Adaptive Interfaces for Massively Multiplayer Online Games

Chris Deaker

This thesis is submitted in partial fulfillment of the requirements for the Degree of Master of Science at the University of Waikato.

March 2013

Abstract

Massively multi-player online role-playing games (MMORPGs), such as World of Warcraft, have become very popular in recent years. These types of games typically feature rich and complex game environments, enabling more engaging game-play experiences. However, the complexity of the underlying game systems can also result in increased interface complexity, which may diminish player enjoyment — a major element of players' game experience. Players may customise their in-game interfaces to deal with this type of complexity and hence improve their performance, but the challenges associated with manual interface customisation may prevent some players from effectively personalising their own game interface.

Players' behavioural models can be used to provide a means of determining potential player in-game behaviour, thus allowing for the automatic adaptation of game interfaces to better suit player needs. This thesis aims to determine whether player-modelled adaptive interfaces can improve players' game experience in MMORPGs.

A survey of MMORPG players was conducted to determine which aspects of player experience may be impacted by interface modification. The findings of this study informed the development of an adaptive interface feedback system which aimed to provide players with relevant information, in order to improve their game experience. This prototype system was then evaluated, in order to determine the impact of the developed system on players' game experience.

Acknowledgements

First and foremost, I would like to express my very great appreciation to my academic supervisors, Dr. Masood Masoodian and Bill Rogers, who provided invaluable guidance during the early stages of this project and insightful comments during the later stages, as well as being a continual source of motivation throughout the bits in between.

Thank you to Dr. Doris Jung, for her academic support during my first year at The University of Waikato, and her on-going friendship since. I will miss our semi-regular catch-ups.

I would also like to extend my thanks to Dr. Neville Churcher, who introduced me to academic research at The University of Canterbury, and has always been a source of honest and helpful advice. If not for that first summer research project, the past few years would have been very different for me.

In addition, I wish to acknowledge the contribution made by the many survey and evaluation participants, without whom this work would not have been possible.

Thank you to my parents, who have always been supportive, and who have worked hard to provide myself and my siblings with the opportunities we have enjoyed. My appreciation cannot be overstated.

Finally, thank you to Jo, for everything.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Since the widespread success of early arcade games during the 1970's, the commercial video game[1] industry has continued to grow, becoming an integral part of popular entertainment and modern culture. Technological advances have allowed for the creation of new types of games and new methods of interaction with those games, enabling more immersive and enjoyable experiences for players (Watt and Policarpo, 2001; Bainbridge, 2007; Microsoft, 2013). Meanwhile, research performed in fields such as cognitive science has allowed for a better understanding of the ways in which players engage with video games (see Section 2.1). Despite the relatively short time since the inception of the medium, progress has been significant.

In addition to the developing academic field of video game research, the commercial video game industry continues to grow financially, with worldwide revenue forecasts for 2012 reaching USD$78 billion (Baker and Nayak, 2012) — a USD$13 billion increase over the preceding year (Baker, 2011) — and individual game genres such as Massively Multi-player Online Role-Playing Games (MMORPGs) servicing tens of millions of active players (Geel, 2012).

As the industry has developed, so too have the ways in which players engage with their game experiences. Where 2-dimensional black-and-white abstractions of game worlds were once necessitated by technical limitations (see Figure 1.1), modern computer hardware allows for players to experience more immersive and realistic environments (see Figure 1.2). Additionally, improvements in computational processing power allow for increasingly complex game systems to be implemented, enabling the creation of more believable and interesting game-play experiences.

[1] Throughout this thesis, the terms *video game* and *computer game* are used interchangeably.

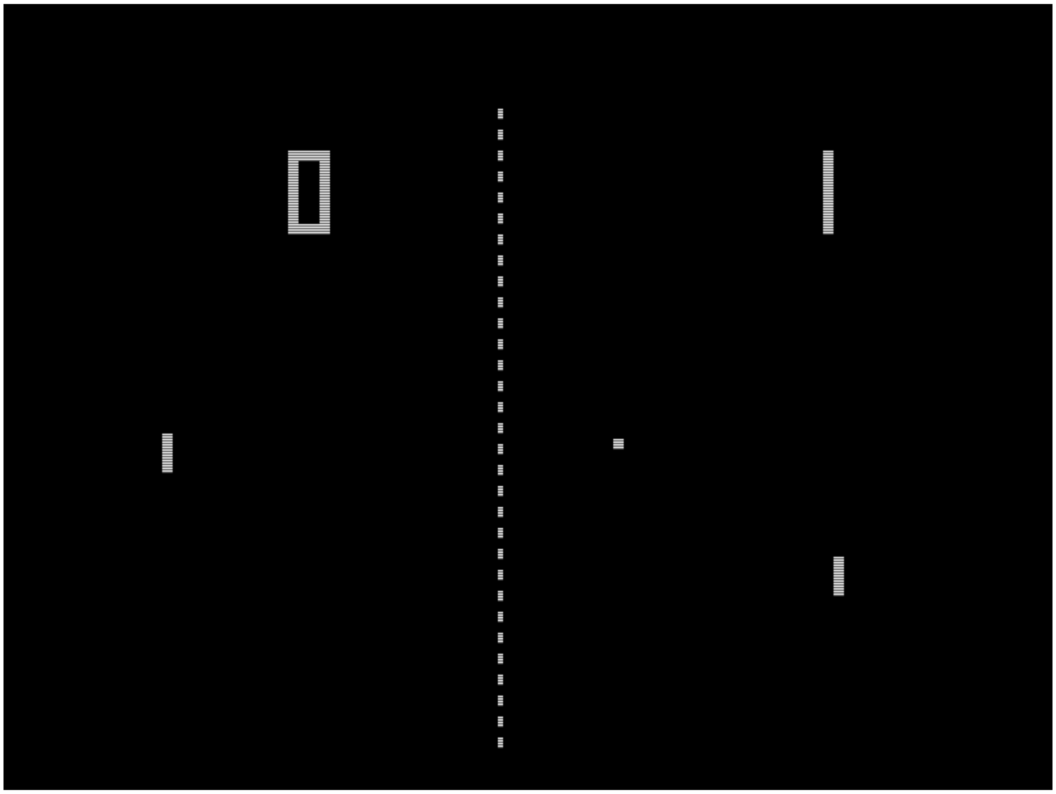


Figure 1.1: The game interface for Pong (Alcorn, 1972).

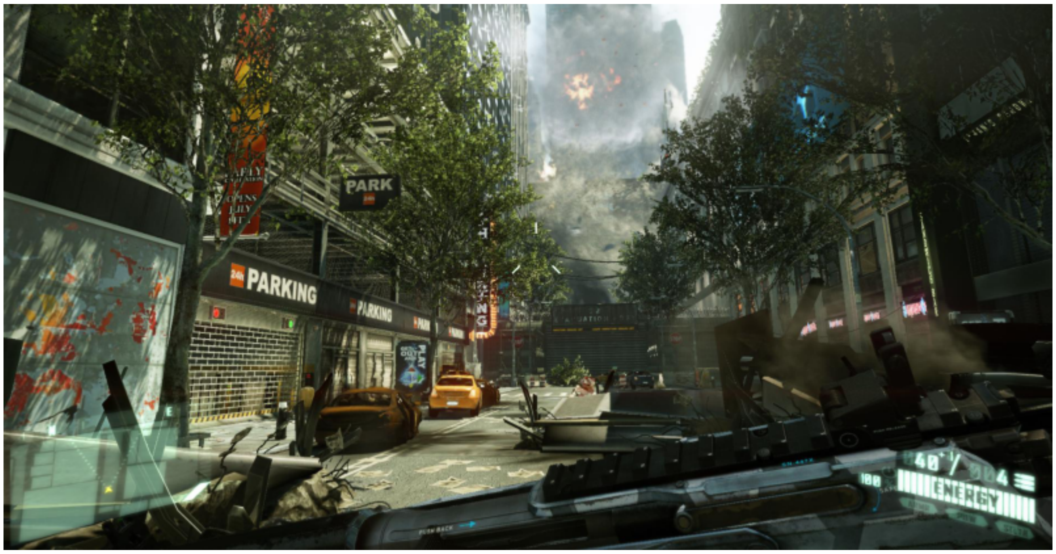


Figure 1.2: The game interface for Crysis 2 (Crytek, 2011).

However, there are potential drawbacks associated with this increase in fidelity. As the underlying complexity of games continues to grow, there is the potential for that complexity to be exposed to the player, especially when attempting to provide the player with important information through the game interface. While challenging but achievable tasks can support player engagement, over-exposing the player to complex or obtuse game-play systems can lead to frustration, harming the player's experience (see Section 2.1.3).

Many games attempt to manage player exposure to complexity by limiting the level of system detail which is shown to the player at early stages in the game, and then gradually introducing the player to other concepts as they continue to play the game. A common implementation of this approach, which is typically seen in MMORPGs, is to limit the number of possible actions the player can perform at early stages in the game[2,3,4]. As the player progresses, they are 'rewarded' by being given access to an increasingly large number of different actions.

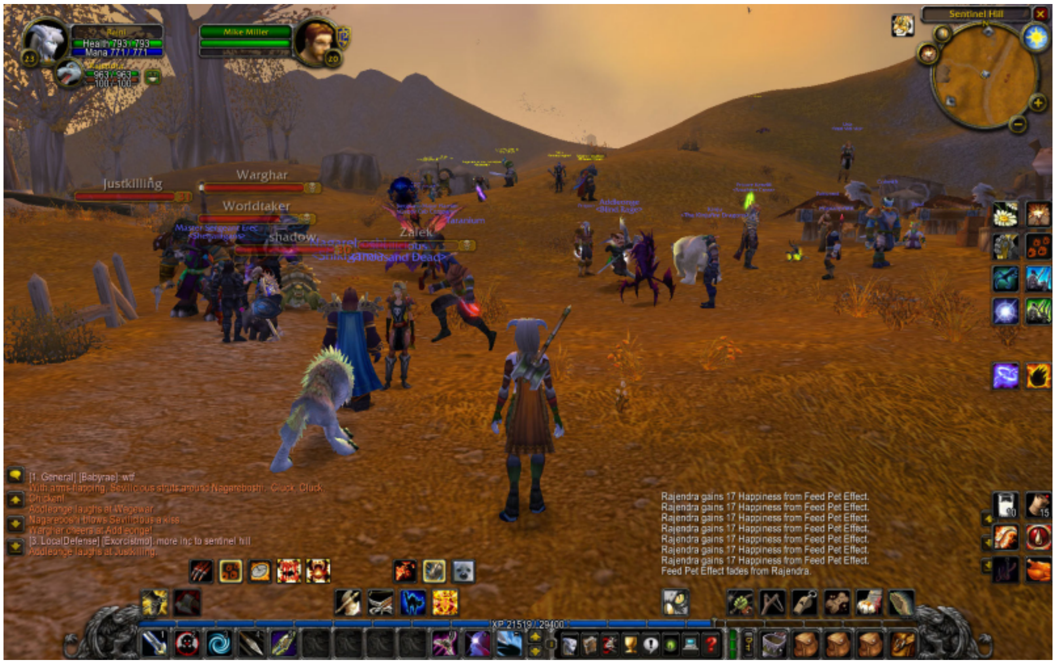
Another method of managing interface-specific game complexity is to allow players to customise their in-game interfaces. This allows individual players to determine which information is displayed to them (and how that information is displayed) according to their own personal preferences regarding the importance of particular elements of the game. Personalised interfaces allow players to reduce distractions caused by the display of unwanted information, and to reduce frustration by ensuring that desirable information is readily available.

One way of supporting interface customisation is to allow players to install packages or plug-ins which alter the game interface in some way. A number of popular MMORPGs provide support for such plug-ins, including World of Warcraft (WoW)(Blizzard Entertainment, 2013). These plug-ins allow players to customise their game interface by modifying its visual aspects (e.g. adjusting the colour, texture, shape and position of the player health bar, etc.), on-screen information (e.g. providing detailed statistics of player damage output, etc.), game-play assistance (e.g. hints and alerts for upcoming events in specific scripted encounters, etc.), and a variety of other aspects of the interface (see Figures 1.3 and 1.4 and Section 2.3.2).

