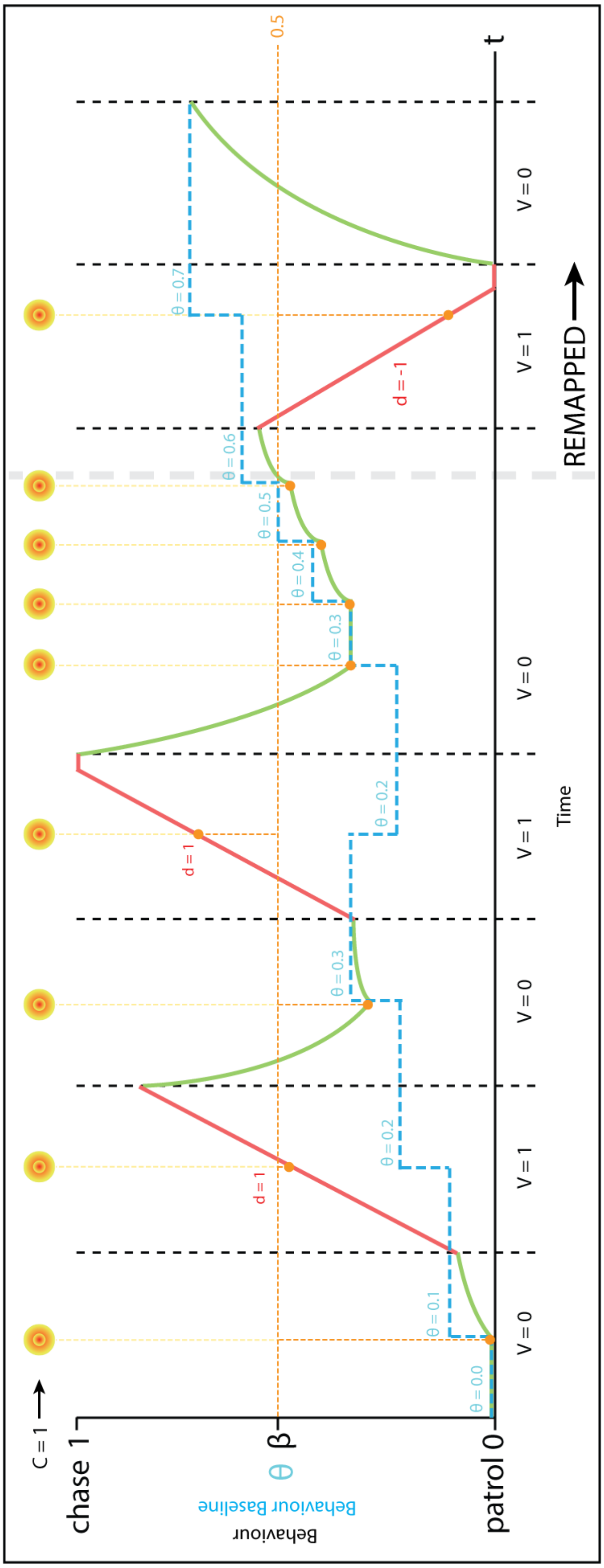


Figure 27 Behaviour illustration of agent type 3 " behaviour remapping"

Figure 27 illustrates the "remapping" behaviour of agent type 3 during a typical trial. The agent initially behaves like agent type 2 in that behaviour bias β is controlled by bias change rate p and the bias decay rate λ. The latter also returns β back to θ. Changes to the behaviour bias β caused by the bias change rate p are coloured red, while the exponential decay of β toward θ caused by λ is coloured green. The baseline θ itself is indicated by the blue dashed line.

The difference to agent type 2 is that θ is affected by the player collecting coins. Each time the player does so (each event is indicated by the coin symbols at the top of the graph), the algorithm checks the current state of β. If β is below the centre threshold of 0.5 (indicated by the orange dotted line through the middle of the graph), then θ is raised by the amount η (0.1 in this configuration). On the other hand, if β happens to be above the centre threshold of 0.5 when a coin is collected, the baseline θ is shifted down by amount η.

The behaviour "remapping" that gives the agents its name occurs when the baseline θ is shifted so much that it is above the centre threshold of 0.5. When this happens, the directional modifier d flips the effect of p to decrease β instead of increasing it. Since the decay rate λ will always return behaviour bias β to the baseline θ, this will now have effectively flipped, or remapped the primary and the secondary behaviours.

Agent 4 Behaviour: Adaptive Behaviour

At the start of a round, this agent type behaves indifferently to the player's presence. It will not switch between the primary and secondary behaviour when the player's agent enters its field of view ($V = 1$). Unlike the "Remapping" type, the "Adaptive" agent's primary and secondary behaviours stay mapped to 0 = patrol and 1 = chase throughout the round. It is essentially an enhanced version of agent type 2 "Continuous Switching" and has the same behaviour change formula:

$$\Delta\beta = (V \cdot p) + \big((V - 1) \cdot \lambda \cdot \beta\big)$$

Two new variables are introduced, that implement two new sentiments that were given the anthropomorphic terms "suspicion" and "forgetfulness". These two new variables affect p and λ, turning them from constants into variables.

"Suspicion" is denoted by s and affects p, which has the effect of increasing the speed at which the behaviour β transitions from the primary patrolling behaviour 0 to the secondary chasing behaviour 1. This only happens when the enemy agent sees the player's agent while they are collecting a Coin ($C = 1\ and\ V = 1$). Change in p is thus denoted by:

$$\Delta p = s(V \cdot C)$$

Where "suspicion" s linearly increases p each time the enemy agent sees the player's agent ($V = 1$) while they are collecting a Coin ($C = 1$). In Boolean (and programmers) terms: "p is increased by amount s if V and C are TRUE". In natural (narrative) language: "The enemy becomes more suspicious, as it gathers evidence that the player is stealing all the coins".

The ranges for the values are:

$0 \leq p \leq 1$	Bias change rate p is a variable real value between 0 and 1 (inclusive) that controls how quickly behaviour bias β changes from the primary behaviour 0 (patrol) to the secondary behaviour 1 (chase).
$p_{\emptyset} = 0$	The bias change rate starts at 0 at time $t = 0$ but changes by amount s when the player's agent collects Coins ($C = 1$) while being seen by the enemy agent ($V = 1$).
$0 < s < 1$	A constant real value between 0 and 1, set to 0.001 in our simulation, linearly increases bias change rate p when the player's agent collects Coins ($C = 1$) while being seen by the enemy agent ($V = 1$).

"Forgetfulness" is denoted by f and affects λ, which has the effect of changing the speed at which the behaviour β decays from the secondary chasing behaviour, back to the primary patrolling behaviour when the enemy agent loses sight of the player's agent ($V = 0$ and $\beta > 0$).

The value for λ is lowered every time the player collects a coin ($C = 1$), regardless of whether the enemy saw the player do so or not. Since λ affects behaviour β by exponential decay, $\Delta\lambda$ the change in λ is also implemented in that way. Change in λ is thus denoted by:

$$\Delta\lambda = C(\lambda \cdot f)$$

Where "forgetfulness" is decreased exponentially each time the player's agent collects a coin ($C = 1$). In natural language: "With every coin that the player collects, the enemy agent will be less likely to forget about chasing them".

The ranges for the values are:

$0.0005 \leq \lambda < 1$	Bias decay rate λ is a variable real value ranging from 0.0005 to 1 that controls how quickly the behaviour β decays toward the baseline/primary behaviour θ.
$\lambda_{\emptyset} = 0.9$	The bias decay rate starts at 0.9 at time $t = 0$ but changes by amount f when the player's agent collects Coins ($C = 1$).
$0 < f < 1$	A constant real value between 0 and 1, set to 0.6 in our simulation. Exponentially decreases bias decay rate λ whenever the player's agent collects a Coin ($C = 1$).

Figure 28 below illustrates the behaviour of the "Adaptive" agent type 4. This agent is also very similar to agent type 2 in that it uses bias change rate p and the bias decay rate λ to control the behaviour bias β. As before, the effect of p and λ on β are indicated by the blue and red colouring of the behaviour bias line. The current values of p and λ are written across the bottom of the chart whenever an event occurs that modifies at least one of them. If only one of them is modified the one not affected is also repeated, but printed in faded colour. Current value of λ is also visualised on the graph.

Compared to agent type 2 this agent starts out with a bias increase p of 0.0 and a high bias decay λ of 0.9. This means that the agent will initially not track the player when it sees them (V=1). After spotting the player collect coins several times ($V \cdot C$)=1 the agent will start to behave like agent type 2.

However, if the player continues to be seen while collecting coins the value of p will eventually get so high that the agent's reaction to seeing the player is almost instant, like agent type 1's binary switching. By the end of the game, the behaviour decay rate λ, (which decreases every time the player collects a coin) will have become so low, that the agent will take a long time to stop chasing the player.

The next section presents an empirical user study using these architectures and behaviours.

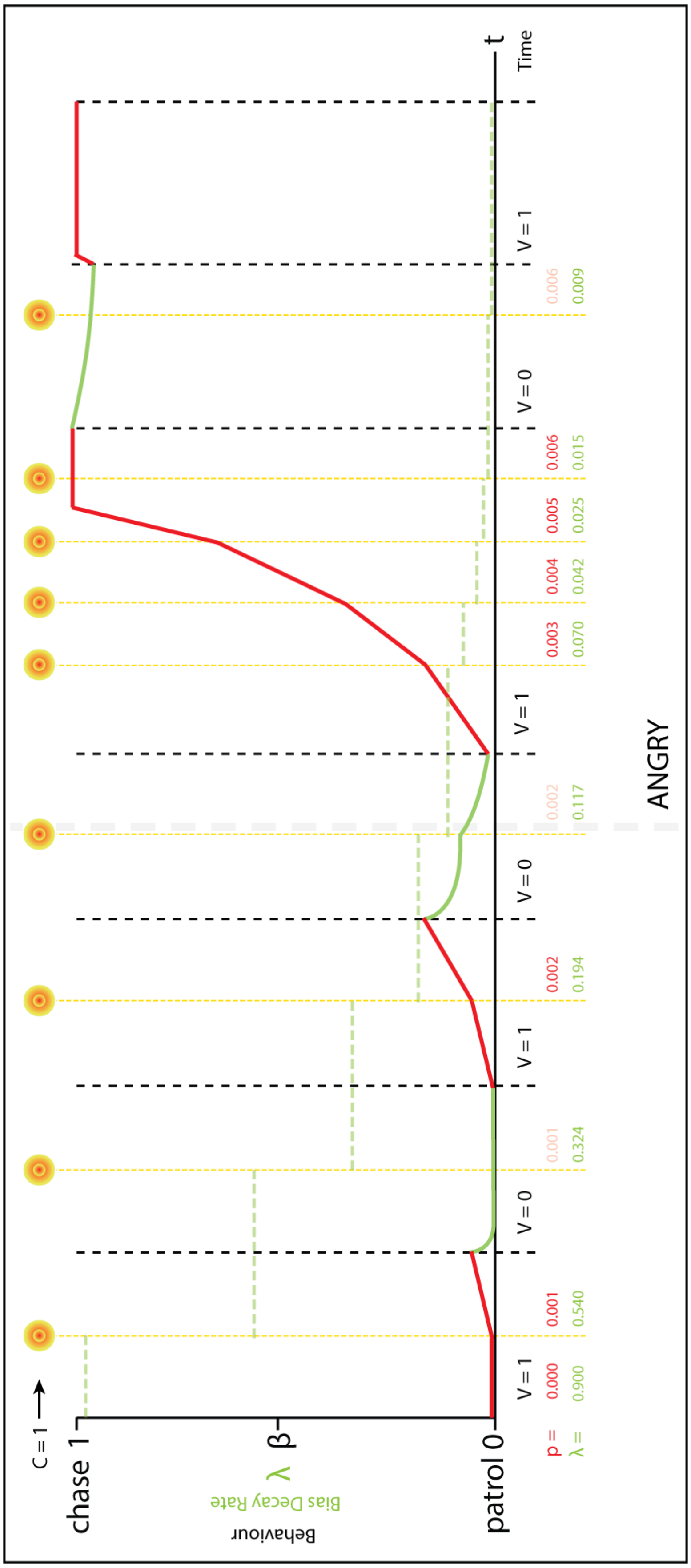


Figure 28 Behaviour illustration of agent type 4 "adaptive behaviour"

4.2.5 *Game Prototype Design and Interface*

The main experimental tool used for this experiment was a game prototype developed in Adobe Flash/Air using the 2D simulation engine introduced in section 3.3.2, which allows for the construction of Braitenberg-inspired Vehicle agents, using a set of body parts, actuators and sensors components.

The entire prototype was implemented in Flash, but the agent architecture and all features could have been implemented in our 3D Game Engine prototype as well. However, the decision was made to use flash and an abstract graphics style in order to simplify the experiment and allow the users to focus on the behaviour of the agents, without being distracted by the graphical representation.

Game Screens

This section gives an overview of the game prototype developed for this study. Note that not all of these screens were shown to the participants, this is indicated where appropriate.

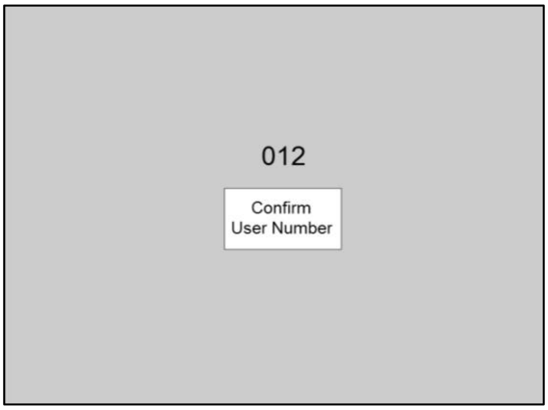


Figure 29 Game prototype: Agent mapping screen

Starting up the game shows the agent mapping configuration screen, shown here in Figure 29. This screen was not shown to participants. It allows the experimenter to configure the mapping of agent behaviour types 0 (Binary), 1 (Continuous), 2 (Remapping) and later 3 (Adaptive) to the labels A, B, and C. Thus the mapping shown of the options shown on the menu screen (see Figure 30 below) is A=0 (Binary), B=1 (Continuous) and C=2 (Adaptive). Note that the behaviour types were labelled differently for the statistical evaluation, where 0 indicates a training trial and the subsequent types have been incremented by one (1 Binary, 2 Continuous, 3 Remapping, 4 Adaptive).

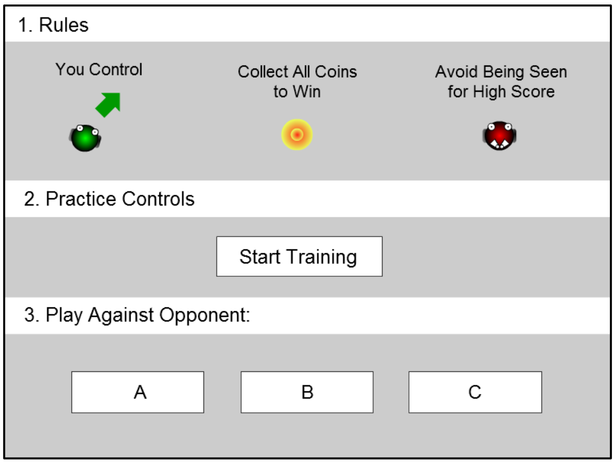


Figure 30 Game prototype: Main menu screen

Confirming the selection opens the main menu screen show here in Figure 30. This screen is split into three sections:

1. Rules
2. Practice Controls
3. Play Against Opponent

These three sections headings were mirrored by the paper instructions given to the participants at the start of the experiment.

The rules are kept purposefully simple and are presented in the style of classic arcade games.

They include the visual representation of the Player agent, Enemy agent and Coins so that players can become familiar with them before playing.

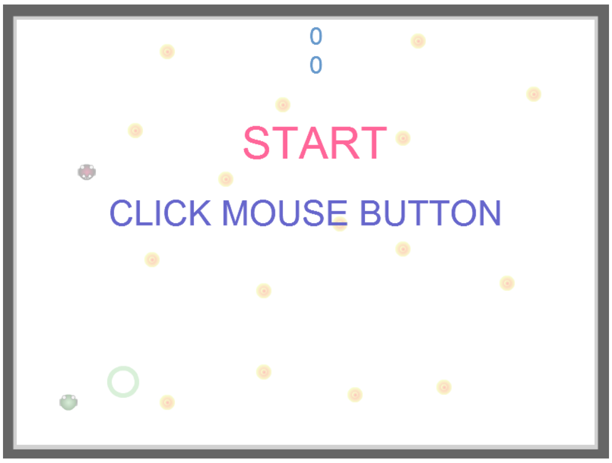


Figure 31 Game prototype: Start screen

Selecting either the training or opponent trials will open up the pre-game "Start" screen, shown here in Figure 31. This screen waits for the user to click anywhere on the screen before starting the round.

In the background the user can already see the position of the coins, the enemy agent and the agent they are controlling. This was designed to allow the users to get their bearings before starting the game.

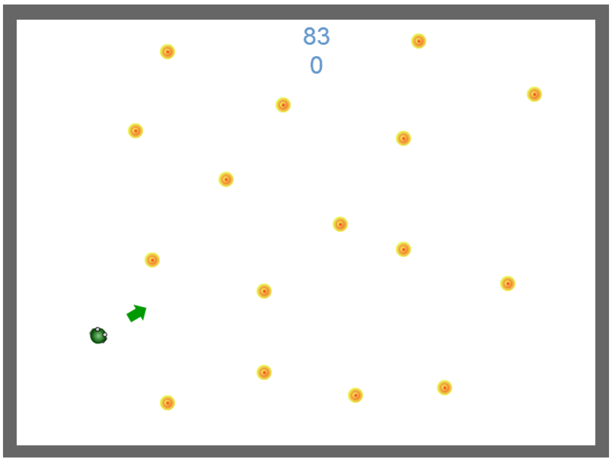


Figure 32 Game prototype: Tutorial arena

Selecting "Start Training" from the main menu lets the user access a training arena, shown here in Figure 32. This arena was included to allow the participants of the study to practice the controls before going up against AI and competing for a high score.

It was also used to give the participants an opportunity to ask questions about the controls or the rules and scoring system of the game.

The objective of this training is simply to collect all the coins on the screen.

The training ends once all of the 16 coins have been collected.

Once the training is complete, the game returns to the main menu screen.

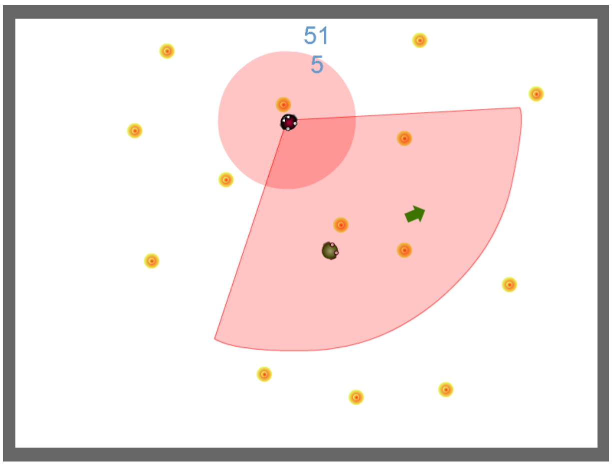


Figure 33 Game prototype: Game against agent

Figure 33 shows a scene from the main game screen, while playing a game against one of the three opponents. The green player agent is currently within the field of view of the enemy agent and is being chased.

The two numbers at the top of the screen indicate the total time that has passed and the time that the participant has spent in being seen (penalty time). Both are indicated in centisecond (10 millisecond) increments, even though the simulation and data output is processed in milliseconds.

Figure 34 Game prototype: Game Over screen

There are two possible outcomes to the game. If the player collects all the coins in the arena, they win. If they get caught by the enemy agent, they lose.

Figure 34 shows the “Game Over” screen, which is shown when the enemy touches the player’s agent. The screen only shows how many coins the Player has collected before they were caught. Clicking a mouse button will return the game to the main menu screen.

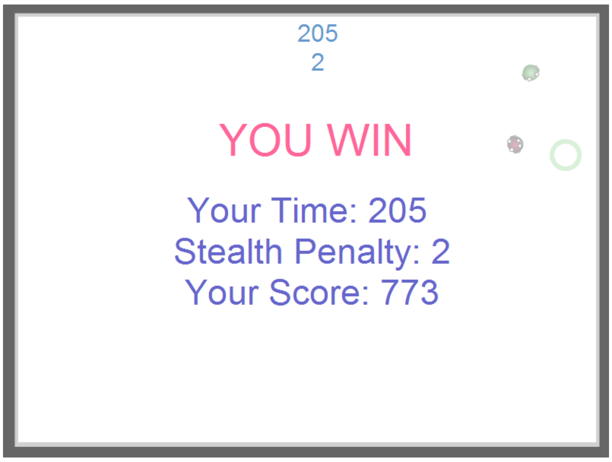


Figure 35 Game prototype: Win screen

If the Player manages to collect all of the coins in the arena without getting caught by the enemy agent, the "Win" screen is shown (see Figure 35).
This screen shows a breakdown of the score including the total time, penalty time and final score, which is based on the both time scores.
Clicking a mouse button will return the game to the main menu screen.

Agent Order and Labels

Traditionally, games are structured in "levels", with subsequent levels becoming more challenging. This can affect the feedback on the agents, since participants might inherently assume the later opponents to be more challenging than the earlier ones. When presenting users with a set of prototypes to evaluate and compare, this sequence effect needs to be considered. Ideally, this study would be performed with enough participants to conduct a between-group study, such that individual participant groups would only test one agent each.

With a total of 7 participants this wasn't feasible, so an attempt was made to minimise bias caused by this sequence effect, by breaking down the perception of linear structure in two ways.

First, the agents are presented to the participants with the generic labels A, B and C instead of numbers. In addition, the mapping between label and agent type varied between participants. Therefore, agent A, the first agent that they played against, was different for each participant. Participants were not told what the actual mapping was.

Another factor that needs to be considered is that participants may judge later agents as easier, due to training and learning strategies. This should not be prevented of course, since we want to see the participants getting better at the game.

So to counter the training effect we let participants retry each agent as many times as they want, after they have played them in sequence once. This ensures that we can look at each participant's "best" trials, to elicit objective data on agent difficulty and participants can go back and compare agents to form their subjective opinion.

Scoring System

A scoring system is used to measure the overall "performance" of the participant when playing against a given agent. It is designed to reflect the participant's ability to identify and exploit the enemy agent's behaviour while completing the objective of the game, collecting all the coins in the playing arena, as quickly as possible.

The score is based on the number of coins collected and the time taken to complete the trial. A time penalty is given when the green player-controlled agent, is within the cone-shaped field of view of the enemy agent. The score formula was designed to encourage the user to follow the rules. It can essentially be described as a combination of coin collection rate and completion percentage. Initially the score did not factor in the completion percentage, which is a fraction of the coins collected over the total number of coins (16 in our game). This resulted in players maximising the score, which was purely the coin collection rate, by quickly collecting a few coins and then driving into the enemy to end the game. With the fraction of coins collected taken into account, completing the game was encouraged.

$$Score = \left(\frac{Coins\ Collected}{Total\ Time + Penalty\ Time}\right) * \left(\frac{Coins\ Collected}{Total\ Coins}\right)$$

4.3 EXTENSION PROTOTYPES

This section presents the extended prototypes that were result of using a Synthetic Psychology approach to develop agents for a variety of typical game scenarios. Some of these were developed alongside the simulation environments to test new features and components. As such they were not specifically designed with user-interaction in mind. However, it was decided to include these in the empirical user study to be evaluated in a purely observational study, where participants were asked to simply describe the behaviours of the agents (see section 5.1 for details on the experimental procedure).

These specifications will provide three types of information. The behaviours will be expressed both in a natural language and mathematical format. In addition, the code from the simulator implementation will be included in appendix 10.1. For the purpose of brevity, a set of utility functions that are re-used in the behaviour definitions are listed below:

Utility Functions

$$d(A,B) = \sqrt{(Ax-Bx)^2+(Ay-By)^2}$$

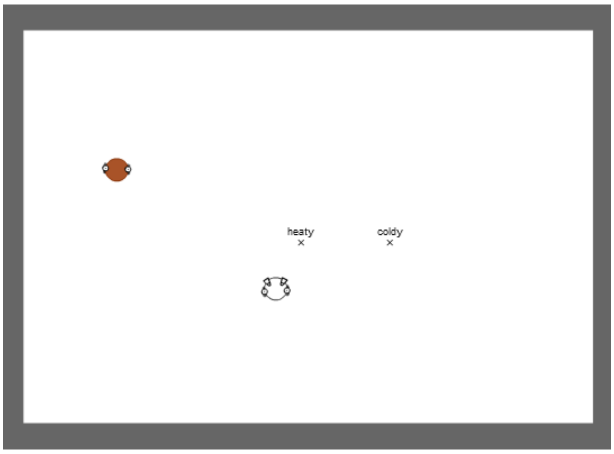
$$farther(P,Q,T) = \begin{cases} 1 \; if\, d(P,T) > d(Q,T) \\ 0 \; otherwise \end{cases}$$

$$larger(a,b) = \begin{cases} 1 \; if\, a > b \\ 0 \; otherwise \end{cases}$$

$$max(a,b) = \begin{cases} a \; if\, a \geq b \\ b \; otherwise \end{cases}$$

Where $d(A,B)$ calculates the distance between two points A and B. The function $larger(a,b)$ returns 1 if a is greater than b. The function $max(a,b)$ returns the higher of the two values. The function $farther(P,Q,T)$ returns 1 if the point P is farther away from point T than point Q, otherwise returns 0.

4.3.1 *HUNTER & PREY PROTOTYPE*



This prototype features two agents. The brown Hunter agent is "aggressive" toward the white Prey agent, which "fears" the Hunter agent and attempts to flee. Both agents "love" the warm source labelled "heaty" and "fear" the cold source labelled "coldy". The Prey agent has additional infrared sensors, which allow it to avoid walls, causing the Hunter agent, who doesn't have these sensors, to bump into walls when chasing the Prey agent.

Hunter Behaviour:

$$v_l = \frac{d(L,W)}{100} + max\left(0, 1 - \frac{d(L,C)}{100}\right) + max\left(0, 1 - \frac{d(R,P)}{100}\right)$$

$$v_r = \frac{d(R,W)}{100} + max\left(0, 1 - \frac{d(R,C)}{100}\right) + max\left(0, 1 - \frac{d(L,P)}{100}\right)$$

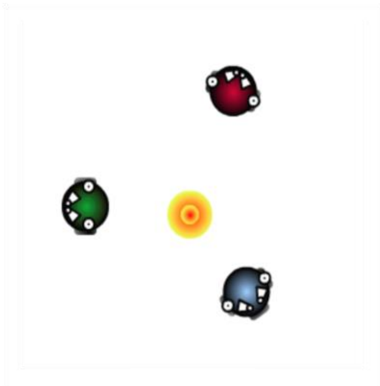
Prey Behaviour:

$$v_l = \frac{d(L,W)}{100} + max\left(0, 1 - \frac{d(L,C)}{100}\right) + max\left(0, 1 - \frac{d(L,H)}{100}\right) + \frac{100 - LeftIR(100)}{100}$$

$$v_r = \frac{d(R,W)}{100} + max\left(0, 1 - \frac{d(R,C)}{100}\right) + max\left(0, 1 - \frac{d(R,H)}{100}\right) + \frac{100 - RightIR(100)}{100}$$

Where v_l and $v_r$denote the velocities calculated for each wheel. L and R are the points denoting the position of the left and right sensors respectively. W is the location of the heat source, C is the location of the cold source, P is the location of the Prey agent and H is the location of the Hunter agent. The values for $LeftIR$ and $RightIR$ are calculated independently using a ray cast interpolation function that takes a $range$ as an input (set to 100 here) tests for collision with walls and objects and returns the penetration distance as an $output$ $(0 \leq output \geq range)$.

4.3.2 CONVERSATION PROTOTYPE



The conversation prototype features three agents and a single source. The agents' behaviour is a competitive mixture of "aggressive" and "love" behaviours. This architecture demonstrates the use of non-linear sensory-motor connections. Aggressive behaviour is strongest within a range of 500 pixels from the agent, while "love" behaviour is strongest within a range of 100 pixels. It inhibits the aggressive behaviour when active. The sensors of the agents measure the average distance to all objects within their range.

The resulting behaviour is that the agents will seek each other out when they are in range using the "aggressive" behaviour. Once in range they will come to a stop (due to the inhibition from the "love" behaviour) and face toward the centre of the group. This behaviour is labelled "attention".

They will also move apart to make space for newcomers to the group. In addition, when an agent moves too close to another, the "love" behaviour's inhibitive effect causes the agent to move backwards. This seemingly defensive behaviour gives the illusion that the agents have a sense of "personal space".

The prototype features an interactive element in the form of a yellow "source" object that users can pick up and dragged around the arena. The agents perceive this object the same way as they perceive each other, allowing users to explore their behaviour.

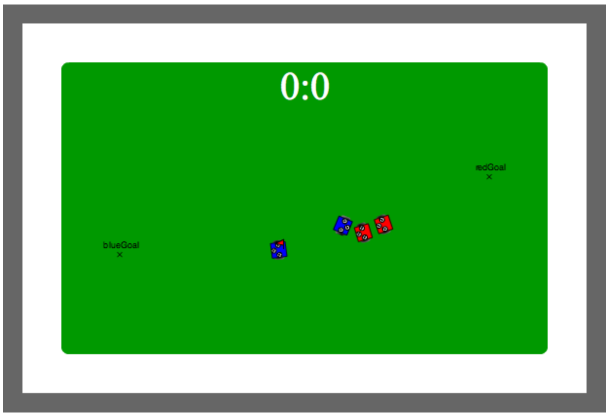
Behaviours

$$v_l = 1 - \frac{\sum_{t=1}^{n} max\big(0, 100 - d(L, T_t)\big)}{20n} + \frac{\sum_{t=1}^{n} max(0, 500 - d(R, T_t))}{350n}$$

$$v_r = 1 - \frac{\sum_{t=1}^{n} max(0, 100 - d(R, T_t))}{20n} + \frac{\sum_{t=1}^{n} max(0, 500 - d(L, T_t))}{350n}$$

Where $d(A, B)$ calculates the distance between two points A and B, v_l and v_r denote the velocities calculated for each wheel. n is the number of targets in the environment. L and R are the points denoting the position of the left and right sensors respectively and T_t is the location of the current target.

4.3.3 *Capture the Flag Prototype*



"Capture the flag" Flash demonstrates the integration of steering behaviours into a finite state machine. As such, the behaviours listed below are activated by discrete changes in the state, denoted by a set of variables. The agents exhibit 3 basic behaviours. Initially the 'aggressive' behaviour steers the agents toward the flag. At the same time, a 'love' behaviour will steer the agents closer toward the area of the enemy goal.

If an agent is the flag carrier, the 'aggressive' behaviour switches to targeting the home goal. Simultaneously the 'cowardly' behaviour steers the flag carrier away from attacking enemy agents. If a teammate is carrying the flag, the agents switch to blocking the opposing team agents by targeting nearby enemies with the 'aggressive' behaviour.

Variables

$T \in \{F, E, B\}$

$C \in \{0,1\}$

$P \in \{0,1\}$

$s = 2.0$

Where T is the location of the Target that the agent is chasing. The Target can be the Flag F, an Enemy E or a Base B. C denotes whether the agent is carrying the flag. P Indicates that a teammate has got the flag and the agent should be protecting. s is the base velocity of the wheels, set to 2.0 pixels per frame in this implementation.

Target Chasing Behaviours

$$v_l = (s \cdot 0.9) \cdot farther(L, R, T) \cdot larger(1, |d(L, T) - d(R, T)|) + s \cdot farther(R, L, T)$$

$$v_r = (s \cdot 0.9) \cdot farther(R, L, T) \cdot larger(1, |d(L, T) - d(R, T)|) + s \cdot farther(L, R, T)$$

This is the basic target chasing behaviour. It modifies the base velocity set for the motors if the difference between the distances to the target measured from sensors L and R is greater than 1. If the sensor L is further from target T than sensor R, then v_l is $(s \cdot 0.9)$.

Flag Carrier Behaviour (+Evasion)

$T = B_h$

$C = 1$

$P = 0$

$$\dot{v_l} = \frac{v_l}{2} + \sum_{t=1}^{Enemies} max\left(0, 1 - \frac{d(L, E_t)}{100}\right)$$

$$\dot{v_r} = \frac{v_r}{2} + \sum_{t=1}^{Enemies} max\left(0, 1 - \frac{d(R, E_t)}{100}\right)$$

When the agent carries the flag, the target T is switched to their home base B_h. The current value of v_l and v_r is halved (to give the flag a "weight") and the effect of the "Evasion" behaviour is added. The "Evasion" behaviour implements a "heat" sensor. The $max(a, b)$ prevents the measurements from going out of bounds. The "heat" values for all $Enemies$ are summed and added to the respective wheel velocities.

Protect State

$T = E$

$C = 0$

$P = 1$

When another agent on the same team is carrying the flag ($C = 1$) the state machine sets $P = 1$ and sets an Enemy E as a target T. In this implementation, an agent will always try to block the Enemy agent with the same team (jersey) number. The behaviour remains the standard chasing behaviour.

Chase Flag Behaviour

$T = F$

$C = 0$

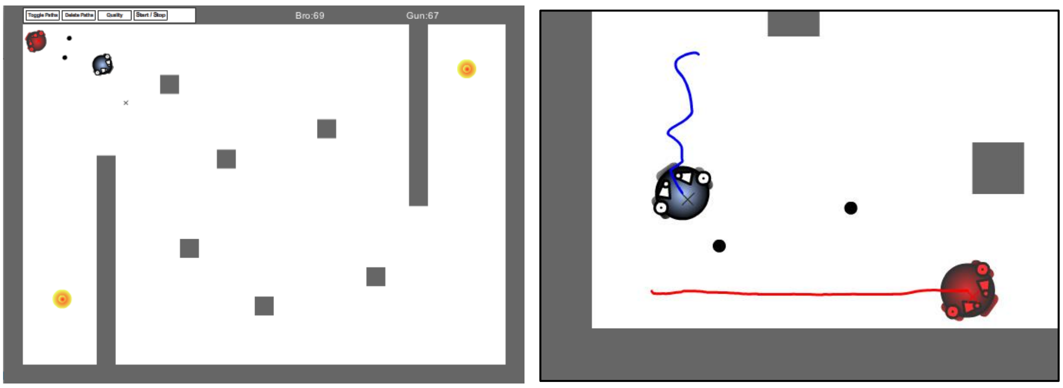
$P = 0$

$$\dot{v_l} = 0.7 \cdot \left(v_l + \frac{d(L, B_e)}{100}\right)$$

$$\dot{v_r} = 0.7 \cdot \left(v_r + \frac{d(R, B_e)}{100}\right)$$

The flag chasing behaviour sets the main target T to the flag F. Simultaneously, a "love" behaviour that targets the enemy base B_e is added to the current velocity on the motors. The overall velocity is scaled by 0.7.

4.3.4 *Combat Prototype*



This "combat" shooting game involved two agents fighting. This prototype implements an additional "strafing wheel", which allows the agents to move laterally to their heading. "Strafing" sideways to avoid bullets, while returning fire, is a typical behaviour for characters in modern shooting games (2D or 3D). The implementation of the strafing wheel is described in section 3.3.2 "2D Flash Simulator Agent Kinematics". While strafing is mainly used to avoid bullets, the normal differential drive is used to navigate the environment.

Two additional systems are included in this prototype. Similar to Walter's tortoises (Walter 1953) the agents return to their "base" sources when their energy level reaches a threshold of 50 points. Touching the "base" gradually recharges their energy. At 100 the agent resumes chasing its opponent.

The second system works to solve a problem that the agents regularly encountered. Focused on attacking each other, the agent would get stuck in corners and keep attacking each other until their energy reaches the "return to base" threshold. To make their behaviour more interesting and dynamic the agents drop a "marker" if they stay stationary for one second. The agents are programmed to "fear" these markers, forcing the agent to move to a new location. This system resolves the majority of stand-off situations between the agents, creating more dynamic behaviour and position changes.

Variables

$H = 100$

$T = \{E, B\}$

$A = \{0,1\}$

Where H is the health of the agent and T is its target. Two variables are controlled by a state machine. Target attribute T determines whether the agent chases their enemy E or returns to base B to recharge its energy. If the health falls below 50, T is switched to B. If health is recharged to 100, T is switched back to E. Attack variable A is 1 when $T = E$ and 0 when $T = B$ and controls the combat-specific behaviours.