[2] RIFT. `http://www.riftgame.come/en/`

[3] Warhammer Online: Age of Reckoning. `http://warhammeronline.com`

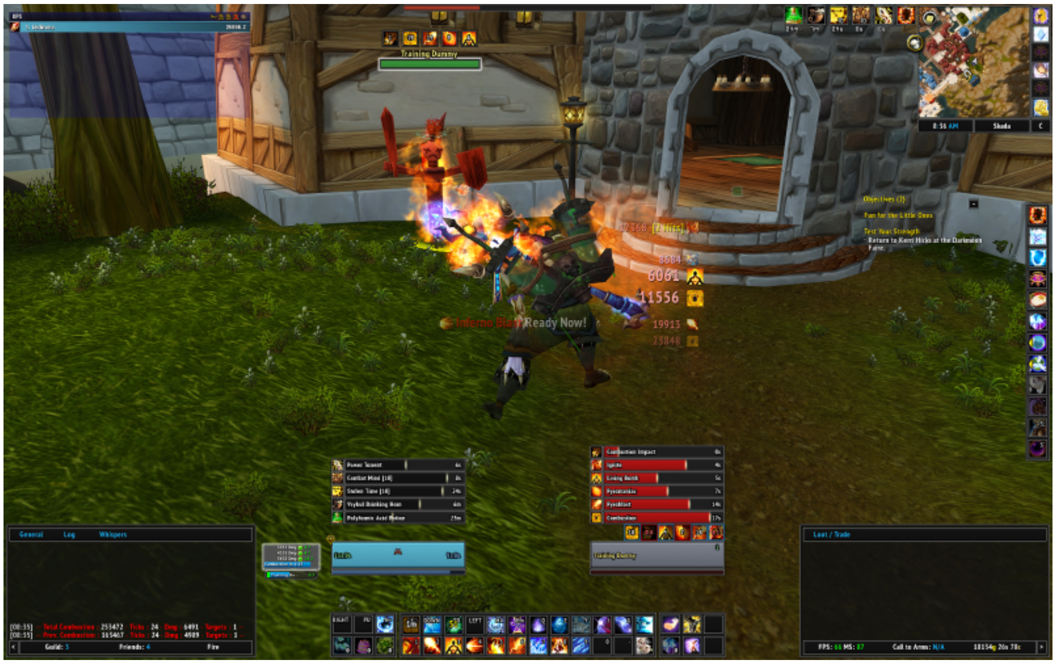
[4] Age of Conan. `http://www.ageofconan.com`



[1. General] [Babyrae]: wtf
With arms flapping, Sevilicious struts around Nagareboshi. Cluck, Cluck, Chicken!
Addleonge laughs at Wagewar.
Nagareboshi blows Sevilicious a kiss.
Warghar cheers at Addleonge!
[3. LocalDefense] [Exorcistmo]: more inc to sentinel hill
Addleonge laughs at Justkilling.

Rajendra gains 17 Happiness from Feed Pet Effect.
Rajendra gains 17 Happiness from Feed Pet Effect.
Rajendra gains 17 Happiness from Feed Pet Effect.
Rajendra gains 17 Happiness from Feed Pet Effect.
Rajendra gains 17 Happiness from Feed Pet Effect.
Rajendra gains 17 Happiness from Feed Pet Effect.
Rajendra gains 17 Happiness from Feed Pet Effect.
Feed Pet Effect fades from Rajendra.

Figure 1.3: The default World of Warcraft interface.



Objectives (3)
Fun for the Little Ones
Test Your Strength
- Return to Kerri Hicks at the Darkmoon Faire.

[08:55] -- Total Combustion : 253472 - Ticks : 24 - Dmg : 6491 - Targets : 1 --
[08:55] -- Prev. Combustion : 165467 - Ticks : 24 - Dmg : 4909 - Targets : 1 --

Figure 1.4: A highly customised World of Warcraft interface.

1.1 Motivations

Despite the existence and use of a large number of MMORPG interface customisation plug-ins, their effectiveness in providing assistance to players by reducing the complexity of the game interface has not been investigated. Exploration of this topic may result in a better understanding of the relationship between different player characteristics and their preferences regarding interface customisation.

Current approaches to interface customisation in MMORPGs require the player to make all their interface modifications themselves (see Section 2.3.2). Even in cases where the player utilises plug-in packages developed by others, they must search for, acquire and install these plug-ins themselves. This presupposes a high level of knowledge, both in terms of an advanced understanding of the capabilities of the default interface — necessary in order to determine where a third-party plug-in is needed — and an awareness of which third-party plug-ins are currently available (and where they can be obtained). Beyond this, many plug-ins require advanced configuration in order to be effective.

Previous research has employed player modelling for the purposes of adapting game difficulty and creating more convincing in-game agents (see Section 2.2.4), yet research into the modelling of player behaviour for the purpose of modifying the game interface is lacking. Such an approach may improve player experience by adapting interface characteristics to individual players, without the need for manual installation and configuration of interface plug-ins.

1.2 Research Goals

With consideration given to the motivations discussed in Section 1.1, the primary research goal of this thesis is: *to determine whether player-modelled adaptive interfaces can improve players' game experience in MMORPGs*.

This core goal can be framed in terms of three key research questions:

1. Which aspects of MMORPG interfaces can be modified in order to improve the player's game experience?

2. How can player actions be modelled and utilised in order to improve the player's game experience?

3. Do interface modifications based on modelling player actions improve the player's game experience?

Section 1.3 discusses the approach taken in this thesis to answer these key research questions.

1.3 Thesis Overview

This thesis begins by providing an overview of the relevant research literature, in Chapter 2. This includes academic work that has aimed to understand the goals and motivations of video game players, providing an insight into potential methods of improving players' game experience. A variety of approaches to modelling player behaviour are discussed, as are various approaches to interface adaptation. Areas of research lacking exploration are identified, providing a basis for the work conducted in this thesis.

In order to determine which aspects of player experience may be impacted by interface modification, a survey of MMORPG players was conducted. The study, discussed in detail in Chapter 3 (also see Deaker et al. (2012)), aimed to identify and characterise the interface customisation habits and motivations of WoW players. In addition to providing insight into motivations for interface customisation, results of the study suggest ways in which players wish to have their game interfaces improved.

The findings of this study informed the development of an adaptive interface feedback system, which aimed to provide players with relevant information, to improve their game experience. The system uses behavioural modelling to provide dynamic and adaptive feedback to players, by predicting their likely future actions. Chapter 4 discusses the implementation of this system, called *WatchAndLearn*, which was developed as a plug-in for the World of Warcraft MMORPG.

Following the development of the prototype system, a user study was conducted. The goal of this study was to determine the impact of the developed system on players' game experience (see Chapter 5). More specifically, this study aimed to evaluate the efficacy of the utilised approach to behavioural modelling, as well as to determine the appropriateness of the prediction visualisation.

The implementation and evaluation of the prototype system provided insight into the potential for adaptive user interfaces to improve the game experience of MMORPG players. Chapter 6 discusses the contributions of this thesis, and outlines a number of potential avenues for future research that have been identified by this work.

Chapter 2

Background and Related Work

This chapter provides an overview of previous research and development relating to this thesis. Section 2.1 discusses previous work that aims to understand and describe the goals and motivations of players. Section 2.2 examines methods of modelling player behaviour for a variety of purposes. Finally, Section 2.3 provides an overview of research related to interface customisation and the development of adaptive interfaces.

2.1 Player Interaction in Computer Games

In order to improve players' game experience, the goals and motivations of players must be understood.

Previous work has attempted to classify Multi-User Dungeon (MUD) players into a number of distinct categories, using common characteristics in order to differentiate player types (see Section 2.1.1). More recently, large scale surveys of MMORPG players have identified a number of over-lapping motivational components, indicating that players may not fall neatly into well-defined categories (see Section 2.1.2).

The widely accepted and empirically proven theory of *flow* has more recently been applied to video game experiences, with some practical applications showing the theory to be relevant (see Section 2.1.3). Research into cognitive models of player experience helps to provide a more complete description of how players engage in video games and other forms of interactive entertainment (see Section 2.1.4), while neuro-psychological research suggests why players interact with video games in these ways (see Section 2.1.5).

2.1.1 Player Types and Motivations in Multi-User Dungeons

The basis of the modern MMORPG genre can be traced back to Multi-User Dungeon (MUD) games, which provided some of the first implementations of large-scale, real-time, virtual multi-player game worlds (Castronova, 2005). While MUDs are often text-based, they share a number of similarities with modern MMORPGs, including common themes (e.g. fantasy, science-fiction), game play styles (e.g. class-based, character-driven role-playing), and methods of interaction within the game world (e.g. focusing on dungeon exploration and loot-driven combat encounters). Due to these similarities, previous research conducted to explore the motivations of MUD players is relevant when considering the motivations of MMORPG players.

In a seminal piece of analysis, *Hearts, Clubs, Diamonds, Spades: Players Who Suit MUDs*, Bartle (1996) presented a "simply taxonomy", intended to capture and describe the motivations of MUD players. In this analysis, Bartle characterised four player types: *achievers*, *explorers*, *socialisers* and *killers*. While some cross-over between areas was expected, Bartle posited that:

> "This research suggests that many (if not most) players do have a primary style, and will only switch to other styles as a (deliberate or subconscious) means to advance their main interest." Bartle (1996)

Bartle further described these four player types using an interest graph, displayed in Figure 2.1. By acknowledging not only the *types* of player, but also how they are defined in terms of their interaction within the game, Bartle argues that it is possible to "change the player type balance" by recognising what type of player an individual is, and reacting accordingly. For example, a game (specifically, a MUD) can emphasise *players* over *world* (and thereby better appeal to *killers* or *socialisers*) by adding communication facilities or decreasing the size of the world. Conversely, reducing available communication facilities, or increasing the size of the world, is suggested as a means of increasing appeal for *achievers* and *explorers*.

While Bartle's taxonomy has been met with some criticism for its simplistic classification methodology (Yee, 2007), it has provided a grounding for discussion of player motivation in large-scale multi-player video games.

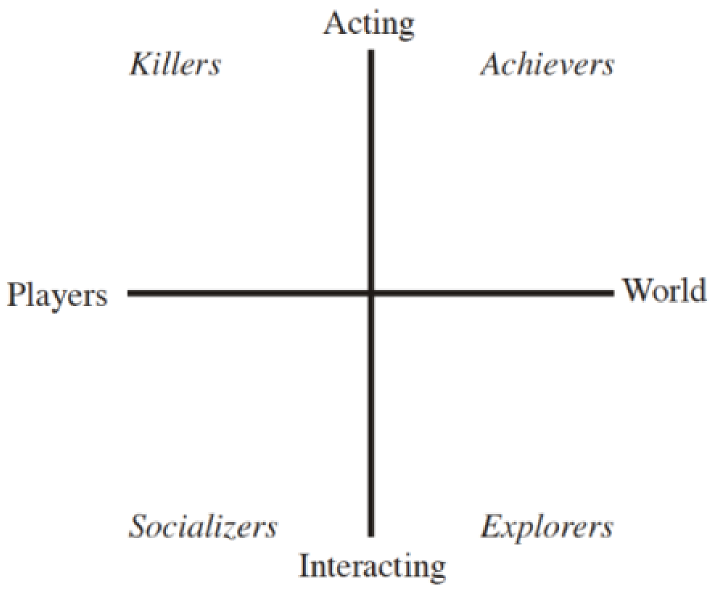


Figure 2.1: Bartle's four player types and their interests.