This behaviour description is split up into components for clarity. See the definitions below.

Differential Drive Behaviour

$$v_l = a_l + l_l + (c_l - c_r) + f_l$$

$$v_r = a_r + l_r + (c_r - c_l) + f_r$$

The differential drive is controlled by a combination of simple behaviours. Note that the wall avoidance "cowardly" behaviour c is applied twice to each wheel, once as a parallel excitatory connection (cowardly) and once as a crossed inhibitive connection (defensive). This causes the agent to turn away sharply from obstacles. The individual behaviours that combine to control the differential drive are detailed below.

Aggressive toward target

$$a_l = \frac{max(0, 500 - d(L, T))}{500}$$

$$a_r = \frac{max(0, 500 - d(R, T))}{500}$$

Defines a "heat" sensor with a range of 500 pixels. L and R are the left and right sensors, T is the current target of the agent.

Love Target

$$l_l = -\frac{max(0, 200 - d(L, T))}{100}$$

$$l_r = -\frac{max(0, 200 - d(R, T))}{100}$$

Defines a "heat" sensor with a range of 200 pixels. L and R are the left and right sensors, T is the current target of the agent.

Cowardly and Defensive Collisions

$$c_l = \frac{50 - LeftIR(50)}{200}$$

$$c_r = \frac{50 - RightIR(50)}{200}$$

$LeftIR$ and $RightIR$ are simulated infrared distance sensors. Their output value is calculated a ray cast interpolation function that takes a $range$ as an input (set to 50 here). If any object collides with the "ray" within this range, the sensor returns the penetration distance as an $output$ ($0 \leq output \geq range$).

Fear Marker

$$f_l = \frac{max(0, 100 - d(L, M))}{100}$$

$$f_r = \frac{max(0, 100 - d(R, M))}{100}$$

Where L and $R$are the left and right distance sensors respectively. M is the marker dropped by the agent if it remains immobile for 1 second.

Strafing Wheel Behaviour

In addition to the differential drive, the combat agents make use of a "strafing" motor that allows it to move laterally and avoid incoming fire. Positive signals sent to this actuator move the agent right, while negative signals move the agent left. The strafing motor is controlled by two competing behaviours.

$$s = c_s - e_s$$

Where s is the signal sent to the strafing wheel, c_s is an additional obstacle avoidance behaviour and e_s lets the agent avoid bullets using its simulated IR sensors.

Additional Obstacle Avoidance

$$c_s = \left(5 - \frac{LeftIR(50)}{10}\right) - \left(5 - \frac{RightIR(50)}{10}\right)$$

This behaviour uses simulated $LeftIR$ and $RightIR$ to provide an additional obstacle avoidance mechanism to cooperate with the c_l and c_r behaviours that were defined for the differential drive. The range at which this behaviour becomes active is set at 50 pixels.

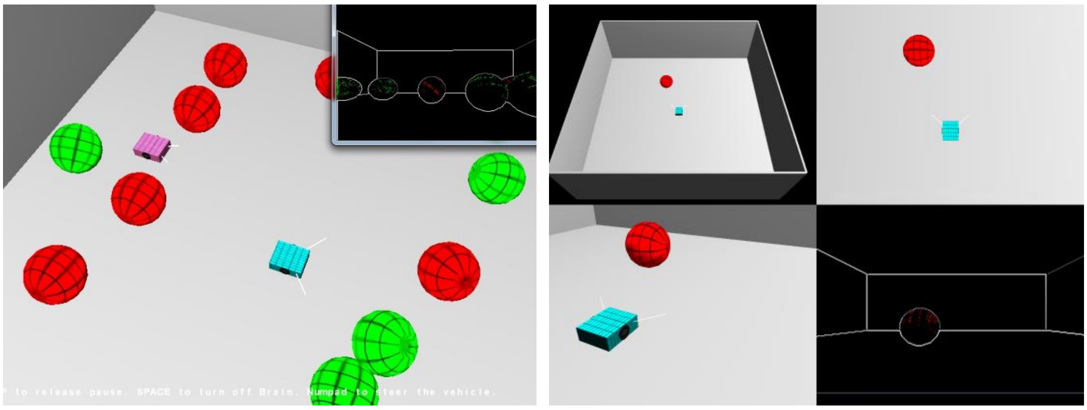
Bullet Evasion

$$e_s = \frac{max\big(0, 50 - d(R, S_E)\big)}{10} - \frac{max\big(0, 50 - d(L, S_E)\big)}{10}$$

This behaviour uses the strafing wheel to evade the shots fired by the enemy agent S_E. It implements a "heat" sensor. The range at which the agent starts evading is set at 50 pixels.

Original in Colour

4.3.5 3D Agents Prototype Video



The 3D agent prototype uses the Java Monkey 3D engine and ODE physics to simulate two agents in a physical environment filled with green and red balls. The features of the 3D simulation engine are described section 3.3.3 “3D Java Simulator”.

The PINK agent explores the environment and avoids obstacles using its distance sensors. This behaviour is implemented using a simple state machine that causes the agent to turn until the distance sensors are not registering an obstacle anymore.

The BLUE agent uses a virtual eye with a Braitenberg inspired lateral inhibition filter to navigate the environment (see section 3.3.3 “The Virtual Eye”). It “fears” GREEN balls and is “aggressive” toward RED balls. Since the pink agent is also slightly red, the BLUE agent will be attracted to it, albeit not as strongly as to the RED balls.

To see if participants can tell the difference between the two agents based solely on their behaviour, they are not told that the agents have different control mechanisms. They are also not shown the virtual eye’s perspective window.

The Pink Agent

This agent uses a simple state machine to control its behaviour. The motor signals are discrete and not influenced by the strength of sensory stimulus. It drives at a constant speed (3) until the simulated infrared sensors collide with an obstacle.

The simulated infrared sensors $LeftIR$ and $RightIR$ output a “distance” measurement when colliding with an object. They are configured to output their set range (22) when not colliding with an object. The state machine has four states:

State Table

Nr.	Action	Condition	Effect
0	Neutral	$RightIR < 5$ and $LeftIR < 5$	$v_l = 3, v_r = 3$
1	Turn Left	$RightIR \leq 5$ and $LeftIR > 5$	$v_l = 0, v_r = 3$
2	Turn Right	$RightIR > 5$ and $LeftIR \leq 5$	$v_l = 3, v_r = 0$
3	Turn Around	$RightIR \leq 5$ and $LeftIR \leq 5$	$v_l = -30, v_r = -60$
4	Back Away	$RightIR \leq 2$ and $LeftIR \leq 2$	$v_l = -20, v_r = -20$

Table 7 State table for simple reactive agent

While states 1 and 2 are implemented to be exclusive, state 4 was implemented to lead into state 3. When the agent drives into a wall, state 4 will cause it to back away far enough for state 3 to trigger and turn the agent around and resume driving.

The Blue Agent

This agent uses the virtual eye that was introduced in section 3.3.3. This sensor renders a view from the perspective of the agent. A lateral inhibition edge-enhancement filter removes "uniformity" from its view and allows the agent to focus on important details. Unlike classical edge *detection* filters, which typically remove colour from their output, lateral inhibition is an edge *enhancement* filter that retains colour information.

The implementation of the sensor used for this agent splits the field of view in half as depicted in Figure 14 (in section 3.3.3). For each half of the viewport the average intensity for each of the three colour components in the RGB (Red Green Blue) spectrum is computed.

$$\overline{Red} = \frac{1}{n * 255} \cdot \sum_{p}^{n} Red_p$$

Where *Red* is the red RGB colour value ranging from 0 – 255, *p* is each individual pixel and *n* is the total number of pixels. The output is three colour component values for each half of the field of view: R_l, G_l, B_l and R_r, G_r, B_r . These were then used to implement a set of simple behaviours.

Behaviours

$$v_l = (R_r \cdot m) - (G_r \cdot m) - (RightIR(5) \cdot 4) + LeftIR(5)$$

$$v_r = (R_l \cdot m) - (G_l \cdot m) - (LeftIR(5) \cdot 4) + RightIR(5)$$

Where v_l and v_r are the signals sent to the left and right motors respectively. R_r, $G_r$and R_l, G_l are the average red and green values of the right and left half of the field of view. m is a scaling factor that was used to tune the colour values. $LeftIR(5)$ and $RightIR(5)$ are the simulated infrared sensors, which are set to have a range (and maximum value) of 5.

4.4 DISCUSSION

The prototypes developed in this chapter show the wide variety of behaviours that are possible with Braitenberg inspired bottom-up agent architectures that go beyond steering behaviours. The focus on combinations of behaviour to create *emergent* behaviour is a constructionist design approach promoted by Braitenberg and lies at the heart of Synthetic Psychology. This has led to the development of interesting, dynamic behaviours, such as the adaptive patrolling behaviour in the gaming agents or the conversational agents' awareness of 'personal space'.

In addition to developing new sensory components and actuators that can be used in other agents, the agent designs themselves have added to the "library of behaviours" that can be used by any agent based on a similar platform. While the focus here was solely on virtual agents, future work should explore whether some of these can inform the development of physical agents as well.

The development of these architectures has shown that modularity and re-usability is one of the greatest strengths of this agent development approach. Starting from the simplest possible behaviours, new components were developed and configurations discovered that were subsequently used in increasingly complex constructs. Where interference between behaviours became a problem, the architecture was easily integrated with traditional finite state machine systems.

Handling behavioural complexity was found to be the main challenge during the development of the architectures. While some formal methods were used to specify and analyse agent behaviour, the most effective tests remained observational. Further work should test the incorporation of new formal analysis methods for Braitenberg agents that are currently being developed by other research groups (Salumae 2012; Rano 2014a; 2014b).

The psychological terms proved helpful as a mnemonic and label for the behaviours, in that they managed to reduce their complexity, while still retaining an intuitive understanding of their dynamic properties. However, these may only be intuitive to the developer. The empirical study that follows investigates to what degree users that interact with these agent architectures without prior knowledge of these labels, are able to identify them solely by interacting with them in a gaming context.

The purpose of that study is to compare the design intent underlying the agents documented in this chapter, with the observations from the users. This comparison will give insight into the degree to which these labels are indicative and intuitive while suggesting refinements and additional interpretations.

5 EMPIRICAL ANALYSIS OF AGENT BEHAVIOUR

This chapter presents the conduct of the empirical study carried out to test the agents created using the "Synthetic Psychology Approach" for their effect on users – players of a game prototype that pits participants against a series of agents in an adversarial context.

The first section describes the procedure and design of the testing materials such as the questionnaire and testing schedule. The second section presents the findings from the experiment. The third section discusses the findings with respect to the initial research questions of the thesis.

5.1 PROCEDURE AND QUESTIONNAIRE DESIGN

This section discusses the procedure and questions used to evaluate the agent prototypes. Further details regarding particular design choices can be found in the Methodology chapter in section 3.2.3, which discusses the pilot study.

Demographics Questions

At the beginning of an experiment session the participant would fill out a demographics form asking for standard demographics data on the participant, including gender, age and education level and professional background.

The form also asked whether the participant had previously worked or studied in a field relating to artificial intelligence, animation or psychology. This data was of interest to the experimenter since according to Ronald and Sipper (Ronald et al. 1999a; 1999b; Ronald and Sipper 2001), prior experience in these fields could affect how "emergent" the agent behaviour may seem to the participants. In our qualitative study this would be reflected in the assumptions that participants make about the agent's internal mechanisms.

Further demographic questions included a series of statements regarding the participants' experience with video games. These were included based on the most common responses gathered during feedback in the pilot study.

- I don't play video games at all
- Someone close to me plays them, but I don't
- I consider myself a "casual gamer" (e.g. Solitaire, Mobile Phone games, Facebook games)

- I play games regularly, but prefer other hobbies
- Games are my favourite hobby
- I work with games, or want to make games

The final question was directed to participants with prior experience in games:

- If you do play games, what would you consider to be your favourite game genres?

Previous experience with certain genres can mean that participants are familiar with the type of controls used or may have played games with similar rules. This can have an effect on how quickly participants develop strategies to complete the game and how much practice they need to master the controls.

Individual Agent Questions

The questionnaire used for the evaluation of the individual agents is based on the format suggested by Gomes (2013). Eight statements are presented on a Likert scale ranging from 0 "Totally disagree" to 5 "Totally agree". A scale with even number of choices was chosen to encourage users to make a decision toward one of the sides.

Each question pertains to one of the believability criteria from Loyall (1997). The letter X was replaced with the particular label of each agent:

- Opponent X perceives the world around him/her
- It is easy to understand what Opponent X is thinking about
- Opponent X has a personality
- Opponent X's behaviour draws my attention
- Opponent X's behaviour is predictable
- Opponent X's behaviour is coherent
- Opponent X's behaviour changes according to experience
- Opponent X interacts socially with other characters

In addition the agent evaluation questionnaires included a set of three open questions, asking the participant to use their own words to describe the enemy agent's behaviour under different game circumstances (see sections 4.2 for a description of the game design:

- When Opponent X didn't see you
- When Opponent X saw you
- When you Collected coins

Game and Comparative Agent Questions

The final page of the questionnaire was to be completed after testing all three agents. It presented questions on the game and comparative questions about the three agents.

The first two questions were presented on a Likert scale in the same format as the believability criteria (0-5):

- My Character was easy to control
- The objective of the game was clear

The Comparative questions about the agents presented three statements about the agents and allowed the participants to choose one of the agent labels as their answer:

- I had the most fun playing against opponent (A,B,C)
- The most difficult opponent was (A,B,C)
- The easiest opponent was (A,B,C)

The first open question about the game was meant to reveal any other reasons for replaying the agent, such as the participant's approach to testing the agents (comparison, unsure about behaviour etc.).

- Which opponents did you choose to retry and why?

The final open question simply asked the participants how they thought the game could be improved.

- How could the game be improved?

Informal Prototype Observation Study

After the formal user observation part of the experiment, participants were asked to evaluate the set of 5 "extension prototypes" specified in chapter 4.3. In this case the participants' task was to simply observe and describe the behaviour of the agents they saw. Even though this was primarily an observational study, some of the prototype scenarios included limited user interaction features. This included interactive "source" objects in the environment to which the agents were sensitive. The participants could pick up and move with these to observe the agent's reactions.

5.2 USER STUDY FINDINGS

This qualitative user study does not aim to make claims of statistical significance, but to collect evidence surrounding the issue of measuring the believability in virtual characters and to form a perspective on the relationship between the AI programmer's design intent and the participant's interpretation of behaviour. Statistical data is presented solely to give an overview of participants' responses and is referred to in the text to support the qualitative data, which is the focus of the study. Due to the sample size the data may not indicative of trends, but the methods for presenting the data inform analysis methods in any future quantitative study.

Two terms are frequently used when discussing the vocal feedback given by the participants. The first is the "mental model" that participants use. When prompted to explain the agents' behaviour, the participants build up a model of the components and mechanisms that produce the behaviour. It is Braitenberg's (1984) theory, that participants will overestimate the complexity and number of components responsible for a given behaviour. The second term that will be used is the "narrative" that participants use to explain the agents' behaviours. The participant "narratives" are stories that participants tell that are inspired by their observations. Normally a game designer will have a narrative in mind to explain the way in which they envisage a game playing scenario.

It is the goal of designers of believable characters to encourage participants to look beyond the objective reality, to be less objective when describing the agents with which they are interacting. The terminology in the responses of the participants are analysed for the ability of each agent type to encourage story-telling and the tendency for the participant to anthropomorphise them. The analysis also looks for agent behaviours that break suspension of disbelief and prevent the participants from seeing the agent as a character.

The section begins with an overview of the participants and then moves on to the present the most significant findings of the experiment.

5.2.1 PARTICIPANT DEMOGRAPHICS

The background of the participants used for this study can have a significant impact on their response to the game. Participants with gaming experience were expected to identify the rules of the game quicker than novice players and also have an advantage with regards to learning the controls. A questionnaire was used to elicit general demographic and background information on the participants. The participants were also interviewed during this process to get a better understanding about their personal interest in games, why they decided to take part in the study and their game genre preferences and experience.

Participant	**1**	**2**	**3**	**4**	**5**	**6**	**7**
General Demographic Data							
Age	35-44	25-34	15-24	15-24	15-24	15-24	15-24
Gender	F	M	M	M	F	M	M
Current Occupation	PhD Student	Student	Student	Student	Student	Student	Student
Country of Residence	UK	England	England	UK	England	England	England
Country of Origin	Kenya	Serbia	England	Greece	England	England	England
Marital Status	Married	Single	Single	Single	Single	Single	Single
Professional and Academic Background							
Education Level	Msc.	BSc CS 2nd Year	BSc CS Final Year	BSc CSG 2nd Year	BSc CS 1st Year	BSc CS 1st Year	BSc CS 2nd Year
Academic or Professional Background in:							
AI	No	AI First Semester	AI module Complete	AI module	No	No	AI module
Animation	No	No	No	No	No	No	No
Psychology	No	No	No	No	No	No	No

Table 8 General demographic participant data

Table 8 shows the general demographics data that was gathered, including the participants' background in professional and academic fields relating to the subject of the experiment. Most of the participants are younger, with 5 participants within the 15-24 age bracket. Two participants could be considered mature with ages between 25 and 44. All of the tested participants were more or less advanced students in technical subjects. All of the participants are currently resident in Brighton. The country of residence was included in the study to allow the experiment to be conducted as an online web-based study in the future. The countries of origin include Kenya, Serbia and Greece. Only one of the participants had children and interestingly this was also the only source of their experience with games.

4 out of 7 participants have some background knowledge of AI, specifically the second year "Intelligent Systems" AI module at the University of Brighton. Participant 2 had only attending a few lectures in the first semester of this course. Participant 3 had completed the module while participants 4 and 7 were in their final weeks of the second semester. None of the participants had a background or other experience with animation or psychology.

The participants were also asked about their relationship to and experience with games. This was an important factor when considering the responses of the participant, their performance in the game and issues they might have understanding the rules of the game or the controls. Most of the participants had significant experience with games and considered it their favourite hobby. Only 2 participants stated that they had no experience with games. Participant 1 said that she never played games, but knew about them through her children, who regularly played action games on their home consoles. It is not unusual for casual gamers to describe themselves as non-gamers, so the participants that stated that they had no experience with games were asked if they were familiar with games like Tetris (Alexey Pajitnov 1987), Microsoft Solitaire (Microsoft, 1989) and Minesweeper (Microsoft, 1990) or ever played games on their smart phone or on Facebook. Somewhat surprisingly, both participants weren't familiar with any of the titles mentioned and re-iterated they had never played games on a computer before.

Apart from these two exceptions, all the participants in this study had substantial experience with games. Participant 4 stated that they considered games to be both their favourite hobby and the reason they are studying Computer Science (Games) at the university. Participants 5 and 6 also stated that they were considering a career in games, but chose to study Computer Science without a focus on games since they did not want to rely on finding work in that specific industry.

Those participants that stated an interest or had experience with games were asked to list their favourite games or genres. This can give more specific insight into what types of control methods or gameplay types with which participants are familiar. The favourite genre among the participants was role playing games (RPG). Participants 4, 5 and 6 cited massively multiplayer online role playing games (MMORPG) as their favourite sub-genre. Participants 3 and 6 cited multiplayer online battle arenas (MOBA) as their favourite genre. This is a sub-genre mix between role playing and real time strategy (RTS) gameplay that is currently considered one of the most popular online multiplayer sports. Surprisingly, the two genres that have arguably the most presence in mainstream media and advertising, the Sports game (football) and Action shooter genres were only cited by participants 3 and 7 respectively.

Of the games cited during the interview with the participants, the MOBA type games were the most significant with regards to the experiment, since the control system used in those games is mouse-driven and very similar to the system used in the game prototype that was tested during the experiment.

5.2.2 Questionnaire Results: Believability Criteria

This section reviews the results from the believability criteria questionnaire that participants completed while testing the three agent architectures. The questions were based on the criteria defined by Loyall (Mateas 1997; Loyall 1997) and presented using the types of questions and format suggested by Gomes (Gomes 2013). Figure 36 shows the believability questions as they were presented to the participants. The letter X was replaced by the given agent label on each sheet.

Q	Questions
1	**Awareness:** Opponent X perceives the world around him/her
2	**Behaviour understandability:** It is easy to understand what Opponent X is thinking about
3	**Personality:** Opponent X has a personality
4	**Visual Impact:** Opponent X's behaviour draws my attention
5	**Predictability:** Opponent X's behaviour is predictable
6	**Behaviour coherence:** Opponent X's behaviour is coherent
7	**Change with experience:** Opponent X's behaviour changes according to experience
8	**Social relationships:** Opponent X interacts socially with other characters

Figure 36 Believability questions as they were presented to the users

The participants were asked to state the degree to which they agree with each of these statements. The ratings were described to the participants as:

0 = Totally disagree
1 = Disagree
2 = Slightly disagree
3 = Slightly agree
4 = Agree
5 = Totally agree

Voting Tendency of Participants

Before looking at the results for the agents, it is interesting to have a look at the distribution of ratings per participant, since the test group did include a variety of demographics, including mature non-gamers, aspiring game developers and PhD, Masters and first year students. Table 9 shows the results. It is notable that the two more mature students who were also the two that had never played games gave the highest scores overall and never gave low ratings to any of the agents they tested.

Participant Rating Frequency

P	0	1	2	3	4	5
1	0	0	1	8	12	3
2	0	0	2	9	11	2
3	2	6	3	6	3	4
4	6	1	3	3	5	6
5	6	0	4	4	7	3
6	3	3	3	5	10	0
7	0	8	1	6	9	0

Table 9 Frequency count of ratings per participant across all agents they tested

Rating Frequency for Each Agent Type

Table 10 through Table 13 below, show the frequency count for the responses from the participants for each of the 8 believability dimensions. Likert scale ratings are listed on the horizontal axes against the questions on the vertical axes. The most frequent ratings per question (row) are shaded in green.

Agent 1 Binary

Q	0	1	2	3	4	5
1	0	2	1	2	1	1
2	0	0	1	3	3	0
3	1	0	0	2	4	0
4	0	0	2	2	1	2
5	1	0	0	0	3	3
6	1	0	0	3	3	0
7	1	3	1	2	0	0
8	2	1	3	0	0	1

Table 10 Believability metric rating frequency count for agent type 1 (Binary)

Agent 2 Continuous

Q	0	1	2	3	4	5
1	0	1	0	2	2	2
2	1	0	0	2	4	0
3	0	0	2	1	4	0
4	0	0	1	2	3	1
5	1	0	0	3	2	1
6	0	0	2	2	3	0
7	2	1	0	3	1	0
8	2	1	0	1	3	0

Table 11 Believability metric rating frequency count for agent type 2 (Continuous)

Agent 3 Remapping

Q	0	1	2	3	4	5
1	1	1	0	0	2	0
2	0	0	0	1	2	1
3	0	1	0	0	3	0
4	0	1	0	0	2	1
5	0	0	0	2	1	1
6	0	0	0	2	1	1
7	0	1	1	2	0	0
8	0	2	0	1	1	0

Table 12 Believability metric rating frequency count for agent type 3 (Remapping)

Agent 4 Adaptive

Q	0	1	2	3	4	5
1	0	1	0	1	0	1
2	0	1	0	0	2	0
3	1	0	1	1	0	0
4	0	0	1	2	0	0
5	0	0	0	0	1	2
6	0	0	0	1	2	0
7	2	1	0	0	0	0
8	2	0	0	0	1	0

Table 13 Believability metric rating frequency count for agent type 1 (Adaptive)

Just looking at the distribution of votes in Table 10 through Table 13 the patterns clearly show that the participants favoured agent types 2 (Continuous) and 3 (Remapping) over the other two. While agent type 1 (Binary) also received high scores the distribution was similarly spread out as with agent type 4 (Adaptive).

Further Score Calculations

Even with few users, looking at the distribution of responses can give further insight into the data and help to steer the review of the qualitative data toward interesting responses. The tables below were also produced to test the analysis method for future quantitative studies using the believability metrics.

Table 14 through Table 17 show the individual scores for the four agents tested by each participant and their total and mean scores. After the initial three tests, the complex agent type 3 “Remapping” was replaced with the “Adaptive” agent type 4.

Agent 1 Binary
Participant →

Q	1	2	3	4	5	6	7	x̄	σ
1	3	3	1	5	4	2	1	2.71	1.39
2	3	4	3	2	4	3	4	3.29	0.70
3	4	3	0	4	3	4	4	3.14	1.36
4	5	3	2	5	2	4	3	3.43	1.18
5	4	5	5	0	5	4	4	3.86	1.64
6	4	2	4	3	4	3	3	3.29	0.70
7	4	4	1	1	0	2	1	1.86	1.46
8	2	2	2	5	0	0	1	1.71	1.58
Total	29	26	18	25	22	22	21	**23.29**	
~~Predict.~~	25	21	13	25	17	18	17	**19.43**	

Table 14 Agent type 1 (Binary) believability metric scores (0-5)

Agent 2 Continuous
Participant →

Q	1	2	3	4	5	6	7	x̄	σ
1	4	5	3	5	4	3	1	3.57	1.29
2	4	4	3	0	4	3	4	3.14	1.36
3	4	4	3	2	2	4	4	3.29	0.88
4	5	4	4	3	2	4	3	3.57	0.90
5	3	3	3	0	5	4	4	3.14	1.46
6	3	3	2	2	4	4	4	3.14	0.83
7	3	3	3	0	0	4	1	2.00	1.51
8	4	3	4	4	0	0	1	2.29	1.75
Total	30	29	25	16	21	26	22	**24.14**	
~~Predict.~~	27	26	22	16	16	22	18	**21.00**	

Table 15 Agent type 2 (Continuous) metric criteria scores (0-5)

Agent 3 Remapping
Participant →

Q	1	2	3	4	5	6	7	x̄	σ
1	4	4	0				1	2.25	1.79
2	4	4	5				3	4.00	0.71
3	4	4	1				4	3.25	1.30
4	5	4	1				4	3.50	1.50
5	3	4	5				3	3.75	0.83
6	3	4	5				3	3.75	0.83
7	3	3	1				2	2.25	0.83
8	4	3	1				1	2.25	1.30
Total	30	30	19				21	**25.00**	
~~Predict.~~	27	26	14				18	**21.25**	

Table 16 Agent type 3 (Remapping) believability metric scores (0-5)

Agent 4 Adaptive
Participant →

Q	1	2	3	4	5	6	7	x̄	σ
1				5	3	1		3.00	1.63
2				4	4	1		3.00	1.41
3				0	3	2		1.67	1.25
4				3	2	3		2.67	0.47
5				5	5	4		4.67	0.47
6				4	3	4		3.67	0.47
7				0	0	1		0.33	0.47
8				4	0	0		1.33	1.89
Total				25	20	16		**20.33**	
~~Predict.~~				20	15	12		**15.67**	

Table 17 Agent type 4 (Adaptive) believability metric scores (0-5)

Numeric Believability Criteria Results

This section attempts to calculate a numeric believability rating using the method defined by Gomes (2011; 2013). This uses the numeric data calculated in Table 14 through Table 17.

According to Gomes (Gomes 2013) it is important to consider "predictability" as separate from the other believability criteria, so these scores will be looked at individually. Based on the mean average scores (without predictability), agent type 3 (Remapping) scored the highest with 21.25 points, followed by type 2 (Continuous) with 21.00 and type 1 (Binary) with 19.43 points and the worst scoring type 4 (Adaptive) with 15.67 points. The maximum total score with 7 dimensions that could have been achieved is 35.00. Table 18 shows the resulting ranking.

Rank	Agent Type (Highest to lowest)	Total Score (out of 35)	Min	Max
1	3 Remap	21.25	14	27
2	2 Continuous	21.00	16	26
3	1 Binary	19.43	17	25
4	4 Adaptive	15.67	12	20

Table 18 Total believability score ranking

According to the feedback from the participants, agent type 3 "Remapping" is the most believable agent, albeit by a small margin over agent types 2 (Continuous). A significant difference can be seen in comparison with agent type 4.

Rank	Agent Type (Highest to lowest)	Average Score (0-5)	Min	Max
1	4 Adaptive	4.67	4	5
2	1 Binary	3.86	0	5
3	3 Remap	3.75	3	5
4	2 Continuous	3.14	0	5

Table 19 Believability score ranking

Gomes (2013) suggests that the ideal value for predictability should be around the middle of the evaluation scale. They justify this by stating that scores near the extremes indicate that an agent's behaviour was too predictable ("robotic" as some if the participants in this trial called it), or so unpredictable that it seemed random to the participants. If a score at the centre of the evaluation scale at 2.5 is ideal, then agent type 2 (Continuous) had the closest and 'best' predictability score with 3.14.

Agent type 4 received the worst score again receiving 4.67 on average. Seeing the low score given to agent type 4 was intriguing, since it was designed to be an improvement over the "Remapping" agent type 3. The reasoning behind the user's ratings is discussed in section 5.2.3 "Believability Criteria User Responses".

Using the Data to Guide Further Analysis

Statistical analysis techniques were used to guide the qualitative research process toward interesting participant responses. Since this is a qualitative study, this was not done to indicate trends in the data. Instead, patterns found in the dataset were used to identify outliers. Where these were found, a look at the audio transcripts provided insight into the reasoning behind the participant's choices.

To find the outliers, standard deviations were calculated for the answers given by the participants. Then the upper and lower limits of the 1-standard-deviation range were calculated for each question. The results of these calculations can be seen in 10.3 in Appendix “Believability Criteria Statistical Analysis”.

Cells which had values below the range of 1 standard deviation were then shaded with a red dotted pattern, while cells with values above 1 standard deviation were shaded with a green diagonal line pattern. This visualisation guided the qualitative analysis in several ways:

1. The fewer values were outside of the standard deviation range, the more unanimous the decision and the more objective and less open to interpretation the question was considered.
2. If deviation was very high i.e. the majority of values are outside the 1-standard-deviation range and very spread out, then the researcher should look at whether participants understood the question or whether they thought it was applicable.
3. Extreme opinions are highlighted. The labelling of values outside the 1-standard-deviation range allows the researcher to identify which participants gave significantly higher or lower than average scores.
    a. If highlighted values are on opposite sides of the range then it should be checked whether those participants interpreted the question differently or observed different phenomena.
    b. If the highlighted values are on the same side of the range, then their opinion should be compared to that of the mean.

Another way of looking at the responses of the participant is to compare the standard deviations of each question instead of a per-agent basis. This overview can be useful as it highlights where the question and not the agent might be the cause for disagreement between participants. Table 20 below, shows the standard deviation of the responses to each of the believability questions per agent type:

		Agents					
	Questions	1	2	3	4	x̄	σ
1	Opponent X perceives the world around him/her	1.39	1.29	1.79	1.63	1.52	0.20
2	It is easy to understand what the Opponent X is thinking about	0.70	1.36	0.71	1.41	1.04	0.34
3	Opponent X has a personality	1.36	0.88	1.30	1.25	1.20	0.19
4	Opponent X’s behaviour draws my attention	1.18	0.90	1.50	0.47	1.01	0.38
5	Opponent X’s behaviour is predictable	1.64	1.46	0.83	0.47	1.10	0.47
6	Opponent X’s behaviour is coherent	0.70	0.83	0.83	0.47	0.71	0.15
7	Opponent X’s behaviour changes according to experience	1.46	1.51	0.83	0.47	1.07	0.44
8	Opponent X interacts socially with other characters	1.58	1.75	1.30	1.89	1.63	0.22
	x̄	1.25	1.25	1.13	1.01	**1.16**	**0.30**
	σ	0.34	0.32	0.37	0.56		

Table 20 Believability questions answer standard deviation

To guide the analysis, values one standard deviation below the average for each agent type were considered an "agreement" between participants and a values one standard deviation above the average a "disagreement". Agreements are marked with green diagonal lines, whereas disagreement is marked in red.

5.2.3 Believability Criteria User Responses

While the previous section analysed the numeric responses to each agent, this section analyses the qualitative responses from the transcript to elicit which of the four agent types presented to the users is the most believable. The participants did not only offer feedback and criticism during the trial, but also recommended improvements to the agent models they were testing.

For each of the 8 believability dimensions (represented by a Likert scale question on the questionnaire) this section highlights significant findings while also looking at how the participants interpreted the questions, whether they were confused by it and how the answers they gave were influenced by their interpretation of the questions. Based on this feedback, modifications to the questions will be suggested where appropriate.

The results therefore not only contribute to the evaluation of these agents, but also contribute to the ongoing discussion on the use and format of believability metrics in the development process of virtual characters. Critically, this experiment does not just use, but also tests the suitability of these believability metrics in the context of early prototyping, when only the behaviour of an agent can be evaluated. It discusses how this evaluation can inform further agent behaviour design choices as well as decisions regarding the character's graphical representation, its interactions with the user and the its interplay with the surrounding narrative of the game.

In short, this section looks at *what* the participants said about the believability of the agents and also investigates *why* participants gave the answers they did.

Dimension 1 – Awareness

Opponent X perceives the world around him/her

Even though dimension 1 "awareness" received some of the highest ratings from the participants, the interpretations of the question by participants varied. Some participants thought that "perceiving the world around it" was obvious- of course the agent had to sense the world to be able to interact with it. For example participant 5 didn't understand why this was even questioned:

> "By perceiving the world around him. Like noticing things? Why would he not notice things?" – Participant 5 (About Agent 2 – Continuous)

In contrast to this, other participants who gave low ratings (like participants 3 and 6) thought of "perceiving" in the literal sense of using some simulation of sight as a sensor and not just a field of view range.

> "I'd say probably he obviously doesn't perceive it's just a circle" – Participant 3 (about Agent 3 – Remapping)

In this case a similar observation leads to different ratings due to differing interpretations of the question.

While Gomes (2013) refers both in his definition of the "awareness" dimension, the phrasing he chooses for this question leans more toward Loyall's definition of "reactive and responsive behaviour" (Loyall 1997) than Lester & Stone's "situated liveliness" (Lester & Stone 1997).

User responses indicate that this question was too generic. Therefore it is suggested that it should be made more specific by including elements of Lester & Stone's "situated liveliness" (Lester & Stone 1997).

Question Refinements

Rephrase Question:

> "Opponent X shows that it is aware of changes in its environment"

Agent Results

The two main observations that were made by the participants with regards to this dimension were the agent's ability to see and chase the player and its ability to avoid walls. Some participants seemed to notice different active ranges for the field of view of the agent and were convinced that some agents could see farther than others. For example, while testing the "Adaptive" agent, participant 4 noticed that this agent had a shorter field of view:

> “His field of vision is much shorter than the other ones. I didn’t notice that before. It takes much longer for him to notice me.” – Participant 4 (about agent type 4 – Adaptive)

This is an interesting perception, since the actual field of view is exactly the same for each agent. So what the participant was observing was in fact the different ways that seeing the player affects the agent’s behaviour. So when the participant said that the field of view for Agent 4 was shorter than that of agent types 1 (Binary) and 2 (Continuous) (the other two that this participant tested), then they are really referring to that particular agent’s slower response when seeing the player.

Overall, agent type 2 (continuous switching) scored the highest in this dimension. The main reason for this was that the participants saw the chasing behaviour being more persistent that in agent type 1 (Binary) as an improvement to the way the agent could see and track the player. Looking at the data that was gathered it could be assumed that the second best agent was the “Binary” agent, but this does not match the participant’s responses. Looking closer at the data, it is clear that this is due to the limited number of participants that got to test agent type 3 (Remapping). Two of the participants that rated agent type 3, gave low scores in this dimension to all the agents. This is also reflected in the high deviation between the ratings of this agent.

Taking this into account, it can be found that in the case of the agent models tested, that those that a more persistent chasing behaviour were rated to have better perception of the world around them. The agent’s ability to keep track of moving objects seems to have been the primary focus of the participants.

From their responses the participants thought that the behaviour could be improved further, by enabling the agent to track the player even better and to anticipate where the player might go next. One suggestion by Participant 5 was particularly interesting, since they suggested adding the sense of hearing to improve the agent:

> “Would there also be like, a noise indicator. Might be useful. So for the coins maybe. Because they do create a sound for the user, so technically the opponent should be able to hear that and go in that very direction.” – Participant 5 (About all agents)

What the participant is describing is a well-established mechanic from the genre of stealth games. A “noise indicator” visually displays the amount of noise that a player makes. If the noise indicator hits certain threshold, enemies in the environment are alerted. Another implementation indicates the range of sounds that the player makes in the environment. If an enemy is within the range, they can hear the sound.