2.1.2 Motivations of MMORPG Players

In an effort to establish an empirical model of player motivation in MMORPGs, Yee (2007) conducted a study of MMORPG players that revealed 10 motivational subcomponents, which could be grouped into 3 over-arching components of *achievement*, *social*, and *immersion* based motivations. These factors are shown in Table 2.1.

The study also presented findings describing relationships between motivations and demographic variables, such as age, gender and usage patterns. The study seemingly confirms stereotypical assumptions regarding the effect

Table 2.1: Yee's motivational factors and overarching components (Yee, 2007).

Overarching component	**Motivational factor**
Achievement	Advancement Mechanics Competition
Social	Socialising Relationships Teamwork
Immersion	Discovery Role-playing Character customisation Escapism

of gender on motivation, implying that female players tend to favour relationship factors more strongly than male players. However, while this is true, Yee indicates that the effect of this single measurement is misleading.

> “...there is a gender difference in the relationship subcomponent but not in the socializing subcomponent although these two subcomponents are highly related. In other words, male players socialize just as much as female players but are looking for very different things in those relationships.” (Yee, 2007)

Finally, Yee’s analysis suggests that different motivational components in MMORPGs do not necessarily exclude each other, as previously posited by Bartle (1996). “Just because a player scores high on the Achievement component doesn’t mean they can’t also score high on the Social component.” (Yee, 2007)

2.1.3 Flow in Computer Games

Flow describes a balanced mental state, where a persons ability allows them to perform an activity successfully, but the level of challenge of the activity is high enough that the person does not become bored (see Figure 2.2) (Csikszentmihalyi, 1991; Csikszentmihalyi and Csikszentmihalyi, 1992). The result is a sensation of complete focus and absorption in a task. “Optimal experience requires a balance between the challenges perceived in a given situation and the skills a person brings to it.” (Csikszentmihalyi and Csikszentmihalyi, 1992)

Through an empirical study, Csikszentmihalyi (1991) identified eight major elements of flow, not all of which need to be present simultaneously in order to facilitate a state of flow:

1. A challenging but achievable task to be completed.
2. Ability to concentrate completely on the task.
3. Task has clear goals.
4. Task provides immediate feedback.
5. Deep but effortless involvement in the task.
6. Exercising a sense of control over their own actions.

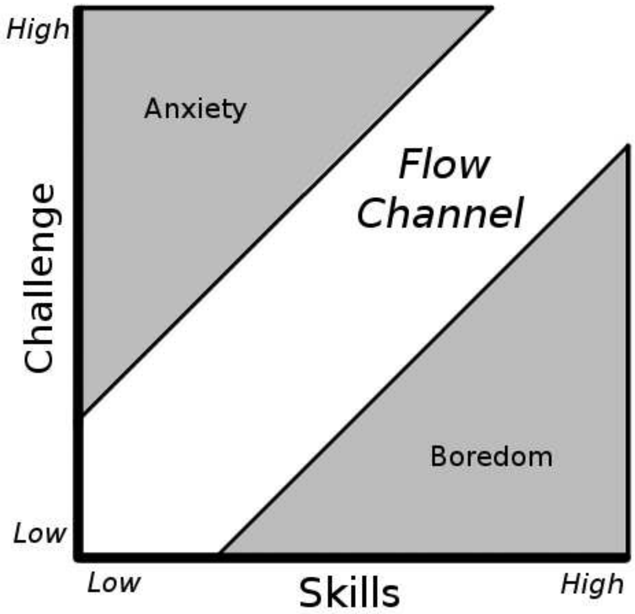


Figure 2.2: Flow occurs where a person's skills are sufficient and the task is appropriately challenging (figure adapted from Csikszentmihalyi (1991)).

7. Loss of self-awareness.

8. Alteration of the sense of time.

Later research by Cowley et al. (2008) aims to provide an interpretation of flow more specific to video games, by framing the player's relationship with a game in terms of an information systems framework, with the intent of enabling a mapping of flow onto game-play. This mapping requires a method of describing flow in terms of game-play elements. Jones (1998) provided examples of possible in-game manifestations for each of Csikszentmihalyi's eight elements, implying that the theory can be applied directly to game design (see Table 2.2).

By recognising the elements required in order to support and encourage a sensation of flow in players, games can be designed to provide players with more immersive and satisfying experiences. However, some approach to measuring and verifying a game's support for flow is required. Sweetser and Wyeth (2005) propose a heuristic-based model they call *GameFlow* which consists of eight elements that directly relate to Csikszentmihalyi's original elements.

Heuristic evaluations of this model were carried out on two commercially

Table 2.2: Eight elements of Flow and corresponding game play attributes, from (Jones, 1998).

Element of Flow	**Manifestation in a game**
Task that we can complete	The use of levels in games provides small sections that lead to the completion of the entire task.
Ability to concentrate on task	Creation of convincing worlds that draw users in. The dungeons and labyrinths in Doom II help suspend your belief systems for a time.
Task has clear goals	Survival, collection of points, gathering of objects and artefacts, solving the puzzle.
Task provides immediate feedback	Shoot people and they die. Find a clue, and you can put it in your bag.
Deep but effortless involvement	The creation of environments far removed from what we know to be real helps suspend belief systems and takes us away from the ordinary.
Exercising a sense of control over their actions	Mastering controls of the game, such as a mouse movement or keyboard combinations.
Concern for self disappears during flow, but sense of self is stronger after flow activity	Many games provide for an environment that is a simulation of life and death. One can cheat death and not really die. People stay up all night to play these games. It is the creation of an integration of representation, problem, and control over the systems that promotes this.
Sense of duration of time is altered	Years can be played out in hours; battles can be conducted in minutes. The key point is that people stay up all night playing these games.

developed video games. Results of these heuristic evaluations were then compared against criticisms and comments that were made during expert reviews of the same games, and showed that these elements could be successfully measured, and that they did in fact have a bearing on players' enjoyment of the games evaluated. While the application of the *GameFlow* model was the goal of this research, the findings also suggest that the core theory of flow can be successfully applied to video games, and may assist game designers in pinpointing parts of a game which require improvements.

Chen (2006a) argues for a less granular adaptation of Csikszentmihalyi's eight elements of flow, choosing instead to digest these from a game design perspective, creating three specific elements necessary to evoke flow in video games:

1. As a premise, the game is intrinsically rewarding, and the player is up to play the game.
2. The game offers the right amount of challenges to match with the players' ability, which allows him/her to delve deeply into the game.
3. The player needs to feel a sense of personal control over the game activity.

Research conducted by Chen (2006a) aimed to utilise flow in the development of a game design methodology which employs dynamic difficulty adjustment as a means of keeping players within a continual zone of flow. Autodynamic difficulty adjustment monitors player performance, and selects appropriate strategies in order to tune game play difficulty appropriately (Bailey and Katchabaw, 2005). Chen notes that typically, dynamic difficulty adjustment has the negative side-effect of reducing the players sense of control over their experience — a core element of flow — and suggests that this can be remedied by providing the player with a number of possible paths through the game, which may be either more or less challenging:

> "Once a network of choices is applied, the Flow experience is very much customizable by the players. If they start feeling bored, they can choose to play harder, vice versa." (Chen, 2006a)

A prototype application of this approach was developed in the form of an online in-browser game called *flOw* (Chen, 2006b). *flOw* aimed to test player-oriented dynamic difficulty adjustment, and featured gameplay which was intended to be "extremely minimal for easily evaluating" the approach. Players were able to immediately increase or decrease the games difficulty, with the intention of providing the player with a sense of control while still allowing for appropriately challenging situations. The game was downloaded over 350,000 times within two weeks of release, and was later released on *PlayStation Network* [1], becoming the most downloaded game on the service in 2007. The critical acclaim and commercial success of *flOw* suggests that the application of flow in video games can be an effective design tool.

2.1.4 Schema Theory and Game Play Experience

Douglas and Hargadon (2001) use schema theory to explain and describe engagement and immersion for readers of hypertext fiction — a digital fiction genre which presents non-linear narratives using hypertext links within a body of text. Schemas are not limited to games, or even narrative structures. Rather, they provide a more general means for application of existing understanding to new situations, by attempting to identify similarities between historical and current experience.

A key element of schema theory is the existence of "scripts", which provide a set of rules which can be applied to particular scenarios. Schank and Abelson (1977) articulate this using the example of a restaurant, where customers enter, order, eat, then exit, with each individual task being performed apparently without hesitation. For customers who have never visited this particular restaurant, the ready execution of these tasks can be attributed to familiarity of the process from other restaurants which they have visited, with the knowledge gained during these previous experiences then being applied to a different situation which appears to share a number of similarities.

The flexibility of scripts allows for versatility, enabling a single script to be adapted according to situational constraints. As there will tend to be a number of obvious or subtle differences between each situation which a single script may be applied to, these scripts must be rapidly modified in order to support additional scenarios (Douglas and Hargadon, 2001). In the restaurant example, this may take the form of a customer entering a buffet restaurant which requires payment in advance. In order to address the situation, and apply this newfound knowledge to future scenarios, the customers "restaurant script" must be adjusted.

[1] Sony Computer Entertainment Europe. `http://nz.playstation.com/`

Schemas provide individuals with frameworks which can be applied in order to understand tasks they are currently undertaking. Douglas and Hargadon explain this using the example of a light fictional novel, where the reader can rely on the basic overall structure of the novel, and is able to focus more attention on the individual details which might differ. Further, the predictability, typical of game schemas (which tend to favour well-established base mechanics where possible), helps to enable a "trance-like" state which has been recognised as a hallmark of reader immersion in fictional writing (Douglas and Hargadon, 2001; Schank and Abelson, 1977). The impact of predictability is also evident in the opposite case: where elements of the reader or player experience defy convention, requiring an adjustment to the applied script, and therefore a greater level of engagement in the activity.

Recognising the importance of player expectation in regards to game schemas provides game developers with the means to manipulate elements of the game in order to either encourage a "trance-like" experience (by meeting player expectations), or a more challenging and engaging experience (by confounding those expectations).

Lindley and Sennersten (2006) propose a cognitive framework which aims to integrate schema theory and attention theory in order to better describe the cognitive processes underlying game play. The framework involves comparative analysis of various player interaction patterns — referred to as *game play gestalts* — intended to identify the relationships between particular game design features and the experience of the player. Lindley and Sennersten argue that this form of analysis allows for a more specific description of player interaction.

> "This theory promises a more explicit mapping from designs to cognitive affects and play patterns than current design practice allows for, providing clearer principles for design decision making and the translation of design intentions into patterns of features." (Lindley and Sennersten, 2006)

Capturing player interaction in video games is a reasonably trivial matter — as this is nearly always managed in some capacity by the game itself. Identifying patterns based upon this interaction, however, is significantly less trivial. Lindley and Sennersten discuss an example where a player might have 30 distinct actions available to them (for example, move left, move right, jump, drink potion, and so forth). When attempting to recognise and examine distinct action patterns, the possibility space quickly becomes unmanageably large:

> “Considering a sequence of 10 moves made by the player, the number of combinations of 10 selections from 30 available moves is 30^{10}, or 59 049 x 10^{10} move sequences. Since a typical single player RPG may be played for a hundred hours or so, often with the player selecting several moves per second, the combinatorial space of possible move sequences over the play time required to finish the game is extremely large (in excess of 10^{5319}).” (Lindley and Sennersten, 2006)

However, while the total possibility space is extremely large, a significant proportion of the actions within this possibility space would be nonsensical. For example, it can be assumed that a player would be unlikely to attempt to drink a healing potion if their character was already at a maximum level of health. This assumption can be made by examining the game mechanics and the overall goals of the player given the current game state, in order to extract a game play schema comprised of a number of individual scripts which “make sense” and are therefore more likely to be used within the given situation.

The goal then is not to simply recognise action sequences that are possible, but to recognise action sequences that are sensible, and transcribe these sequences into game play scripts which can be matched against those scripts being employed by the player. By recognising these scripts, game play schemas can be consulted when determining ways in which the game should respond to, and interact with, the player; either by supporting previously recognised scripts — for example, by providing a player with access to health potions when their character’s health is low — or by intentionally confounding those scripts in order to encourage new script generation, requiring a higher level of active engagement from the player.