A similar feature was implemented in a previous version of the game prototype, but was not used for the purpose of simplifying the agent behaviour for the experiment, based on feedback during the pilot studies. A description of the implementation is discussed in section 5.2.8 “Further Findings” under “Differential Drive Issues”, since it also addresses some of the other issues caused by the agent’s vehicle-like movement.

Dimension 2 - Behaviour understandability

It is easy to understand what Opponent X is thinking about

Ratings for question 2 “behaviour understandability” had a mixture of high and low deviations between agents, which indicate that participants agree on the ratings of some agents, but not on others. The reasons for these ratings can be investigated when looking at the transcript. While answering question 2 for Agent 2, Participant 4 stated:

> “I don’t think he’s thinking about anything really”- Participant 4 (About Agent 2 – Continuous)

However, the same participant was also unsure about the question itself. This is revealed when looking at their response when first encountering the question, where they said:

> “Easy to identify what he (A - Binary) is thinking about..uhm, is that in terms of where he’s gonna go next, or? I think it’s quite hard to know what he’s (A - Binary) thinking about, to see where he’s going to go or how fast he’s going to move” – Participant 4 (About Question 2, while testing Agent 1 - Binary)

This response shows that the participant was struggling to figure out how to rate the agent according to this believability dimension. Another example is participant 1 who gave Agent 2 a relatively high score for this question, Participant 1 proposed a way for improving the agents in this regard:

> “Not sure about that [marks between 2 and 3 out of 5]. Ok, what else there is about, maybe like what I said about trying to make it more clear about some emotions so that one can be able to see the emotion and know that.

> So I know that if he's angry, then he's really coming after me, so I try as much as possible to be (far away). So put in like that emotion's images." – Participant 1 (about all agents)

Comparing these two shows two interpretations of the questions:

1. It is easy to tell what the agent will do next
2. I can tell what the agent is thinking

These interpretations show that question 2 can seem similar to question 5 "predictability" to some participants.

Since the question is trying to elicit whether participants are able to "create a model of an agent's behaviour *motivations*" (Ortony 2003; Gomes 2013) and not whether they can predict the agent's behaviour the question should be more specific include the notion of emotions.

Question Refinements

Rephrase Question:

> "It is easy to understand what Opponent X is thinking and feeling"

Agent Results

Overall the ratings were comparatively high for this dimension with average scores of 3.29, 3.14, 4.00 and 3.00 for agent types 1,2,3 and 4 respectively.

The highest rating was given to agent type 3 (Remapping) and the lowest to agent type 4 (Adaptive). This was actually surprising, since the way that agent type 3 adapts to the player collecting coins is technically less plausible than the way agent type 4 works. The difference between these two is that Agent 3 will incrementally change its behaviour tendency to be more aggressive each time the player collects a coin, regardless whether the agent sees the player steal it or not. Agent type 4 will only change its behaviour when it sees the player collecting coins. However, as reported before, this differentiating feature of agent type 4 was overlooked by the majority of the participants. In fact, Agent 4 prompted several puzzled responses from the participants, often because they were not able to get the agent to chase them on repeated trials. This happened because they didn't let the agent spot them stealing the coins, but none of the participants figured this out.

The behaviour of agent type 3 technically "cheats" by getting information about the player's behaviour, without using its senses. This is a common technique used in game AI development as a shortcut to creating more sophisticated AI. Agent type 4 is therefore the more "realistic" of the two.

With regards to the mental models that the participants built, there were a variety of interesting responses, which will be described in detail in the following section "Questionnaire Results: Behaviour Description Open Questions". These varied from participants trying to guess the algorithms controlling the behaviour, to entire background stories for each of the three agents.

Dimension 3 – Personality

Opponent X has a personality

All of the participants focused on the chasing behaviour and not the patrolling when discussing personality. Emotions were at the centre of comments about personality. Especially when discussing the agents that attained the highest rating, participants were keen to emphasize the apparent determination and irrational (some called it "mad") anger that the agent had toward the player when chasing them.

Participants 3, 4 and 5 that gave low ratings in this category seemed to focus on the comparatively slow switch from patrolling to chasing observed in agent types 2 and 3 attributing the definition "curiosity" to that slow reaction to the player instead of "anger", which they used for the "Binary" agent type 1. None of the participants mentioned the persistent chasing behaviour that agent types 2, 3 and 4 display once the player had been spotted.

The perception of personality seems to be primarily linked to the display of *strong emotions* and is enhanced when participants are *surprised* by the occasional irrational behaviour.

In addition, participants frequently used comparisons between the agents when describing their personality. This indicates that personality is *enhanced by contrasting characters* and could further benefit from having multiple agents in the scene.

Agent Results

Since the game presented to the participants was very limited with regards to its narrative, the participants had to develop their own "story" around the scenario. One component of these user-generated-narratives is the "personality" of the different enemy agents. According to Loyall (Mateas 1997; Loyall 1997) these are the characteristics that make each agent unique from the others.

While the user-generated-narratives and personality descriptions will be detailed in the next section "Questionnaire Results: Behaviour Description Open Questions", this section looks at the scores that the participants gave.

The highest ratings were given to the "Continuous" agent (type 2) with 3.29 and the "Remapping" agent (type 3) with 3.25. The main feature of these two agents that was mentioned with regards to their personality was that they "stayed angry" at the player and were "more persistent" when chasing them. This was the participants' interpretation of these agents' slow decay of the chasing behaviour, back to the patrolling behaviour after losing sight of the player. The "Binary" agent (type 1) on the other hand immediately lost "interest" in the player when they manage to escape the field of view. Participants gave this agent a 3.14 on average.

The lowest rating was again given to the "Adaptive" agent (type 4) with 1.67, which participants had difficulties testing and creating a mental model for its behaviour.

With regards to the agent behaviours that were that participants referred to when describing the "personality" of the agents, two factors were considered:

1. A prompt reaction to seeing the player was seen as strong emotions and termed "aggressive" or "mad at".
2. A persistent chasing behaviour, without "losing interest" in the player when they leave the field of view.

When both of these factors were present in the agents behaviour, participants gave high personality ratings. If one of them was missing, as was the case with the "Binary" switching (type 1) agent, then ratings tended to be slightly lower. If both were missing, or only happened occasionally, as was sporadically the case with the "Adaptive" (type 4) agent, then the ratings tended to be low.

From the statements that participants made while evaluating these agents, it becomes clear that participants were surprised by the persistence of agent type 2 and 3's chasing behaviour. The participants attributed this to an "eagerness" of the agent to "not let go" of the player when they had seen them. It was also the root cause for participants attributing the emotion "anger" to the agent. For example, Participant 1 seemed surprised by the difference to the prior agent (type 2 – Binary):

> This one's really aggressive, he's after me! [laughs] He's really angry or what. – Participant 1 (while testing Agent 2 - Continuous)

The gameplay data shows that during the particular trial that the participant was playing, they did not get spotted by the enemy agent very often. An issue discussed in "Dimension 2 - Behaviour understandability" was that good players may be able to avoid seeing certain behaviours by playing "too well". Thus a player, who is caught more often, will have to deal with the agent's more persistent tracking behaviour, while a player that manages to mostly avoid the agent and evades its field of view before the agent can fully switch to the chase behaviour, will experience a different, much more lenient behaviour.

Dimension 4 – Visual Impact

Opponent X's behaviour draws my attention

According to Gomes (2013), question 4 "Opponent X's behaviour draws my attention", asks the participant to rate the visual impact that the agent's behaviour has on the observer. It is suggested that this question is asked about the entire agent, yet the responses from the participants in this study show that this question only makes sense when there are more than a single agent for the player to pay attention to.

Participant 1 stated that they had to look at the agent in order to avoid it:

> "Yes of course. Because I'm there I'm trying to look, really see and avoid him. So he does." –Participant 1 (about Agent 1 – Binary)

Participant 6 also interpreted the question in this way and was not sure what rating to give:

> "Not sure about draws my attention. It draws my attention because I would like to figure out how it works. Generally how I approach a lot of games." – Participant 6 (about Agent 4 – Adapt)

In fact, it seems that Gomes (2013) misinterpreted Lester and Stone's (1997) definition of visual impact, since they indicate that an agent must not simply maximise, but be able to *control* the visual impact of individual behaviours.

Question Refinements

It is therefore recommended that the visual impact dimension should be taken out of the Likert scale and evaluated separately in a format similar to dimension 9 “emotional expressiveness”. The participants should be asked to rate the visual impact of individual behaviours. Their answers should then be compared to the visual impact that the agent designer or animator *intended* that behaviour to have.

A Likert scale or multiple choice questions could be used. Suggested choices/boundary values for the question “Rate the visual impact of behaviour X”:

LOW	MEDIUM	HIGH
I didn’t notice	It drew my attention	It was distracting/annoying

Agent Results

The highest rating was given to the “Continuous” agent (type 2) with an average of 3.57 points closely followed by the “Remapping” agent (type 3) with an average of 3.50 points. Both of the agents beat the “Binary” agent (type 1) with 3.43 and the “Adaptive” agent (type 3) with 2.67.

The low score of the “Adaptive” agent (type 4) can again be attributed to the difficulty that participants had with testing the agent, particularly getting it to chase them, since this required them to first collect coins in front of it. Nevertheless, even this agent rated relatively high on this dimension.

From the responses of the participants in the transcript it is clear that the main reason that participants gave high ratings in this dimension was due to two elements of the game design:

1. There is only one autonomous agent besides the player controlled agent.
2. The game rules require the player to keep track of the enemy agent’s position in order to avoid it.

This is also one of the reasons why the “Adaptive” agent type 4 received the lowest rating. If participants successfully avoided the agent when they picked up coins, the agent would never chase them even when it saw them afterwards. They could therefore just ignore the agent, unless it actually spotted them stealing a coin, resulting in a lower visual impact rating.

The visual impact dimension is strongly related to dimension 2 “behaviour understandability”. While giving feedback on that question the participants showed a desire for the agent to have additional visual ways of expressing their emotions and thoughts.

According to the participants, the main feature that currently fulfils this purpose, which also differentiates agent types 2 and 3 from the “Binary” agent, was the way that the field of view dynamically fades in and out when the agent sees the player. Several participants (1, 4 and 7) made positive remarks about the animated field of view, showing a preference over the immediate switching displayed by the “Binary” agent’s field of view.

It was interesting that participant 1 also mentioned this, since it is a staple of modern stealth-oriented games to have visual indicators on screen that indicate whether the enemies can see the player. The fact that the participants liked the smooth (visual) transition is also interesting, since the most recent examples of stealth-gameplay-mechanics also feature smooth indicators instead of discrete indicators. For example, while “Metal Gear Solid 4” (Kojima 2008) still had the famous “!” indicator appear above the enemy’s heads when they spotted the player, while the modern shooter “Far Cry 3” (Ubisoft 2012) has a gradually increasing directional arrow indicating how aware enemies in the surrounding area are. “Dishonoured” (Arkane 2012) has a mix of the two, with 3 discrete states, that gradually increment.

Dimension 5 – Predictability

OpponentX’s behaviour is predictable

The “Predictability” dimension defines a participant’s ability to recognise patterns in the agent’s behaviour. Gomes (2013) and Ortony (2003) point out that a high score in this dimension will negatively affect believability, while Lester and Stone (1997) showed that too much unpredictable, seemingly random behaviour that participants cannot anticipate or explain using the mental model they have developed of the agents “inner workings” can also negatively affect believability by making the behaviour appear incoherent.

Looking at the overall participant responses it is clear that participant 4 was the main outlier in the results and tended to vote against the other participants. Participant 4 rated the predictability of agent types 1 and 2 at zero. The averages for agent types 1 and 2 are 3.9 and 3.1. Without participant 4, these would have been 4.5 and 3.6.

The transcript shows that participant 4 was apparently torn about what rating to give the agent since they found the patrolling behaviour to be unpredictable and somewhat incoherent, while at the same time finding the agent's reaction to the player very predictable and easy to anticipate:

> "So was it (A-Binary) coherent? Well I don't know because it's kind of related to predictable. It's the fact that you really don't know where he's gonna go, makes it not very coherent, but then the way he follows you, is completely coherent."- Participant 4 (about Agent 1 – Binary)

In this case the participant clearly understood the question and the participant's response was well justified. However this does reveal a potential weakness the question which asks the participant to rate the overall predictability of the agent's behaviour. Participant 4 clearly wanted to give different ratings for the different behaviours. It is also notable that in the same response participant 4 describes the relationship between predictability and coherence in a similar way to Gomes (2013) and Lester and Stone (1997).

Question Refinements

Rate predictability of individual behaviours instead of entire agent.

Agent Results

In the responses gathered from the participants, Agent 2 "Continuous" with 3.1 and Agent 3 "Remapping" with 3.8 have the best predictability scores, since they are closest to the Likert scale's median 2.5 rating.

The highest score (out of a possible 5) in the predictability dimension was given to Agent 1, the binary behaviour switching agent, which had a mean average of 3.9 or "agree". The only outlier was participant 4 who gave this agent a 0 in this category.

The other participants' scores confirm what Gomes et al suggested about predictability scores (Gomes 2013). Participants who gave the agents a high predictability score often stated that they had figured out the agent's behaviour quickly. After their second trial, Participant 3 said:

> "Well, whenever he (Agent 3-Remapping) gets you into line of sight, he just kind of chases you. But it's still in a circular motion." – Participant 3 (about Agent 3 – Remapping)

Giving similar feedback, after several trials, Participant 5 was more explicit and pointed out that they considered the predictability of the agents an issue:

> "So in a way it's still a bit too predictable. You know that when they're facing there, like straight ahead, you know that they will just go there. They won't just suddenly turn around."- Participant 5 (about all agents)

This refers to an unexpected emergent agent behaviour that was observed by several participants, which saw agents "flee" from the player if they approached it directly from behind.

Dimension 6 - Behaviour coherence

Opponent X's behaviour is coherent

Looking at the overview in Table 20, question 6, "Opponent X's behaviour is coherent" received the most unanimous responses from the participants. This initially suggests that the participants both interpreted the question in the same way and agreed on the ratings for the agents based on that interpretation. However, it should be noted that the transcript shows that Participants 2 and 5 asked what "coherent" meant when they first read the question. The researcher had to describe its meaning using other terms such as "the behaviour *makes sense* and is *consistent* with its personality". When that didn't help (participant 2), "does *not act crazy*" seemed like the closest colloquial term. Perhaps one of these terms instead of "coherent" should be considered in the future.

The participants found this dimension to be strongly linked with dimension 5 "predictability". This is also reflected in the questionnaire responses, which show that votes were very close (and resulted in the same average score) in this dimension for agent types 2 and 3.

Participants drew a correlation between high predictability and high coherence, stating that all coherent behaviour must be predictable. In other words predictability is a *condition* for perceived coherence. The results show that where participants gave an agent a high score for "coherence", they also gave it a high score for "predictability".

The reverse is not true, however. The participants did not state that all predictable behaviour was necessarily coherent. In two cases (Agent 1, participant 2 and Agent 4, participant 5) the participants thought the behaviour was still predictable, but not coherent.

This data indicates that high coherence can only be perceived if the participant is able to create a mental model that allows them to anticipate the agent's actions.

Question Refinements

Rephrase question:

> "Opponent X's behaviour *makes sense* and is *consistent* with its personality".
> "Opponent X acted *sanely* (and not *crazily*)"

Agent Results

As with predictability, some participants preferred to discuss the coherence of certain behaviours and not the agent as a whole.

Looking at the overall results, agent type 3 (Remapping) had the highest mean score in this dimension. This is an interesting result, since agent type 3 architecture actually has the ability produce very "incoherent" behaviour. As described in the section titled "Agent Behaviour Description Predictions" in 3.2.5, the way this agent "remaps" the primary and secondary behaviours by shifting the baseline of the behaviour bias up and down, can result in the agent chasing the player when it doesn't see them and switching to the patrolling behaviour when it spots them. It was expected (by the designer) that participants would find this behaviour incoherent.

However, the evaluation shows that the participants rated this particular agent the most coherent with a mean score of 3.75, closely followed by agent type 4 with 3.67. Both of these agents are more complex than agent types 1 and 2, with the main difference being that they change their behaviour when the player collects coins.

Participants noticed a difference between Agent 3 and the simpler types, stating that the "Remapping" agent changed its behaviour over the course of the game. For example Participant 1 explained:

> "He (C) seems like he doesn't really want to chase me too much. Then all of a sudden, he's fast." – Participant 1 (about agent type 3 - Remapping)

Participant 1 could not develop a mental model for this change in the behaviour in the agent and also said that they did not notice any other differences between agent type 2 and 3. Participant 2 also noticed that the agent "was quite guarded against you taking the coins" (Participant 2, testing agent type 3), but did not further explain how they thought the agents behaviour changed when the player collected coins.

While the Participants did not notice the effect that collecting coins had on the behaviour switching in the complex agent types 3 and 4, they did notice how collecting coins affected the patrolling behaviour. All participants noticed that the agents stay close to the remaining coins once the player had collected some of them. This and the fact that the agent started chasing the player to "protect" the coins were seen as a "coherent" way to act.

The low scores given to agent type 2 (Continuous) by Participants 3 and 4 were also investigated. A closer look at the feedback that the participants gave while reviewing this agent reveals an unintentional emergent behaviour of the agent that can certainly be considered incoherent. While discussing the coherent behaviour dimension, Participant 4 described something they noticed while they tried to "sneak up" on the enemy agent from behind:

> "Coherent. Well… What was weird was that when I go from behind him [pointing at the position behind the RED agent on the screen], I'd expect him to turn around really quickly, but he does like a huge circle, which makes him quite slow to catch up with me, because I'm already out of his field of vision that time. So he doesn't know where I am anymore. So coherent, not that much I'd say.
>
> I recon Opponent A (Binary) was the same though. Like he (RED agent) did also big circles around him (GREEN player agent)." - Participant 4 (about agent type 2 – Continuous)

What is interesting about this finding is that it is not isolated to that particular agent type, but is a result of the implementation of the differential drive. It is an emergent behaviour resulting from the interaction between the sensory and steering systems, which are unable to detect the difference between something sensed in front, or behind of the agent. Further investigation of this showed that Participant 3 also had discovered the same issue. The two participants did no let this affect the scores they gave to the other agents and did not go back to their previous ratings to amend them.

The emergent behaviour described here also affected the steering of the player character in some instances. This is discussed separately in Chapter 5.2.8 "Further

Findings” in the “Differential Drive Issues” section.

Dimension 7 – Change with Experience

Opponent X’s behaviour changes with experience

The participants were asked whether they believe that the enemy agent adapts over the course of a play session – whether it changes based on its experience. Looking at the responses across the four agent types, this is among the lowest rated believability dimensions. Only two of the participants (1 and 2) agreed with the statement that Agent 1 learnt. These were both of the participants with no experience with games. Aside from these outliers, none of the participants “agreed” with this statement.

The applicability of this question to the scenario was not immediately apparent to the participants. All of the participants read the question aloud and had to actively look for behaviours that could be deemed adaptive. Interestingly, all of the participants settled on the notion that the enemy agents adapted by staying near the last few coins instead of patrolling the entire environment. While this is certainly a change that happens in the agent’s behaviour as the game is played, the participants were not sure if this could be called “learning from experience”.

Participant 2 explicitly proposed a different type of adaption, stating that “changing with experience” should involve the enemy agent figuring out better strategies for chasing the player and predicting where the player would go:

> “Does it have the intelligence to see how am I controlling and then he used another trick to follow me.” – Participant 2 (Defining Dimension 7 – Change with Experience)

Looking through the responses of the participants it is clear that they were struggling to apply this metric to the behaviour of the agent. In fact, the response by participant 2 shows that they were expecting more from the agent and wanted it to clearly learn and optimise its behaviour to become better at finding and chasing the player.

The most surprising result was that agent type 4, the “Adaptive” agent received such a low score of 0.3 points. This agent was specifically designed to change its behaviour when it detects the player collecting coins. However, this behaviour was not detected by participants who were able to avoid being seen by the agent when they collected coins. Agent type 3 (remapping behaviours) also changes its behaviour when the player collects coins, but it does not need to *see* the player collecting the coins to do so. Even though agent type 3 had the highest result in this dimension with a mean of 2.3 points, this still indicates that participants slightly disagreed that it learns from experience.

Question Refinements

Participants could not apply dimension to tested agents. Further review necessary.

Agent Results

Gomes (2013) describes the requirement that a character is (permanently) affected by events in the game as the “essential building block of the classical plot arch”. When developing the questionnaire that was used for this empirical study, Gomes’ focus was on interactive narratives. Thus the context this dimension was intended for is different from the context that this experiment present its virtual agents in. However, the believability requirement that the agent should “change with experience” still applies, albeit in the abstract sense that the agent should have some way of permanently changing its behaviour, based on key events in the game.

In this sense, agent type 3 (Remapping) and type 4 (Adaptive) are the only agents that support some form of permanent change, since they have the ability to shift their base behaviour, which affects how they react when they see the player character.

From the designer’s perspective, these two agents were specifically designed to score high in this dimension. Therefore it was surprising to see that agent type 4 only scored a low average of 0.33 in this category.

None of the agents were rated highly in this dimension since none of the Participants were able to identify any permanent change in the agent’s behaviour.

The only adaption that participants noticed and were able to explain and create a mental model for was that the agents tended to stay in areas with more coins. The participants saw that the agent would patrol the area around the last remaining coins toward the end of the game. However, none of the Participants saw this as adaptive behaviour, merely as a decently functioning, dynamic “protective” behaviour in its different permutations/interacting with a changing environment, which in fact, is exactly what it is.

Agent type 3 (Remapping) changes its base behaviour and the way it reacts to the player each time the player collects a coin. The Participants interpreted it as the agent "getting angry" or being generally more aware of the player than the other agents.

When Participants were asked which agents they preferred to play against at the end of the trial, several Participants cited the ability of agent type 3 (Remapping) to start chasing the player even when it didn't see them as something positive. Even though they did not identify this change in behaviour as adaptive, they liked that it made the game more challenging. What this result also shows is that the participants did not mind that this agent effectively "cheated" when it started chasing the player without having spotted them previously.

Another surprising result was that agent type 4 received such low scores in almost every dimension including this one (0.33 score average). This agent changes its behaviour when the player collects coins, but in contrast to agent type 3, it doesn't "cheat". It need to *see* the player pick up a coin before it starts chasing them. While this was intended to be more *realistic* and *plausible* than agent type 3's "cheating" approach, the participants seemed entirely confused by it.

The main issue was that agent type 4 starts out the round ignoring the player when it sees them and for a change in behaviour to occur, the player had to make a *mistake* by being *seen while collecting* coins. Since this was against the rules, players that abided by them and out-played the agent never triggered the behaviour change. Consequently the agent would continue to ignore the player, until they made a mistake. When this happened late in the game, the participants *did* notice the sudden change, but could not figure out what they had done differently.

Dimension 8 – Social Relationships

Opponent X interacts socially with other characters

The question with the highest distribution of ratings is question 8 "Opponent X interacts socially with other characters". It was found that the Participants had difficulty applying this dimension to the game scenario they were given. The main reason for this was that their interpretations of "social behaviour" required more than two characters interacting with each other. Participants 1, 2, 3 and 7 directly commented on this issue, stating that more non-player characters would be required for this metric to apply. In addition, Participant 3 remarked that they did not consider "aggression" a social interaction, further suggesting that the non-player characters present must be interacting cooperatively for the interaction to be called "social".

Question Refinements

Participants could not apply dimension to tested agents. Question requires multiple non player characters to be present in scene. Further review necessary.

Agent Results

This question was one that participants were unsure about and didn't know how to apply in the context of the experiment, which explains why the interpretations and answers deviated so much from each other.

The participants' feedback ended up falling into two groups: Those that considered attacking the player character a social interaction (Participants 1, 2, 4) and those that believed that more diverse forms of interaction were necessary to justify a higher score in this dimension (Participants 3, 5, 6, 7).

Within the first group that did consider attacking a form of social interaction, the Participants rated the agents that were more aggressive higher than those which were less persistent in chasing the player. Thus the "Continuous" (type 2) and "Remapping" (type 3) agents received higher scores than the "Binary" agent. The exception here was Participant 4, who considered the instant aggressiveness of the "Binary" (type 1) agent to be more aggressive than the persistence of "Continuous" (type 2) and "Remapping" (type 3) agents.

5.2.4 Questionnaire Results: Behaviour Description Open Questions

This set of questions informs the ninth believability dimension "Emotional Expressiveness", a concept which was defined by Loyall (Mateas 1997; Loyall 1997) and Ortony (2003) and adopted as a metric by Gomes et al (2013). In order to assess this question Gomes (2013) suggests providing participants with a multiple choice list of basic emotions and tasking them to attribute these to the behaviours they observed the agent perform in specific situations. They further suggest enumerating the result by counting how many choices were "correct" according to the specification of the system.

While this study followed Gomes' format, it was also the intention of the experiment to encourage the participants to offer their own views of what constitutes believability.

Therefore the last question on each agent questionnaire was used to initiate an open conversation about the agent with the participant. The purpose of these questions is to learn about the mental model that the participants had developed in order to understand and explain the enemy agent's behaviour.

The Open questions asked the participants to describe the behaviour of the agent in their own words. Three typical scenarios relating to interactions with the enemy agent were given:

1. When the agent didn't see the participant
2. When the agent saw the participant
3. When the participant collected coins

Participants were asked to describe the behaviour of the agent in the specified situations in two ways:

1. Write down simple emotional or objective terms that appropriately describe the behaviour of the agent.

2. Talk about the agent's behaviour with regard to the three scenarios presented. These responses were recorded and later transcribed for content analysis.

This section looks at the participants' responses in two steps. In the first part, the questionnaire (paper) responses will be summarised and presented in groups formed of similar terms. This is useful in getting an overall impression of how the participants interpreted the behaviour of the agents and applies the method suggested by Gomes (2013). It also serves as the starting point for the next step of the analysis.

In the second part, the vocal responses (audio transcript) from the participants will be analysed for each agent type. Particular attention was also paid to the methods that the participants used to communicate their mental model of the agent's behaviour. These are discussed in Chapter 5.2.8 "Further Findings" in the section "Participant's Methods of Describing Their Mental Model".

Questionnaire Responses

Agent 1 Binary — **Participant Responses**

Situation	1	2	3	4	5	6	7
When agent didn't see you	n/a	Curiosity	Curiosity	Careless	Curious, Relaxed	Nothing	Relaxed
When agent saw you	n/a	Anger	Hate	Hungry, mad	Alarmed, Anger, hate	Anger/Hate	Curiosity
When you collected coins	n/a	Didn't Care	Fear	Careless	Relaxed	Nothing	Not bothered

Agent 2 Continuous — **Participant Responses**

Situation	1	2	3	4	5	6	7
When agent didn't see you	n/a	Curiosity	Fear	Careless	Curious, Relaxed	Nothing	Didn't care
When agent saw you	Aggressive	Anger	Hate	Curious	Hate, Anger	Anger/Hate	a little angry
When you collected coins	Aggressive	Hate	Curiosity	Careless	Relaxed	Nothing	Didn't care

Agent 3 Remapping — **Participant Responses**

Situation	1	2	3	4	5	6	7
When agent didn't see you	Same as Agent 2	Fear	Curiosity				Curiosity
When agent saw you	Same as Agent 2	Anger	Anger				Very Angry
When you collected coins	Same as Agent 2	Hate	Fear				Didn't Care

Agent 4 Adaptive — **Participant Responses**

Situation	1	2	3	4	5	6	7
When agent didn't see you				Careless	Curious, Relaxed	Nothing	
When agent saw you				Confused	Hate, Anger	Curiosity, Anger/Hate	
When you collected coins				Careless	Relaxed	Nothing	

Figure 37 Questionnaire responses: Agent open questions

Figure 37 shows the terms that participants attributed to the behaviour of the agents in the three different situations that can occur in the game. Participants were encouraged to use their own words, but the example terms Fear, Anger, Hate and Curiosity were presented to the participant on the form. The results show that some participants initially focused on the example terms. Since they were encouraged to use their own words some added their own terms as well.

One of the main research questions to be answered by this experiment was: "Q1: How do users interpret behaviour and is this different from the designers' interpretation?". To get an overall sense of the participants' interpretations of the agents' behaviours, the responses of the Participants were grouped into similar terms. These were then juxtaposed with the definition of the behaviours as they were intended by the designer of the agent architecture. Figure 38 compares the participants' summarised responses, with the designer specification.

Agent 1 Binary						
	When Agent didn't see you		**When Agent saw you**		**When you collected coins**	
Participants	Curiosity	3	Anger, Hate, Mad	7	Didn't Care, Careless, Not Bothered	3
	Relaxed	2	Curiosity	1	Fear	1
			Hungry	1	Relaxed	1
Design	Patrolling, aware, safe		Angry, Aggressive. Chase Player		Unaware, adjusts path to guard leftover coins	

Agent 2 Continuous						
	When Agent didn't see you		**When Agent saw you**		**When you collected coins**	
Participants	Curiosity	2	Aggressive, Anger, Hate, A little angry	8	Aggressive, Hate	2
	Fear	1	Curious	1	Careless, Didn't Care	2
	Careless, Didn't Care	2			Curiosity	1
					Relaxed	1
Design	Usually patrolling, safe.		More Aggressive. Chase Player. Tracks player after loosing sight. Sees player for a short time = suspicious.		Unaware, adjusts path to guard leftover coins	

Agent 3 Remapping						
	When Agent didn't see you		**When Agent saw you**		**When you collected coins**	
Participants	Fear	1	Aggressive, Anger, Very Angry	4	Aggressive, Hate	2
	Curiosity	2			Fear	1
					Didn't Care	1
Design	Usually patrolling, safe. Tracks player after loosing sight. Initially patrols. After remapping, chases (angry at player)		Angry, Aggressive. Chase Player Tracks player after loosing sight. Sees player for a short time = suspicious. Initially chases player. After remapping, it patrols (incoherent)		Adjusts behaviour. Get's Angry at player Remapts behaviours, swtiching primary and secondary behaviours	
	Combination of reversed chase and patrol mapping will cause the agent to keep the player on the edge of its field of view, "keeping an eye on" them.					

Agent 4 Adaptive						
	When Agent didn't see you		**When Agent saw you**		**When you collected coins**	
Participants	Careless	1	Confused	1	Careless	1
	Curious	1	Hate, Anger	3	Relaxed	1
	Relaxed	1	Curiosity	1		
Design	Usually patrolling, safe. When Chasing, will track player after loosing sight. When Hunting, will find player when it doesn't see them		No Coins: Ignores Player. Friendly Few Coins: Suspicious, will approach More Coins: Chase, but forgive (agent 3) Most Coins: Hunt, won't forgive		Adjusts behaviour w/ No. of Coins taken Few Coins: Becomes suspicious at player More Coins: Becomes angry, but forgives Most Coins: Hates player, won't forgive	
			This agent will only start chasing the player once it has seen them steal a coin with each coin stolen it gets angrier at the player until it won't stop chasing them.			

Figure 38 Agent open questions: Grouped response

Note that the design differentiates between two types of aggressive behaviour, *chasing* and *hunting.* In the context of the game, *chasing* refers to the enemy agent following the player's agent when it sees them (i.e. $V = 1$ in the behaviour formula) and *hunting* refers to the enemy agent trying to find and chase the player's agent when it cannot see them (i.e. $V = 0$ in the behaviour formula).

Looking at the participant's responses first, it is apparent that the strong emotion was easily identified by the participants. Almost all of the participants used terms denoting strong aggressive emotions to the chasing behaviour displayed by the agents when they saw the player.

The more subtle emotions displayed by agent types 3 and 4, like suspicion were identified less consistently. Participants used a variety of terms that didn't indicate particular activities, but more general moods, like "relaxed" or "careless". The latter term in particular was used often when the participants were not sure how to define the behaviour, such as Agent 3's reaction to collecting coins.

Agent Type 1 – Binary

With regards to the different agent types, the descriptions of the "Binary" agent were almost identical to the design specification.

When the Agent didn't see you

According to the specification of agent type 1, the agent will not seek out the player when it cannot see them. It will patrol the environment, stay near coins and avoid walls.

For a while, Participant 1 seemed to be under the impression that the agent reacted to the player clicking on the screen and would start chasing them if they did so, even if it didn't currently see them.

> "I've realised that he comes after me when I click, so I try to be as far from him as possible." – Participant 1 (about agent type 1- Binary)

Participant 1 only mentioned this behaviour for the first agent. The reason why this did not come up later is that the Participant switched to the direct control method, where the mouse button is held down instead of clicking, for the later trials. They realised then, that clicking was not causing the agent to chase them.

Another behaviour that was identified, but didn't actually exist was that Participant 6 found that agent type 1 would "search" for the player when they were hiding in a corner of the arena.

> "It's quite a good balance because from C, if you do take everything from one corner, you can just hide in the other corner and they will never find you, but here (Opponent B), if you are trying to hide in the corner, they will go out just enough, to see you." Participant 6 (about Agent C type 2 – Continuous, and Agent B type 1 – Binary)

The patrolling behaviour of agent types 2 and 1 are identical, but the participant was sure that they spotted a difference in the agents' patrolling behaviours. They even went back to test it and found that the area that agent type 1 (Binary) covered while patrolling was greater than the area agent type 2 covered. According to Participant 6, agent type 1 (Binary) would also check areas that had no coins nearby, in an effort to find the player, while agent type 2 (Continuous) would not "search" for the player and instead stay near the coins.

What had actually happened was that Participant 6 had noticed a very subtle difference between agent types 1 and 2. When an enemy agent sees the player only briefly, something which can happen when the player is far way and the edge of field of view passes through them, an agent of type 2 (Continuous) will not immediately turn to chase the player. Since the switch from patrolling to chasing is not immediate, the patrolling behaviour will initially dominate, which can turn the agent away and cause it to lose sight of the player. It will thus seem to "ignore" the player if it catches passing glimpse. On the other hand, an enemy agent of type 1 will immediately switch to the chase behaviour. If an agent of this type sees the player, even if it is only briefly, it will turn to face the player and start chasing. So what the participant perceived as a difference in the strategy and patrolling radius of agent type 1 (Binary) can actually be attributed to a simple difference in the speed of transition between behaviours.

Both of these are good example of Braitenberg's "law of uphill analysis" (Braitenberg 1984), which states that observers tend to overestimate the complexity of the underlying components of a system when analysing it by looking at its behaviour. Participant 1 thought they saw the agent react to their clicks and incorporated this notion into their mental model of the mechanisms behind the agent's behaviour, making it more complex than it actually was. Similarly, Participant 2 observed agent type 1 deviating from its patrol path at some point and thought that this was due to a separate "search" mechanism.

When the Agent saw you

All of the participants indicated that the agent reacted to the player's presence when it saw them. 7 participants stated that this reaction to the player was in some way "aggressive".

Participant 4 made an interesting comparison between the agent's behaviour and a hungry animal saying that it looked like the agent was "hungry" and wanted to "eat" the player. This was also a type of "aggressive" behaviour, but was less objective, indicating that this participant was using animal behaviour in their mental model of the agent's behaviour.

Participant 5 had an interesting take on describing the behaviour of the three agents they tested. They used a narrative featuring three security guards. According to them, agent type 1 was a guard at the "start of their shift" since they seemed more alert and reacted faster than the other two agents. This participant also often acted out the behaviour of the agent and talking in the agent's voice. For example, to describe the reaction of agent type 1 to seeing the player they said:

> "C is definitely immediately alarmed when they see the player, like 'oh no, get them!'" Participant 5 (about agent type 1 – Binary)

Participant 1 was the only participant in the group that could not clearly define the behaviour of the agent when it saw the player. From the demographics, this was the participant with the least amount of background experience with regards to games and AI, but their reasons for why they could not decide on descriptors for the behaviour were clearly expressed. The main issue they had was that the agent did not clearly indicate its emotions. They knew the behaviour seemed aggressive, but did not want to settle on that term, since the agent did not have any visible indicator of an emotion that would justify that aggressive behaviour. So while the participant said they would have written "aggressive" as the behaviour for when the agent is chasing them, they wanted the agent to show emotions and not just actions.

> "So if there was something more to define that behaviour or the kind of emotion, then I would be able to see that more. So like in this case I'm not sure.
>
> Yea I was not sure about the emotion that he would be displaying. So I'm just like, we are chasing each other and I'm just trying to avoid being in his way" – Participant 1 (about agent type 1 – Binary)

The most significant part of these comments is that the Participant says that they want to have a visual indicator of the agent's emotions and that this is what is preventing them from building up a more complex mental model, or narrative for the agent's behaviour.

When you Collected Coins

None of the Participants were sure how the agent reacted to the player collecting coins. Most of the comments were questions asking the experimenter whether the agent did react.