Lindley and Sennersten propose an additional application of schema theory, where players are encouraged to “create their own retrospective story of the play experience”, by applying their own story schemas (based upon personal experience) to game stories which are only partially described by pre-defined story elements. The intention is to allow the player to experience what appears to be a story which is tailored to their personal experience, and appeals to their own biases and expectations. The schema applied by the player to “fill in the blanks” may differ from that of the game designer, allowing for a more personalised experience which would not be achievable if all story elements were exhaustively described by the game.

2.1.5 Closure

Holopainen and Meyers (2000) present a neuro-psychological description of video game experience, employing the concept of closure. Comprised of two distinct types — predictive and dramatic — closure describes the ability to observe part of something, but perceive its whole. “Predictive closure is the capacity of the mind to suggest consistent completion of a mental model”, while dramatic closure “appears to stem from the property of consciousness that requires formation of a story structure, or internal dialouge [sic]”.

Holopainen and Meyers use the popular game *Tetris* as an example of dramatic closure being used in video games. In *Tetris*, players experience a continuous series of minor dramas as they attempt to deal with a never-ending tide of falling blocks. Each completed row provides a small amount of closure — a lower level of a hierarchical structure of closures:

> “However, Tetris never permits the final, highest level of closure - the game only ends when the player has failed at a series of smaller closures. This is the root of the addictiveness of the game - it causes a state of tension that can never be fulfilled, but can be temporarily sated by further small closures.” (Holopainen and Meyers, 2000)

This theory fits neatly with Lindley and Sennersten’s description of hierarchically structured game play schemas, described in Section 2.1.4. The relationship between the two can be identified without difficulty. While dramatic closure may provide players with the desire to attain a sense of completion, game play schemas may be generated and employed in order to achieve that sense of completion. Specifically, each script within a game play schema applies to a particular level of a hierarchical closure structure — where the completion of that script also marks the achievement of closure by the player. Recognising where the points of closure are in a game may allow game developers to recognise and design in response to the game play scripts that must be employed in order to attain that closure.

2.2 Player Modelling

This section discusses the purposes and methods of modelling player behaviour. While a theoretical understanding of what impacts player enjoyment is valuable, in order to apply this information dynamically, a game must develop an understanding of the current player. Player modelling allows player interaction to be captured and described in such a way that the characteristics of the player can be identified.

Cowley et al. (2008) discuss the need to understand and describe "the player as well as the game and the experience" in order to analyse how games can enable a state of flow for the player. While an understanding of flow may be the primary goal in this case, there are other applications of player modelling which can be used for a more general purpose approach to game design:

> "...player modelling and adaptive technologies may be used alongside existing approaches to facilitate improved player-centred game design in order to provide a more appropriate level of challenge, smooth the learning curve, and enhance the gameplay experience for individual players regardless of gender, age and experience." (Charles et al., 2005)

Section 2.2.1 discusses the differences that must be considered when modelling game players, as opposed to users of general purpose interfaces. Following this, various methods of modelling and player description are discussed, including the Elo rating system (see Section 2.2.2), factorial models of player behaviour (see Section 2.2.3), and modelling techniques intended to support player imitation (see Section 2.2.4).

2.2.1 Modelling Players, Not Users

While user modelling is a common approach to providing useful feedback in Intelligent Tutoring Systems (ITS) (Corbett et al., 1997; Sleeman et al., 1982; Ross, 1987), the goals and constraints of ITS differ significantly from those of video games. The domain being taught by an ITS is generally analysed and described to a high level of detail, allowing for the factors which the system should track to be determined with relative ease. In contrast to this, while game domains are usually highly constrained (since they are artificially constructed, the entirety of the domain must be described within the game), it is more difficult to formulate reasonable expectations about player behaviour within an artificially constructed environment which they may have not encountered before (Beal et al., 2002).

Further, ITS are primarily concerned with educating students, making the process of evaluating user input simpler — a correct answer is a successful result, while an incorrect answer signals the need for further assistance from the system. In the context of gaming, however, the delineation between a "good" and "bad" result is less obvious. The player losing a round, crashing their vehicle, or having their character die does not necessarily indicate a negative experience, and may in fact contribute to enhancing the enjoyment of the player in the long term. This makes the tracking of "correct" and "incorrect" actions, or formulation of a model representing the *knowledge space* of the player insufficient if player/user experience is to be modelled with the intended goal of enhancing the enjoyment of the player.

However, certain aspects of ITS user modelling may also be relevant to modelling of game players:

> "Certain player behaviors can be identified that would have predictive significance for game progress and user engagement. For example, quit behaviour may be a strong indicator of user frustration; analysing the precursors to "quit" would be revealing. In evaluation studies with our *AnimalWatch* middle school math tutor, we learned that rapid re-entry of an answer within a brief time interval was indicative of student frustration and boredom; this behaviour served as a signal that the problems being presented were too repetitive and that the student was ready for a new math topic." (Beal et al., 2002)

It is important that factors impacting enjoyment are identified in advance for the purpose of modelling. By capturing and modelling aspects of the game which are relevant to player enjoyment, this data can later be used to enable strategies that help to prevent player frustration and increase their engagement.

Beal et al. (2002) also suggest an application of ITS techniques for generating models built using data from a large number of users, which could represent an estimate of standard player behaviour. This approach was used for the *AnimalWatch* ITS, and helped to reduce the training time required before the ITS was able to deliver useful suggestions (Beck et al., 2001). This approach can easily be applied to on-line video games, where usage statistics from thousands or even millions of players may enable the construction of generic "starting-point" models.

2.2.2 The Elo Rating System

The Elo rating system is an established technique for evaluating player skill (Elo, 1978), used by a number of popular multi-player online games (Riot Games, 2009). The Elo system, initially developed as a system for rating chess players, provides numerical ratings which express the relative skill of two players. If a player with a low Elo score beats a player with a high Elo score, the low-scored player receives a relatively large amount of points, while the high-scored player loses the same amount of points. However, if the low-scored player loses, he or she is docked a relatively small amount of points, while the high-scored player again receives the same amount that was docked from the low-scored player. The result is that any player that beats a highly skilled player can expect to have their rank drastically improved, and vice versa.

The Elo rating system is a simplistic but effective method of measuring relative skill for the purposes of match-making, though it is incapable of measuring more detailed statistics. For example, the Elo system does not take into account *how* a player wins or loses, making it unreliable in terms of providing accurate predictions of the outcome of future matches. Furthermore, the system only measures wins and losses, and has no way of measuring or representing detailed factors contributing to enjoyment.

2.2.3 Factorial Models

Charles et al. (2005) suggest a factorial approach to player modelling, where designers "manually partition data space to attach different meanings to various aspects of the data". Players are profiled by measuring the strength of their tendencies towards particular factors, with a complete player model being comprised of a series of numerical values — one for each pre-determined factor. The result is a profile similar to the profiles generated by Bartle's taxonomy of MUD player types (see Section 2.1.1), describing the player in terms of their tendency towards characteristics.

This approach allows for application of the model wherever the recorded factors are relevant, with the major drawback being that factors must be predetermined in order to track data for later inspection and usage. While the proposed approach requires active manual partitioning to be performed in advance by a game designer, data mining tools or techniques such as factor analysis (Yee, 2005) can also be employed to allow for automatic factor identification.

2.2.4 Modelling for Player Imitation

Recent research by Van Hoorn et al. (2009) discusses a modelling technique which is intended to support "the creation of controllers for computer game agents which are able to play a game in a manner similar to a particular human player". This represents a departure from typical approaches to artificially intelligent non-player game characters, where the goal is to learn to play the game as well as possible, rather than learning to play the game "realistically", in the same way as a typical human player would.

This is described by Bryant and Miikkulainen (2007) as *visible intelligence*, where the goal of developing visibly intelligent autonomous agents is to "devise agent behaviours that display the visible attributes of intelligence, rather than simply performing optimally". Visible intelligence was achieved in a strategy game by monitoring player decisions, and using the collected player decision examples to help train the controller of the game agent, thus creating a game agent which was visibly more intelligent (Bryant and Miikkulainen, 2007).

A commercial implementation of player behaviour imitation can be found in the racing game for Microsoft XBox$^{\text{TM}}$, *Forza Motorsport* [2], that allows for the creation of *drivatars*, which drive similarly to the current player. *Drivatars* are created by having the player drive on a number of test tracks, where each track is divided into a number of shorter segments. The game observes the player's approach to each individual segment, and then applies that approach to similar segments in future tracks. The result is a player model that can be successfully applied to any track composed of segments which have been previously observed. The major drawback of this approach is that the model does not support any track which does not contain previously observed segments, thereby either limiting the variety of future tracks, or drastically increasing the

[2] Microsoft Game Studios. `http://forzamotorsport.net`

variety required within the models' training tracks (Togelius et al., 2006, 2007; Van Hoorn et al., 2009).

Direct, Indirect and Multi-objective Modelling

Behavioural player modelling can be accomplished in a number of ways, not all of which are practical, feasible or appropriate for all domains.

Direct modelling "uses supervised learning to associate observations (sensor data) with actions, and then uses the trained function approximator directly as a controller" (Van Hoorn et al., 2009). An example implementation of direct modelling is given by Togelius et al. (2006), where the authors aimed to model player behaviour in a racing game in order to allow for automatic generation of more enjoyable racing tracks. Players were observed while driving on an in-game race track, with their vehicle speed and distance to the edges of the track being continually recorded. This sensor data was used as input to feed-forward neural networks, with acceleration (forward and backward) and steering provided as outputs. The performance of this applied model was poor, with controllers quickly crashing into walls, and being unable to recover (Togelius et al., 2006).

> "This behaviour points to two shortcomings of direct modelling: the complexity of the function to approximate (the human player), and the inability of the model to generalise to unseen situations (e.g. if the human player never crashed into a wall (the behaviour of backing away from a wall is not in the dataset) a controller based on direct modelling of the players behaviour will not know how to back away from walls)." (Van Hoorn et al., 2009)

In response to this limited success, Togelius et al. (2006) then used an *indirect modelling* approach, where measurable elements of player behaviour were identified, and then used to adapt an evolved controller, in order to reproduce those characteristics. The aim was for the controller to drive "well" while still seeming human-like. While performance improved, the "visible intelligence" of the controller decreased, and it became more obvious to human observers that the vehicles were being artificially controlled.

Multi-objective modelling is suggested as a technique which may combine the advantages of direct and indirect modelling, while alleviating their drawbacks (Van Hoorn et al., 2009). The most notable drawbacks in the discussed cases are that direct modelling fails to produce adequately generalisable controllers without extensive domain knowledge and training, while indirect modelling fails to produce an appropriate level of visible intelligence in the controller. It should be noted that a lack of visible intelligence is only a matter of concern where the aim is to create believable autonomous agents, or where it is otherwise important for the controller to exhibit human-like behaviour. For the purposes of simply predicting likely outcomes, where realism is not important, indirect modelling may therefore be sufficient.

Multi-objective modelling is achieved by selecting objectives that are related to both optimal performance and performing in a human-like manner. Indirect modelling allows for more generalised optimal performance, while direct modelling provides more human-like controllers, with the combination allowing for selection of solutions which are "maximally similar to the modelled human while still performing acceptably well" (Van Hoorn et al., 2009).

Results of research into the application of this approach by Van Hoorn et al. (2009) were mixed, with the evolutionary multi-objective optimisation algorithm used being capable of identifying the trade-off between similarity to human-like behaviour and optimal performance, but the employed learning algorithm being unsuccessful in accurately modelling human behaviour. The findings suggest that while this implementation was not successful, application of more powerful sequence learning algorithm may provide more visibly intelligent agents.