> “He didn’t care did he (C), that I was collecting coins? He (C) wasn’t caring when I was collecting the coins. He (C) was getting angrier, when it was seeing me.” – Participant 2 (about agent type 2 – Binary)

The experimenter did of course not reveal any details, but encouraged the participants to test the agents for this particular reaction. What they found was that the agent would tend to stay closer to the last remaining coins, which matches the specification of the agent.

Participants 3 – 7 noticed that the agent stays closer to the coins when there are only a few remaining. None of them concluded that the act of collecting coins was noticed by the agent.

Participant 6 noticed that the patrolling behaviour can get into conflict with the wall avoidance behaviour. On one occasion the agent tried to circle around the last remaining group of coins, but approached the wall at an angle that caused it to turn away and follow the wall further away from the coins. Initially Participant interpreted this as a new behaviour stating:

> “Huh. Seems like when there are not many coins left, he just stops caring.”
> – Participant 6 (about agent type 1 – Binary)

However, after observing the agent a while longer, they concluded correctly that this was not a new behaviour, but indeed an emergent behaviour. They even pointed out potential issues that could arise from this:

> “Seems like he’s drawn to coins and pushed away from edges. Which can lead to him getting stuck in a corner for example.” – Participant 6 (about agent type 1 – Binary)

This example shows that some participants will not linger on their initial interpretation of the agent’s behaviour and will experiment and observe to find out what is actually causing it. In this case Participant 6 correctly identified an emergent behaviour, instead of attributing what they saw to a new control mechanism. This example stands to counter Braitenberg’s “Law of uphill analysis”, since the participant was eventually able to reduce the number of components in their mental model and understand how the dynamic interactions between the components they already knew existed, caused this new phenomenon.

Agent Type 2- Continuous

Similar to agent type 1, the participants were very good at identifying the main behaviours of agent type 2. Their perception of the behaviour did not deviate much from the specification of the agent.

When the Agent didn't see you

The main difference between agent type 1 (Binary) and agent type 2 (Continuous) is that agent type 2 smoothly transitions between the patrolling and chasing behaviour. This has two consequences. The first is that the agent sometimes does not fully transition to the chasing behaviour if it sees the player's agent only briefly. The second is that the agent will keep chasing the player for a while, even when it has lost sight of them, making it harder for the player to evade.

The first differentiating aspects in the behaviour was noticed by Participants 4 and 6, who noted that agent type 1 (Binary) would immediately turn to chase the player if it spotted them, while agent type 2 (Continuous) would instead return to protecting the coins. Participant 6 concluded that agent type 2 was more interested in guarding remaining coins, instead of chasing the player, than agent type 1.

Participant 6 also noticed the second differentiating aspect, pointing it out as an issue with the way agent types 1 and 2 would both "forget" about chasing the player when they exited their field of view. Participant 6 used a narrative to illustrate how this appeared to them:

> "For the sake of game mechanics, I can see why they stop searching for you once you get out of range, but it doesn't seem very, real.
>
> […]If they're trying to chase you and you go around a corner, they're not just suddenly go 'Oh well thank you for talking' [hand waving gesture]." – Participant 6 (about agent types 1 and 2 – Binary and Continuous)

Participant 4 and 6 identifying the two ways that the behaviour was affected revealed an ambiguity in the question, which should be addressed by making the questions more precise and addressing each scenario separately.

When the Agent saw you

When Participant 4 noticed that the agent type 2 spotted them, but didn't start chasing them immediately, they were puzzled and thought something was wrong with the agent:

> "Oh he saw me at one point and he didn't react [participant looks slightly puzzled, disappointed by this]." – Participant 4 (about agent type 2)

Initially they did not interpret this behaviour as the agent being "suspicious", as was intended by the design. Participant 4 later states the agent seemed "[…] more curious when he sees me now. Didn't seem as angry" (Participant 4) compared to agent type 1. As was pointed out before, there was some ambiguity with the way the questionnaire addressed the two behaviour state transitions that can occur with these agents (entering and exiting the field of view). This comment by Participant 4 describes the behaviour of the agent during the state transition where the player's agent is *entering* the field of view.

In contrast, the other participants besides participant 4 and 6 did not differentiate between entering and exiting the field of view. They all focused on the latter transition (exiting the field of view), which led them to interpret the behaviour of the agent as overall *more* aggressive than agent type 1.

When you Collected Coins

Similar to agent type 1, this agent does not include mechanisms that are triggered by the player collecting coins (unlike agents of type 3 and 4). However, the patrolling behaviour will adapt to the number of coins present in the environment by causing the enemy agent to be attracted to clusters of coins.

Participant 6 noted a difference between agent type 2 and the other agents that they tested (1, Binary and 4, Adaptive), stating that agent type 2 was would stay closer to remaining coins and would be less likely to chase the player than the other agents. The other participants did not identify this difference. Their comments reflect that they thought the agent did not show any direct reaction triggered by the player collecting coins.

It should also be noted that the mature, non-gamer participants attributed aggressive behaviour to agent types 2 and 3 when they couldn't identify other behaviours. Where the other participants used less aggressive terminology that indicated that the agent didn't seem to have a reaction, these two participants saw the aggressive state as persistent.

Agent Type 3 – Remapping

According to the believability criteria ratings from the participants, agent type 3 was the most believable agent out of the 4 types tested. The open feedback from the participants also generally matched the description in the specification. The fact that the agent "cheats" by reacting to the player collecting coins even when it doesn't see the player, did not seem to have a negative impact on the narratives participants told to describe the behaviour, nor did it negatively affect the believability ratings.

When the Agent didn't see you

This was one of the biggest differences between agent type 3 (Remapping) and the simpler agents, 1 (Binary) and 2 (Continuous). While the implementation of the patrol behaviour is the same as in those two, the ability of this agent to shift its primary behaviour from patrolling to aggressive chasing when the player collects coins has a strong effect on how the agent behaves when it can't see the player.

It was expected that the way that agent type 3 can track player without seeing them will be perceived as cheating or unfair behaviour by the participants. However, none of the participants perceived it that way, or used negative terms at all. Generally participants thought that the ability to react to the player, even when it didn't see them made the agent more "exciting" (Participant 2) and fun to play against.

None of the participants were particularly negative about the fact that the agent "cheats" and can start tracking the player when they are still outside of the agent's field of view. All of the participants simply attributed this to the agent "getting angrier" as the player collected more coins.

Only participant 7 was initially confused, wondering how the agent could track them from far away. The participant initially assumed that the agent had an additional sense of smell:

> "How are you, is it like smell or something, are you like. I don't see, how does it know where I am? Is it like smell? Something like smell?" – Participant 7 (about agent type 3 – Remapping)

Toward the end of the trial, the participant eventually settled on explaining the agent's ability to track the player as a result of having a larger field of view than the other agents and not an additional sense.

Participant 7 also used one of the most unique narratives to compare the behaviour of this agent to the others:

> "Sort of like, if a grown up baby. Yea, just like a grown up and this one is more like a child because a grown up won't follow you around, but a child will, like when it's a baby, it will follow its parents around, whereas this one is not following you around, so maybe it's like a teenager or something." – Participant 7 (about agent type 3- Remapping)

In this quote Participant 7 compares the behaviour of the agent with that of a child wanting to stay close to its parent. This was interesting and surprising, since this narrative implies a friendly, positive relationship between the player character and the agent chasing it.

When the Agent saw you

According to the specification, initially the agent will react to the player just like agent type 2 (Continuous). It will be "suspicious" of the player when it sees them briefly and starts chasing them when it sees them longer. When the player picks up coins, the agent "remaps" the primary patrolling behaviour and the secondary chasing behaviour, so that their triggers are reversed. With fully reversed behaviours, the agent will chase the player when it can't see them and stop chasing them when it sees them. The agent will therefore favour keeping the player at the edge of its field of view i.e. "keeping an eye on" the player. Note that the agent will remap the behaviours every time the player picks up a coin, regardless of whether it saw the player at the time.

When the behaviours are fully reversed (after the player has collected several coins), the agent will actually never catch the player, it will just look at them). It was expected that this may be perceived as somewhat incoherent behaviour, but none of the participants noticed this, even though the gameplay telemetry data shows that the all of them played trials where this situation occurred.

Since the agent stops chasing the player, they can easily evade the field of view. However, as soon as the play exits the field of view, the agent begins to chase again. The resulting behaviour, is that the agent will "keep an eye on" the player. This behaviour resulted in some interesting situation in the game, where the participants were surprised that the agent would turn around and find them again after they had evaded. This was different to them, since the strategy to defeat agent types 1 and 2 didn't work anymore.

When you Collected Coins

Participant 1 differentiated between the "aggressive" and "angry" behaviour of the agent when it saw the player and the "hate" it developed when the player collected coins. Hate is typically a more persistent emotion that "anger", so it was good to see that this participant made an attempt to differentiate between the two and identified the remapping behaviour correctly.

It was interesting to see that Participant 7 said that the agent "cares more" about protecting the coins than the other agent types, while at the same time stating that the agent "didn't care" about the player collecting coins. It seems they assumed that the agent behaved like agent types 1 and 2, with the difference that agent type 3 was just more persistent when chasing the player.

It seems that the subtleties of the remapping behaviour were not clearly visible to the participants. None of them noticed that the direction in which the behaviour bias is shifted depends on the agent's current behaviour at the time when the player collects a coin. Participants did observe an overall change in behaviour, but only described this as the agent getting more aggressive over time. There were occasions, when the participants seemed slightly puzzled by what they perceived as some randomness in the level of aggressiveness in the agent, but none of the participants made any statements about this.

Agent Type 4 - Adaptive

The patrolling behaviour of agent type 4 was the same as the other agents. However, this agent has the ability to change its behaviour based on its observations. Two variables differentiate this agent from the other 3 types. When the agent sees the player collect a coin, its "suspicion" of the player will increase. Every time the player collects a coin, whether or not the agent sees them do it, the agent's "forgiveness" factor will decrease. In other words, the less coins there are, the less likely the agent is to stop chasing the player once it spots them. Unlike agent type 3, this agent does not have the ability to "cheat", meaning it will <u>not start</u> chasing the player when it cannot see them. However, It will continue to chase them after it has lost sight of them for a period of time determined by the "forgiveness" value.

As previous results in the analysis of the agents using the believability criteria in section 5.2.3 have shown, agent type 4 was difficult to analyse for the participants. None of the participants were able to create a mental model of this behaviour from their interactions with the agent. Most participants did not notice that the agent adapted to the player's behaviour at all, since this only became apparent if the player made a "mistake" and let themselves be spotted stealing coins. This resulted in agent type 4 receiving the lowest ratings in the believability questionnaire and some of the most negative responses from the participants. The examples in this section represent some of the most significant differences between the design intent and how the agent was perceived by the participants.

When the Agent didn't see you

One of the key behaviour features of agent type 4 is that it will initially not chase the player when it sees them. It needs to build up "suspicion" based on "evidence" (seeing the player collect coins), before it takes up chasing the player.

Participant 4 stated that it seemed like the agent "didn't care about anything in the world" (Participant 4), since they were able to collect coins behind its back, without the agent starting to track them (like agent type 3 did). Participants 5 and 6 both noticed that the agent wouldn't instantly switch to the chasing behaviour when it saw the player and wondered why this was the case.

The main issue with the interpretation of this agent was that participants thought that the agent behaved as if it did not see them, when in fact it should have seen them. Participants tended to attribute this to a physical shortcoming, instead of a mental process. It was the intention of this agent design to seem as if the agent initially "did not care" about the player, until they were suspicious of them, but participants repeatedly stated that the agent "did not see" them instead.

When the Agent saw you

Participants were confused by this initial inactivity of the agent and stated that it was the "slowest" (Participant 5) or that "His field of vision is much shorter than the other ones" (Participant 4). The participants seemed to attribute the behaviour to some physical inability, instead of a mental process of gathering evidence (as intended in by the design). This was also reflected in the narratives they used. Participant 5 stated that agent type 4 behaved "like they had a long day at work" and "one that has to work long hours".

Participant 6 seemed especially puzzled by agent type 4, stating that the agent had "No reaction to seeing me at all" (Participant 6). When they were eventually spotted collecting coins by the agent and it started chasing the, the participant seemed confused stating "Hmm, why didn't it react the first time?" (Participant 6). Similar to participant 5, participant 6 ended up attributing this puzzling behaviour to a difference in the physical capabilities of the agent, instead of a difference in the mental process by stating that the agent has a "much shorter sight range" (Participant 6).

Initially, Participant 7 was not sure whether the agent could see them, saying "Oh it's blind – it's not". They later noticed that the decay rate of the aggressive behaviour became so low, that the agent would not stop chasing them:

> "It seems like it's getting more eager to catch me the longer I stay inside (the FOV). Yea, that's what's happening. Yea it gets really (aggressive). I'm guessing if I stay in there for longer, it would be like impossible to get out of it (the FOV)." – Participant 7 (about agent type 4)

However, they were not able to explain how this happened; only stating that this was due to the duration that he player stayed within the field of view and not correctly identifying the actual mechanism that involves a separate "forgiveness" factor that is decreased when the player collects coins.

When you Collected Coins

When the participant were asked to describe how they thought the agent reacted to them collecting coins, their answers revealed that none of them had discovered that the agent changes its behaviour permanently when the player collects coins. In contrast participants that tested agent type 3 (Continuous) had clearly identified the coin collecting as the trigger for the agent getting "angrier". While the participants noticed that agent type 4 changed its behaviour, they were unsure about the reasons for this change.

Participant 6 stated that the agent "doesn't seem to be aware of me collecting coins at all" (Participant 6). Participant 4 seemed to be teasing the agent when they said "I am picking your Coins [grins]". While investigating how what the agent did when it sees the player, Participant 7 ran several trials to figure out what caused the sudden changes in the agent's behaviour they were observing, but never figured out that collecting coins while being seen by the agent was the trigger. Somewhat ironically, at the end of the trial Participant 5 even suggested that it would be an improvement to the game if the agents were able to react to the player collecting coins:

> "It would also be a bit more interesting if they would change their behaviour in any way if you collect coins. Because their technically supposed to guard them I assume. It's like ladida.[laughs]" – Participant 5 (about all agents)

After filling out all the questionnaire forms, the experimenter interviewed the participants about their answers. During the interview, some of the participants stated that they had noticed that agent type 4 adapted in some way, but could not determine what situations triggered the change. While discussing how the adaptive behaviour of agent type 4 actually worked with to participant 6, they suggested an interesting way of addressing the issue of players not noticing certain behaviours that require a certain interaction from them:

> "I played a game where there was a similar sort of thing (mechanic). They were not interested in you until you've done something wrong, but I remember them basically telling[…]you about it before you ever came across them. And it was just like ah, yea that. When you are aware of it, suddenly it counts a lot more. When you are not aware of it you're like, well, this is easy." – Participant 6 (about Agent 4)

Participant 6 said that they had encountered agents in another game (they were not sure at the time, but thought it was a game from the "Final Fantasy" series by Square Enix), where the agents would not react aggressively toward the player initially. Just like agent type 4, the enemies in that game had to be "provoked" by the player, before they would pursue any aggressive action against them.

This suggests that the inclusion of a narrative in the game can serve an additional the purpose as a way of emphasising or highlighting "hidden" behaviours (or character traits in the narrative) that would otherwise be under the risk of being overlooked. Using agent type 4 as an example, a simple narrative could be "this guard is friendly at first, but if he sees you steal, he will get very angry at you". Even if the player still avoids being spotted and doesn't see the agent's adaption behaviour, they would likely be less confused and more willing to suspend disbelief, since the narrative made them aware that the more complex adaptive behaviours exist and clearly stated how they are triggered. If this player now became curious and choose to test whether the narrative was "telling the truth", they could perform the necessary actions to purposely trigger the adaptive behaviours.

5.2.5 *Number of Trials played by Participants*

After playing against each agent once, participants were allowed to replay games against the previous agents in order to be able to compare their behaviours and/or attempt to improve their score in the game.

It is often regarded an important quality of good games that they encourage repeat plays (especially in traditional monetisation models such as arcade games). In video game reviews, this is often referred to as "replay value". It is therefore interesting to look for reasons why a specific agent type was played against more than others. One reason could be that this was simply determined by the order that the participants were presented with each agent type. Another reason would be the agents themselves, that some types were more intriguing, puzzling, or fun. The latter would provide evidence pointing at architectural design recommendations, while the prior is a methodological concern. Table 21 below shows the breakdown of trials per agent and the total number of trials:

	Number of Trials						
Partic.	0 Tutorial	1 Binary	2 Contin.	3 Remap	4 Adapt	Total (Agent)	Total (All)
1	2	**<u>7</u>**	3	3		13	15
2	1	**<u>1</u>**	**<u>1</u>**	**<u>1</u>**		3	4
3	1	**<u>9</u>**	**<u>9</u>**	4		22	23
4	1	**<u>4</u>**	2		3	9	10
5	1	1	**<u>7</u>**		5	13	14
6	1	**<u>5</u>**	4		**<u>5</u>**	14	15
7	3	**<u>9</u>**	6	6	3	24	27
Totals	10	36	32	14	16	98	108

Table 21 Number of trials per agent type

Table 21 shows that the agent types 1 and 2 were played most. It needs to be determined whether the "popularity" of the "Binary" and "Continuous" agents was due to participant preference, the order of presentation, or the update from agent type 3 to 4 as a result of the user feedback. Table 22 shows the same trial data arranged by agent label instead of agent type. The labels A, B and C correspond to the sequence in which participants were asked to play against the agent:

Partic.	Number of Trials Tutorial	A	B	C	Total (Agent)	Total (All)
1	2	**7**	3	3	13	15
2	1	**1**	**1**	**1**	3	4
3	1	4	**9**	**9**	22	23
4	1	**4**	2	3	9	10
5	1	**7**	5	1	13	14
6	1	**5**	**5**	4	14	15
7	3	**9**	6	6	21	24
Totals	10	46	31	27	95	105

Table 22 Number of trials per agent label

As described in the subsection "Agent Order and Labels" of section 4.2.5, using letters instead of numbers to represent the order and numbers instead of letters to represent the type was a deliberate measure intended to test for participant bias caused by the *impression* that they are playing against a series of agents increasing in difficulty and complexity.

Table 23 below shows the mapping of agent types to labels per participant. Table 24 summarises this data, listing how many times a given agent type was assigned to each label:

Participant	Agent Label mapping A	B	C
1	1 Binary	2 Continuous	3 Remap
2	2 Continuous	3 Remap	1 Binary
3	3 Remap	1 Binary	2 Continuous
4	1 Binary	2 Continuous	4 Adapt
5	2 Continuous	4 Adapt	1 Binary
6	4 Adapt	1 Binary	2 Continuous
7	1 Binary	2 Continuous	3 Remap

Table 23 Agent type to label mapping per participant

Agent Type	A	B	C	Total Trials
1 Binary	**3**	2	**2**	7
2 Continuous	2	**3**	**2**	7
3 Remap	1	1	**2**	4
4 Adapt	1	1	1	3

Table 24 Agent type to label totals

The sum of trials in Table 24 shows that there was a methodological bias with regards to the sequence in which the agents were presented that was caused by the number of participants. Keeping the current experimental design, a minimum of 2 more participants would be needed to remove sequence bias from agent types 1 and 2, such that each agent was associated with each label 3 times. A total of at least 24 participants would be needed to have sequence effects completely removed from the experiment, though this would see each agent played in each sequence spot only once. To be statistically relevant the number of participants would increase drastically. For example, to have each sequence tested by 3 participants, 72 participants would be needed.

Due to the sequence bias, the data set with the agent labels is the more relevant, since it shows that participants played against the initial agents more often than the later agents. The reasons for this cannot be derived from this data set and a separate study should be conducted to measure the effect of sequence bias separately from believability criteria.

However, from the observations made during the participants trials, it can be hypothesised that it is likely that participants were still "in training" during the initial trials against the first agent. None of the participants spent many trials practicing the controls in the tutorial and preferred to quickly move on with the experiment. As a consequence in a revised study, it may be preferable to make the training phase a game itself, perhaps even including an agent to play against so that the participants may learn the controls in context. However, this form of tutorial would have to be carefully designed to make sure there are still significant differences between the training agent and the agent types under investigation.

5.2.6 Questionnaire Results: Game Questions

Following the individual agent trials, the participants were asked to rank the tested agents in terms of difficulty and pick the agent that they found to be the most "fun" to play against. In the final part of the experiment, the participants were then asked to answer a set of open questions about their approach to testing the agents and suggest improvements that they think could make the agents and the game as a whole better.

This section focuses on the feedback that the participants gave on the agents, with a particular focus on the reasoning behind the participants' choices. More general feedback on the game design, controls, and experimental design are presented in the "Further Findings" that follows.

Fun

The participants were asked which of the three agents they tested was the most "fun" to play against. The "fun" factor is the most widely used and often criticised metric used in the games industry. In game review literature it is typically comprised of separate ratings that are then combined into a score out of 100% (although alternative presentations i.e. scores out of 10 or "star" ratings are used in some publications). Examples of the metrics that are usually used are controls, graphics, sound, difficulty and longevity among others. For this experiment, the definition of "fun" was left completely up to the participants and they were asked to explain their reasoning.

Agent Type	Participant								
	1	2	3	4	5	6	7	Total	Voters
1 Binary	-	-	-	X	X	X	-	3	7
2 Continuous	X	-	X	-	-	-	-	2	7
3 Remap	-	X	-				X	2	4
4 Adapt				-	-	-		0	3

Table 25 Participant votes: Agent most fun to play against

The results in Table 25 show that participants favoured the behaviour of both the "Binary" agent type 1 (3 out of 7 possible votes) and the "Continuous" agent type 3 (2 out of 4 possible votes). Agent type 1 had the most votes in total, but was also tested by more participants. As stated before this data is not indicative of a trend, but it is still apparent that in a direct comparison, the participants preferred agent type 1 over agent type 2 (3 to 2 votes, both tested by all 7 participants) and that none of the participants thought that playing against "Adaptive" agent type 4 (0 out of 3 possible votes) was the most fun.

After the participants answered the question, the experimenter asked them what the reasons for their selection were. The reasons that participants gave for their choices varied. Participant 1 selected agent type 2 because it was the only agent they could defeat, however they also said that this was the most difficult agent for them, but they liked the challenge. Participant 4 managed to defeat all the agent types, but preferred agent type 1 because "He has faster reaction time". According to them the agent was also the most difficult. Participant 5 justified their choice in a similar way, saying they liked agent type 1 "because it's more alarming". Participant 6 also picked agent type 1 "because they were harder to just ignore".

Looking at the responses, it seems that the difficulty and the degree to which the enemy agent automatically engages with the player are the main factors behind the participants' sense of fun. While none of the participants commented on agent type 4 (Adaptive) in this section of the questionnaire, their responses while testing it showed that the fact that that agent would often times stay "passive" and not visibly respond to the player's actions was the main reason for its low scores in several categories, including their judgement of overall "fun".

Difficulty

Participants were asked which of the agent types they found to be the hardest to defeat and which was the easiest. The vote distribution in Table 26 and Table 27 below shows that some participants were not too clear when determining which agent was hardest to defeat, while it was easier for them to decide, which agent type was the easiest to defeat. In cases where they could not decide, participants were allowed to vote for multiple agent types instead of just one. While most participants were only a little unsure, Participant 5 voted for all three agents for both questions.

A fair way of considering multiple votes is to count them as "partial votes". Using this system, each participant has a single vote point that is divided between the agent types that the participant voted for.

The "partial vote" counting method involves several steps. First the "partial value" of a participant's vote is calculated by dividing their single vote point by the number of votes they gave. In the next step all the partial values for each agent type are added up giving the "partial vote sum".

To compensate for the uneven number of participants that tested the agent types a proportionate per-participant point value was also calculated by dividing the partial vote sums by the number of participants that tested that agent. This resulting value was then rounded up and converted to an easily readable percentage value, giving the final voting result.

Participant	Agent Type: 1 Binary	2 Continuous	3 Remapping	4 Adaptive	Partial Value
1	-	x	x		0.5
2	-	-	x		1
3	-	x	-		1
4	x	-		-	1
5	x	x		x	0.3
6	x	x		-	0.5
7	-	-	x		1

Partial Vote

Partial vote Sum	1.80	2.30	2.50	0.33
Testers	7	7	4	3
Points per Partic.	0.2571	0.3286	0.6250	0.1111
% Vote (partial)	**26%**	**33%**	**63%**	**11%**

Table 26 Results: Hardest to defeat agent

Participant	Agent Type: 1 Binary	2 Continuous	3 Remapping	4 Adaptive	Partial Value
1	x	-	-		1
2	x	-	-		1
3	-	-	x		1
4	-	-		x	1
5	x	x		x	0.3
6	-	-		x	1
7	x	-	-		1

Partial Vote

Partial vote Sum	3.33	0.33	1.00	2.33
Testers	7	7	4	3
Points per Partic.	0.4757	0.0471	0.2500	0.7777
% Vote (partial)	**46%**	**5%**	**25%**	**78%**

Table 27 Results: Easiest to defeat agent type

Table 28 and Table 29 below show the ranking of the agent types based these results of using both the full and partial vote counting methods.

Rank	Hardest – Partial	
1	3 Remap	63%
2	2 Continuous	33%
3	1 Binary	26%
4	4 Adaptive	11%

Table 28 Ranking: Hardest to defeat agent.

Rank	Easiest – Partial	
1	4 Adaptive	78%
2	1 Binary	46%
3	3 Remap	25%
4	2 Continuous	5%

Table 29 Ranking: Easiest to defeat agent.

These results can be combined by subtracting the percentage values of the *Easiest* ranking in Table 29 from *Hardest* ranking in Table 28. The final "Difficulty" ranking seen in Table 30 splits the ranked agents into two groups. Agents that are considered *Hard* by the participants are indicated by positive vote values, while *Easy* agents have negative values.

Rank	Difficulty – Partial	
1	3 Remap	38%
Hard 2	2 Continuous	28%
Easy 3	1 Binary	-20%
4	4 Adaptive	-67%

Table 30 Difficulty of agents

The final results in Table 30 show that the participants were sure that agent type 4 (Adaptive) was the easiest to defeat, giving it a combined result of -67% for the partial and full vote count. Participants were less confident to identify agent type 3 (Remapping) as the hardest to defeat agent giving it a combine result of 38% for the partial and 50% for the full vote count.

Once again, the partial vote counting method makes a clearer distinction between the *Hard* and *Easy* votes, highlighting the fact that participants were more confident in their votes for agent type 4 (Adaptive), compared to all the other agent types.

There is also a clearer distinction between agent types 1 (Binary) and 4 (Adaptive) within the *Easy* group, than there is between agent types 3 (Remapping) and 2 (Continuous) within the *Hard* group.

Correlation between "Hardest to defeat" votes and "Most Fun" votes

Looking back at the results for the vote asking the participants which agent they thought was the most "fun", a correlation between the participants choices for the hardest agent seen in Table 26 and Table 27 and the most "fun" agent, seen in Table 25.

This correlation persists even if "partial" votes are considered, where the participants could not decide which agent was the most difficult. In no case did the participants vote that an easier to defeat agent was the most fun.

Participant	Difficult	Fun	Match?
1	2,3	2	Yes (partial)
2	3	3	Yes
3	2	2	Yes
4	1	1	Yes
5	1,2,4	1	Yes (partial)
6	1,2	1	Yes (partial)
7	3	3	Yes

Table 31 Comparison between difficulty and fun

This observation supports an old game design trope from the age of arcade games in the 1980's (Koster 2013). At that time, video game hardware often limited the amount of content that could be featured in a game. As a consequence, the best way to maximise profit by increasing the longevity and replay value of the game was to make it more difficult. The "difficult but fair" design rule is still widely used today and in fact "old-school style arcade gameplay" as it is sometimes called has had a recent resurgence in recent years that started with the release of Terry Cavanagh's "VVVVVV" (2010) and Team Meat's "Super Meatboy" (2010).

Correlation between perceived difficulty rankings and game scores

A further match can be found between the participants perceived difficulty rankings and the average best scores that the participants attained while playing against each agent.

Best Score Average Ranking (low – high)

Rank	Agent	Type	Avg. Score
1	3	Remapping	353
2	2	Competitive	494
3	1	Binary	498
4	4	Adaptive	597
5	0	Training	563

Table 32 Best score average ranking

Comparing the rankings in Table 32 and Table 30 shows that the participants' impression of the agents' difficulty matched their scores. The distinction between agent types 2 and 1 was not as strong in the score data as it was in the votes however.

The reason for the match could either be because participants had a good "feeling" for their own performance, or due to the fact that they saw their score at the end of each trial. This seems more plausible initially, but since the participants never saw an overview of their previous scores (note that there was no high score table), they would have based their impression on significant events, such as defeating a particular opponent for the first time. This was reportedly the case for some participants, in particular participant 1, who stated that they remembered that agent type 2 was the hardest, since they practiced playing against it the most and eventually defeated it.

Agent Order and Perceived Difficulty

The experiment tried to minimise the perception that agent would become increasingly difficult. Responses from the Participants show that they assumed agents would become more difficult with subsequent trials, even though alphabetic instead of numeric labels were used and the scenario (of being a member of a game development studio) stated that the agent prototypes were three equal alternatives to be evaluated. Nevertheless participants assumed an increase in difficulty before the trials:

> "Because the answer to "can predict", I'm just seeing if I'm doing the answers right. <u>Because the difficulty is going up</u> (between Opponent A and B), so I'm trying to think about how to (rate them)." – Participant 7 (about agent type 2) [emphasis added]

Looking at the results, the participants did indeed rate the agents in order of increasing difficulty.

Game Prototype Improvement Suggestions

This section presents the feedback from the participants that directly addressed the research tool developed for the experiment. While the previous sections focused on feedback on the agents, this section looks at comments that the participants made about the game itself and lists the problems that they had that may have prevented them from focusing on testing the believability of the agents.

Two participants (2, 3) stated that the game could be improved by including maze-like structures and obstacles for the agents to navigate. While this would make the game more interesting, it would increase the number of 'situations' that the player could get into with the agents, thus providing more variability. Whether including maze-like structures would have improved the usefulness of the prototype for evaluating the believability of the agents is doubtful. Having obstacles would add a level of complexity to the seen/not-seen gameplay mechanic that may be difficult to understand for some users. Navigating the environment would also be more difficult and could cause potential issues such as agents or player getting stuck.

One of the most common suggestion offered by four participants (3, 5, 6, 7) was to let the player compete against all three agent types at the same time. This is an interesting suggestion in that it could potentially address the applicability issue of the "social interaction" believability metric. Participant 3 suggested that the behaviour of the agents could be designed to complement each other, where each agent has defined weaknesses which the other agents compensate for, creating interesting cooperative dynamics. Participant 7 elaborated on this idea, but emphasised the importance of making the characters easy to distinguish, both behaviourally and visually. They suggested that the agents could have different colours to signify their "difficulty" level or personality. Knowing beforehand which agent is more aggressive or difficult to avoid would create an additional layer of strategy in the game dynamics. One concern that Participant 6 offered regarding the inclusion of multiple agents was that the arena would have to be bigger to accommodate for the greater coverage of the three agents fields of view. Otherwise it would be too difficult for the player to find a hiding spot.

Another set of popular suggestion offered by 4 participants (1, 3, 5, 7) was concerned with the control options that the game offered players. These were not suggestions for improving the prototype per se (none of the participants outright criticised or reported serious issues with the controls), but rather improvements to the game dynamics that could make the game more interesting and add additional layers of strategy. Among these additional options were allowing for keyboard controls (participants 1, 3, 7), touchscreen controls (participant 1) and giving the player an additional ability to drive faster for a short period of time (participants 5, 7).

5.2.7 *Results for the Informal Prototype Observation Study*

After the participants evaluated the agent prototypes in the interactive game scenario, they were presented with a series of 5 prototypes that show the agents in a variety of different typical gaming scenarios. In this part of the experiment the participants were not directly tasked with interacting with the agents, but were instead engaged in an informal discussion with the experimenter about the behaviour of the agents presented.

As a guideline for this informal interview-style discussion, the experimenter often referred back to the believability questionnaire criteria. However, this was not done strictly and the participants were encouraged to speak freely and use their own terms to describe what they observed.

Some of the prototypes that were shown to the participants had interactive elements. These never constituted any challenge, meaning that the participant could not "win" or "lose" any of these demos. Rather, the interactive elements, usually in the form of movable "sources" in the environment were included to let the participants "test" the agents' reactions to their environment.

This section discusses the feedback given about each of the 5 prototypes in the order that they were presented to the participants. Significant and common feedback will be highlighted.

1. Hunter & Prey Prototype

The participants seemed very amused by the hunter and prey demo. They were initially confused by the prey being pushed out of the arena, but the experimenter informed them that this was intentional behaviour.

The participants used some interesting comparisons when describing the behaviour of the agents. Participant 5 said they move like "magnets", as if the agents both had the same polarity and were repelling each other, while being attracted to the "heaty" source which must be of the opposite polarity.

All of the participant were able to identify that the agent both wanted to stay close to the target called "heaty" and were essentially fighting over territory. Only participant 1 used the word "territory" and compared the agents to animals however, the other participants were more objective in their descriptions.

Participant 4 was confused by which of the agents was hunting which. It was not clear to them that the brown agent was chasing the white agent, since they thought that the white agent had "teeth". In fact, what the participant identified as teeth were the additional distance sensors that the white agent had, but this visual cue was enough to make the participant confuse the roles of the agents.

None of the participants were able to clearly identify the white (prey) agent's ability to avoid walls and turn tighter corners. Participant 5 noticed that it was "odd" when the brown agent drove in a long arc around the white agent and bumped into a wall, instead of just turning toward it.

2. Conversation Prototype

All of the participants noticed that the attraction toward the group was stronger than the attraction between individual agents. The participants tested this by trying to pull one of the agents out of the group using the movable source in the demo.

Participant 1 seemed very confused by this prototype and was not able to find a fitting analogy to describe the agents' behaviour. Participant 4 immediately said "So they love each other" when the demo started. At the end they concluded "It's like a life. Like a life lesson. Instead of looking for money, you could look for love". This conclusion was likely based on the visual representation of the movable source that could be used to interact with the agents. The source looked the same as the "coins" in the game prototype. Similar to their observations for the "Hunter & Prey" prototype, participant 4 seemed to be very affected by the visual representation of the objects and characters.

Participant 5 initially compared the behaviour of the agents to that of a "molecular structure", with bonds of attraction between the agents. Testing the group's reaction to the movable source, this participant found that the agents reacted to it as if it were a stranger entering their personal space, by backing away from them. They also noticed that "everyone stared" at the stranger when they got close. This interpretation by participant 5 matched the intention of the designer precisely. They concluded with a social analogy "Yea, they always group up together. So it's like none of them want to be alone, in a way. And they always make room for a fourth. So it is kind of like, social even" (Participant 5 about the conversational agents).

Interestingly, it was participant 7 who initially compared the behaviour of these agents to how magnets behave, whereas it was participant 5 who had previously used a similar analogy to describe the behaviour of the "Hunter & Prey" agents. Participant 7 concluded that the behaviour of the agents reminded them of the fish swarm simulations found in the game "Call of Duty: Ghosts" (Infinity Ward et al, 2013), which exhibit flocking behaviour when the player character swims near to them. They noted that even though these agent didn't "scare" like the virtual fish in the game, they seemed to be aware of the interactions of the user in a way that reminded them of flocking.

While participant 6 liked how the agents reacted when a new agent joined the group, by making space for them, they noted that it seemed "rather creepy" how an agent could end up "chasing" another agent when there are only two in "conversation" and one of them pulled away.

3. Capture the Flag Game Prototype

Participant 1 said that the physical struggle between agents fighting for the flag gave them more personality than the agents from the interactive game demo that they had tested. Their awareness of each other and the team play made them more believable as well.

Participant 3 said that the agents move in "a very circular" motion, which implies some criticism for the movement model. This participant had previously offered similar criticism of the enemy agent's movement patterns in the interactive game experiment.

Several participants initially interpreted the "rugby scum" that the agents could occasionally get into as an error. While participant 6 said it was "a little messy", participants 3 and 5 thought that the game had broken or reached a deadlock situation from which it couldn't recover. Participant 5 noticed that the flag-recovery system, which resets the flag to the middle of the playfield if it exits the arena to the side, was a good fail-safe to ultimately resolve this situation. However they criticised how the flag was recovered, saying that playing the flag toward the centre can result in one team getting an unfair advantage.

Participants 1, 4, 6 and 7 said they saw team tactics being used by the agents. All except participant 1 correctly identified the "blocking" behaviour of teammates attempting to protect their flag carrier from enemy agents.