2.3 Adaptive Game Interfaces

This section provides an overview of research related to interface customisation and the development of adaptive interfaces. The effects of interface complexity in games are discussed in Section 2.3.1. A common method of dealing with interface complexity is manually customising the game interface (see Section 2.3.2). The benefits, drawbacks and major considerations involved in the implementation and study of general-purpose adaptive interfaces are discussed in Section 2.3.3. Finally, Section 2.3.4 provides examples of interface adaptation in existing games.

2.3.1 Interface Complexity in Games

In typical video games, user interface complexity increases as the player progresses through the game and new character abilities and skills become available. The entirety of the player's character abilities are generally unavailable in the early stages of the game, with unlocked abilities serving as a method of rewarding the player for progression and encouraging continued engagement with the game. As more abilities become available, the complexity of the interface necessarily increases, in turn increasing the player's cognitive load.

In a study of cognitive loads on MMORPG players, Ang et al. (2007) identified "user interface overload" as one of a variety of cognitive loads experienced by players, where the player "failed to attend to important information as the game screen contained a lot of other irrelevant information". It was observed that in response to this, players "learn to prioritise tasks and information" and "learn not to attend to irrelevant information", since this allows them to "become more aware of important information that requires constant attendance such as health status" (Ang et al., 2007).

Depending on the goals of the designers, learning to master the complexity of an interface may be considered a core game goal. However, in cases where the interface is intended to provide the player with a means to interact with the game world, and not intended to confound them, high levels of interface complexity may serve as a barrier to immersion. While players can react by prioritising certain tasks, this may also result in certain tasks or aspects of the game being left unattended by the player, causing them to miss important information or game events, adversely impacting their game-play experience.

2.3.2 Interface Customisation in MMORPGs

To support players in improving their experience (for example, by reducing cognitive load), many games offer configurable interface customisation options, where players may modify common interface characteristics such as the scale, position and visibility of key interface elements. Such games, including WoW, RIFT[3], Warhammer Online: Age of Reckoning[4], and Age of Conan[5], also feature robust application programming interfaces (APIs), enabling the devel-

[3] http://www.riftgame.come/en/

[4] http://warhammeronline.com

[5] http://www.ageofconan.com

opment of game plug-ins or add-ons by community members, in order to provide players with the ability to make more significant alterations to their game interface. In a recent survey of 871 MMORPG players, Adinolf and Turkay (2007) found that interface quality and customisability are considered important and desirable features when establishing and maintaining player engagement and motivation, with 90% of the participants that played WoW indicating that they use third-party interface add-ons when playing the game.

WoW add-ons are typically shared amongst the player community through publicly accessible repositories, such as Curse[6], WoWInterface[7] and WoWAce[8]. For instance, Curse lists over 5,000 available add-ons, many of which have been downloaded millions of times. *QuestHelper*[9] alone has received over 42 million downloads[10], with the total number of downloads for the top 50 ranked add-ons exceeding 420 million.

While interface customisation and installation of third-party add-ons may assist players in simplifying their game interfaces, it necessitates a high level of domain knowledge before customisation will be effective — understanding the relative importance of individual interface elements requires an extensive understanding of the game. Furthermore, certain interface elements may only be important in certain scenarios. For example, character health is generally irrelevant to the player unless they are currently, or expect to soon be, involved in a combat situation, where their character's health is important. While this is an extreme example, there are many interface elements which will become either more or less important throughout the course of the game, with their importance potentially simultaneously impacted by a number of different factors. These combined issues may make manual interface customisation difficult for novice players.

2.3.3 General-purpose Adaptive Interfaces

A significant body of established research exists involving the study and evaluation of various methods of adapting graphical user interfaces in order to improve user experience. Much of this research is focused on general-purpose

[6] `http://www.curse.com/addons/wow`. Last accessed February 19, 2013.

[7] `http://www.wowinterface.com/addons`. Last accessed February 19, 2013.

[8] `http://www.wowace.com/addons/`. Last accessed February 19, 2013.

[9] *QuestHelper*, available from `http://www.curse.com/addons/wow/quest-helper`. Last accessed February 19, 2013.

[10] Including multiple downloads by the same player, as the add-on is updated regularly.

interfaces, such as selection menus (Tsandilas et al., 2005; Cockburn et al., 2007; Gajos et al., 2008) and mobile devices (Bridle and McCreath, 2006). The specific implementation details of such interfaces are not relevant to this thesis — instead, this section discusses adaptive interfaces in terms of more general benefits, drawbacks and considerations.

Findlater and Gajos (2009) found that successful application of adaptation can serve to reduce visual search time, cognitive load, and motor movement. However, the authors note that challenges exist:

> “Often, an adaptive mechanism designed to improve one aspect of the interaction, typically motor movement or visual search, inadvertently increases effort along another dimension, such as cognitive or perceptual load”. (Findlater and Gajos, 2009)

Langley and Fehling (1998) discuss the importance of identifying dependent measures which may be used when studying adaptive user interfaces, suggesting that four general types of measures are appropriate: efficiency, quality, user satisfaction and predictive accuracy. Findlater and Gajos (2009) suggest that the predictability and frequency of adaptive algorithms also bears consideration, citing previous studies (Bridle and McCreath, 2006; Cockburn et al., 2007) raising predictability or stability as a potential factor. When modelling menu performance, Cockburn et al. (2007) found that users' target selection time decreased as they became more experienced with the menu, but only where the interface supported this through stable and predictable item placement.

In response to criticisms of the potential of adaptation to disorient users, Gajos et al. (2008) conducted a study examining the relative effects of predictability and accuracy on the usability of adaptive interfaces. The authors found that while both predictability and accuracy affected user satisfaction, only accuracy had a significant effect on user performance or usage of the adaptive portion of the interface. Additionally, it was found that in certain cases, machine learning algorithms may improve user satisfaction, despite producing inscrutable behaviour:

> “Specifically, if a machine-learning algorithm can more accurately predict a user's next action or parameter value, then it may outperform a more predictable method of selecting adaptive buttons or default values”. Gajos et al. (2008)

Despite this positive outcome, Gajos et al. also note that predictability and accuracy affect different aspects of user satisfaction, and that improvements to one factor may not necessarily completely offset losses to the other. This indicates that application of adaptation necessitates an understanding of what users require from the interface.

In a study testing the effects of varied levels of suggestion accuracy in a menu, Tsandilas et al. (2005) found that higher levels of accuracy resulted in lower selection times, as did reducing the number of suggested items. Additionally, employing a fish-eye lens approach to shrinking non-suggested items resulted in higher levels of selection accuracy from participants. The authors note that, while shrinking irrelevant information is likely to be useful, the effectiveness of such an adaptation technique is dependant upon the accuracy of the prediction mechanism. In situations where suggestion accuracy was low, performance suffered, though selection time was still lower even with low levels of adaptive accuracy than when no adaptation was applied.

Research by Lavie and Meyer (2010) also notes the importance of the accuracy of any algorithm used for interface adaptation. In a study of interface adaptation in an in-vehicle telematic system, adaptivity was found to be beneficial to all users when in familiar or routine situations. However, in unfamiliar situations, which the adaptation algorithm had not sufficiently adjusted to, cognitive load substantially increased, thus negatively affecting performance. The authors therefore suggest utilising a partial application of adaptivity, rather than introducing adaptivity either completely or not at all:

> "Intermediate levels of adaptivity keep users involved in the task and help them become more proficient when performing both routine and non-routine tasks." Lavie and Meyer (2010)

2.3.4 Adaptive Interfaces in Commercial Games

The popular racing series for Microsoft XBox^{TM}and Microsoft XBox 360^{TM}, Forza (see Section 2.2.4), provides players with a selection of optional driver-assist features, which simplify driving for inexperienced players and eases the transition from novice to expert. These options include features such as braking and steering assist, and stability and traction control. Rather than taking control away from the player, the game reacts to situations where the player is struggling, by engaging any activated assist controls only when they are

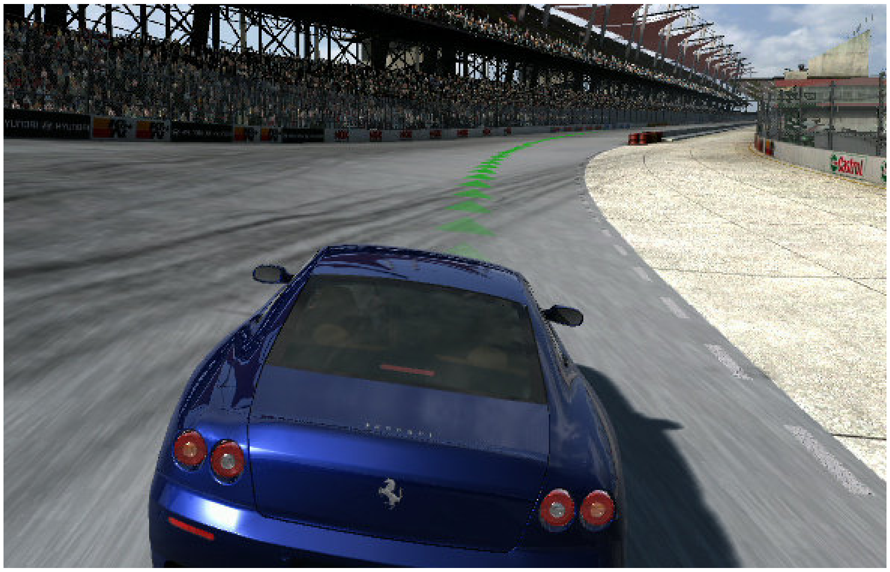


Figure 2.3: Forza 3's "suggested line".

needed. This adaptive approach to difficulty tuning allows novice players to experience the game with minimal frustration, encouraging them to continue playing and enhancing their skills, until they may be able to drive effectively with the automatic assistance disabled.

A more visually focused assistance feature is the *suggested line* driver assist, which renders a series of colour-coded arrows on the track, providing the player with a suggestion of an optimal racing line for the upcoming corner, based upon the player's current position and speed (see Figure 2.3). The colour of the line remains green in areas where the game suggests that the player should accelerate or maintain their speed, and turns yellow or red where the optimal speed is lower than the player's current speed. This provides the player with a comprehensive strategy for approaching corners, indicating both when they should be braking, and the ideal path to follow. By adapting to the player's position and speed, the suggested line can be useful even when players finds themselves in unexpected situations — e.g. having recently crashed or otherwise lost speed, the line will not advocate braking on an upcoming corner if the player's current speed does not warrant it.

This approach to adaptive suggestion does not require robust player modelling, since suggestions are based upon immediately available factors (player position, direction, and speed) and an optimal approach which can be calculated using the same approach used by the game to calculate routes for AI drivers.

2.4 Summary

Section 2.1 discussed previous research which aimed to characterise how players experience and enjoy video games. Models of player experience such as schema theory (see Section 2.1.4) and flow (see Section 2.1.3) provide insight which allows for the development of strategies that can positively affect player enjoyment.

Understanding player experience requires techniques which can accurately describe the player. A number of such techniques are discussed in Section 2.2. However, having an understanding of the individual player and of their gameplay experience is insufficient if the goal is to improve the player's game experience. In order to impact the experience of different players, games must utilise this understanding and adapt the game accordingly.

While Section 2.3 has discussed approaches to interface adaptation, examples of applications of adaptive interfaces in video games are lacking. In particular, there has been little research or commercial development that has focused on applying adaptive interfaces for the purpose of improving player experience. This thesis aims to explore the potential benefits of such an approach.

Chapter 3

Survey of Opinions on Game Interface Customisation

Chapter 2 identified a lack of research exploring the application of interface adaptation as a means of improving players' game experience.

In order to determine ways in which computer game interfaces can be adjusted to support such a goal, it is important to better understand how interface adjustment impacts game-play and player motivation. Manual interface customisation provides players of games such as WoW with the ability to tailor their in-game interface to better match their own preferences, therefore allowing them to express which aspects of the interface improve their own game-play experience.