Participants 2, 3 and 5 also did not identify tactical behaviour. Unlike participants 4 and 7, these participants did not notice any coordinated team behaviour or intentional blocking behaviour, stating that all the agents just go for the flag all the time. Participant 5 described their movement as "swarming". The only tactical move that this participant noticed was that the flag-carrier would occasionally try to rush in between two approaching opponent agents instead of driving around them.

Participant 5 critically noted that it seems like an opponent would willingly "pass" the flag to an attacking opponent sometimes. The participant said it would be better if the flag-carrying agent would fight to keep the flag.

Both participants 5 and 6 noted that the flag transfer seemed to be dependent on the angle at which the agents collided, stating that collision from the side rarely resulted in a transfer, while collisions to the back of the carrier were more likely to result in a steal.

4. Combat Prototype

The main issue with the behaviour of the agents in the combat prototype was that they would repeatedly get into a stand-off situation, where one agent was stuck in the corner, while the other would keep firing at it. The agents were unable to get out of this situation on their own and would only change their behaviour if the energy of one of them reached a value below 50, which triggers a fleeing response.

The participants' negative comments were usually directed at the RED agent, since it seemed to be the more defensive of the two agents and tended to get stuck more often. Participant 1 interpreted the more aggressive BLUE agent as an angry male and the RED agent as female. Initially participant 5 said the RED agent *looked* very aggressive, saying that the colour is usually associated with aggressiveness, as it was in the game prototype they had previously played. However, the participant noticed that the agent behaved more "causal" and "sluggish" than the blue agent.

Participants 5 and 7 noticed the "X" markers that the agents left trailing behind them and wondered what the purpose for them was. Participant 5's first thought that the markers just showed where the agents had been, but the participant kept investigating what effect the markers might have on the agents. Participant 7 got closest to the actual purpose of the markers, stating that they were some kind of target. Discussing this with the experimenter both participants eventually noticed that the agents were avoiding the "X" markers, but were still not sure exactly what the purpose of this behaviour was. From a design perspective, this mechanism was implemented to prevent the agents from getting stuck in corners for too long, but as participant 5 humorously pointed out, this was clearly not working as intended.

Participant 5 also described the movement of the agents as "laggy". This observation refers to the agents' ability to move laterally. This was a feature implemented to make the movement of these against more human-like and less like the agents in the other demos, which only use a differential drive for manoeuvring. The term "laggy" has negative connotations, since it is usually used to describe the stuttering seen in network-multiplayer games when the connection is bad. In this case it can also be describe as the opposite of "smooth". Asked to elaborate on that statement the participant said that it was hard to describe, but that they found the movement of the agents to be smooth at times, but then to be interrupted by movement that seemed abrupt, "like pixel movements" and looked as if they were being controlled by someone using a digital input device like a keyboard over an analogue device like a joystick.

On the other hand, participant 7 liked the agent's ability to perform sharp turns. They stated that this was the kind of reaction that they would include in the design for the agents in the game prototype that they tested before.

The shooting mechanic was also criticised. Most of the participant pointed out that it was odd that the agents were shooting all the time and seemingly "randomly". Participant 4 said the BLUE agent was a "newb" or inexperienced player who did not know how to shoot. Participant 6 also pointed out that the BLUE agent seemed to be able to corner the RED agent, but would constantly keep missing their shots, even though the RED agent could not move.

Participants liked the low health mechanic, which sent agents fleeing to return to their bases for a health recharge. Participant 4 called this "clever" behaviour, since it showed the agent "cared" about its health. Participants 5 tested what would happen when the agents were not able to return to their bases by placing the movable recharging stations outside of the arena. This resulting in the agents frantically driving up and down the wall closes to their stations. The participant interpreted this as the agents "panicking".

In some instances both agents would get stuck trying to drive around a wall. Participant 4 said this made them look like "a fly" that was seemingly unaware of a window. Participant 5 also noticed this, criticising that the agents would try to go through walls instead of around them. They made humorous remarks about how the agents seemed to be “talking to the wall” and cursing it.

5. 3D Agents Prototype Video

Unlike the other prototypes presented to the participants in the observation study, this prototype does not represent a typical situation found in a game. Instead, the video presented to the participants shows a comparison between two agents; a BLUE agent navigates the environment using a virtual eye sensor and a PINK agent that simply drives around blindly, while avoiding obstacles. The test here was to see whether participants notice a difference in the behaviour of the two agents. The experiment also looked at how the participants describe the behaviour of the agents, with a particular focus being on how they interpret the more subtle behaviours that emerge from the interaction between the BLUE agent and the various coloured objects in its environment.

The responses of the participants to this prototype varied greatly. Participants 2 and 3 both interpreted the behaviour of the agents as mostly random, but did notice that the BLUE agent pushed the coloured balls around more than the PINK agent, which also seemed to be more likely to get stuck around corners.

The other participants developed more complex mental models to explain the behaviour of the agents. Participant 5 compared the behaviour of the BLUE agent to that of dung beetles they had seen on TV in a show called “Insect World”. To them it looked like the agent was foraging for food. Initially the participant thought the agents were competing:

> “It’s basically like an insect fight. In a way you have to wait for a long time for insects to find each other and start fighting.” – Participant 5 (about the virtual 3D agents)

At one point the BLUE agent bumps into the PINK agent, which starts to first move away and then racing toward the BLUE agent. Participant 5 interpreted this scene as a playful interaction between the agents, describing the BLUE agent taunting the PINK agent saying “come on, you can’t get me” to the PINK agent which turns away and replies “I’m scared!”. The participant seemed to be very amused by these interactions between the agents.

Participant 7 thought that the BLUE agent was "counting" the balls by moving them all out of the middle to the sides. This was, of course, not an intentional property of the BLUE agents design, but an interesting interpretation by the participant. Participant 4 had a similar theory initially, stating that it looked like the BLUE agent was trying to push all the balls to one place.

Participant 1 also interpreted the behaviour of the BLUE agent as some form of foraging behaviour. However they also criticised that it was not very clear what the agents were thinking and difficult to discern what they were trying to do. The participant assumed that this scene, like the other prototypes, was something found in typical games and wondered if it was their inexperience with games that prevented them from understanding what the agents were doing. The participant focused on the interaction between the agents and found that the BLUE was trying to keep the food away from the PINK agent, even pushing the other agent away from the coloured balls at times. However, the participant was confused when the BLUE agent occasionally pushed a ball toward the PINK agent, which seemed to disagree with their previous interpretation of the BLUE agent's motivations.

All of the participants noticed a clear difference between the behaviour of the BLUE and the PINK agents, stating that the latter seemed mostly dormant, while the prior was actively roaming around the environment and deliberately interacting with objects. Participant 5 said that it was "So weird that the PINK one just keeps going up and down".

One of the more surprising results of this experiment was that the BLUE agent's reaction to the different colours was unclear to the participants. Even though the agent was programmed to avoid Green balls and push only the Red balls, none of the participants noticed a difference. The reason for this was likely that the speed at which the BLUE agent pushed the red balls was so dependent on the environment around the ball. In crowded environments or near walls, the agent would slow down or even stop mid-push when its distance sensors triggered as they touched walls or other nearby balls. This seemed to "dilute" the identification of the pushing behaviour, ultimately also interfering with the illusion that the BLUE agent was trying to achieve something *with* the balls. Participants were scanning through the video to find a good example of the BLUE agent interacting with the balls to identify its objective, but were not able to find one that was consistent.

The 3D graphics and the realistic physics were also a factor in the interpretations of the participants. Participant 4 seemed to take the 3D graphics more "literal" than the 2D graphics used in the other prototypes, asking "Are those trucks?". Participant 5 commented on the visible antennas, saying that these made the agents look like "electric plugs" with wheels.

5.2.8 FURTHER FINDINGS

During the study several findings were made and questions were raised that didn't directly address the research questions that the experiment was designed to answer. However, many of these findings are interesting in their own right and contribute to the field of study.

Participants' Testing Methods

Participants were told to evaluate and compare the agents. They used different approaches. Some followed the suggested script for the experiment and tested each agent once. The participants that did this also had the least engagement with the experiment itself. A particular example of this is participant 2 who did not seem engaged or inquisitive at all.

The two major behaviours observed were "competing" and "probing". When "competing", the participant is trying to "defeat" the agent in the context of the game. While "probing", participants interact with the agent to find behaviour patterns to exploit. From the participant's perspective, "competing" has the goal of performing well and achieving a higher score, while the "probing" serves the purpose of learning about the agent, even if it means that a higher score is sacrificed.

When "competing" the player is guided by the game rules. They mean to player well or to "defeat" the opponent. When players want to learn about a game's mechanics (including agent behaviours) they test the game by "probing". This activity sees players wanting to answer a specific question or test for certain agent behaviours. The probing approach is something commonly found in computer game design, where "trying out" or trial and error methods of learning are a fundamental building block of what is commonly referred to as "gameplay", the way that players engage and interact with a game. Experienced players will have previous "gameplay" experience to draw from and will have certain expectations that they then test or "probe" for. These may be related to any element of the game mechanics, including controls or agent behaviours that they may have observed in other similar games prior to the test. An example of this is participants referring to other "stealth" games, where it is typical that the player character is allowed to "sneak up" on enemy agents, without being noticed by them.

It is important to consider what a participant is trying to achieve when they are playing a trial against an agent. What might seem like competitive play, may in fact be probing behaviour. In a revised experiment, both approaches should be considered and perhaps tested separately. So the competitive trials should be a separate event. Competitive events should also be limited in the number of tries a player gets. If the number of tries are not limited, the players are not reacting to the agents in the game anymore, since they will have learned what patterns can be used to exploit enemy behaviour. While it would be possible to prevent rote learning by introducing random enemy behaviours, this could potentially cause issues with regards to testing and comparing participant results since the difficulty level might be affected.

A good example of a technique that is used to eliminate rote learning in competitive games, without introducing random agent behaviour, is a mode of play referred to as a "daily challenges". For example, in the platform action game "Spelunky" (Mossmouth. 2009) new, special levels are made available to the players to compete on, but they only get to attempt them once. This way, even experienced players, who may know how to exploit the game in every other level they played, will only be able to use their experience, but not their pattern memory.

Participant's Methods of Describing Their Mental Model

During the analysis using Nvivo 10 it was found that participants described the agents' behaviour in several ways.

Technique	**Participant Examples**
Describe the behaviour dynamics **objectively.** Compares agent to an object.	He's **moving away** from this. (P3) They **move like magnets**. (P7)
Ethological comparison **-** Comparing behaviour to animals.	He wants to own that **territory**. (P1) It looks like **a bug collecting food**. (P5)
Using **psychological** / emotional terms	The agent is **afraid** of this. (P1) Oh he **likes** that one. (P5)
Anthropomorphising the agent and Speaking *to* the agent	No **you're** not getting me (P1) I'm picking **your** coins [grins] (P4)
Acting out of demonstrating a behaviour	He's like **"OK, I'm gonna get him**!**"** (P4)

and speaking *for* the agent	"**Oh well thank you for talking**" [hand waving gesture] (P6)
Storytelling – developing a narrative to describe the behaviour	"They **are like guards** and **that one is bored** with his job." (P5)

Table 33 The main techniques for describing agent behaviour

This re-enforces the notion that video games are a combination of classical media and require for a combination of different forms of expression (that are related to other media) to be adequately described.

Participant Play Performance Anxiety

After the tutorial trials and before starting their first trial against the opponents, participants 3, 4 and 7 made statements showing concern about the difficulty of the task ahead. Two assumed they would play badly, stating "Oh I'm gonna do awfully" (3) and "I feel it's going to be really difficult and I'm not good at the game." (4), while one seemed concerned asking the experimenter "Is it difficult?" (7).

The scenario made clear that it was not their performance that was being evaluated, yet it seems the experiment being carried out in the context of a game presents itself as a challenge to be met to some users. Interestingly the three participants who made these statements were second year (4, 7) and final year (3) Computer Science Games students, all with strong background experience in playing games.

While it could be assumed that the performance anxiety expressed here may be due to being used to taking games "seriously", the first year games students that participated in the study (5, 6) noted similar regular gaming habits, yet didn't express concern about the difficulty.

Differential Drive Issues

When a source that a Braitenberg vehicle is targeting is placed directly behind and between its two sensors, the resulting movement trajectory will be a large arc instead of a quick turn. This is due to the implementation of the differential drive, which directly converts the difference between the sensor measurements to actuators.

The reaction of the enemy agent to the player approaching it from behind is one of the main points of criticism identified by several participants. Participant 3 was the first to exploit this behaviour against Agent C (Continuous), but merely remarked it was "interesting". While answering questionnaire question 6 on "coherence" about the same agent (B - Continuous), Participant 4 described the issue more thoroughly:

> "Coherent. Well… What was weird was that when I go from behind him [pointing at the position behind the RED agent on the screen], I'd expect him to turn around really quickly, but he does like a huge circle, which makes him quite slow to catch up with me, because I'm already out of his field of vision that time. So he doesn't know where I am anymore. So coherent, not that much I'd say.
>
> I reckon Opponent A (Binary) was the same though. Like he (RED agent) did also big circles around him (GREEN player agent)." - Participant 4 (about Agent 2 - Continuous)

A previous game prototype saw the agent creating "points of interest" in the location of coins that the player picked up. The player had to be within a given "hearing" range for the point of interest to be created. If multiple points of interest existed in the arena, the agent would investigate them in the sequence that they were created in. In addition to creating points of interest to detect coin collecting players, it was also used to implement the agent "turning around" when the player came too close to them from behind. The agent had a "personal space" which would create a point of interest if the player entered it, simulating the agent "feeling" the player behind them. When a point of interest was created behind the agent, it would stop patrolling and turn around to investigate the point, usually spotting the surprised player.

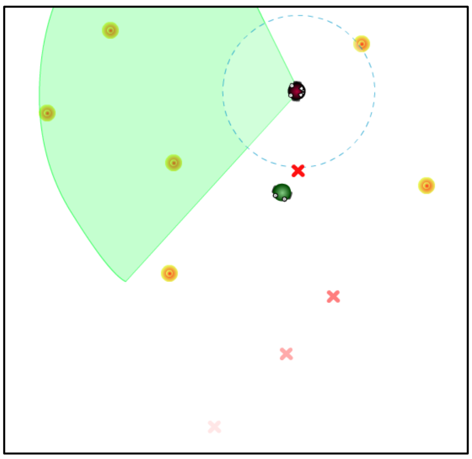


Figure 39 Game prototype including hearing and points of interest

Figure 39 above shows a screenshot from that earlier prototype. The red crosses are visualisations of the sounds that the enemy agent could hear. The dashed outline around the enemy agent was an area in which the agent would "hear" the player if they came "within earshot".

Steering Issues

A subtle issue with the controls that was revealed is related to the Vehicle-style movement of the player's green agent. The way that movement is implemented does not allow the agent to turn around on the spot, without moving forward as well. There is a minimum turning radius, which can interfere with the controls in two ways.

Participant 6 found that rapidly changing directions and pulling the mouse "over" the green agent's body can cause unexpected behaviour. If the player clicks or drags the target directly behind and between the two distance sensors, the agent will continue in the direction it was going in, instead of turning around. This behaviour is rare and only occurred once during 108 trials. Figure 40 below shows the "Target" (mouse position) and "Player" agent's position during this instance from time 186 to 194 during Trial 10, playing against Agent 2 (Continuous).

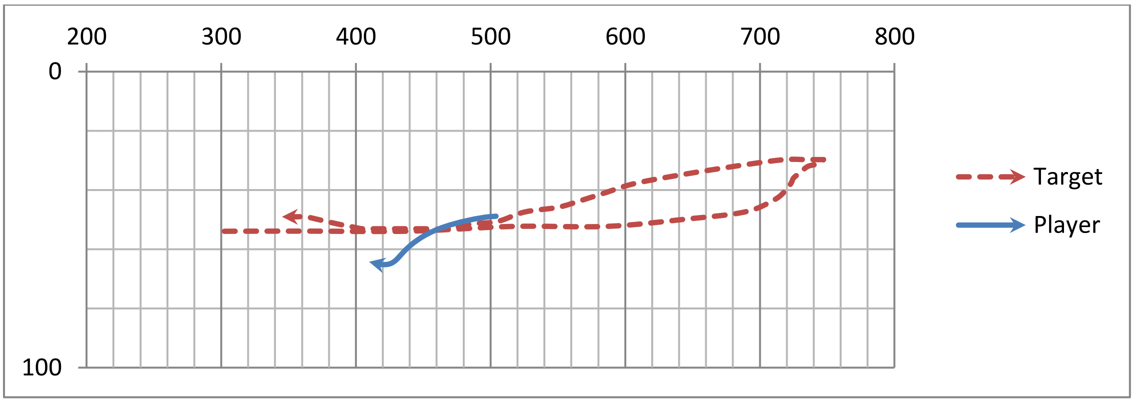


Figure 40 Player drags target over the agent, but agent does not turn around.

Figure 41 below illustrates the same instance as Figure 40 in storyboard form. It shows how the player agent continues in a straight line when the user drags the movement target directly over the agent. This unexpected behaviour from the player's agent causes the player to be spotted by the enemy agent.

Figure 41 Storyboard illustration of the control issue that can occur

At 1, the player notices that they are on a collision course with the enemy agent. At 2, they attempt to change direction and drag the target (green arrow) right, over their agent at 3. At 4, the agent should have turned around, but keeps driving left. At 5, the player notices this unexpected behaviour. At 6, they react as they notice that the enemy agent has spotted their agent. At 7, they attempt to regain control of their agent by dragging the movement target back over and in front of it. At 8, the participant has regained control, but has been spotted by the enemy agent.

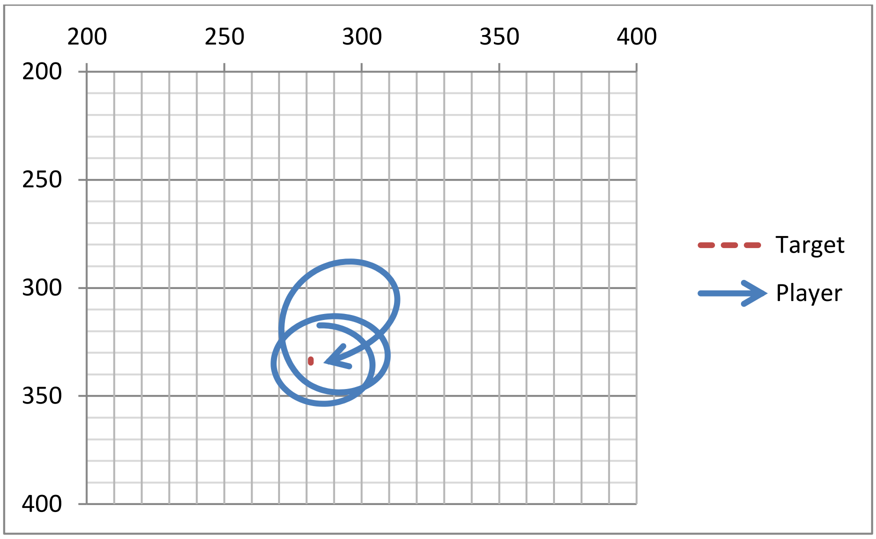


Figure 42 The player's agent orbits the stationary target instead of stopping

Figure 42 shows another issue that was observed is that the agent will occasionally not come to a stop on the movement target as expected, but will instead orbit around it. From time 182 to 218 in trial 15, Participant 3 noticed their agent doing this and was only able to regain control by moving the target. This behaviour is the result of the player agent's speed being too high when near the movement target and the distance threshold that marks the area around the movement target in which the player agent will stop completely, being set too small.

Analysing adjectives to find implied criticism

During the analysis a set of codes that was found to be particularly useful stems from the participants use of the word "just" and its synonyms. When participants were describing the behaviour of agents and their experience of playing the game, they used this term in a variety of ways. A text search of the transcripts for the use of "just" and similar adjectives and adverbs like "only" revealed many of the useful statements cited in the findings. The word "just" was used the most out of these, with 147 references found in total. Table 34 lists the different ways that participants used the word "just", sorted by the number of references:

Subject	Description	Example	Refs.	Par.
Agent	Limited, Criticism	Just does this	79	7
Player	Justify, Focus	I am just checking	16	5
Player	Criticism, Not enough, Wrong	Before I was just	15	6
Agent	Focus, Emotion	Just wants this	12	4
Concept	Realise Something Complex is Simple	So it's just this	6	2
Agent	Can't explain behaviour, without reason	It just did this	6	2
Experiment	Expectation, Procedure	I need to do just this?	4	3
Agent	Comparison (Negative inflection)	It is just the same	3	3
Concept	Comparison, Exactly like	Just like	3	2
Concept	Assume something Simple is Complex	Is it just this?	3	2

Table 34 Uses of the word "just" sorted by number of references

Searching for "only" (23 references) found even more critical statements. While this word was used differently than "just" and mostly to defined quantities and disjunctions, many uses similar to those found for "just" and were in fact interchangeable. Table 35 shows the different usage types and the number of references found for each.

Subject	Description	Example	Refs.	Particip.
Quantity	Few, Not Enough, Only NUMBER	Only one of these	9	4
Exclusive	Limited to, Only CRITERIA	It is only in terms of	4	2
Agent	Exclusive, Only GROUP, THIS	Only this one	4	3
Time	Exception, Only WHEN, Only AT	Only when this happens	3	3
Conditional	State change, Only IF, Only TO	It does this, only to do this	2	2
Disjunction	Apart from group, Only THIS among	It's the only thing besides	1	1

Table 35 Uses of the word "only"

After categorising these, the researcher looked for references where the use of word "only" was a synonym for one of the uses of "just". Only references that didn't change in meaning when expressed using "just" were included in this list, show in Table 36. In total, 13 of the "only" references were considered synonyms of "just".

Subject	Description	Example	“Just” Synonym	Refs.	Particip.
Quantity	Not Enough	Only this many	Just this many	4	2
Quantity	A small amount	Only a few	Just a few	3	1
Feature	Limited, Criticism	Only does this	Just does this	3	2
Concept	Limited to, Exclusive case	Only in this case	Just in this case	2	1
Quantity	Enough	Only these	Just these	1	1

Table 36 Uses of the word "only" as a synonym for "just"

When searching for emergent topics using content analysis tools such as Nvivo, it is often surprising what searches provide the most useful results. In this case the majority references listed above were valuable, critical feedback on the prototypes presented to the participants. Along with the codes based on the questionnaire questions and the initially assumed themes, these references were one of the main tools for finding significant information in the data set.

5.3 Discussion

The results of this empirical study addressed three questions that were set to address the research question pertaining to the human factors that affect the development of believable virtual agents. The results address the questions of design intent and the use of labels to describe behaviour, the use of heuristics to evaluate agent behaviour and give insight into further factors that affect suspension of disbelief in a gaming context.

The comparisons between design intent and user interpretations of behaviour have shown that users are able to correctly identify the labels for strong emotions such as “fear” and “aggression” easily, while the labels for “exploring” and “love” were not as indicative. “love” was often confused with “aggression”, while “exploring” was regarded more as the absence of other behaviours, rather than a behaviour itself. From an architectural perspective this is notable, since both “fear” and “aggression” are excitatory base behaviours while “exploring” and “love” are inhibitory behaviours that require the presence of another passive drive. Further study could explore whether there is a connection between forward-driving excitatory connections and the user perception of agent *agency*.

The use of heuristics for agent evaluation, in particular “believability metrics” was tested in this study. The results show that there is still a need for refinement before the metrics can give a more reliable indication of the impressions that the users had. In the current study, the informal feedback that users gave, often contradicted their formal answers on the questionnaire. Possible reasons for this were discussed and refinements to the questions that address these issues were proposed. The most common among these were misinterpretations of the terminology used in the questions (1-awareness, 2-understandibility, 6-coherence), the inapplicability of the questions to the agent being investigated (7-experience, 8-social) and the need to summarise observations made about individual behaviours into one statement about the whole agent (4-visual impact, 5-predictability).

The participants' responses to question "8-visual impact" led to the discovery of a misinterpretation by Gomes (2013) of the concept of "controlled visual impact" by Lester and Stone (1997), which was the inspiration for this metric. Lester and Stone define "visual impact" as a property of individual *behaviours* and the ability to *control* the visual impact of their behaviours as an important quality of *agents* (Lester and Stone 1997). Therefore, question "8-visual impact" should not be used to evaluate complete agent architectures and should instead be used to evaluate individual behaviours. A question and evaluation format similar to that of dimension "9-emotional expressiveness" was suggested.

Even though there were issues surrounding the use of heuristics, the qualitative approach provided useful insight into human factors that affect the perception of believability and suspension of disbelief in gaming agents. Findings included the preference of users for (and more accurate labelling of) simple forward-driving excitatory behaviours over complex inhibitive behaviours. They also revealed a major issue surrounding the use of subtle "hidden" behaviours that rely on specific user interactions as a trigger. The participants' believability criteria ratings showed what catastrophic effect these can have on believability, but the qualitative content analysis revealed suggestions how this issue can be addressed. Users suggested methods for preventing this issue, which include *visualising* the internal state that controls the behaviour, *demonstrating* the behaviour using a set-piece or *telling* the user the behaviour exists through the use of narrative.

The qualitative approach itself proved to provide excellent insight into the factors affecting the interaction between the users and the game as a whole. Detailed feedback and recommendations were given by the users that was not limited to the agents. Further findings revealed issues with the player controls (which informs the use of bottom-up architectures for avatar agents), the players approach to trial and error learning in games and their methods of communicating gaming experiences and their mental model. The usefulness of certain keywords that reveal hidden criticism was highlighted to inform future work that uses a qualitative content analysis approach in the evaluation of gaming agents.

6 EXTENDED ARCHITECTURES

This chapter discusses the possibilities and limits of the bottom-up architectural approach used to develop the agents in this thesis. The previous chapters have demonstrated how simple agent architectures can produce complex and interesting emergent behaviour and that using a Synthetic Psychology development approach can lead to novel solutions.

6.1 NEW SENSORS AND ACTUATORS FOR GAMES

During the development of the simulation environments and the agent simulations described in Chapter 4 several extensions to the basic architecture were developed.

Strafing wheel

One of the most immediate concerns when adapting the differential drive "Vehicle" platforms for games was that the virtual agent would all move like cars. The strafing wheel, specified in section 3.3.2 and used in the "combat" prototype in section 4.3.4 allows the agents to move laterally at the same time as the differential drive is active. This is an elegant approach to addressing the issue that exploits the fact that virtual simulations are not restricted by the rules applying to physical agents.

It also shows that the theoretical model behind the "Vehicles" architecture "Taxis, Kinesis and Decussation" (Braitenberg 1965) extends to more complex platforms than the differential drive used in the experiments in this thesis.

"Pheromone" Markers

During the development of the "combat" prototype in section 4.3.4 it was quickly discovered that the agents had difficulties escaping from long corridors and corners. Inspired by previous work on the "Hunter and Prey" prototype (see section 4.3.1), a "pheromone-like" system was developed. This saw the agent dropping a trail "markers" behind it that it would "flee" from. It was surprising that this simple, intuitive mechanism seemed to in fact implement a basic form of depth-first search; forcing the agent to go forward and explore new areas and (temporarily) preventing it from back-tracking. While this could also be solved using more traditional path-finding approaches, this solution remains "embodied" in the sense that it uses only the existing sensory system of the agent.

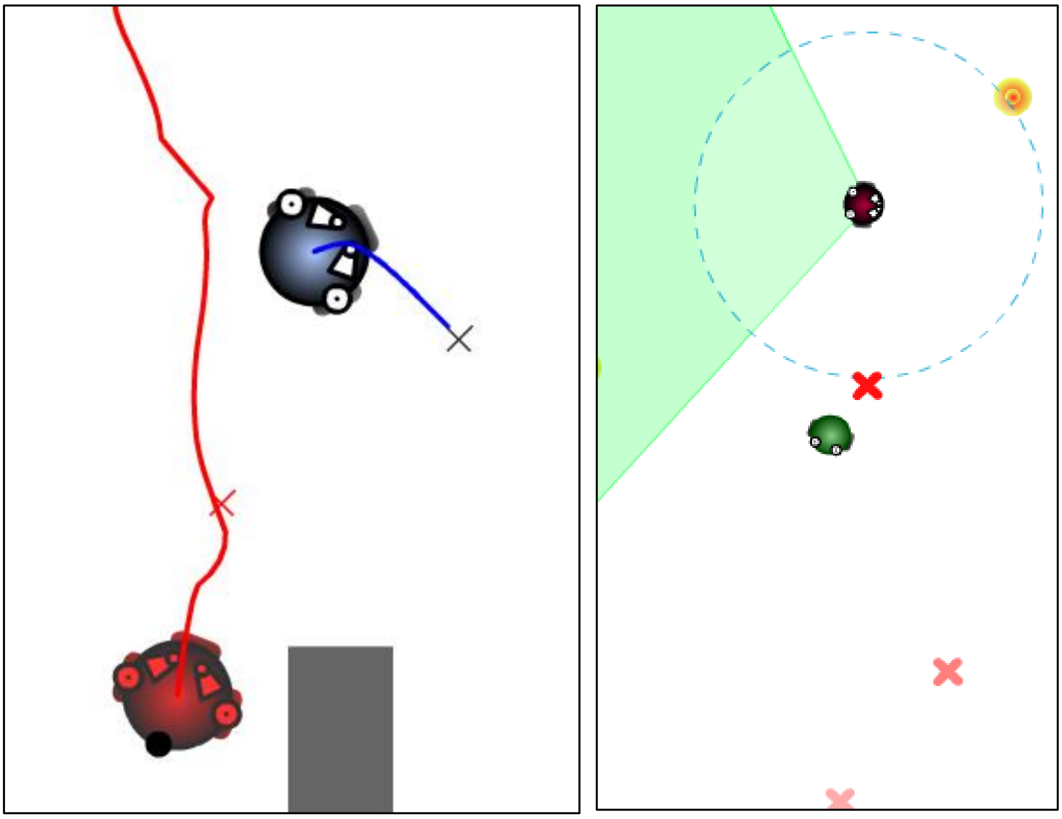


Figure 43 Combat and Game prototype showing "Marker" concept

Figure 43 shows the two prototypes that utilise the marker concept to solve navigational problems. The first picture shows the agents in the “combat” prototype using their markers to get around obstacles and avoid getting stuck. The markers drop automatically when the agent doesn’t move and the agents will “flee” from their own markers. The second picture shows an early prototype of the “game prototype” that was used in the user study. This version used markers to indicate “points of interest” that the enemy agent is attracted to. These two examples demonstrate how the same concept can be applied in opposite ways, as an attractor in the game prototype or as a deterrent in the combat prototype.

Hybrid Systems

The “Capture the Flag game” prototype (see section 4.3.3) demonstrated how easy it is to integrate these kinds of bottom-up architectures with existing finite state machine approaches. The agent architectures presented in this thesis are essentially a more complex form of the “steering behaviours” that are commonly used in games development, so this didn’t come as a surprise. However, proof-of-concepts like the discovery of an emergent “personal space” behaviour when developing the “conversational agents” (section 4.3.2) or the “interactive game prototype” (Chapter

4.2) show that the merit of this design approach is that it facilitates the discovery of emergent behaviours and incremental *evolution* of an agent design that top-down design approaches lack.

The Virtual Eye

The 3D Simulator was a test platform developed to explore some of the more advanced sensory systems that rely on a first person perspective in conjunction with a much more realistic physics simulation. The fact that the virtual camera and physics add "noise" to both the sensing and the behaviour part of the architecture adds another layer of possible *emergence*.

It did, however, also serve to show the current technical limits of the simulation system. The way the current implementation handles the processing of visual data was too inefficient to allow for higher resolutions. In addition, it currently relies on individual frame buffers on the graphics card. This means that individual simulations are actually directly hardware-limited, making them difficult to port to other systems.

For future work an implementation using graphics card shaders could certainly provide benefits and replace the current system. This would also allow of the integration of connectionist systems, into the virtual eye. This was something that was attempted (see section 6.3 in this chapter), but was not feasible with the current simulation engine.

6.2 ARTIFICIAL EVOLUTION

One of the major topics in relation to Braitenberg inspired architectures is the automatic tuning and configuration of the sensory-motor connections using artificial evolution. This was discussed in the review of research on that topic in Chapter 2.2.5.

While they were not used in this thesis, genetic algorithms have great potential as a tool for game AI development. Several research projects seeking to develop human-like behaviour in virtual agents have already successfully employed this approach to tune their systems (Magerko 2004; Thurau 2004; Livingstone 2006; Wang 2009; Laird 2012). One of the reasons for this success, is that these systems have a model to be compared against – other human players.

However, when the goal is to create believable non-human agents, without the aid of existing models, this approach becomes much more difficult to apply. This is one of the reasons that this thesis seeks to further refine our understanding of what aspects of agent behaviour facilitate the perception of the "illusion of life". A theoretical approach to defining and measuring the "fitness" of believable agents may be a viable approach.

Another approach would be to build on the success of the groups that study "human-like" believable agents and try to adapt for non-human characters. This would still use human-behaviour as an input model, however, it would see the human players "act" in character. An apt comparison would be that the developer acts as a "puppeteer", who teaches the agents how to act.

As a proof of concept, Anil Seth's (1998) system for artificial evolution, which was also developed on the basis of Braitenberg Vehicles, was implemented in the 2D Flash simulation environment. Despite being very slow to process the generations (10 generations with a population of 50 agents taking roughly 20 seconds), the test was successful. The result can be seen in Figure 44, which shows the paths that every agent took in its lifetime and the final run of the fittest specimen. In the bottom-right of the screenshots the behaviour graph shows the main difference between this implementation and Seth's. While the sensory-motor connections in Seth's model consist of links made from three straight line components, the model used in this prototype uses Hermite curves with 4 parameters.

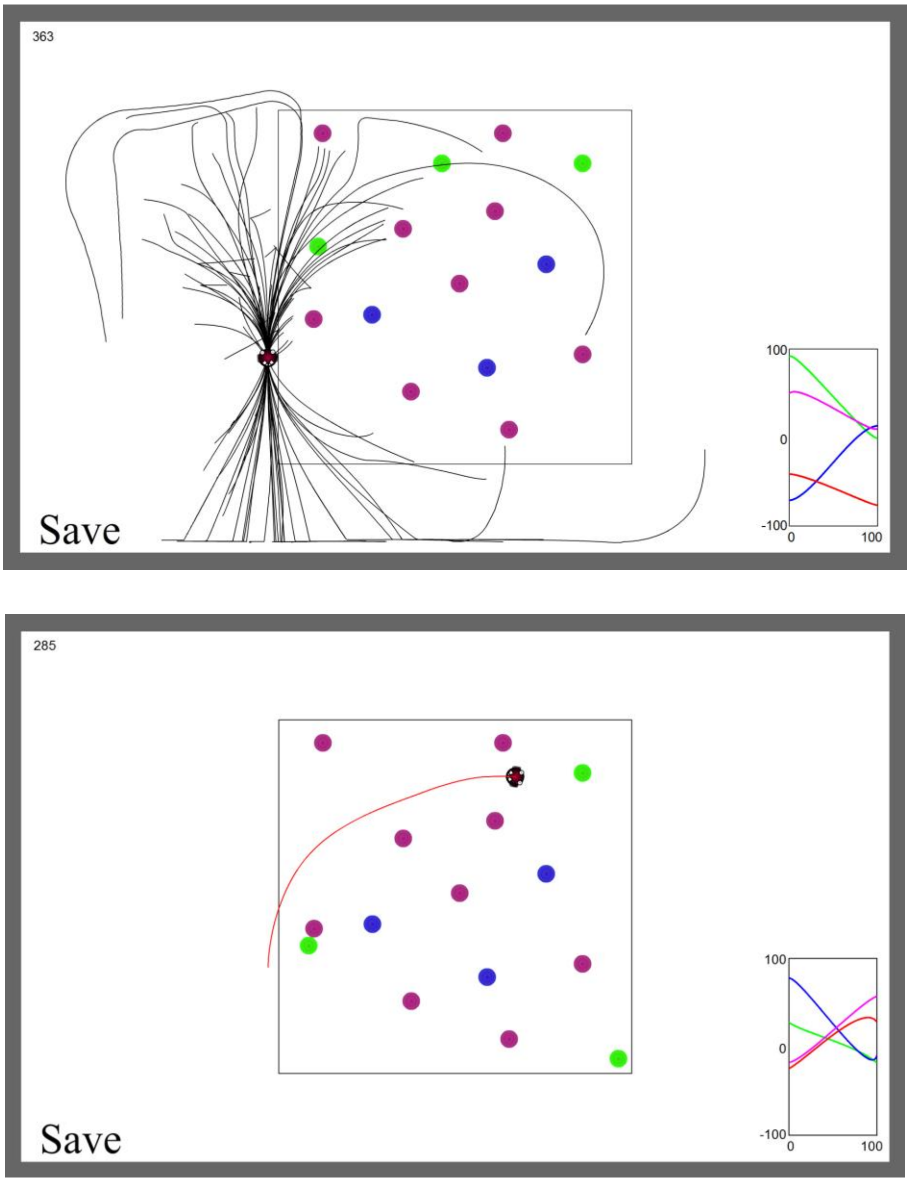


Figure 44 Genetic algorithm 2D Flash simulation test

6.3 Connectionist Components

As this thesis developed and explored the possibilities of using Synthetic Psychology to develop bottom-up architectures of ever-increasing complexity, the work quickly became involved with connectionist architectures. The literature review of research projects that had built sophisticated systems with Vehicle-like agent platforms (see Chapters 2.2.5 and 2.2.6) served as an inspiration to integrating connectionist mechanisms into the agent model developed for this thesis. The later thought experiments in Braitenberg's "Vehicles" provided further inspiration.

In his thought experiment, Braitenberg proposes a holistic model of the mind that was incredibly intriguing. In addition, the process which he presented to develop such a model seemed startlingly simple, yet the literature review of associated work did not appear to conclusively reveal why other projects had not attempted to design an agent architecture based on it.

This led to the development of a series of prototypes that attempted to interpret Braitenberg's theoretical model to form a specification and create an implementation of his model. The following sections summarise this process, which shows what potential for further development and which shortcoming this architectural approach has.

From Instincts to Prediction

All of the architectures that were previously discussed in this thesis essentially fall into the category of "instinct-driven" agents. This is not to say that the behaviours have to be basic. But what it does mean is that while instinctual behaviours may change according to new surroundings (like the Coin-patrolling behaviour of the game agents presented in Chapter 4.2.3) a purely instinctual agent cannot learn to override these behaviours with something completely new.

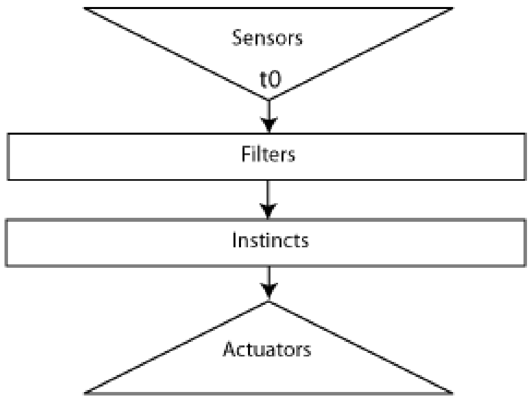


Figure 45 Instinct-driven, reactive brain model

Figure 45 illustrates a purely instinctual model. The term "filters" is used here to indicate any sort of processing that is done to the sensory stimulus before it is passed on to the rules that govern what to do with the input. These rules are the "instincts" or "behaviours" as they were called previously in this thesis.

Another example for purely instinctual agent architectures are the evolved agents discussed in the previous section 6.2. Even though these might evolve sophisticated non-linear sensory motor connections and exhibit behaviour that seems adaptive they still don't possess that critical quality of forming new connections.

Adapting Braitenberg's Connectionist Associative Network

This introduces the concept of "connectionist" architectures. A "connectionist" architecture is one that is able to re-wire its internal sensory-motor connections. Typically, and this was also Braitenberg's viewpoint, it will do so based on patterns in the incoming sensory stimulus.

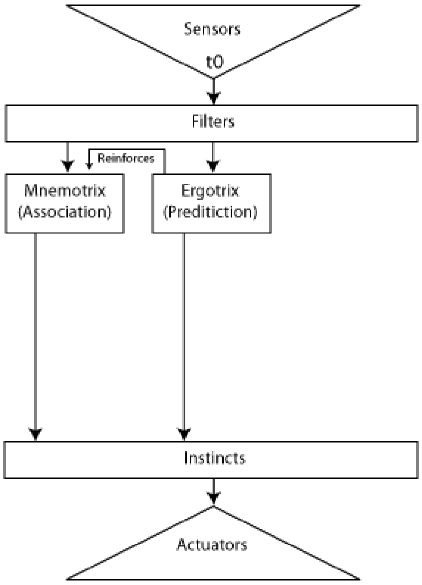


Figure 46 Braitenberg architecture with predictive Ergotrix network

Figure 46 shows the previous purely instinctual architecture with the addition of the two connectionist components that Braitenberg (1984) proposed. While the "Mnemotrix" component represents a system for associating disparate sensory stimuli into phenomenological sets, the "Ergotrix" components is able to remember and reproduce temporal patterns that occur.

A series of experiment were conducted to replicate this behaviour and a series of prototypes were developed, culminating in a prototypical "predictor network" that showed some potential for application in a gaming context, albeit not in the way that it was initially expected.

The reason why they ultimately failed and could not be included in any agent architectures was that the model did not present a suitable method for dealing with feedback between the recurrent connections.

Threshold Control

To address this issue which he refers to as "epilepsy", Braitenberg suggests including a separate device that manages the activity in the system by dynamically adjusting the thresholds of all the neuron-like elements in the Vehicle brain. This Threshold Control Device, or TCD for short, would monitor the "number of active brain elements" and raise or lower the thresholds to prevent escalation, or encourage activity. He adds that "in real life, the input for this threshold control device might be the rate of change of the number of active elements, in order to give it an opportunity to foresee the catastrophic explosion of activity before it happens" (Braitenberg 1984 p.63).

These descriptions leave a lot of room for interpretation. This comes as no surprise, since Braitenberg's thought experiment was never meant to be a formal specification. Consequently, several approaches to threshold control were tested that were interpretations of Braitenberg's description of threshold control and also informed by similar theories from the literature:

1. Global threshold control based on overall activity or connectivity in the network. This can be looked at in two ways, which were both suggested by Braitenberg (1984):
    a. Current total activity/connectivity in the network
    b. The rate of change in total activity/connectivity in the network.
2. Local threshold control with different thresholds set for each device. This approach also had two separate themes.
    a. Thresholds based on the activity or connectivity of individual threshold devices.
    b. Taking relative timings of activity into account to promote linear/didactic network organisation was also considered in this approach. This approach was inspired by spike timing dependent plasticity (Bi and Poo 1998; 2001).
3. Completely decouple internal association of the networked devices from the internal activation signals they send to each other. Differentiate between internal "associative" activity and external "sensory" activity and prevent associations from being made based on internal activity.

At the time, approach 2 was deemed the most plausible from a biological standpoint, since a device that connects to every individual neuron in the brain (in addition to the existing inter-connections between them) to control their threshold made approach 1 seem implausible. Approach 3 was considered a radical approach, since it completely eliminates the possibility of associations caused by internal activity.

"Mnemotrix" Associative Networks

A significant amount of effort was invested in developing and testing a connectionist architecture inspired by Braitenberg's description of Mnemotrix wires. Since he simply describes Mnemotrix wires as a connection that gets stronger when two connected devices are "at the same time traversed by an electric current" (Braitenberg 1984 p.29), he leaves a lot of room for interpretation. Even though the "Biological Notes" at the end of his book suggest that he essentially meant to describe the traditional McCulloch Pitts artificial neuron model (McCulloch and Pitts 1943), his descriptions of the *behaviour* of these connections suggested a much more sophisticated kind of connection. In fact, it turns out that the closest relative to the Mnemotrix wires are continuous time recurrent neural networks (CTRNNs) (Beer 1990).

This led to the development of several prototypes that represented different interpretations of Braitenberg's model and attempted to replicate the behaviour from his descriptions.

Global Threshold Control

The behaviour of the networks produced using approaches 1a and 1b followed two basic patterns. Either the threshold devices were too aggressive and prevented the organisation of devices into associated groups, or the TCD failed to prevent "epilepsy", leaving interconnected groups to sustain their activity indefinitely.

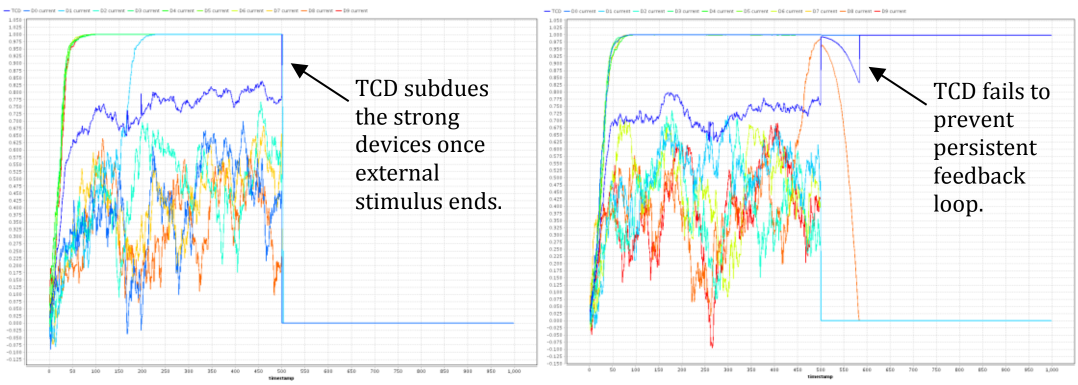


Figure 47 Example output of connectionist network with threshold control

Figure 47 shows these two behaviour patterns in an example of the output of a network that implements approach 1a, where the threshold value set by the TCD is simply the mean of the activity of active elements.

In the first image, the threshold control device subdues all activity after the external stimulation of the network is stopped half way through the trial. In the second image, the threshold control device fails to subdue a group of devices that are stuck left stuck in a persistent feedback loop as a result.

Using approach 1b that uses the “rate of change” in activity instead of current average activity did not bring better results. Figure 48 shows that this approach causes the output of the network to “dither” between high and low values. This happens because of the high number of devices controlled by a TCD with global threshold control, which means that even small changes to the threshold has a huge effect on overall activity. Attempts to “smooth out” these strong reactions failed as well, since this simply resulted in the strongest interconnected device groups “taking over” the network again, as was the case with approach 1a.

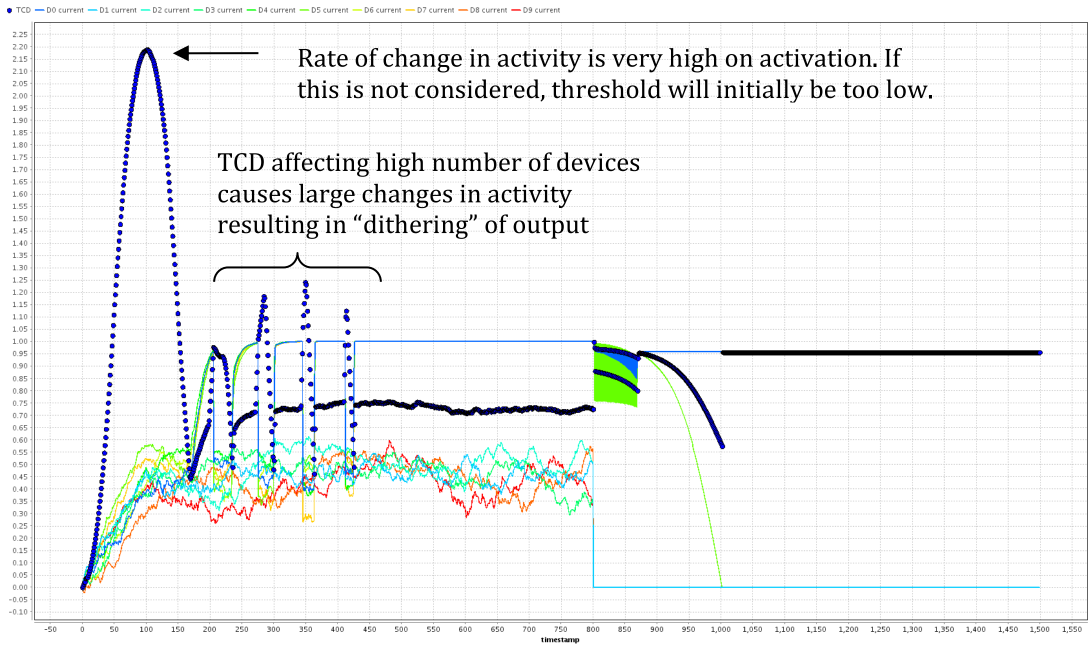


Figure 48 Behaviour of a network using a TCD based on rate of change

Several more attempts were made at using global threshold control to prevent feedback loops, but the behaviour patterns remained the same. The focus eventually shifted toward approaches 2a and 2b which considered local threshold control to be a possible solution.

Local Threshold Control

Experiments using local threshold control i.e. adjusting the individual thresholds of connected devices based on their activity and/or connectivity showed promising results at first. A version of approach 2a saw the threshold of the devices temporarily increased to its maximum (1) after they fired and gradually return to its default (0.70) on subsequent time steps. This effectively stopped any chance of groups of devices re-activating each other in turn.

Figure 49 shows the local threshold control of this network to be more "stable" than the global threshold control illustrated in Figure 47 and Figure 48. However, all activity still ceases as soon as the external input from the sensors is turned off halfway through the trial. This eventually leads to the "starvation" of the network activity and weights as illustrated in Figure 50. Figure 49 shows that this approach at least manages to keep the connection weights under control, ensuring that coincidence, rather than feedback loops are responsible for weight increases.

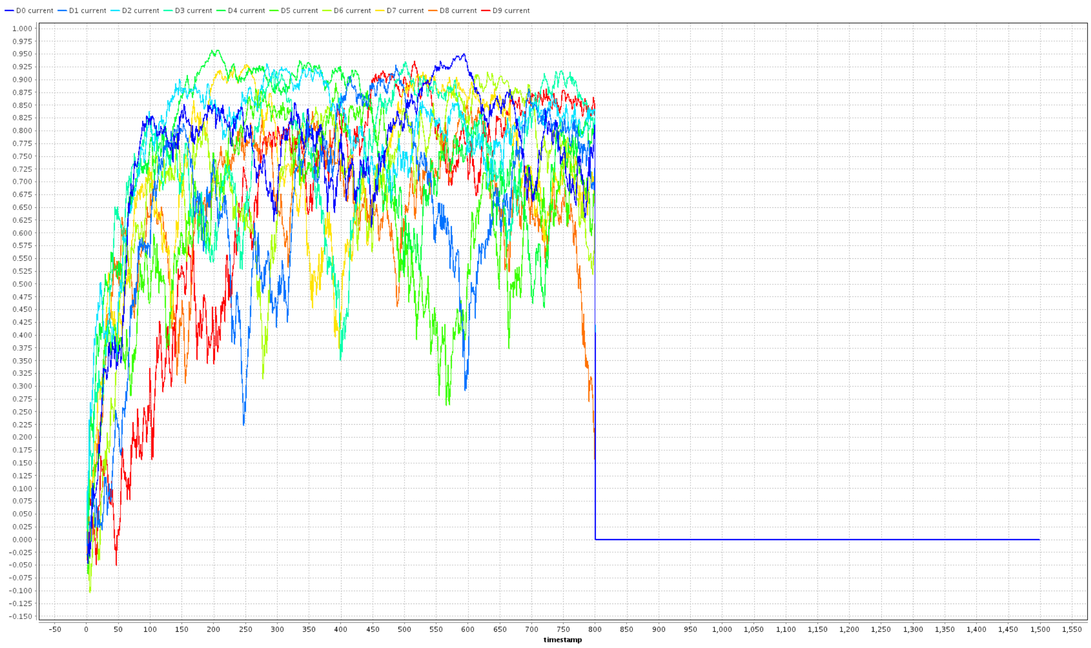


Figure 49 Local temporal threshold control prevents activity escalation

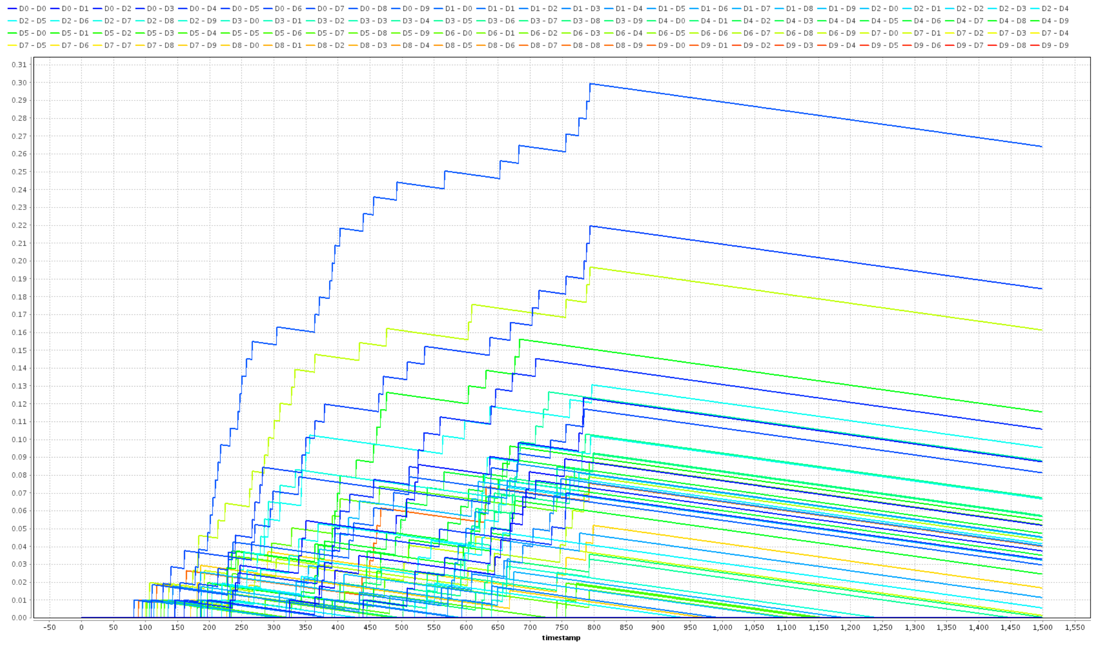


Figure 50 Local temporal threshold control fails to induce internal activity

This model was considered the most successful of the Mnemotrix wire adaptions since it at least prevented the network activity from escalating and reach an irrecoverable state.

"Ergotrix" Associative Networks

Similar to the description of "Mnemotrix" wires, the concepts behind Braitenberg's "Ergotrix" wires (Braitenberg 1998 p.56) left much room for interpretation. Essentially a form of "Temporal Pattern" memory, two functional requirements outlined the behaviour that this connectionist system should be capable of:

1. The temporal pattern memory network associates only elements that are active in succession within a brief delay and not those, which are active simultaneously. This differentiates it from a basic associative connectionist network (Mnemotrix).
2. Memorized patterns can be reproduced at an arbitrary speed. If they are reproduced at a more rapid pace than they are likely to occur as sensed via the sensory system, the network acts as a predictor.

Especially the second requirement potentially introduces an enormous amount of complexity into the model. The first requirement initially seemed fairly simple to implement and the initial model did so with success.

Simple Predictor Model

The first change from the previous associative network model was connection weights now needed to be bi-directional, such that device A could have the ability to send activity to device B, without B being able to send activity back.

The second change was that the network needed to be aware of its previous states. At any given time step, the network should have the ability to "look back" at the previous step to see which devices where active then. For each device that became active on the current time step, it would then establish a connection *from* every device that was active on the previous time step.

While this system worked, the resulting network was only able to "predict" one time step ahead. In addition the feedback problem that was previously observed when developing the associative network reappeared.

Dynamic Feedback Loops

While the local threshold control approach 2a was successful in controlling the activity of a connectionist network that associated connected devices that were active *at the same time*, it did not prove quite so successful in preventing escalation in a system that associated devices that were active *in sequence*.

The reason for this was that the "predictor" allowed devices to associate themselves in "chains". These were groups of devices that became active in sequence. The trouble began when a chain of at least three devices "looped back" to the first device. Since this would re-trigger the initial chain, a dynamic feedback loop was formed.

Neither the global, nor local approaches to threshold control were able to prevent these dynamic feedback loops. This eventually led to the admission that controlling a system with temporal dynamics such as these would require a far more sophisticated model than had previously been anticipated.

If even a simple model such as this had stability issues, adding the functionality that Braitenberg proposed with the second requirement – being able to reproduce patterns at an arbitrary speed – seemed infeasible.

However, a radical change to the previous approaches to threshold control eventually led to a usable pattern prediction model.

Decoupling Internal Association from Internal Stimuli

The first change from the previous models was that the *predicted* activity was separated from the *actual* activity in the network. Instead of adding the internal activation energy to the total charge of each device (the *activity,* which determines whether it will fire on the next time step), the energy sent internally from other devices in the system is instead accumulated in a new property, the *prediction*.

To test and illustrate this concept a prototype predictor was developed in Adobe Flash. The "Predictor Grid Prototype" allowed users to interact with the network by "drawing" input patterns onto the grid. This input "stimulus" pattern was displayed in green, the "prediction" values were displayed in red.

By separating device "activity" and internal association from the "prediction", the network was stable and able to visually predict patterns, based on what the user had previously drawn. However, this initial model was still only capable of predicting one time step ahead. Another issue was revealed when a user interacted with the network for a prolonged period of time. As the weight values accumulate, the network quickly becomes saturated. The predictor can then no longer differentiate between patterns, which leads to false predictions.

Pattern Differentiation by Using Short Term Memory

The solution that was developed introduced the concept of a short term memory (STM) to the predictor model. Instead of storing a discrete list of previously active devices, the model simply added decay to the sensory input "activity". This current "activity" value determines the prediction values sent out to associated devices.

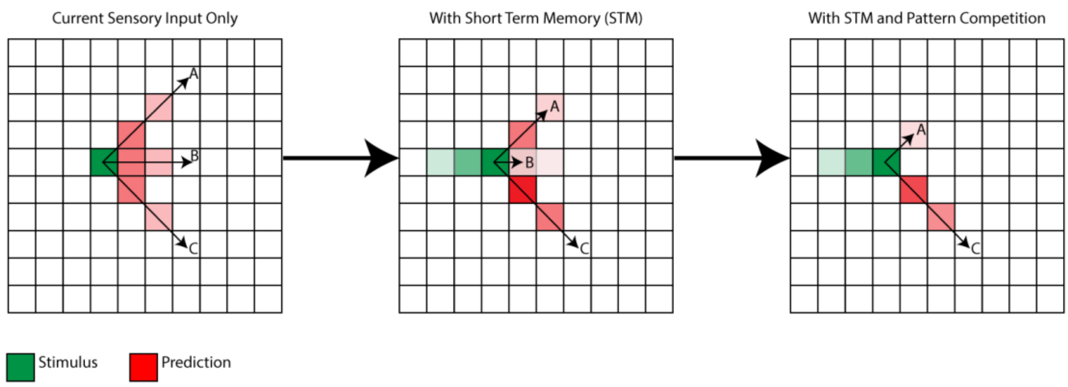


Figure 51 Predictor with short term memory and pattern competition

Figure 51 illustrates how multiple predictions from a series of "past" devices (in the STM) now accumulate, meaning that predicted devices that have multiple previously active devices (in the STM) "pointing" to them, will have significantly higher prediction values.

To emphasize the effect of the devices in STM even more, their activity also inhibits all other devices in the network that they are not associated to. With both the STM and pattern inhibition in place, the amount of interference between patterns was greatly reduced. The final grid in Figure 51 shows the predicted pattern with but STM and pattern competition applied. Figure 52 shows a screenshot of the temporal pattern predictor implemented in Adobe Flash and highlights the decaying *activity* values (grey), the current *stimulus* (green) and the *prediction* (red).

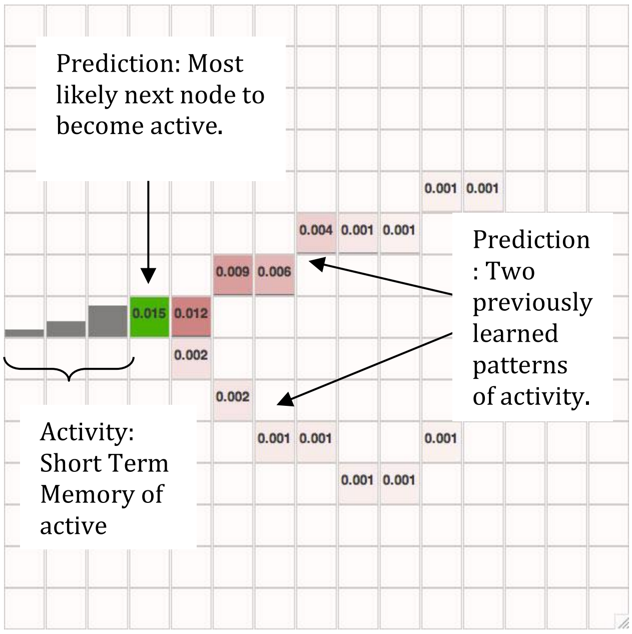


Figure 52 Screenshot of the temporal pattern predictor

Testing the Predictor in an Agent

After this successful predictor prototype, the model was tested as a component of an agent architecture. Two approaches were tested:

1. Use the temporal pattern predictor to predict the location of a tracked object. Inspired by the use of "probabilistic occupancy maps" in (Isla and Blumberg 2002).
2. Use the predictor to predict incoming sensory data.

The implementation of the first approach used the predictor to help an agent anticipate the movement of a moving target. To do so, the arena that the agent and the target were in was divided into a coordinate grid with the same number of cells as the predictor. The model would then receive the current location of the moving target object as its input. As described in the previous section the output of the predictor is typically a set of "predicted" cells, with their *prediction* level corresponding to the amount of "evidence" accumulated from the active devices in short term memory. To make this output usable for the agent in the prototype, the *prediction* values of the devices were compared and only the location of the "most likely" next cell was returned. This location was then used to set the position of another target object that the agent could switch to chase instead of the "real" target.

Similar to the results reported by Isla and Blumberg for the user of probabilistic occupancy maps (Isla and Blumberg 2002) the predictor was able to improve the behaviour of the agent, by allowing it to anticipate movement patterns. Figure 53 shows that this resulted in the agent being better at following the target. The main limitation of the current predictor model is that it can only track the location of a single point on the map.

The current prototype shown in Figure 53 demonstrates the potential use of the predictor network as a form of probabilistic occupancy map. In addition, the feedback from users in the empirical study in Chapter 5 confirmed Isla and Blumberg's assertion that the ability to predict movement patterns is beneficial to perceived believability.

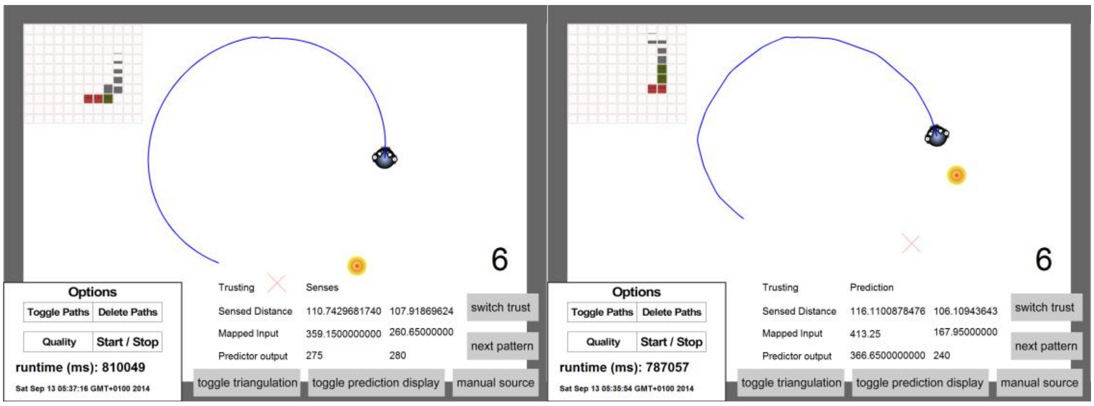


Figure 53 Predictor network used as a probabilistic occupancy map in an agent

The second approach to using the predictor as a component in an agent architecture attempted to use it to predict the sensory input of the two "distance" sensors that the agent uses to track a moving target. This implementation worked when the agent was stationary and the target did not move behind the agent, but failed in every other case.

When the agent moved, it was not able to differentiate between the change in sensory stimulus caused by the movement of the target and that induced by self-movement. This is a general problem that can only be addressed by introducing the concept of movement to the model and compensating for the effect of self-movement on the sensory system.

The second issue was a simple triangulation problem. Since the agent uses only two distance values to determine the position of the object, there are always two possible locations that the target could be in. Figure 54 illustrates this issue, showing that the indicated sensory reading could mean that the target is in front or behind the agent. In both cases the sensory stimulus would be the same.

For future will attempt to address both issues by incorporating the current position and movement of the agent into the triangulation calculations.

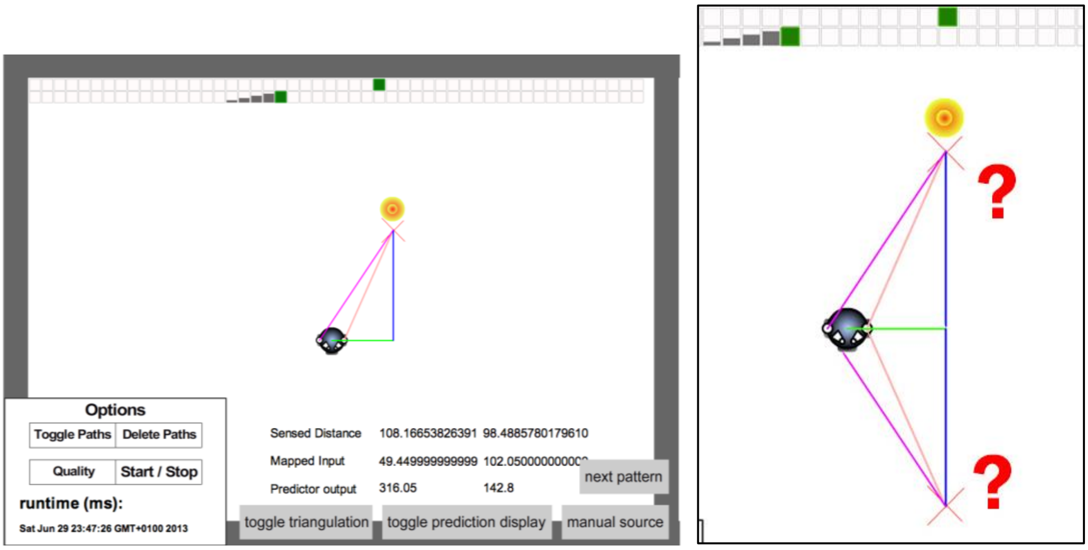


Figure 54 Using the predictor network to predict sensory input

6.4 Discussion

This chapter has explored some of the further work that was carried out to explore the limits of the architectures and simulation environments that were used in this thesis.

The new architectural components such as the strafing wheel, distance sensor types and pheromone markers were all additions to the basic agent model that were developed intuitively in order to solve problems in the development of the gaming agents. The ease with which these additions could be made and how well they integrate with the control theory behind the agent model showed the strength of the synthetic approach, as often time the development of a new component, or in some cases even happy accidents, were able to lead to new design ideas and solutions.

The connectionist and artificial evolution models presented in this chapter were proof of concepts that show that the simulation environment can be used in conjunction with these. However, these tests also showed the limits of what the platform, in particular the Flash simulation engine can handle from a performance perspective. Nevertheless they too highlight the potential that still rests with this agent architecture

7 CONCLUSIONS

The starting hypothesis was that the Synthetic Psychology approach has the potential to create complex behaviours from simple components that are interesting, believable, effective and computationally efficient.

Using Synthetic Psychology as a design approach and extended Braitenberg Vehicles as a platform, this thesis reports on the design and implementation of a set of agents in a variety of gaming contexts. Experiments were then formulated in the attempt to test these and investigate the hypothesis.

Mathematical proof is not yet tractable - and cannot address the subjective responses involved in interpreting agent behaviour - as an approach to testing the hypothesis. User testing is a complex area due to the subjectivity of the field. The most recent approaches from the sectors of believability evaluation and user experience analysis were used to investigate the hypothesis. The agents tested were simple enough to use components that are mathematically formulated, yet complex enough to exhibit interesting emergent behaviour characteristics.

This chapter addresses each of the major parts of this thesis and underlines how these contributed to answering the research questions. It then highlights the contributions of the thesis to related fields and suggests further work that builds on the results presented.

7.1 USING A SYNTHETIC PSYCHOLOGY APPROACH

The agent development process documented in Chapter 4 demonstrated that "Synthetic Psychology" is a viable approach to designing and building agents for games. It shows how this approach provides an alternative perspective on agent design that promotes synthetic, rapid prototyping with a focus on observation to create *emergent,* dynamic behaviour. Further work may see this development approach becoming even more viable with the assistance of new formal models of Braitenberg's Vehicle architectures that are currently being developed by other research groups (Salumae 2012; Rano 2014a; 2014b).

The empirical study in Chapter 5 compared the *design intent* formulated in Chapter 4 with the *interpretations* from a group of users interacting with the agents in a gaming context. This served to elicit the underlying human factors that contribute to the perception of believability in virtual agents in general and the use of mnemonic psychological “labels” to identify complex agent behaviours as Synthetic Psychology proposes it in particular. The results showed that labels relating to overt emotions (anger, fear) were easily identified by observing their behaviour, while subtle behaviours (explore, love) need to be assisted by other expressive means. Feedback from the users gave suggestions into what means are appropriate.

Chapter 6 reviewed the new sensor and motor components developed for the agent models presented in this thesis. It further explored Braitenberg’s later architectures, which gave insight into the range and limits of what is possible with the current toolset and looks forward to what extensions may be added in future work.

7.2 Contributions

This thesis directly contributes to the fields of game AI and the study of believability in virtual characters. The findings indirectly inform robotics, artificial life and computer game design.

Insights into the discrepancy between design intent and user perception gained from the user study (Chapter 5) inform the on-going debate in games development between users and developers arguing over whether users actually *want* more realistic and complex AI (see Chapter 1.2 for an overview). Both the numeric and qualitative analysis showed that this debate is justified. In particular, the results from the evaluation of the “Adaptive” agent type 4 demonstrated that important agent behaviours can easily be missed if they rely on the user performing specific actions, and how this can lead to an irrecoverable break in suspension of disbelief.

The user feedback on and the refinements to the believability metrics suggested in Chapter 5 inform the current discussion on the use of heuristics for the evaluation of virtual agents (Chapter 2.1.7). Specifically, five modifications to the research tool that was originally proposed by Gomes (2013) were proposed that make the questionnaire easier to understand, more specific and potentially more accurate in determining the believability of virtual agents.

The agent architectures and tools developed for the thesis serve as a proof of concept for promoting the use of a Synthetic Psychology approach for the development of gaming agents. The agent architectures demonstrate how this approach can be applied to a variety of gaming contexts while allowing developers to approach typical challenges in game AI from a novel perspective. The "combat" prototype shows how "marker" objects can be used to implement a simple, yet effective navigation strategy. The "conversational" agents use only two cooperating behaviour configurations, but display at least four emergent, distinct and identifiable behaviours. The "capture the flag" game prototype integrates cooperative and competitive behaviours with a finite state machine. The enemy agents in the game prototype used in the empirical study implement a dynamic exploring behaviour that lets them find efficient patrol routes that adapt to changes in the environment.

All the architectures and the software developed for this thesis have been and will continue to be used as a constructionist teaching tool. This provides a simple to use platform for experimenting with bottom-up architectures in robotics and games development.

7.3 Final Considerations and Future Work

The work presented in this thesis can be expanded upon from both a methodological and architectural standpoint.

The use of GA's to optimise the agents for a particular gaming context should be further explored. The issue here remains the need for fitness criteria that take into account the subjective qualities "difficulty" and "fun". The refinements of the believability metrics suggested could inform the development of a more robust set of metrics in future work.

From a methodological standpoint, the inclusion of more sophisticated formal analysis and behaviour visualisation tools for Braitenberg-inspired agents (Salumae 2012; Rano 2014a; 2014b) into the Synthetic Psychology development process may further aid design decisions. The review of Braitenberg's later architectures (Braitenberg 1984) in Chapter 6 highlighted several formal issues that that this approach may address.

The new additions and refinements to the believability metrics presented in Chapter 5 address several of the current issues with interpretation and applicability of specific questions. Additional user tests involving a large enough group of participants to ensure statistical significance should be conducted where the original and refined version of the believability metrics are compared.