Therefore, an online survey of WoW players' ratings of a set of motivational factors relating to interface customisation in WoW was carried out. WoW was specifically selected due to its popularity amongst MMORPG players, and the fact that third-party plug-ins and add-ons are commonly used by a large proportion of WoW players (see Section 2.3.2).

This chapter discusses the procedure and outcome of the survey (see also Deaker et al. (2012)). Section 3.1 provides an overview of the survey methodology, while Section 3.2 describes demographic characteristics and game-play habits of the study participants. Section 3.3 discusses the results and implications of survey. The overall outcome of the study is summarised in Section 3.4 and Section 3.5.

3.1 Methodology

For this study, an online survey consisting of three parts was developed[1]. The first part obtained basic demographic information about the gender and age of the survey participants. The second part of the survey asked participants to answer a set of questions relating to their general game-play habits, preferences and motivations when playing WoW. This included the number of years they had played the game (answer: between 0-8 years), the average number of hours they play each week (answer: open-ended), whether they currently had any characters that had achieved the maximum in-game level (answer: yes/no), and a rating of their perceived level of knowledge of in-game mechanics (answer: on a 5-point Likert scale with anchors 1 being *Very low* and 5 being *Very high*).

Participants were asked to rate the importance of a number of motivational factors for playing WoW. The ratings were collected using a 5-point Likert scale, with anchors 1 being *Not important* and 5 being *Very important*. The factors used were those identified by Yee's taxonomy of motivational factors in online games (Yee, 2007). These factors and their overarching components, as identified by Yee, are shown in Table 2.1. Participants were also asked to comment on any other aspects of WoW which they felt were important, but had not been mentioned in this part of the survey.

The third part of the survey asked the participants to answer a set of questions pertaining to interface customisation within WoW. This included whether the participants used third-party interface modifications or add-ons (answer: yes/no), and whether they felt that some add-ons gave players an unfair game advantage (answer: yes/no). Participants were asked to describe any functionality which they felt was not provided by existing add-ons, and to list any add-ons which improved their game-play experience (answer: open-ended).

A 5-point Likert scale was then used to rate the importance of a number of effects of interface customisation when playing WoW. These effects were:

1. Removing unnecessary information from the default interface.

2. Providing additional information not available in the default interface.

3. Improving the look and feel of the default interface.

[1] The survey can be found at `http://tinyurl.com/cg2b3tx`, and is also included in Appendix B.

4. Providing easier access to important game functions or features.

Finally, the participants were asked to rate the importance of interface customisation in relation to the motivational factors rated in the previous section of the questionnaire using a similar 5-point Likert scale, with anchors 1 being *Not important* and 5 being *Very important*. These ratings were subsequently compared during the analysis with previously collected importance ratings for those factors.

Approval for this evaluation was obtained from the Ethics Committee of the Faculty of Computing and Mathematical Sciences at The University of Waikato (see Appendix A).

3.2 Survey Participants

Requests for participation in the survey were posted on a number of WoW-related internet forums[2]. In total 158 participants completed the survey, of whom 78% were male and 20% female (2 participants did not provide their gender). The average age of the participants was 24.7 years old (standard deviation of 8.1). Participants reported an average weekly play-time of 19.7 hours per week (standard deviation of 15.7). The average participant age and play-time were similar to those of participants in a survey of 30,000 WoW players by Yee (2006), where the average participant age was 26.57 years, with an average weekly play-time of 22 hours. Figure 3.1 shows the distribution of age and weekly play-time of the study participants.

Active participation in the forums from which the participants were recruited indicates a strong level of interest in the game. Therefore it was expected that in general, respondents would be knowledgeable about game mechanics and available add-ons. This was confirmed by the survey results which indicated that most of the survey participants could be considered as being above average in their expertise in WoW. On average they had played the game for 4.6 years (standard deviation of 2.0), and 95% had achieved the maximum game level with at least one character (level 85 at the time of the survey). Over 90% of the participants reported a high level of knowledge of in-game mechanics (either 4 or 5 on a 5-point Likert scale), as shown in Figure 3.2.

[2] Including the official WoW game forum at: `http://us.battle.net/wow/en/forum/`.

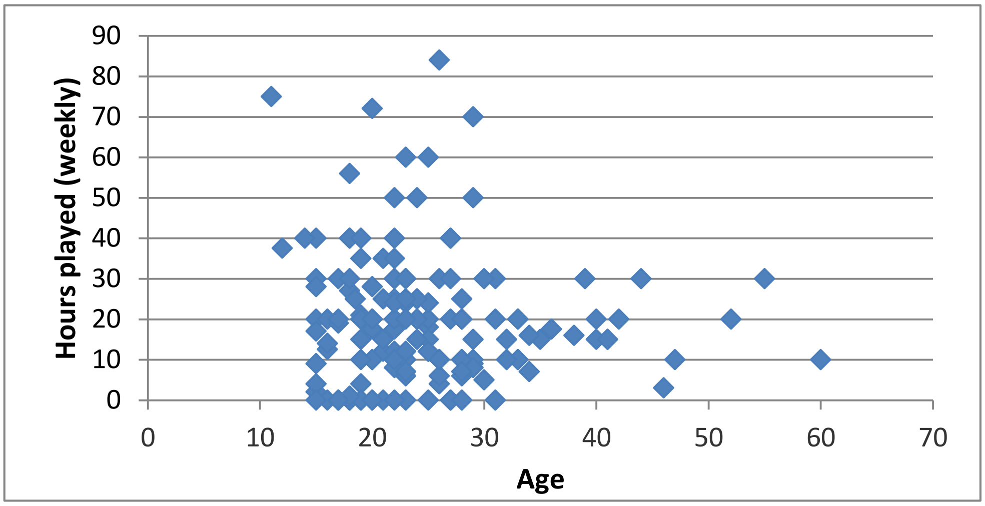


Figure 3.1: Survey participants' age and weekly play-time.

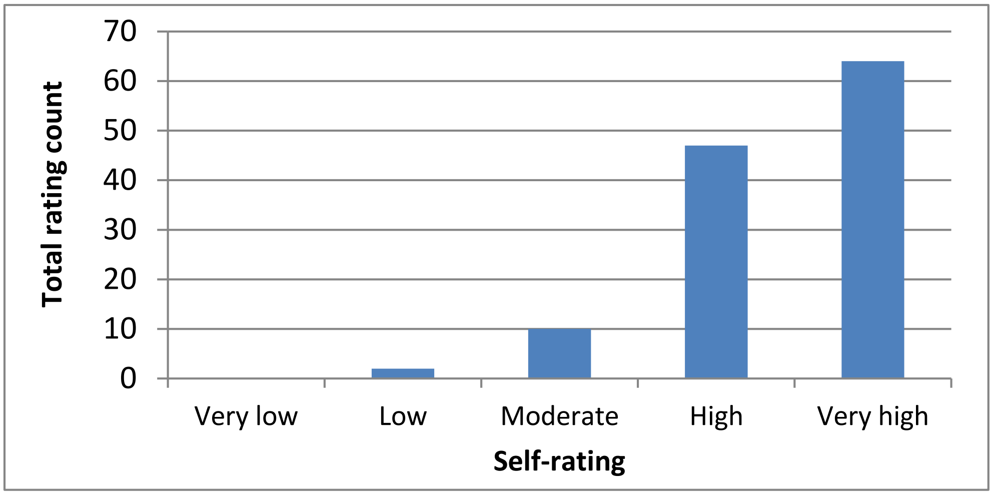


Figure 3.2: Participants' self-ratings of level of knowledge of game mechanics.

3.3 Results

This section discusses the findings of the analysis carried out on the collected survey responses. Firstly, Section 3.3.1 outlines the participants' habits in terms of interface add-on usage. Section 3.3.2 then discusses the participants' ratings for the importance of different motivational factors, both in general terms, and specific to interface customisation. Section 3.3.3 and Section 3.3.4 discuss the effects of interface customisation on player enjoyment, and the initial goals of the participants when they are customising their game interface. Finally, Section 3.3.5 provides an overview of the impact of interface add-ons on game-play, with a focus on fairness.

3.3.1 Usage of Interface Add-Ons

In terms of interface customisation, 92% of the participants reported that they currently modify their game interface using third-party interface add-ons — confirming the results of a previous survey by Adinolf and Turkay (2007) where 90% of the participants that played WoW reported using add-ons. While the WoW Terms of Use[3] prevent add-ons from providing significant "game-breaking" competitive advantages, 23% of the participants indicated that they believe that some add-ons give players an unfair advantage in-game. However, only 22% of the participants claimed that they use add-ons that they believe can provide an unfair in-game advantage. This indicates that the perception of potential issues with fairness does not discourage add-on usage by at least some of the players.

3.3.2 Ratings of Motivational Factors

Table 3.1 shows the participants' rating counts for the importance of motivational factors in general, while Table 3.2 shows the participants' rating counts for the importance of customisation in relation to those motivational factors.

Table 3.3 shows the average ratings for the importance of motivational factors in general along with average ratings for the importance of customisation in terms of those factors. With the exception of competition, all motivational factors were rated higher in general than in terms of customisation specifically.

[3] World of Warcraft Terms of Use: `http://us.blizzard.com/en-us/company/legal/wow_tou.html`

Table 3.1: Rating counts for the importance of motivational factors in general.

	Importance *1: Not important, 5: Very important*				
Factor	**1**	**2**	**3**	**4**	**5**
Advancement	2	2	28	68	58
Mechanics	1	6	28	64	59
Competition	12	21	47	45	33
Socialising	3	15	45	55	40
Relationships	14	26	56	33	29
Teamwork	1	6	24	64	63
Discovery	5	14	47	44	48
Role-playing	76	43	23	6	10
Character customisation	5	14	34	56	49
Escapism	11	17	51	36	43

Table 3.2: Rating counts for the importance of customisation for motivational factors.

	Importance *1: Not important, 5: Very important*				
Factor	**1**	**2**	**3**	**4**	**5**
Advancement	19	5	34	51	49
Mechanics	9	4	29	45	71
Competition	20	13	34	32	59
Socialising	64	46	32	13	3
Relationships	91	32	26	8	1
Teamwork	19	13	43	51	32
Discovery	59	29	40	22	8
Role-playing	88	27	31	9	3
Character customisation	55	32	35	22	14
Escapism	66	22	39	18	13

Table 3.3: Average ratings for the importance of motivational factors in general and for customisation.

	Average rating (S.D.)	
Factor	**General**	**Customisation**
Advancement	4.1 (0.8)	3.7 (1.3)
Mechanics	4.1 (0.9)	4.0 (1.1)
Competition	3.4 (1.2)	3.6 (1.4)
Socialising	3.7 (1.0)	2.0 (1.1)
Relationships	3.2 (1.2)	1.7 (0.9)
Teamwork	4.2 (0.9)	3.4 (1.2)
Discovery	3.7 (1.1)	2.3 (1.2)
Role-playing	1.9 (1.2)	1.8 (1.1)
Character customisation	3.8 (1.1)	2.4 (1.3)
Escapism	3.5 (1.2)	2.3 (1.3)

However, the difference between the means of competition ratings was not statistically significant (paired T-Test, T_{157} = 1.69, p = 0.09).

The ratings for individual motivational factors were grouped into the over-arching factors of *achievement*, *social*, and *immersion*, as shown in Table 2.1. Sub-component factors were given equal weightings. Table 3.4 shows the average ratings and standard deviations for the combined sub-factor ratings.

Table 3.4: Average ratings for the over-arching motivational factors.

	Average rating (S.D.)	
Factor	**General**	**Customisation**
Achievement	3.9 (1.0)	3.8 (1.3)
Social	3.7 (1.1)	2.2 (1.3)
Immersion	3.3 (1.4)	2.2 (1.3)

A two-way analysis of variance was carried out on participants' ratings against the motivational factors and area of importance (general or customisation-specific). Statistically significant main effects were observed for motivational factor, $F_{9,1413} = 110.19$, $p < 0.001$, and area of importance, $F_{1,157} = 247.90$,

$p < 0.001$. More interestingly, there was a statistically significant interaction between motivation and area of importance, $F_{9,1413} = 46.81$, $p < 0.001$. Figure 3.3 shows the average ratings for all motivational factors, both for the general importance and the importance of customisation.