To gain further insight into the use of a synthetic psychology approach a new experiment could be developed that casts one half of the participants in the role of the developer and the other half as users. This would provide further insight into the intuitiveness and veracity of behaviour labels in bottom-up agent design.

8 BIBLIOGRAPHY

Ackermann, E. (2001). “Piaget’s constructivism, Papert’s constructionism: What’s the difference”. Future of learning group publication, **5**(3): 438.

Aesop. (2003). Fables. In Ashliman, D.L. (Ed.) “Aesop’s Fables”. Barnes & Noble Classics, New York.

Altheide, D.L. (1998). “Qualitative Media Analysis”. Thousand Oaks, California, Sage.

Arbib, M. A. (1987). “Brains, Machines, and Mathematics”. Springer-Verlag, New York, NY, USA.

Arbib, M. A. (2003). “Rana computatrix to human language: towards a computational neuroethology of language evolution”. Philosophical Transactions: Mathematical, Physical and Engineering Sciences, **361**(1811):2345-2379.

Arbib, M.A. (2010). “From Mirror Writing to Mirror Neurons”. From Animals to Animats 11: 11th International conference on Simulation of Adaptive Behaviour. Paris, Clos Luce, France. Springer. **6226**: 1-12

Arkane Studios SA. (2012) “Dishonored” [Software]. PS3, Windows PC, Xbox360. Bethesda Softworks LLC. Information available from http://www.mobygames.com/game/dishonored [Accessed: 24.06.14]

Autodesk. (2008). “Ubisoft Assassin's Creed: Autodesk 3D Pipeline Helps Bring Centuries-Old Tale Into the Modern Age”. Available from: http://images.autodesk.com/adsk/files/cust_success_assassin_s_creed_v3.pdf [Accessed 20.03.09]

Autodesk. (2010). “Autodesk in Games”. Available from: http://images.autodesk.com/adsk/files/fy11_games_brochure_us.pdf [Accessed 03.10.11]

Bates, J. (1997). “The Role of Emotion in Believable Agents.” Communications of the ACM **37**(7): 122-125.

Beam, W. (2008). “Mark Tilden's Three Laws of BEAM Robotics”. Available from: http://www.beam-wiki.org/wiki/Three_laws [Accessed 23.05.09]

Beer, R. D. (1990) “Intelligence as adaptive Behaviour: an experiment in computational Neuroethology”. Academic Press Professional, Inc., San Diego, CA.

Beer, R.D. (1995). “On the dynamics of small continuous-time recurrent neural networks”. Adaptive Behaviour **3**(4):469-509.

Beer, R.D. (1995b). “Computational and dynamical languages for autonomous agents”. in R.F. Port, T. van Gelder (eds), Mind as Motion: Explorations in the Dynamics of Cognition. Cambridge, MA. MIT Press.

Beer, R.D. (1997). “The Dynamics of Adaptive Behaviour: A Research Program”. Robotics and Autonomous Systems **20**(2-4):257-289.

Benedek, J., & Miner, T. (2002). Measuring Desirability: New methods for evaluating desirability in a usability lab setting. Proceedings of Usability Professionals Association, 2003, 8-12.

Berning, S., Willig K., Steffens, H., Dibaj, P., Hell S. (2012) “Nanoscopy in a Living Mouse Brain” Science **335**(6068):551.

Bernardet, U., Verschure, P., Blanchard, M., Wyss, R., Lechon, M., Sanchez-Palencia, J., (2010). “iqr: Simulator for Large Scale Neural Systems”, University of Pompeau Fabra, Barcelona. [Software], Linux/Windows PC. Available from: http://iqr.sourceforge.net/ [Accessed 15.09.2010]

Bernardet, U., Verschure, P. (2010). “iqr: A Tool for the Construction of Multi-level Simulations of Brain and Behaviour.” Neuroinformatics **8**: 113-134.

Bernhaupt, R., Eckschlager, M., & Tscheligi, M. (2007, June). Methods for evaluating games: how to measure usability and user experience in games?. In Proceedings of the international conference on Advances in computer entertainment technology. ACM: 309-310

Bi, G.-q. and M.-m. Poo (1998). “Synaptic Modifications in Cultured Hippocampal Neurons: Dependence on Spike Timing, Synaptic Strength, and Postsynaptic Cell Type.” Journal of Neuroscience **18**(24):10464-10472.

Blumberg, B. (1994). “Action-Selection in Hamsterdam: Lessons from Ethology”. SAB94 Proceedings of the 3rd International Conference on the Simulation of Adaptive Behavior, MIT Press, Cambridge, MA, USA. 108-117

Blumberg, B. (1997). “Old tricks, new dogs : Ethology and interactive creatures”. Thesis, (PhD). Massachusetts Institute of Technology. Dept. of Architecture. Program in Media Arts and Sciences.

Boden, M. (2006). “Mind as Machine: A History of Cognitive Science”. Clarendon Press, Oxford, United Kingdom

Bongard, J. (2003). “Incremental Approaches to the Combined Evolution of a Robot's Body and Brain”. Thesis, (PhD). University of Zurich, Switzerland.

Braitenberg, V. (1965). “Taxis, Kenesis and Decussation.” Progress in Brain Research **17**: 210-222.

Braitenberg, V. (1984). “Vehicles: Experiments in Synthetic Psychology”, The MIT Press. Cambridge, MA, USA

Browning, B., Fanaswala, I., Forlizzi, J., Kirby, R., Kyung Lee, M., Makatchev, M. et al. (2011). “Believable Robot Characters”. AI Magazine (Winter 2011). 32 (4):39.

Brooks, R. A. (1986). “A Robust Layered Control System For a Mobile Robot”, IEEE Journal of Robotics and Automation **2**(1):14-23

Brooks, R. A. (1991). “Intelligence without representation”. Artificial Intelligence **47:** 139-159.

Brooks, R.A. (1997). “From Earwigs to Humans”. Robotics and Autonomous Systems 20: 291-304.

Bryman, A. (2008). “Social Research Methods”. 3rd Ed., Oxford: Oxford University Press

Bubic, A., von Cramon, D. Y., Schubotz, R. (2010). “Prediction, cognition and the brain.” Frontiers in Human Neuroscience **4**(25)

Bullfrog Productions (1994). “Theme Park” [Software]. 3DO, Amiga, Amiga CD32, DOS, FM Towns, Genesis, Jaguar, Macintosh, Nintendo DS, PC-98, PlayStation, PlayStation 3, PSP, PS Vita, SEGA CD, SEGA Saturn, SNES, Windows. Electronic Arts Inc., Information available from: http://www.mobygames.com/game/theme-park [Accessed 20.10.15]

Bungie Studios. (2004). “Halo 2” [Software]. Xbox, PC. Microsoft Game Studios. Information available from: http://www.mobygames.com/game/xbox360/halo-3 [Accessed 13.09.15]

Bungie Studios. (2007). “Halo 3” [Software]. Xbox 360. Microsoft Game Studios. Information available from: http://www.mobygames.com/game/xbox360/halo-3 [Accessed 13.02.12]

Burke, R. and B. Blumberg (2002). “Using an ethologically-inspired model to learn apparent temporal causality for planning in synthetic creatures”. AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems Part 1, ACM Press: 326-333

Campbell, J. (1993). “The Hero with a Thousand Faces”, Fontana Press.

Cavanagh, T. (2010). “VVVVVV” [Software], PC/Mobile/Nintendo 3DS/Ouya. Information available from: http://www.mobygames.com/game/super-meat-boy [Accessed 08.08.14]

Champandard, A. (2008). “NaturalMotion's euphoria Technology: The Honeymoon is Over”. Available from: http://aigamedev.com/open/editorial/naturalmotion-euphoria/ [Accessed 15.01.09]

Champandard, A. (2008). "Top 5 Trends and Predictions for Game AI in 2008". Available from: http://aigamedev.com/open/article/predictions-game-ai-2008/ [Accessed 23.12.08]

Chang, S.M. & Sung, H.C. (2013). "The effectiveness of Paro robot therapy on mood of older adults: a systematic review". International Journal of Evidence-Based Healthcare. **11**(3): 216

Chen, J. (2007). “Flow in games (and everything else)”. Communications of the ACM, 50(4): 31-34.

Cliff, D. and G. Miller. (1995). “Co-evolution of Pursuit and Evasion II: Simulation Methods and Results”. Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behaviour (SAB96). MIT Press Bradford Books : 506-515

Coleridge, S. T. (1817). “Biographia Literaria”. Project Gutenberg EBook. Available from: http://www.gutenberg.org/dirs/etext04/bioli10.txt [Accessed 12.02.2009]

Costa, P., R. McCrae, et al. (1991). “Facet scales for agreeableness and conscientiousness: A revision of the NEO personality inventory.” Personality and Individual Differences **12**(9): 887-898

Creative Assembly (2009). “Empire: Total War”. [Software], Windows PC. Information available from: http://www.mobygames.com/game/windows/empire-total-war [Accessed 03.06.09]

Cushing, J., S. Luland, et al. (2010). “Nottingham University MSc Project List - Braitenberg Vehicles Simulator” Available from: http://www.cs.nott.ac.uk/~nhn/G52GRP/G52GRP-2010-2011-Projects.html#gp10-bsl [Accessed 02.11.10]

Daman, E. (2008). “Collaborative Robot Agents”. Thesis, (MSc), Available from: http://www.kbs.twi.tudelft.nl/docs/MSc/2008/Daman/thesis.pdf [Accessed 23.01.09]

Delphine Software International (1991). “Another World”. [Software], AMIGA PC. Interplay Productions Inc., USA. Information available from: http://www.mobygames.com/game/out-of-this-world [Accessed 01.12.10]

Delphine Software International (1992). “Flashback: The Quest for Identity”. [Software], AMIGA PC. U.S. Gold Ltd., Birmingham, UK. Information available from: http://www.mobygames.com/game/flashback-the-quest-for-identity. [Accessed 01.12.2010]

Dreamworks (2010). “How to Train Your Dragon”. [Film]. Directed by DeBlois D. and Sanders C. Dreamworks Animation, Glendale, California, CA, USA

Egri, L. (2004). “The Art of Dramatic Writing: Its Basis in the Creative Interpretation of Human Motives”. Simon & Schuster.

Ekman, P. and E. Rosenberg (1998). “What the Face Reveals: Basic and Applied Studies of Spontaneous Expression Using the Facial Action Coding System (FACS)”. Series in Affective Science, Oxford University Press, USA.

Eldawy, M. (2006). “Trends of Character Animation in Games”. Report for Academic ADL Co-Lab August 2006

El-Nasr, M. S., Drachen, A., & Canossa, A. (2013). “Game analytics: Maximizing the value of player data”. Springer.

ESA, Entertainment Software Association. (2014). “Essential Facts About the Computer and Video Game Industry: 2014 Sales, Demographic, and Usage Data”. ESA. Available from: http://www.theesa.com/facts/pdfs/esa_ef_2014.pdf [Accessed: 20.07.14]

Farmer, J. and Belin, A. (1992). “Artificial Life: The Coming Evolution.” In Langton, C. G. (Ed.), Artificial Life II. Addison-Wesley, Redwood City: 815-840

Floreano, D. and F. Mondada (1994). “Automatic creation of an autonomous agent: genetic evolution of a neural-network driven robot”. From Animals to Animats 3: Proceedings of the Third International Conference on Simulation of Adaptive Behaviour. Brighton, United Kingdom. MIT Press, Cambridge, MA, USA : 421-430

Floreano, D. and J. Urzelai (2000). “Evolutionary Robots with on- line self-organization and behavioral fitness”. Neural Networks **13**: 431-443.

Freud, S. (1919). “Das Unheimliche”. Zeitschrift für Anwendung der Psychoanalyse auf die Geistenswissenschaften **5**: 297-324

Fujita, M. (2001). “AIBO: Toward the Era of Digital Creatures”. The International Journal of Robotics Research **20**(10): 781-794.

Gerken, C. (1998). “NetWorks on Wheels.” Available from: http://www.mindspring.com/~gerken/vehicles/ [Accessed 04.11.09]

GIA and Ueda F. (2001). "Fumito Ueda Interview with GIA." Available from: http://www.psy-q.ch/mirrors/thegia/sites/www.thegia.com/features/f010831.html [Accessed 01.06.09]

Glaser, B.G. and Strauss, A.L.(1967), “The Discovery of Grounded Theory: Strategies for Qualitative Research”. Chicado, Aldine

Gomes, P.F., Martinho, C., Paiva, A. (2011). “I've been here before! location and appraisal in memory retrieval”. In Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems, IFAAMAS 2011, Taipei, Taiwan:1039-1046

Gomes, P., Paiva, A., Martinho, C., Jhala, A. (2013). "Metrics for Character Believability in Interactive Narrative". Proceedings on the International Conference on Interactive Digital Storytelling. Lecture Notes in Computer Science. Springer. **8230**:223-228

Gorman, B., Thurau, C., Bauckhage, C., & Humphrys, M. (2006). "Believability testing and bayesian imitation in interactive computer games". In From Animals to Animats 9 (pp. 655-666). Springer Berlin Heidelberg.

Gorniak, P. and B. Blumberg (2005). "Sequence Learning by Backward Chaining in Synthetic Characters." AIIDE 2005: 51-56.

Grand, S., Cliff, D., Malhotra, A. (1997). "Creatures: Artificial Life Autonomous Software Agents for Home Entertainment". Proceedings of the First International Conference on Autonomous Agents (Agents '97), ACM Press: 22-29

Grand, S. and Cyberlife Technology. (1997). "Creatures for Windows 3.x." [Software], Windows PC. Information available from: http://www.mobygames.com/game/win3x/creatures [Accessed 02.04.08]

Guerrilla B. V. (2009). "Killzone 2" [Software]. PS3. Sony Computer Entertainment America. Information available from: http://www.mobygames.com/game/ps3/killzone-2 [Accessed 15.09.15]

Hernandez, P. (2014). "Maye we can't handle smart enemies in our games". Available from: http://kotaku.com/maybe-we-cant-handle-smart-enemies-in-our-games-1542300804 [Accessed 20.04.14]

Harvery, I., Husband, P., Thompson, A., and Jakobi, N. (1997). "Evolutionary Robotics: The Sussex Approach". Robotics and Automation Systems. 20:205-224

Hardwidge, B. (2009). "How AI in Games Works." Available from: http://www.bit-tech.net/gaming/2009/03/05/how-ai-in-games-works/1 [Accessed 03.06.09]

Harrington, K., Freeman, J. and Pollack J. (2014) "Coevolution in Hide and Seek: Camouflage and Vision." Fourteenth Conference on Simulation and Synthesis of Life, New York NY: 25-32

Harvey, I., P. Husbands, D.Cliff (1994). "Seeing the light: artificial evolution, real vision". From Animals to Animats 3: Proceedings of the Third International Conference on Simulation of Adaptive Behaviour. Brighton, United Kingdom. MIT Press, Cambridge, MA, USA : 392-401

Hassabis, D. and Lionhead Studios (2001). "Black & White" [Software], Windows PC, Macintosh. Electronic Arts Inc., Information available from: http://www.mobygames.com/game/black-white [Accessed 08.02.15]

Hebb, D. O. (1949). "The Organization of Behavior", Wiley and Son.

Herold, C. (2001). “Game Theory: When a Tiny Taut Gesture Upstages Demons and Noise”. Available from http://www.nytimes.com/2001/10/18/technology/game-theory-when-a-tiny-taut-gesture-upstages-demons-and-noise.html [Accessed 05.09.08]

Hirose, M. and T. Takenaka (2001). “Development of Humanoid Robot ASIMO.” Honda R&D Tech Rev **13**(1): 1-6

Hirose, M. and Ogawa, K. (2006). “Honda humanoid robots development”. Philosophical Transactions of the Royal Society **365**(1850): 11-19

Hogg, D., Martin F., Resnick, M. (1991). “Braitenberg creatures”. Techreport, MIT Media Laboratory **2002**(13)

Hopfield, John J. (1982) “Neural Networks and Physical Systems with Emergent Collective Computational Abilities”. Proceedings of the National Academy of Sciences **79**(8): 2554-2558

Isbister, K. (2006). “Better game characters by design: a psychological approach”, Morgan Kaufmann, San Francisco.

Isla, D. and Blumberg, B., (2002). “Object Persistence for Synthetic Creatures”, Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS): Part 3, ACM, New York, NY, USA: 1356-1363

Isla, D., Burke, R., Downie, M., Blumberg, B., (2001). “A Layered Brain Architecture for Synthetic Creatures”. IJCAI’01 Proceedings of the 17th international joint conference on Artificial Intelligence, Volume 2, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA: 1051-1058

Isla, D., (2005) “GDC 2005 Proceeding: Handling Complexity in the HALO 2 AI”. Proceedings of the International Game Developers Conference. Available From: http://www.gamasutra.com/view/feature/130663/gdc_2005_proceeding_handling_.php [Accessed: 15.11.15]

Jentsch, E. (1997). “On the psychology of the uncanny (1906).” Angelaki: Journal of Theoretical Humanities **2**(1): 7-16

jMonkey-Community. (2010). “jMonkey Engine”. [Software], Java. Available from http://www.jmonkeyengine.com/ [Accessed 02.03.09]

Jones, C. (1989). “Chuck Amuck: The Life and Times of an Animated Cartoonist”. Farrar, Straus and Giroux.

Kim, J. H., Gunn, D. V., Schuh, E., Phillips, B., Pagulayan, R. J., & Wixon, D. (2008, April). Tracking real-time user experience (TRUE): a comprehensive instrumentation solution for complex systems. In Proceedings of the SIGCHI conference on Human Factors in Computing Systems. ACM: 443-452

Klein, J. (2002). “BREVE: a 3D Environment for the Simulation of Decentralized Systems and Artificial Life”. Proceedings of Artificial Life VIII, the 8th International Conference on the Simulation and Synthesis of Living Systems, The MIT Press, Cambridge, MA, USA: 329-334

Koball, B. R. and J. H. Wharton (1988). “A test vehicle for Braitenberg control structures”. Compcon Spring '88: Thirty-Third IEEE Computer Society International Conference, Digest of Papers: 394-398

Kohler, E. (1999). "A Braitenberg vehicle simulator with pseudo-physics". Available from http://www.lcdf.org/xbraitenberg/proj.pdf [Accessed 03.04.09]

Koike, A. and T. Koshiishi (2001). “Styling Design of Humanoid Robot ASIMO”. Honda R&D Tech Rev **13**(1): 7-16

Kojima Productions (2008). “Metal Gear Solid 4: Guns of the Patriots” [Software]. PS3, Konami Digital Entertainment. Information available from: http://www.mobygames.com/game/ps3/metal-gear-solid-4-guns-of-the-patriots [Accessed 25.05.13]

Komosinski, M. and S. Ulatowski. (1998a). “Framsticks”. [Software], Windows PC. Available from: http://www.framsticks.com/a/doc_opengl.html. [Accessed 20.04.09]

Komosinski, M. and S. Ulatowski. (1998b). “Framsticks”. ECML 98 Demonstration and Poster Papers. Chemnitzer Informatik-Berichte, Chemnitz: 7-9

Koster, R. (2013). “Theory of fun for game design”. O'Reilly Media, Inc.

Kowall, C. (2005). “Braitenberg simulations as vehicles of evolution”. GECCO '05: Proceedings of the 2005 workshops on Genetic and evolutionary computation, ACM, New York, NY, USA: 398-400

Kuniavsky, M. (2003). “Observing the User Experience: A Practitioner’s Guide to User Research”. Morgan Kaufmann Publishers.

Lafave, S. (2000). “Notes on Braitenberg Vehicles.” Available from: http://instruct.westvalley.edu/lafave/Vehicles_online.html [Accessed 12.05.09]

Laird, J. E. (2002). Research in human-level AI using computer games. Communications of the ACM, **45**(1): 32-35.

Lambrinos, D. and Scheier C. (1995). “Extended Braitenberg architectures”. AILab Technical Report. Computer Science Department, University of Zurich, Zurich, Switzerland. **95**(10):1-10

Lambrinos, D. (1999). "Braitenberg Vehicles". Available from: http://www.ifi.uzh.ch/ailab/people/lambri/mitbook/braitenberg/braitenberg.html [Accessed 20.03.09]

Langton, C. (1990). "Artificial Life". Artificial Life: Proceedings of the Santa Fe Institute Studies in the Sciences of Complexity Volume 6, Addison Wesley.

Lasseter, J. (1987). "Principles of Traditional Animation Applied to 3D Computer Animation." SIGGRAPH '87: Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques. ACM, New York, NY, USA:35-44

Lee, J., Takehashi, H., Nagai, C., Obinata, G. & Stefanov, D. (2012). "Which Robot Features Can Stimulate Better Responses from Children with Autism in Robot-Assisted Therapy?". International Journal of Advanced Robotic Systems. **9**

Lester, J.C. and Stone, B.A.. (1997)."Increasing believability in animated pedagogical agents". In Proceedings of the Frst international conference on Autonomous agents. ACM, 1997: 16-21

Leyzberg, D., Avrunin, E., Liu, J., & Scassellati, B. (2011, March). „Robots that express emotion elicit better human teaching". In Human-Robot Interaction (HRI), 2011 6th ACM/IEEE International Conference on. IEEE: 347-354

Livingstone, D. (2006). "Turing's test and believable AI in games". Computers in Entertainment (CIE) January 2006, **4**(1)

Loyall, B. (1997). "Believable Agents: Building Interactive Personalities." Thesis (PhD). Carnegie Mellon University, Pittsburgh, PA

Lucic, V., A. H. Kossel, et al. (2007). "Multiscale imaging of neurons grown in culture: from light microscopy to cryo-electron tomography." Journal of Structural Biology **160**: 146-156.

MacDorman, K. F., Green, R. D., Ho, C.-C., & Koch, C. (2009). "Too real for comfort: Uncanny responses to computer generated faces". Computers in Human Behavior, **25**(3): 695–710

Maes, P. (1995). "Artificial life meets entertainment: lifelike autonomous agents." Communications of the ACM **38**(11): 108-114

Maes, P., Darrell, T., Blumber, B., Pentland, A., (1995). "The ALIVE System: Wireless, Full-body Interaction with Autonomous Agents". Multimedia Systems - Special issue on multimedia and multisensory virtual worlds **5**(2): 105-112

Magerko, B., Laird, J., Assanie, M., Kerfoot, A., & Stokes, D. (2004). AI characters and directors for interactive computer games. Ann Arbor, **1001**(48): 109-2110.

Mateas, M. (1999). “An Oz-Centric Review of Interactive Drama and Believable Agents”. Artificial Intelligence Today. Springer Verlag, Berlin, Heidelberg, Germany: 297-328

Mattiussi, C., Marbach, D., Dürr, P., Floreano, D. (2008). “The Age of Analog Networks – Infoscience”. AI Magazine **29**(3): 63-76

McAllister, G. (2011). “Uncovering player experience: Utilising Biometrics data to Reveal Latent Playability Issues”. In: Under the Mask.

McCulloch, W. and Pitts, W. (1943). “A logical calculus of the ideas immanent in nervous activity”. Bulletin of Mathematical Biology **5**(4): 115-133

McMillen, E. and Refenes, T., (2010). “Super Meat Boy” [Software], XBOX 360/PS3/PC. Team Meat. Information available from: http://www.mobygames.com/game/super-meat-boy [Accessed 08.08.14]

Mechner, J. (1989). “Prince of Persia”. [Software], AMIGA PC. Broderbund Software, California, USA. Info Available from: http://www.mobygames.com/game/prince-of-persia [Accessed 01.12.10]

Meier, S. and M. P. S. Labs. (1991). "Sid Meier's Civilization". [Software], Windows PC. MicroProse Software Inc. Info available from: http://www.mobygames.com/game/dos/sid-meiers-civilization [Accessed 02.01.09]

Microsoft (1989). “Solitaire” [Software]. Windows PC. Microsoft. Information available from: http://www.mobygames.com/game/win3x/microsoft-windows-31-included-games [Accessed: 12.10.13]

Microsoft. (1990). “Minesweeper” [Software]. Windows PC. Microsoft. Information available from: http://www.mobygames.com/game/minesweeper [Accessed: 12.10.13]

Microsoft (2008) “Microsoft Robotics Developer Studio 2008”. Available from: http://www.microsoft.com/robotics/ [Accessed 01.02.2009]

Microsoft Research, Cambridge UK. (2013). "Drivatar". [Software]. Information available from: http://research.microsoft.com/en-us/projects/drivatar/ [Accessed 20.05.14]

Mirza-Babaei, P., Nacke, L., Fitzpatrick, G.A., White, G., McAllister, G. and Collins, N. (2012) “Biometric storyboards: visualising game user research data”. In: CHI EA '12 Proceedings of the 2012 ACM annual conference extended abstracts on Human Factors in Computing Systems Extended Abstracts. ACM, New York, NY, USA: 2315-2320.

Mobus, G. and Fisher, P. (1994). “MAVRIC's Brain”. IEA/AIE '94: Proceedings of the 7th international conference on Industrial and engineering applications of artificial intelligence and expert systems, Austin, Texas, United States, Gordon and Breach Science Publishers, Inc. : 15-320

Mobus, G. (1994). “Toward a Theory of Learning and Representing Causal Inferences in Neural Networks”. In Levine, D. S., Aparicio M. (Eds.) Neural Networks for Knowledge Representation and Inference. Lawrence Erlbaum Associates, Hillsdale, New Jersey: 339 - 374

Molyneux, P. and L. Bullfrog Productions. (1989). “Populous" [Software], Windows PC. Electronic Arts. Information available from: http://www.mobygames.com/game/dos/populous [Accessed 02.01.09]

Monolith Productions. (2005). “F.E.A.R.: First Encounter Assault Recon” [Software], Windows PC, Playstation 3, Xbox 360. Vivendi Universal Games.

Mori, M. (1970). “Bukimi no tani [The uncanny valley].” Energy **7**(4): 33-35

Mossmouth. (2009). “Spelunky” [Software]. Windows PC, Mac. Information available from: http://www.mobygames.com/game/spelunky [Accessed: 11.08.14]

Nacke, L., Lindley, C., & Stellmach, S. (2008). Log who’s playing: psychophysiological game analysis made easy through event logging. In Fun and games. Springer Berlin Heidelberg: 150-157

Namco (1980), “Pac-Man” [Software], Android/Apple II/Arcade/Atari 2600/Atari 5200/Atari 8-bit/BlackBerry/Channel F/Commodore 64/FM-7/Game Boy/Game Boy Advance/Game Gear/Intellivision/iPhone/MSX/Neo Geo Pocket Color/NES/Nintendo 3DS/Palm OS/PC-88/PC-98/PC Booter/Sharp X1/TI-99(4A)/VIC-20/Wii/Wii U/Windows Phone/Xbox 360/ZX Spectrum. NAMCO Ltd. Information available from: http://www.mobygames.com/game/pac-man [Accessed 08.08.14]

Nareyek, A. (2004). “AI in computer games”. Queue, 1(10), 58.

Nielsen, J., Mack, R. (1994). “Usability Inspection Methods”, John Wiley & Sons. New York, NY, USA

Nielsen, J.(2011). “How to Conduct a Heuristic Evaluation” [Online]. Available: http://www.useit.com/papers/heuristic/heuristic_evaluation.html [Accessed 01.05.13].

Naturalmotion. (2005). “Natural Motion”. Available from: http://www.naturalmotion.com/products.htm [Accessed 22.05.09]

Nolfi, S. and Floreano, D. (2000). “Evolutionary Robotics: The Biology, Intelligence and Technology of Self-Organizing Machines”. MA: MIT Press/Bradford Books.

Oppenheimer, D. (2012) "Artificially Intelligent Game Bots Pass the Turing Test on Turing's Centenary". Austin, Texas, USA. Available from: http://www.utexas.edu/news/2012/09/26/artificially-intelligent-game-bots-pass-the-turing-test-on-turing's-centenary/ [Accessed: 12.02.13]

Ortony, A., Collins A.,Clore G.L., (1988). "The cognitive structure of emotions". Cambridge University Press, Cambridge, UK

Ortony, A. (2003). "On making believable emotional agents believable". In R. Trappl, P. Petta & S. Payr (Eds.), Emotions in humans and artifacts. Cambridge, MA.: MIT Press.

Osfield, R. P. M., Pinson, C., Platings, M. (2010). "Open Scene Graph". [Software], PC/Mac. Available from: http://www.openscenegraph.org/ [Accessed 02.04.10]

Pajitnov, A. (1987). "Tetris" [Software]. Amiga, Amstrad CPC, Amstrad PCW, Apple II, Apple IIgs, Atari ST, BBC Micro, Commodore 64, DOS, Electron, Macintosh, MSX, TRS-80 CoCo, ZX Spectrum. Spectrum Holobyte. (1988). FM-7, NES, PC-88, PC-98, Sharp X1, Sharp X68000. Bullet-Proof Software. Arcade, Genesis/MegaDrive. SEGA Enterprizes. Arcade, NES. Tengen. (1989) NES, GameBoy. Nintendo. (1991). Dragon 32/64. Pulser Software. (1992). CD-i. Philips Interactive Media. (1995). Virtual Boy. Nintendo. (2002). Wonderswan Color. Vanguard Works. (2006). VIC-20. (2007). ZX81. (2010). Android, iPad, iPhone. EA Mobile. (2011). Nintendo 3DS. Nintendo. Information available from: http://www.mobygames.com/game/tetris [Accessed: 12.10.13]

Papert, S. (1980). "Mindstorms: Children, Computers, And Powerful Ideas", Basic Books, New York, NY, USA

Pavlov, I. P. (1903) "The Experimental Psychology and Psychopathology of Animals" From Nobel Lectures, Physiology or Medicine 1901-1921, Elsevier Publishing Company, Amsterdam, 1967

Perlin, K., & Goldberg, A. (1996). "Improv: A system for scripting interactive actors in virtual worlds". In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques. ACM: 205-216.

Perlin, K. (2009). "Actor Machine". Available from: http://www.actormachine.com/manifesto.html [Accessed 03.08.09]

Peschke, A. (2013). "Absichtlich Dumm". In: Gamestar 11/2013. IDG Verlag, Munich: 106 – 109.

Piaget, J. (1929). "The Child's Conception of the World", Harcourt, Brace and Co.

Pixar (2003). “Finding Nemo”. [Film]. Directed by Stanton A. and Unkrich L.. Pixar Animation Studios, California, CA, USA

Pixar (2006). “Wall-E, Pixar”. [Film]. Directed by Stanton A.. Pixar Animation Studios, California, CA, USA

Pocketwatch Games (2013) “Monaco: What’s Yours is Mine”. Information available from: http://www.mobygames.com/game/monaco-whats-yours-is-mine [Accessed: 12.11.15]

Pollick, F. (2009). “In Search of the Uncanny Valley”. The Vienna Series in Theoretical Biology, MIT Press.

Purchase, H.C. (2012). “Experimental Human-Computer Interaction: A Practical Guide with Visual Examples”. Cambridge University Press. New York, NY

Quinton, J.-C. (2009). “Coordination implicite d'interactions sensorimotrices comme fondement de la cognition”. Thesis (PhD), Institut de Recherche en Informatique de Toulouse (IRIT).

Quinton, J.-C. (2010). “Exploring and Optimizing Dynamic Neural Fields Parameters Using Genetic Algorithms”. IEEE World Congress on Computational Intelligence 2010, Barcelona, Spain.

Rabin, S. (2006). “AI Game Programming Wisdom 3”, Charles River Media.

Rabin, S. (2008). “AI Game Programming Wisdom 4”, Charles River Media.

Rañó, I., & Smithers, T. (2005). Obstacle avoidance through Braitenberg’s aggression behaviour and motor fusion. In Proc. of the 2nd European conf. on mobile robots: 98-103.

Salumae, T., Ranó, I., Akanyeti, O., & Kruusmaa, M. (2012, May). Against the flow: A Braitenberg controller for a fish robot. In Robotics and Automation (ICRA), 2012 IEEE International Conference on (pp. 4210-4215). IEEE.

Rañó, I. (2014a). “Biologically inspired navigation primitives”. In: Robotics and Autonomous Systems. **62**(10):1361-1370

Rañó, I. (2014b)."Results on the convergence of Braitenberg vehicle 3a". Artificial Life **20**(2): 223-235.

Rapid-I. (2010). “Rapidmine”. [Software], PC/Mac. Available from: http://rapid-i.com/[Accessed 03.09.10]

Regelous, S. (2009). “Massive Software”. Available from: http://www.massivesoftware.com/stephen-regelous/ [Accessed 01.02.09]

Reynolds, C. (1987). “Flocks, herds and schools: A distributed behavioral model.” SIGGRAPH Computer Graphics **21**(4): 25-34

Reynolds, C. W. (1999, March). Steering behaviors for autonomous characters. In Game developers conference. **1999**: 763-782.

Rockstar Games (2008). “Grand Theft Auto IV”. [Software], XBOX360/PS3/PC. Information available from: http://www.mobygames.com/game/grand-theft-auto-iv [Accessed 22.11.08]

Rollings, Andrew and Dave Morris (2000), “Game Architecture and Design”. Scottsdale, Arizona: Coriolis

Ronald, E. M. A., Sipper, M., Capcarrère, M. S. (1999a). “Design, observation, surprise! A test of emergence.” Artificial Life **5**(3): 225-239

Ronald, E. M. A., Sipper, M., Capcarrère, M. S. (1999b). “Testing for Emergence in Artificial Life”. ECAL '99: Proceedings of the 5th European Conference on Advances in Artificial Life, Springer-Verlag.

Ronald, E. M. A. and Sipper M. (2001). “Surprise versus unsurprise: Implications of emergence in robotics.” Robotics and Autonomous Systems **37**(1): 19-24

Rossignol, J. (2012). "Custom UT Bots Pass 'Turing Test', Win BotPrize". Available from: http://www.rockpapershotgun.com/2012/09/28/custom-ut-bots-pass-turing-test-win-botprize/ [Accessed: 12.02.2013]

Rundle, M. (2012). "BotPrize 2012: Programmers 'Beat Turing Test' By Making Video Game Bots Seem 'More Human Than Human". Available from: http://www.huffingtonpost.co.uk/2012/10/01/botprize-2012-programmers_n_1928349.html [Accessed: 12.02.2013]

Sakagami, Y., R. Watanabe, et al. (2002). “The intelligent ASIMO: system overview and integration”. Intelligent Robots and System, 2002. IEEE/RSJ International Conference on Intelligent Robots and Systems. **3**: 2478 - 2483

Sakaguchi, H. and M. Sakakibara. (2001). “Final Fantasy: The Spirits Within.” [Film]. Directed by Sakaguchi, H. and M. Sakakibara. Square Company. Information available from: http://www.imdb.com/title/tt0173840/ [Accessed 11.03.09]

Salumae, T., Ranó, I., Akanyeti, O., & Kruusmaa, M. (2012). “Against the flow: A Braitenberg controller for a fish robot”. In Robotics and Automation (ICRA), 2012 IEEE International Conference on Robotics and Automation, RiverCentre, Saint Paul, Minnesota, USA: 4210-4215.

Sergi, B., Pyk, P., Verschure, P. F. M. J. (2007). “A fly-locust based neuronal control system applied to an unmanned aerial vehicle: the invertebrate neuronal principles for course stabilization, altitude control and collision avoidance.” International Journal of Robotics Research **26**(7): 759-772

Seth, A. (1998). “Evolving Action Selection and Selective Attention Without Actions, Attention, or Selection”. From Animals to Animats 5: Proceedings of the Fifth International Conference on Simulation of Adaptive Behaviour. Brighton, United Kingdom. MIT Press, Cambridge, MA, USA: 139-146

Sevan, F. and Pollack, J. (2007). "Evolutionary Dynamics of Finite Populations in Games with Polymorphic Fitness-Equilibria." Journal of Theoretical Biology **247**(3): 426-441

Shao, W. and Terzopoulos D. (2005). “Autonomous pedestrians”. SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation, ACM Press: 19-28

Sims, K. (1994). “Evolving 3D Morphology and Behavior by Competition” Artificial Life **1**(4): 353-372.

Sony Global (1999). “Press Release - Sony Launches Four-Legged Entertainment Robot AIBO Creates a New Market for Robot-Based Entertainment”. Available from: http://www.sony.net/SonyInfo/News/Press_Archive/199905/99-046/index.html [Accessed 01.08.09]

Sony Global (2005). "Q3 FY2005 Sony Group Earnings Announcement" Available from: http://www.sony.net/SonyInfo/IR/info/presen/05q3/qfhh7c000008adfe.html [Accessed 01.08.09]

Stanislavski, C. (1968). “Stanislavski's Legacy: A Collection of Comments on a Variety of Aspects of an Actor's Art and Life”, Routledge: A Theatre Arts Book.