A second two-way analysis of variance was carried out on participants' rat-

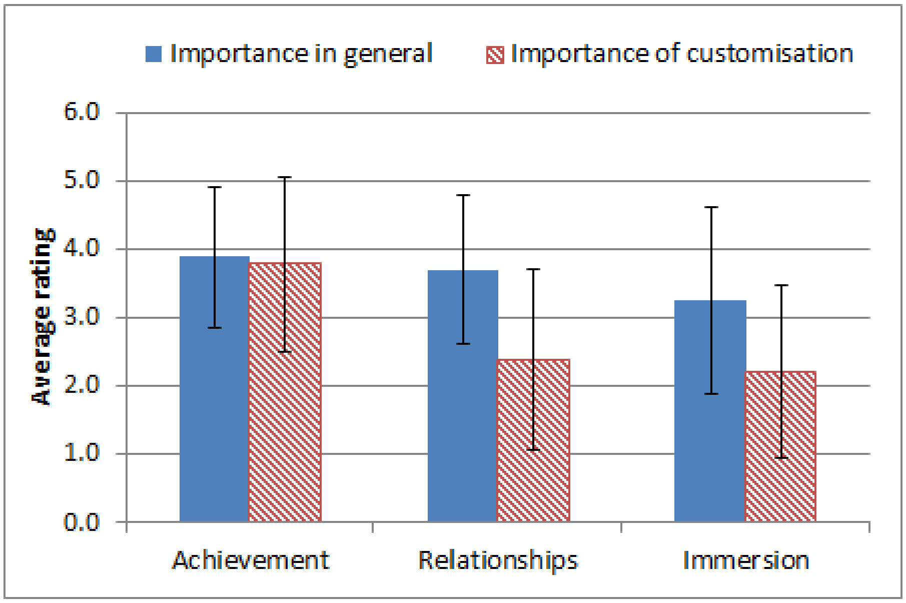


Figure 3.3: Average ratings of the importance of grouped motivational factors.

ings against the area of importance and over-arching motivational factors (see Table 2.1 and Table 3.4). Again, statistically significant main effects were observed for motivational factor, $F_{2,314}$ = 138.93, $p < 0.001$, and area of importance, $F_{1,157}$ = 232.72, $p < 0.001$. Finally, there was a statistically significant interaction between motivational factor and area of importance, $F_{2,314}$ = 75.11, $p < 0.001$.

The effect of this interaction can be seen in Figure 3.3, showing a notable difference between the importance of customisation for *achievement* factors when compared to *relationship* and *immersion* factors. The difference between the three factor groups for general importance is more subdued. Average ratings for all motivational sub-factors are shown in Figure 3.4.

To view this in more detail, Figure 3.5, Figure 3.6 and Figure 3.7 show line plots of rating counts for general importance, and for the importance of customisation, for *achievement*, *relationship*, and *immersion* factors respectively.

One interesting observation is that there is a strong positive correlation between the importance rating of *achievement* for both general importance and interface customisation (r = 0.89). In contrast, correlations between general and customisation-specific ratings are negative for both *relationship* and *immersion* factors (r = -0.77 and r = -0.53 respectively), as well as being weaker for *immersion* specifically.

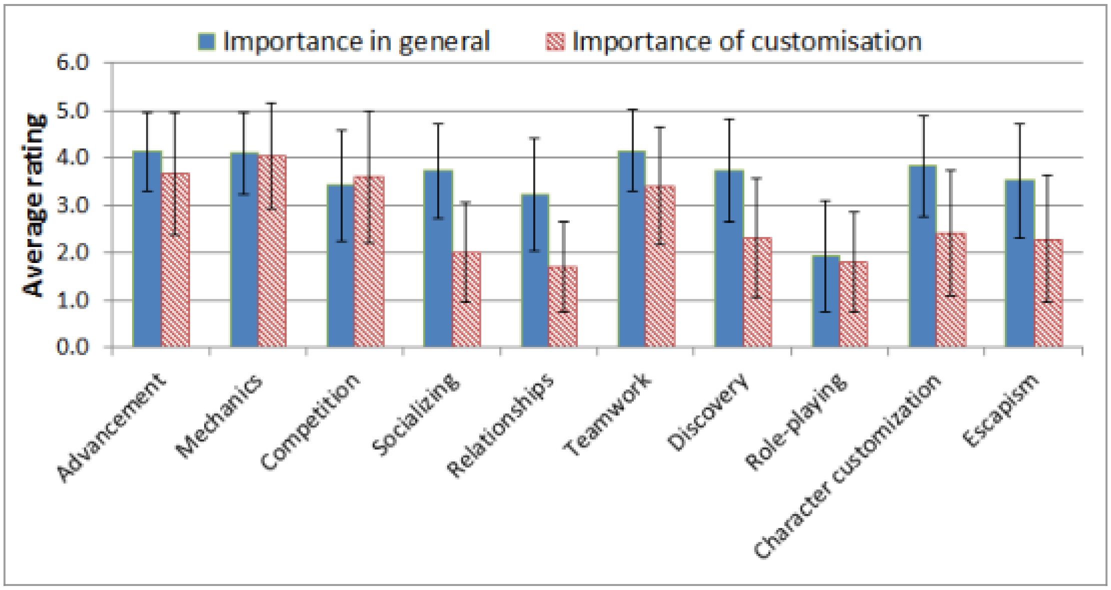


Figure 3.4: Average ratings of the importance of all motivational factors in general, and for customisation.

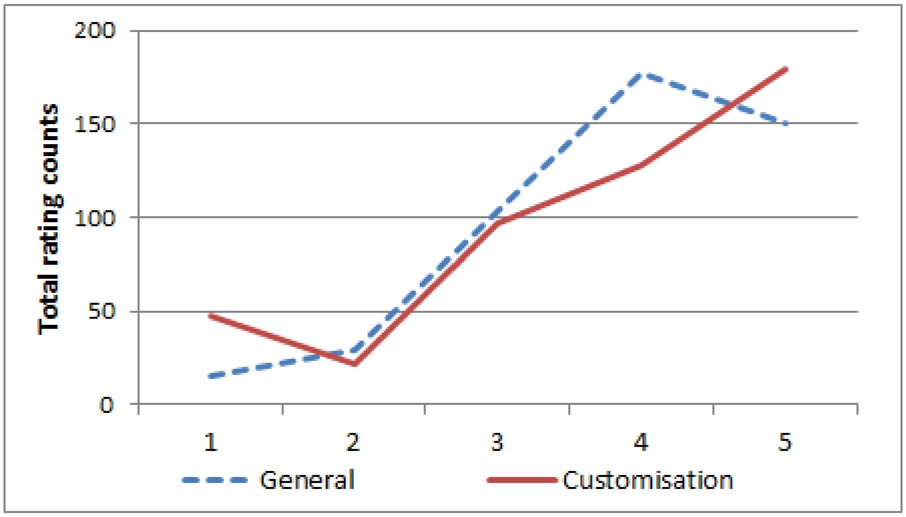


Figure 3.5: Total rating counts for the importance of achievement related motivational factors.

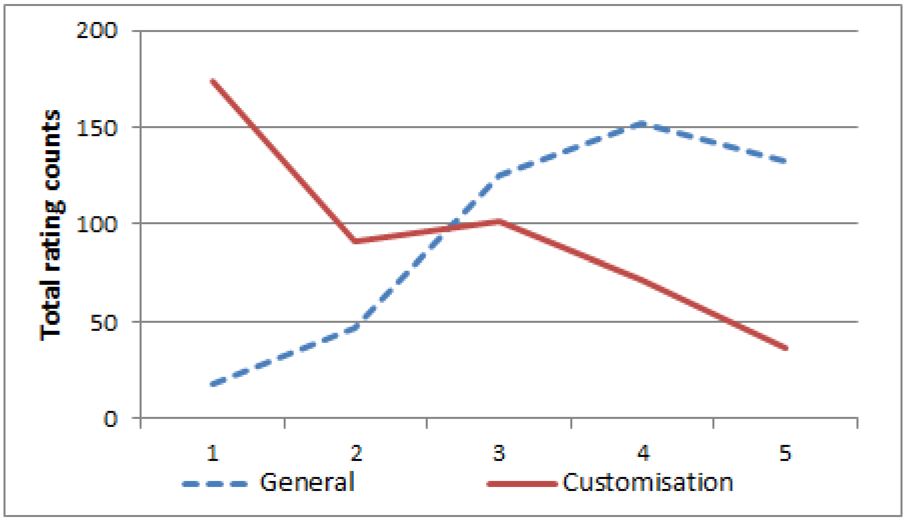


Figure 3.6: Total rating counts for the importance for relationship related motivational factors.

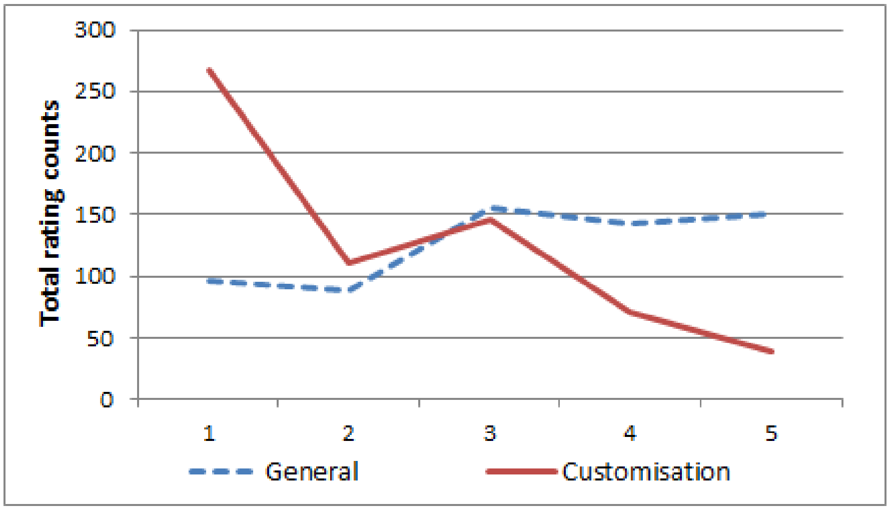


Figure 3.7: Total rating counts for the importance for immersion related motivational factors.

3.3.3 Effects of Interface Customisation

Participants were also asked to rate the importance of interface customisation in WoW. Most of the participants (132) indicated that they considered the ability to modify their game interface important (ratings of either 4 or 5 on a 5-point scale), with 109 participants (69%) selecting a rating of very important (see Figure 3.8). This result was not surprising, given the potential benefits and advantages to be gained from interface customisation (discussed in sections 3.3.4 3.3.5).