Stanislavsky, C. and H. Popper (1961). “Creating a Role, Theatre Arts Books”. Routledge

Stern, A. and P. F. Magic. (1995). “Dogz: Your Computer Pet”. [Software], Windows PC. Information available from http://www.mobygames.com/game/win3x/dogz-your-computer-pet [Accessed 03.07.09]

Stern, D., and Candela, J.Q,n. (2009) “Playing machines: Machine learning applications in computer games”. In CIG’09: Proceedings of the 5th international conference on Computational Intelligence and Games. Piscataway, NJ, USA, 2009. IEEE Press:1–1

Strauss, A. and Corbin, J.M. (1998). “Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory”. Thousand Oaks, California. Sage Publishing.

Streeting, S. O. (2010). “OGRE 3D Engine”. [Software], PC/Mac. Available from: http://www.ogre3d.org/ [Accessed 20.01.11]

TCLWiki. (2007). “Braitenberg Vehicles Simulator in TCL.” Available from: http://wiki.tcl.tk/9047 [Accessed 12.04.10]

Terzopoulos, D., X. Tu, et al. (1994). "Artificial fishes: Autonomous locomotion, perception, behavior, and learning in a simulated physical world." Artificial Life **1**(4): 327-351.

Terzopoulos, D. and T. Rabie (1997). “Animat Vision: Active Vision in Artificial Animals”. Videre: Journal of Computer Vision Research **1**(1): 2-19

Terzopoulos, D. (1999). “Artificial life for computer graphics.” Communications of the ACM **42**(8): 32-42

Terzopoulos, D. (2002). “Artificial Life Cinematography.” YLEM **22**(12).

Thomas, F. and Johnston (1981). “Disney Animation - the illusion of life”, Abbeville Press.

Thompson, C. (2007). “Halo 3: How Microsoft Invented a new Science of Play”. In: Wired Magazine, Issue 15.09.

Thornton, C. (2004). “BUGART Experimental Robot Art.” Available from: http://www.informatics.sussex.ac.uk/users/christ/bugworks/bugartHome.html [Accessed 12.02.10]

Thurau, C., Bauckhage, C., & Sagerer, G. (2004). Learning human-like movement behavior for computer games. From animals to animats, 8: 315-323.

Tu, X. and D. Terzopoulos (1994). “Artificial fishes: physics, locomotion, perception, behaviour”. SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques, ACM Press: 43-50

Turn 10 Studios, Microsoft Studios. (2013). "Forza Motorsport 5" [Software] Xbox One. Information available from: http://www.mobygames.com/game/xbox-one/forza-motorsport-5 [Accessed 12.02.14]

Tychsen, A. (2008). “Crafting user experience via game metrics analysis". In Proceedings of the Workshop: Research Goals and Strategies for Studying User Experience and Emotion at the 5th Nordic Conference on Human-computer interaction: building bridges (NordiCHI), Lund, Sweden: 20-22.

Ubisoft Divertissements Inc. (2012). “Far Cry 3” [Software]. PS3, Windows PC, Xbox 360. Information available from http://mobygames.com/game/far-cry-3 [Accessed: 24.11.13]

Ubisoft Toronto. (2013). “Tom Clancy’s Splinter Cell: Blacklist”. PS3, WiiU, Windows PC, Xbox 360. Information available from: http://www.mobygames.com/game/tom-clancys-splinter-cell-blacklist [Accessed: 24.11.13]

Ueda, F. (2001). “ICO”. [Software], PlayStation2. Sony Computer Entertainment. Information available from: http://www.mobygames.com/game/ps2/ico. [Accessed 12.03.09]

Ueda, F. (2005). "Shadow of the Colossus". [Software], PlayStation2. Sony Computer Entertainment. Information available from: http://www.mobygames.com/game/shadow-of-the-colossus. [Accessed 12.03.09]

Valve (2004). "Half Life 2." [Software], Windows PC/Mac. Valve Software. Information available from: http://www.mobygames.com/game/windows/half-life-2 [Accessed 17.06.09]

Van J. Wedeen, Rosene, D. L., Wang, R., Dai G., Mortazavi, F., Hagmann, P., Kaas, J. H. and Tseng, W-Y I. (2012). “The Geometric Structure of the Brain Fiber Pathways”. Science 30 March 2012 **335** (6076): 1628-1634

Walter, W. G. (1953). “The Living Brain”. Duckworth, London

Wang, D., Subagdja, B., Tan, A. H., & Ng, G. W. (2009, July). “Creating human-like autonomous players in real-time first person shooter computer games”. In Proceedings, Twenty-First Annual Conference on Innovative Applications of Artificial Intelligence:173-178

Weizenbaum, J. (1966). “ELIZA---a computer program for the study of natural language communication between man and machine.” Commun. ACM **9**(1): 36-45.

White, G., McKay, L., Pollick, F. (2007). “Motion and the uncanny valley”. Journal of Vision **7**(9): 477-477

White, G. R., Mirza-Babaei, P., McAllister, G., & Good, J. (2011, May). “Weak inter-rater reliability in heuristic evaluation of video games”. In CHI'11 Extended Abstracts on Human Factors in Computing Systems. ACM: 1441-1446

Will, T. (1999). “Braitenberg Vehicle Simulator.” [Software], Windows PC. Available from: http://www.geocities.com/Colosseum/3141/Braitenberg.html. [Accessed 10.11.10]

Winograd, T. (1980). “What does it mean to understand language?” Cognitive Science: A Multidisciplinary Journal **4**(3): 209-241.

Wiseman, J. (1999). "Braitenberg Vehicles." Available from: http://www.cs.uchicago.edu/~wiseman/vehicles/ [Accessed 10.03.09]

Wright, W. (2008). “Spore: Galactic Adventures for Windows”. [Software], Windows PC. Electronic Arts. Information available from: http://www.mobygames.com/game/spore_ [Accessed 23.01.10]

Wright, W. and I. Maxis Software. (1989). "SimCity." [Software], Windows PC/Mac. Maxis Software. Information available from: http://www.mobygames.com/game/simcity [Accessed 22.03.09]

Wright, W. and I. Maxis Software. (1990). "SimEarth: The Living Planet for DOS". [Software], Windows PC/Mac. Maxis Software. Information available from: http://www.mobygames.com/game/dos/simearth-the-living-planet. [Accessed 02.03.09]

Wright, W. and I. Maxis Software (1991). "SimAnt: The Electronic Ant Colony" [Software], Windows PC/Mac. Ocean Software. Information available from: http://www.mobygames.com/game/simant-the-electronic-ant-colony [Accessed 02.03.09]

Wright, W. and I. Maxis Software. (1992). "SimLife". [Software], Windows PC/Mac. Maxis Software. Information available from: http://www.mobygames.com/game/win3x/simlife. [Accessed 02.03.09]

Wright, W. and I. Maxis Software. (2000). "The Sims" [Software], Windows PC/Mac/PS2/GameCube. Informartion available from http://www.mobygames.com/game/windows/sims [Accessed 22.03.09]

Wyeth, P. and H. Purchase (2001). "Exploring the learning potential of an artificial life simulation." International Journal of Continuing Engineering Education and Life Long Learning **11**(3): 229-241.

Yamauchi, B. M. and Beer, R. D. (1994). "Integrating Reactive, Sequential, and Learning Behaviour Using Dynamical Neural Network". In Cliff, D., Husbands, P., Meyer, J.-A., and Wilson, S. W., (Eds.), From Animals to Animats III: Proceedings of the 3rd International Conference on Simulation of Adaptive Behaviour: 382-391

Zeki, S. and L. Marini (1998). "Three cortical stages of colour processing in the human brain." Brain **121**(9): 1669-1685.

Zemeckis, R. (2004). "The Polar Express.". [Film]. Directed by Zemeckis R. Castle Rock Entertainment. Information available from: http://www.imdb.com/title/tt0433622/. [Accessed 11.01.09]

Zemeckis, R. (2007). "Beowulf". [Film]. Directed by Zemeckis, R. Paramount Pictures. Information available from: http://www.imdb.com/title/tt0442933/ [Accessed 11.01.09]

9 PUBLICATIONS RELATING TO THIS THESIS

Rosenkind, M., Winstanley G. and Blake, A. (2008). “Creating believable characters using a Synthetic Psychology approach”. Proceedings of CGAMES'2008, 13th international Conference on Computer Games: AI, Animation, Mobile, Educational and Serious Games. Hosted by Light House Media Centre, Wolverhampton, UK and organised by the School of Computing and Information Technology, University of Wolverhampton, 3-5 November, 2008:22-26. ISBN-13: 978-0954901660. Available from: http://eprints.brighton.ac.uk/6792/ [Accessed: 12.08.14]

Rosenkind, M., Winstanley, G. and Blake, A. (2012) “Adapting Bottom-up, Emergent Behaviour for Character-Based AI in Games”. M. Bramer, M. Petridis (Eds.). Proceedings of the 32nd SGAI International Conference on Artificial Intelligence. Peterhouse College, Cambridge, UK, 11-13th December, 2012: 333-346. ISBN 978-1-4471-4738-1, Springer. Available from: http://link.springer.com/chapter/10.1007/978-1-4471-4739-8_26 [Accessed: 12.08.14]

Rosenkind, M., Arnold, D. and Winstanley, G. (2013) “A temporal pattern predictor for virtual characters”. In: ECAL 2013: 12th European Conference on the Synthesis and Simulation of Living Systems, Taormina, Italy, 2-6 September, 2013. (P. Liò, O. Miglino, G. Nicosia, S. Nolfi and M. Pavone eds.) MIT Press: 79-85. ISBN 978-0-262-31709-2. Available from: http://mitpress.mit.edu/sites/default/files/titles/free_download/9780262317092_Advances_Artificial_Life_ECAL_2013.pdf [Accessed 12.08.14]

10 APPENDIX

The DVD attached to the back cover of this thesis contains the simulation environment developed for the experiments, including the modified JMonkey engine (which is distributed under the new BSD License), code samples and experimental data.

A list of the full contents and instructions on how to use these are included on the DVD in the text file "How to use this Disc".

10.1 EXTENSION PROTOTYPES: IMPLEMENTATIONS

Hunter & Prey (Adobe Flash ActionScript 3.0)

Hunter Code:

```
//LOVE INFLUENCE
loveLeftSpeed = _parent.leftTarget.getDistance(_parent._parent.heaty)*0.01;
loveRightSpeed= _parent.rightTarget.getDistance(_parent._parent.heaty)*0.01;

//FEAR INFLUENCE
fearLeftSpeed =Math.max(0,(1-(_parent.leftTarget.getDistance(_parent._parent.coldy)*0.01)));
fearRightSpeed=Math.max(0,(1-(_parent.rightTarget.getDistance(_parent._parent.coldy)*0.01)));

//HATES Prey
hateLeft= Math.max(0,(1-(_parent.rightTarget.getDistance(_parent._parent.prey)*0.01)));
hateRight=Math.max(0,(1-(_parent.leftTarget.getDistance(_parent._parent.prey)*0.01)));

//TOTAL BEHAVIOUR
leftSpeed = (loveLeftSpeed + fearLeftSpeed + hateLeft);
rightSpeed= (loveRightSpeed+ fearRightSpeed+ hateRight);

_parent.leftWheel.setVelocity(leftSpeed);
_parent.rightWheel.setVelocity(rightSpeed);
```

Prey Code:

```
//LOVE INFLUENCE
loveLeftSpeed = _parent.leftTarget.getDistance(_parent._parent.heaty)*0.01;
loveRightSpeed= _parent.rightTarget.getDistance(_parent._parent.heaty)*0.01;

//FEAR INFLUENCE
coldLeft = Math.max(0,(1-(_parent.leftTarget.getDistance(_parent._parent.coldy)*0.01)));
coldRight= Math.max(0,(1-(_parent.rightTarget.getDistance(_parent._parent.coldy)*0.01)));

//FEARS Hunter
hunterLeft = Math.max(0,(1-(_parent.leftTarget.getDistance(_parent._parent.hunter)*0.01)));
hunterRight= Math.max(0,(1-(_parent.rightTarget.getDistance(_parent._parent.hunter)*0.01)));

fearLeftSpeed = Math.max(0,(coldLeft + hunterLeft));
fearRightSpeed= Math.max(0,(coldRight+ hunterRight));

//TOTAL BEHAVIOUR
leftSpeed = (loveLeftSpeed + fearLeftSpeed);
rightSpeed= (loveRightSpeed+ fearRightSpeed);

//AVOID WALLS
LeftIR = _parent.L_IR.getOutput();
RightIR = _parent.R_IR.getOutput();

leftSpeed += ((100-LeftIR)/100);
```

```
rightSpeed += ((100-RightIR)/100);

_parent.leftWheel.setVelocity(leftSpeed);
_parent.rightWheel.setVelocity(rightSpeed);
```

Conversational Agents (Adobe Flash ActionScript 3.0)

Code for all Agents:

```
var vel_L = 1;
var vel_R = 1;

//Agressive
vel_L-=(ev.target.dist_L.measureHeatTargets([robot1,robot2,robot3,heat],100)/20);
vel_R-=(ev.target.dist_R.measureHeatTargets([robot1,robot2,robot3,heat],100)/20);

//Coward
vel_L+=(ev.target.dist_R.measureHeatTargets([robot1,robot2,robot3,heat],500))/350;
vel_R+=(ev.target.dist_L.measureHeatTargets([robot1,robot2,robot3,heat],500))/350;

ev.target.leftWheel.setVelocity(vel_L);
ev.target.rightWheel.setVelocity(vel_R);
ev.target.powerOn();
```

Capture the Flag (Adobe Flash ActionScript 3.0)

Blue Team Code (Both teams use the same code)

```
//Target is set by events, get output from sensors
lD = _parent.leftTarget.getOutput();
rD = _parent.rightTarget.getOutput();

//Set Starting Velocity for wheels
leftVelocity = 2;
rightVelocity = 2;

//Turn Towards Target
if(Math.abs(lD-rD) > 1){
    if(lD > rD){
        rightVelocity *= 0.9;
    }else if(rD > lD){
        leftVelocity *= 0.9;
    }
}
if(hasFlag == true){

    //Carry Flag to Goal
    _parent.rightTarget.setTarget(_parent._parent.blueGoal);
    _parent.leftTarget.setTarget(_parent._parent.blueGoal);
    leftVelocity /= 2;
    rightVelocity /= 2;
    fearLeft = 0;
    fearRight= 0;

    //Avoid Opponents
    for(opponent in _parent._parent.redTeam){
        fearLeft+=Math.max(0,
        (1-(_parent.leftTarget.getDistance(_parent._parent.redTeam[opponent])*0.01)));
        fearRight+=Math.max(0,
        (1-(_parent.rightTarget.getDistance(_parent._parent.redTeam[opponent])*0.01)));
    }
    leftVelocity +=fearLeft;
    rightVelocity+=fearRight;
} else{
```

```
    //Check if Team Member has the Flag
    chasingFlag = true;
    for(teamMember in _parent._parent.blueTeam){
        if(_parent._parent.blueTeam[teamMember].brain.hasFlag == true){
            chasingFlag = false;
        }
    }

    //If they don't, chase flag
    if(chasingFlag == true){
        _parent.rightTarget.setTarget(_parent._parent.flag);
        _parent.leftTarget.setTarget(_parent._parent.flag);

        //Also try to get to oponents goal to block them
        loveLeftSpeed = _parent.leftTarget.getDistance(_parent._parent.redGoal)*0.01;
        loveRightSpeed= _parent.rightTarget.getDistance(_parent._parent.redGoal)*0.01;
        leftVelocity +=loveLeftSpeed;
        rightVelocity+=loveRightSpeed;
        leftVelocity *= 0.7;
        rightVelocity *= 0.7;
    }else{

        //If they Do, block opponents
        if(_parent._parent.redTeam[myNumber] != undefined){
            _parent.rightTarget.setTarget(_parent._parent.redTeam[myNumber]);
            _parent.leftTarget.setTarget(_parent._parent.redTeam[myNumber]);
        }else{
            _parent.rightTarget.setTarget(_parent._parent.flag);
            _parent.leftTarget.setTarget(_parent._parent.flag);
        }
    }
}
```

Combat Prototype (Adobe Flash ActionScript 3.0)

Code for "Gun" agent (Same for "Bro" agent):

```
var vel_L = 0;
var vel_R = 0;
var shift = 0;

//Agressive toward target
var chase_L = (ev.target.dist_R.getHeat(targetGun, 500)/500);
var chase_R = (ev.target.dist_L.getHeat(targetGun, 500)/500);

//Love Target (a little)
vel_L -= (ev.target.dist_L.getHeat(targetGun, 200)/100);
vel_R -= (ev.target.dist_R.getHeat(targetGun, 200)/100);

//Coward and Defensive Collisions
var coll_L = (50-ev.target.ir_L.measure())/200;
var coll_R = (50-ev.target.ir_R.measure())/200;

//Fear Marker
vel_L += ev.target.dist_L.getHeat(mark2,100)/100;
vel_R += ev.target.dist_R.getHeat(mark2,100)/100;

//Final Differential Drive Signal
vel_L += chase_L+coll_L-coll_R;
vel_R += chase_R+coll_R-coll_L;

//Strafing Motor
shift+=(5-(ev.target.ir_L.measure()/10))-(5-(ev.target.ir_R.measure()/10));
shift-=(ev.target.dist_R.getHeat(shotGun,50)/10-ev.target.dist_L.getHeat(shotGun,50)/10);

//Apply Motor Signals
ev.target.shift_motor.setVelocity(shift);
ev.target.leftWheel.setVelocity(vel_L);
```

```
ev.target.rightWheel.setVelocity(vel_R);

//Collision Detection for bullet hits
if(ev.target.hitTestObject(shotBro)){
    energyGun -= 1;
}
if(ev.target.hitTestObject(baseGun)){
    energyGun += 1;
}

//State machine to switch targets
eGun.text = "Gun:"+energyGun;
if(energyGun < 50){
    targetGun = baseGun;
}else{
    targetGun = bro;
}
ev.target.powerOn();
```

3D Simulator Demo (Java / Java Monkey Engine 2.0)

Code for PINK agent

```
float leftDrive = 3;
float rightDrive= 3;
float distance = 5;

if(LIR > distance && RIR <= distance){
    //Turn left
    leftDrive -= 3;
    rightDrive+= 3;
}else if(LIR <= distance && RIR > distance){
    //Turn Right
    leftDrive += 3;
    rightDrive-= 3;
}else if(LIR <= distance && RIR <= distance){
    //Back up and Turn Right
    leftDrive = -30;
    rightDrive= -60;
}else if(LIR <= 2 && RIR <= 2){
    //Back away
    leftDrive = -20;
    rightDrive= -20;
}
```

Code for BLUE agent

```
//Initial Drive - Simple Cooperative
leftSpeed  = 0f;
rightSpeed = 0f;
int m = 10;

//Crossed-Excitatory: Hates RED
leftSpeed  += RR*m;
rightSpeed += RL*m;

//Crossed-Inhibitory: Disgusted by GREEN
leftSpeed  -= GR*m;
rightSpeed -= GL*m;

//HATES WALLS (even more)
float IRDistance = 5;        //Maximum Value of IRSensor
vehicle.default_infity = 99; //Use for IR that acts like a whisker
```

```
RIR = Math.max(0, (IRDistance-vehicle.rightIR));
LIR = Math.max(0, (IRDistance-vehicle.leftIR));

//Crossed
leftSpeed -= RIR*4;
rightSpeed -= LIR*4;

//Parallel
leftSpeed += LIR;
rightSpeed += RIR;

//Send to Motors
vehicle.drive.leftAxes.setDesiredVelocity(leftSpeed);
vehicle.drive.rightAxes.setDesiredVelocity(rightSpeed);
```

10.2 User Experiment: Refined Codes

Name	Sources	References
By Agent Types	7	805
Game 1 Binary	7	118
Game 2 Continuous	7	82
Game 3 Remapping	4	40
Game 4 Adaptive	4	68
Game Enemy Agent	6	53
Game Player Agent	6	22
HP All	6	36
HP Hunter	7	38
HP Prey	7	33
Shoot All	5	49
Shoot Blue	4	17
Shoot Red	5	21
Social All	7	43
Social Green	4	11
Social Red and Blue	5	12
Team All	7	39
Team Blue	6	20
Team Red	4	10
Virtual 3D All	7	29
Virtual 3D Blue	7	40
Virtual 3D Pink	6	24
Interactive Study	7	853
Gameplay Analysis	7	667
What Participants Did	6	46
Actions Inside the Game	1	6
Applying	0	0
Learning	1	4

Name	Sources	References
Switch Control Type	1	2
Actions Outside of Game	6	40
Display of Emotions	6	40
Annoyed	2	2
Apprehensive	1	2
Concentrated	1	4
Confusion	2	4
Despondent	1	1
Frustration	2	5
Panic	2	2
Positive	2	13
Tense	1	1
Tired	2	3
Unresponsive Participant	1	3
What Participants Said (Topics)	7	621
About themselves	5	12
Their performance	5	12
Say they want to do better	1	2
Think they will do badly	4	4
Agent	7	184
Adaption	3	12
Area Preference	1	1
Behaviour	7	70
1 Binary	3	8
2 Competitive	1	4
3 Remapping	3	6
4 Adaptive	0	0
Bugs Weird Behaviour	1	2
Emotions	4	14
FOV Visualization	2	4
Graphics	1	2
Memory	2	2
Movement	6	13
Personality	2	2
Predictability	4	7
Proximity to Player	2	3
Reaction to Coins	2	6
Reaction to Walls	2	2
Reactivity	4	10
Senses	6	25
Speed	3	3
Anthropromorphism	5	26
Address the Enemy Agent	2	9
Address the Player Agent	1	1
Speak for the Enemy Agent	5	12
Telling a story	1	4

Name	Sources	References
Comparison and contrast	6	41
Agents with other Agent	6	33
Games observed and other Games	2	5
Situation observed with Reality	1	2
This study to other Studies	1	1
Controls	6	48
Player Controls	6	33
Player Speed	2	4
Turning Speed	1	1
User Interface	4	10
Developer	3	5
Experiment	6	54
Participant's Evaluation Approach	5	21
Prior knowledge	1	1
Questionnaire	6	17
Finding the right words	1	3
Likert Scales	1	2
Options Presented	4	7
Question Applicability	3	4
Space for Answers	1	1
Testing Procedure	6	14
Gameplay	7	93
Coin Placement	1	1
Game Design	2	4
Number of opponents	1	1
Rule definition	6	24
Correct descriptions of rules	2	5
Incorrect description of rules	3	10
Reflect on Understanding the Rules	1	1
Unsure - Asking about rules	3	5
Scoring System	4	8
Strategies	6	51
JUST	7	133
Agent - Cant explain behaviour - it just did this	2	6
Agent - Comparison - BAD - just the same	3	3
Agent - Criticism - Limited - BAD - just does this	7	79
Agent - Focus - Emotion - GOOD - just wants this	4	12
Experiment - Expectation - Do I need to do this	3	4
Explain - Comparison - exactly like - Just like	2	3
Explain - Realise something COMPLEX is SIMPLE - So its just	2	6
Explain - Realise something SIMPLE is COMPLEX - I thought that its just	2	3
ONLY Synonym	4	13
Enough - just enough	1	1
Exclusive - Limited - just in this case	1	1
Limited - just does this	2	4
Not Enough - just this many	2	4

Name	Sources	References
Quantity - just a few	1	3
Player - Criticism - not enough - BAD - I am only	6	15
Player - Justify - FOCUS - I am just checking	5	16
ONLY	5	23
Agent - Specify - Single out - only this one	3	4
Conditional - State Change - Only IF, Only TO - it does this, only to do this	2	2
Disjunction - Apart from - its the only thing besides	1	1
Exclusive - Limited to - it is only in terms of	2	4
Quantity - Few - Not Enough - only one of these	4	9
Time - Exception - Only WHEN, Only AT time - only in this situation	3	3
Recommendations they made	2	2
Questionnaire	7	186
Agent Questions	7	116
1 Perceives the World	6	9
2 Thinking about	3	6
3 Personality	3	8
4 Draws Attention	4	7
5 Is Predictable	5	9
6 Behaves Coherent	2	5
7 Experience	6	15
8 Interacts Socially	6	11
Open Questions	6	45
o1 Not Seen by Agent	4	8
o2 Seen by Agent	5	17
o3 Collected Coins	5	19
Demographics	4	11
AI experience	1	1
Education	1	1
Favourite Games	1	1
Gaming Experience	3	6
Marital Status	1	1
Game Questions	7	56
Agent Easiest	3	3
Agent Most Difficult	5	12
Agent Most Fun	4	5
Agent Retried most	6	8
Easy to Control	4	4
Game Improvements	7	20
Controls	1	2
Multiple Agent at once	1	2
Game Objective Clear	4	4
Observation Study	7	309
1 Hunter Prey	7	54
Agent Behaviour	3	10
Agent Emotions	2	5
Agent Movement	2	2

Name	Sources	References
Agent Pushed out	3	5
Compare to Animals	1	1
Effect of Coldy Source	1	4
Effect of Heaty Source	1	1
Moving Sources	1	1
Naming the Agents	1	1
Who Hunts who	2	3
2 Conversation	7	46
Agent Appearance	1	2
Agent Movement	2	2
Agents Looking around	1	1
Compare to animals	1	1
Compare to games	1	1
Compare to non-animal	1	3
Group Behaviour	3	11
Individual Agent Behaviour	1	3
Interactivity	1	1
Reaction to Source	2	6
User Interface	1	1
3 Capture the Flag	7	57
Carrying Flag	1	2
Collision system	1	2
Compare Teams	1	1
Flag Behaviour	2	4
Interactivity	2	2
Movement	2	2
Rules	1	2
Scrum	2	4
Taking Flag from Opponent	2	5
Team Behaviour	3	10
Team Preference	1	2
4 Combat Demo	5	84
Address Agents	1	2
Affect of other games	1	1
Agent Behaviour	3	11
Agent gets Stuck	1	1
Agent Health	3	6
Agent Movement	3	10
Bullet Dodge	2	2
Compare Agents	1	2
Emotions	1	1
Human Playable	1	1
Interactivity	1	1
Shooting	2	11
Sources	3	8
Strafing Wheel Effect	1	1

Name	Sources	References
Visual Representation	1	2
Walls	1	2
x Markers	3	12
5 3D Prototype	7	68
Agent Behaviour	3	18
Agent Movement	2	3
Agent Objective	1	1
Agent relationship to each other	2	5
Agent Sensors	3	8
Colour of Balls	2	2
Compare Agents to animals	1	2
Compare Agents to Robots	1	1
Compare to other games	1	1
Engine	1	3
Participant Previous experience	1	1
Speaking for Agents	1	1
User Interaction	2	2

10.3 Believability Criteria Statistical Analysis Results

Agent 1 Binary

Questions		Avg. $\bar{x}$	Std. Dev. σ	1 Std. Dev. Range $\bar{x}-\sigma$	$\bar{x}+\sigma$
1	Perceives the world	2.7	1.4	1.3	4.1
2	Easy to understand what agent is thinking	3.3	0.7	2.6	4.0
3	Agent has personality	3.1	1.4	1.8	4.5
4	Agent behaviour draws my attention	3.4	1.2	2.3	4.6
5	Agent behaviour is predictable	3.9	1.6	2.2	5.5
6	Agent behaviour is coherent	3.3	0.7	2.6	4.0
7	Agent behaviour changes according to experience	1.9	1.5	0.4	3.3
8	Agent interacts socially with other characters	1.7	1.6	0.1	3.3

Agent 2 Continuous

Questions		Avg. $\bar{x}$	Std. Dev. σ	1 Std. Dev. Range $\bar{x}-\sigma$	$\bar{x}+\sigma$
1	Perceives the world	3.6	1.3	2.3	4.9
2	Easy to understand what agent is thinking	3.1	1.4	1.8	4.5
3	Agent has personality	3.3	0.9	2.4	4.2
4	Agent behaviour draws my attention	3.6	0.9	2.7	4.5
5	Agent behaviour is predictable	3.1	1.5	1.7	4.6
6	Agent behaviour is coherent	3.1	0.8	2.3	4.0
7	Agent behaviour changes according to experience	2.0	1.5	0.5	3.5
8	Agent interacts socially with other characters	2.3	1.7	0.5	4.0

Agent 3 Remapping

Questions		Avg. $\bar{x}$	Std Dev. σ	1 Std. Dev. Range $\bar{x}-\sigma$	$\bar{x}+\sigma$
1	Perceives the world	2.3	1.8	0.5	4.0
2	Easy to understand what agent is thinking	4.0	0.7	3.3	4.7
3	Agent has personality	3.3	1.3	2.0	4.5
4	Agent behaviour draws my attention	3.5	1.5	2.0	5.0
5	Agent behaviour is predictable	3.8	0.8	2.9	4.6
6	Agent behaviour is coherent	3.8	0.8	2.9	4.6
7	Agent behaviour changes according to experience	2.3	0.8	1.4	3.1
8	Agent interacts socially with other characters	2.3	1.3	1.0	3.5

Agent 4 Adaptive

Questions		Avg. $\bar{x}$	Std Dev. σ	1 Std. Dev. Range $\bar{x}-\sigma$	$\bar{x}+\sigma$
1	Perceives the world	3.0	1.6	1.4	4.6
2	Easy to understand what agent is thinking	3.0	1.4	1.6	4.4
3	Agent has personality	1.7	1.2	0.4	2.9
4	Agent behaviour draws my attention	2.7	0.5	2.2	3.1
5	Agent behaviour is predictable	4.7	0.5	4.2	5.1
6	Agent behaviour is coherent	3.7	0.5	3.2	4.1
7	Agent behaviour changes according to experience	0.3	0.5	-0.1	0.8
8	Agent interacts socially with other characters	1.3	1.9	-0.6	3.2

Table 37 Mean, standard dev. and 1 st. dev. range for each question per agent

10.4 User Experiment: Consent Form

Micah Rosenkind
PhD Student
University of Brighton
Watt Building –CEM
Moulsecoomb
Brighton, BN2 4GJ
T: 01273 642598
M: 07950 500404

Dear participant,

This is an agreement between you and me, Micah Rosenkind to assist in tests to be included in my PhD thesis on virtual agents.

You will be asked to:

- Provide details of your age, educational, professional qualifications and your experience with computers in general and computer games in particular.
- Play 3 rounds of a game against computer controlled opponents
- Answer questions about your opponent and the game
- Observe 5 demonstrations of autonomous agents
- Associate labels and describe the behaviour of those agents

A sound and video recording will be made of your participation in the test. This will be used for analysis purposes only and will be destroyed once work on this project is complete.

You will not be identified in any report or publication arising from this test. Your data will only be used for the purpose of this study and will not be handed to any third parties, sold or used for commercial purposes.

The testing session will take approximately 45 minutes to complete.

If at any time you wish to leave, you may do so without explanation.

Participant's Signature	Date

Experiment Director's Signature	Date

10.5 User Experiment: Handout Script & Questionnaire

Dear participant,

thank you for taking part in my study. In this experiment, **you take the role of a games tester**. Your task is to evaluate a game in terms of controls, graphics, gameplay mechanics, difficulty and fun.

In the next few minutes you will be playing a game **against three computer-controlled characters and asked to compare them**.

During the test, try to **be as vocal (speak loudly) as you can** about any thoughts you have regarding the experiment, the game itself and the behaviour of the opponent character.

The script below will advise you what to do. If you have any questions feel free to ask the experimenter.

Step 1: Read the instructions to the game.

They should be displayed on the screen in front of you, but are also included below:

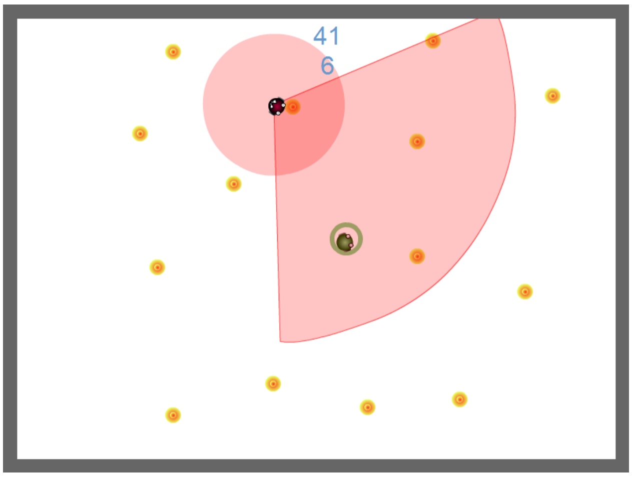
Your objective is to **collect all of the coins on the screen as quickly as possible**, while **avoiding being seen or captured** by the patrolling enemy.

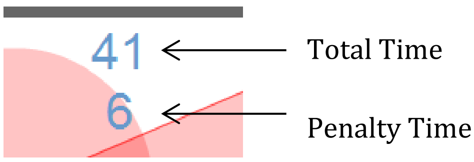
The game can end in two ways:

1. You **collect all the coins** and win

2. The **enemy captures you** and you lose.

If you get seen by the Opponent, penalty points will be subtracted from your score. These are shown underneath the time display at the top of the screen.



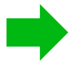


Step 2: Practice the controls of the game.

Click on the **Start Training** button on the title screen

The game will initially be paused, left-click anywhere on the screen to start the game.

This training exercise will familiarise you with the controls. There are two methods of steering your character:

- **Indirect Control**: Left-click to place target marker circle.  Your character will drive to the target marker and stop.
- **Direct Control**: Hold the left mouse button. An arrow marker  will appear and your character will drive in this direction. When you let go of the mouse button, the character will drive to the location where you let go.

There is no enemy in this world, so feel free to **practice both control methods**.

To finish the exercise, collect all the coins. Your score will be shown. Left-click again to return to the title screen, or repeat the training if you wish to practice more.

Step 3: Playing against opponents

Play against each of the three opponents **in the order A, B and then C.**

From the title screen, start by clicking on the button labelled **A**

The game will **initially be paused**.

To Start the Game, when you are ready, left-click anywhere on the screen to start the game.

Try to reach a score of around 500

While you are playing the game please **comment on:**

- Any **strategy or exploit** you may use to win against the opponent
- Anything **unexpected** about the game or the opponent
- Any **Problems** you find with the controls or the game

Once the trial completes, the game over screen will show your score

Clicking on the game over screen will return you to the main menu.

After playing each of the three games, answer the matching questionnaire on the following pages.

Step 3a: Opponent A Questions

Please answer the following questions after you have played against opponent **A. If any of these are unclear or do not apply, say so out loud and explain how you are interpreting them.**

	Totally disagree					Totally agree
Opponent A perceives the world around him/her	0	1	2	3	4	5
It is easy to understand what Opponent A is thinking about	0	1	2	3	4	5
Opponent A has a personality	0	1	2	3	4	5
Opponent A's behaviour draws my attention	0	1	2	3	4	5
Opponent A's behaviour is predictable	0	1	2	3	4	5
Opponent A's behaviour is coherent	0	1	2	3	4	5
Opponent A's behaviour changes according to experience.	0	1	2	3	4	5
Opponent A interacts socially with other characters.	0	1	2	3	4	5

How would you describe the behaviour of Opponent A in different situation? What emotions would you attribute to the behaviour of Opponent A in these specific situations and Why?

Describe in your own words or use terms like Fear, Anger, Love, Hate, Curiosity etc. where appropriate

When Opponent A didn't see you:

When Opponent A saw you:

When you Collected Coins:

Step 4: Questions about the Game

After playing against each of the three opponents and filling out the questionnaires above, **you may retry any of the games**.

Please continue to be vocal about any thoughts you have while you play and answer the questions below when you are ready.

	Totally disagree					Totally agree
My character was easy to control.	0	1	2	3	4	5
The objective of the game was clear.	0	1	2	3	4	5

	Opponent		
I had the most fun playing against opponent	A	B	C
The most difficult opponent was	A	B	C
The easiest opponent was	A	B	C

Which opponent did you choose to retry and why?

How could the game be improved?

Step 5: Observation Study

Now you can relax. In this task you will not be challenged to play a game, but are instead asked to **describe the behaviour of the agents** in a series of game demos. This task is meant to be done intuitively and quickly and is much less formal.

Some of these demos are slightly interactive (the experimenter will tell you what you can do).

Watch the 5 Demos and write down what kind of behaviours you are seeing or if anything catches your interest. Use the spaces below to take notes:

1. Hunter & Prey Demo

2. Conversation Demo

3. Capture the Flag Game Demo

4. Combat Demo

5. 3D Agents Video

Thank you very much for your participation!