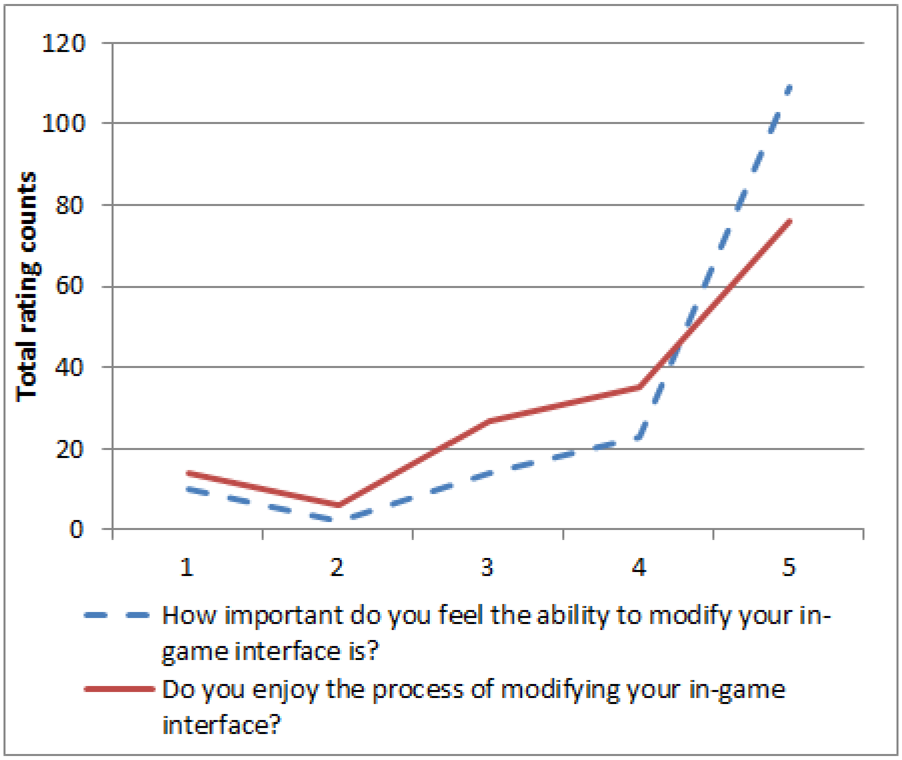


Figure 3.8: Total rating counts for the importance and enjoyment of interface customisation, on a 5-point Likert scale.

More unexpected was the proportion of the participants who indicated that they enjoy the process of modifying their game interface. 111 participants (70%) reported a positive experience when modifying their game interface (see Figure 3.8). Section 3.3.4 discusses more detailed feedback from the participants indicating why they enjoy the interface customisation process.

Figure 3.9 shows a plot of the participants' ratings for the importance of four different effects of interface customisation. Over 60% of the survey respondents gave each effect a rating of 4 or 5, indicating that all the proposed

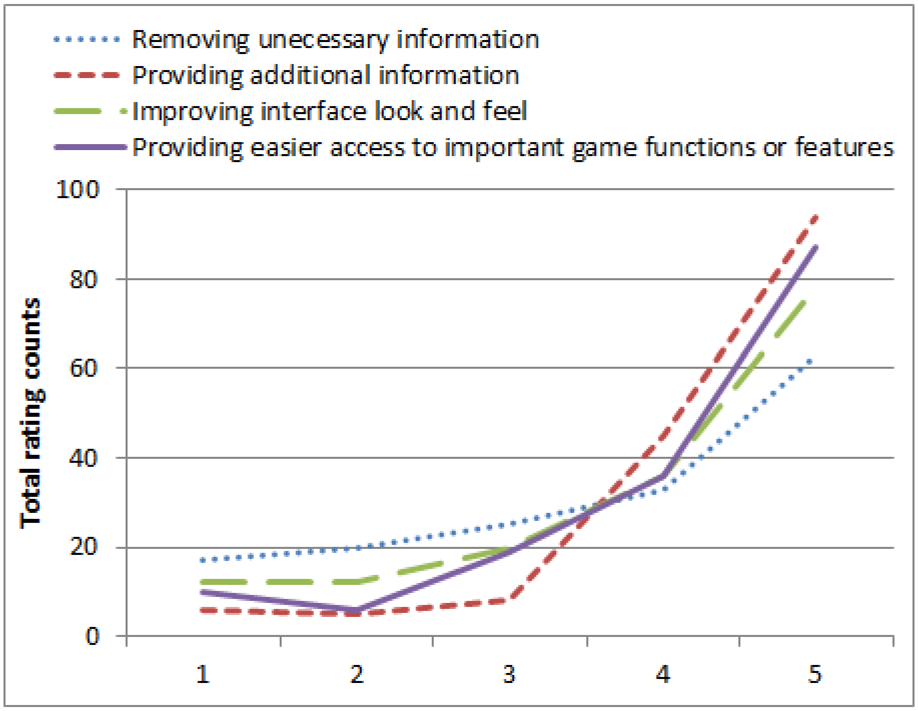


Figure 3.9: Total rating counts for the importance of four effects of interface customisation, on a 5-point Likert scale.

effects were considered important to some degree. Furthermore, this indicates that players may modify their interfaces for a range of reasons simultaneously, rather than focusing heavily on one area (for example, improving accessibility) at the detriment of other areas (for example, improving interface aesthetics).

3.3.4 Reasons for Interface Customisation

The survey also sought the participants' feedback, through several open-ended questions, on the main reasons for using the interface customisation add-ons they use, or would like to use if they were available. In general, the responses to these questions can be classified into a number of categories: simplification and consolidation of information, performance advantages and functionality, aesthetics, and enjoyment of the customisation process. These are discussed in more detail below.

Simplification and Consolidation of Information

The most prevalent theme within the participants' feedback was the need to maximise the availability of relevant and important information, while removing or decreasing the focus given to less important information[4]:

> “I try to take a minimalist approach to my customisation, displaying the highest amount of relevant information in the smallest, most unobtrusive space as possible.”

Participants frequently cited the need for critical information to be shown in the centre of the screen, where the player character is shown:

> “One of the primary things that is extremely lacking in the default interface is clustering and localization of important information. Ideally, and this is how I design my interfaces, all the most important information should be near where you are most often looking: your character.”

> “Because the focus of our play experience happens at the center of the screen, where our toon [character] is, the most important information should be relatively close to that area. From there, the information should radiate out from the center as it becomes of less immediate importance. Ultimately, if the information is not necessary during the majority of gameplay, it should be hidden from the screen and accessed through keybind or mouseover.”

Comments on consolidation generally mentioned the need to improve the visibility of “important” information, without discussing what determines the relevance or importance of a particular feature at any given time. As WoW supports character classes of various roles and archetypes, as well as a number of different in-game goals, distinguishing important information from extraneous data may be dependant upon individual player character attributes and goals. This subject was not investigated in this study, but may benefit from further studies in the future.

Performance Advantages and Functionality

A number of performance-focused interface add-ons exist, providing functionality such as the ability to track detailed character damage statistics, monitor effects that assist and harm characters, and provide players with alerts for

[4] Participants’ comments are given in quotes.

scripted events during dungeon encounters. These add-ons can provide players with a competitive advantage, either over other players (if those players choose not to utilise similar add-ons), or over non-player characters within the game:

> “I optimize my interface like I optimize my gear or my talents, because it has a measurable impact on how well I’m capable of playing.”

> “It’s much easier to have an add-on ready to go, instead of having to alt-tab before every flight [to obtain information from outside of the game], or figuring out the best route to take when farming herbs.”

While most participants remarked that add-ons made their role “easier to perform”, feedback from those participants that played healers (i.e. characters that focus on aiding and supporting allies in battle) indicated that they felt that add-ons were necessary in order to perform their role effectively:

> “I’m a healer and must be observing to help find important information which can either save or [fail] the attempt.”

> “I play healers almost exclusively and I need to know Buffs, HOTs, dispellables, and health in full without other distractions.”

Aesthetics

Aesthetic preferences were often informed by or related to other considerations, such as immersion, and practical aspects such as efficiency:

> “Allowing the player to personalize their game-play experience based on their needs (i.e. the type of content they do, the level of play they are at and how much information they require to function, and personal preference regarding look and feel).”

Minimalism and “cleanliness” was repeatedly discussed — “Minimalist UI’s result in greater screen real estate which is important for immersion and spacial awareness.” — as was the removal or modification of default interface elements or textures, which participants described as “ugly” or “unessential clutter”.

Finally, participants expressed a degree of satisfaction with the ability to tailor their interface to their personal preferences:

> "I just want to make it pretty and intuitive."
>
> "Making my screen look the way that I want it specifically."
>
> "Making the game experience my own."

Enjoyment of the Customisation Process

While including some overlap with feedback on aesthetics, enjoyment of the act of customisation was an unexpectedly popular effect discussed by the participants. This supports the ratings discussed in Section 3.3.3. It should be noted that the survey did not differentiate between customisation in the form of players writing their own add-ons or in-game scripts, and a higher-level approach, such as players simply installing and using add-on packages created by other players. The relevant participant comments tended to focus on the satisfaction of creating or building a personalised interface:

> "The customisation for its own sake is a big deal to me. That is, I am quite certain that I could find pre-built interfaces that do what I want better than what I have, but what I have is MINE."

Beyond self-expression, feedback indicated that participants enjoyed frequently updating and redesigning their game interfaces. Presumably, as there are realistic limits on the practical benefits between custom interface configurations, reconfiguring an interface may not always improve its functionality, and may in some cases result in equivalent, or even inferior functionality. Besides cases where players are replacing deprecated add-ons with improved or updated versions, this indicates the strong appeal of novelty (relative to the appeal of improved functionality) when players are considering modifications to their interface.

3.3.5 Interface Customisation and Gameplay

The survey also aimed to find out if interface customisation gave the players any clear game-play advantages. As mentioned earlier, a considerable number of the participants (22%) reported that they use add-ons which they believe can provide them with an unfair in-game advantage. Only 2 participants (1% of all participants) reported that they believe add-ons can provide an unfair advantage to players, and that they did not personally use interface add-ons. These results indicate that the perception of potential issues with fairness does not discourage interface customisation by using add-ons.

One possible explanation for this is the feeling that in order to remain competitive, any available in-game advantage must be utilised. Participants saw add-on usage as another factor influencing success, similar to character itemisation or ability usage:

> "Everyone has the same access to the same add-ons, so a player who chooses to use default raid frames or not run a raid/PVP add-on is only giving themselves a disadvantage. No different than choosing not to use the best gear available to you."

> "Most mods give people an advantage over people who prefer the default interface. I wouldn't call it unfair though, since these mods are readily available for everyone."

Several participants identified a previously available add-on, which they felt provided an unreasonable advantage to players. This add-on, AVR (Augmented Virtual Reality)[5], allowed players to draw lines, circles and other markings onto objects within the game world. This was typically used to identify and communicate locations of interest or importance to other players, in order to aid players with positioning in dungeon encounters. In some of these encounters, correct player positioning was integral to success, and the use of this add-on allowed players to greatly simplify the encounter by reducing their need to communicate directly with other players. The WoW interface API was eventually modified in such a way that AVR was effectively disabled. The participant response to this (which was largely critical of the existence of AVR) indicates that there are limitations to the degree of simplification of game-play which players are willing to accept as reasonable.

3.4 Discussion

Survey results show that there are several motivational factors for playing WoW — supporting findings of previous studies on MMORPGs. Furthermore, this study has shown that the players surveyed have different motivations for

[5] Available from: `http://www.wowace.com/addons/avr/`

customising the game interface of WoW. These are to: simplify and consolidate information, gain performance advantage and functionality, alter aesthetics, or simply enjoy the customisation process. Although these motivations are not mutually exclusive, they have varying levels of importance to different players, with simplification and consolidation of information featuring prominently as a common goal for interface customisation.

In addition the study shows that there are interface customisations that some players would not consider undertaking because this would give them an unfair game-play advantage, despite the fact that these modifications are permitted under the terms of use of WoW. However, the boundaries distinguishing what modifications are acceptable or not are rather subtle.

Of particular importance to the goal of this thesis — to establish whether player-modelled adaptive interfaces can improve MMORPG players' game experience — are insights into motivations for interface customisation. Participant feedback most commonly expressed the importance of simplifying and consolidating information within the interface, allowing players to focus on important and relevant information, and reducing their need to attend to unimportant or irrelevant information. Additionally, motivational factor ratings showed that participants felt that interface customisation was most important for achievement-related factors (advancement, mechanics and competition). This was further supported by feedback indicating that participants customised their interfaces in order to secure a performance advantage and to access additional functionality.

Considered together, these insights suggest that MMORPG players want their interfaces to provide them with more simplified but highly relevant information which gives them with a mechanical game-play advantage. This information can be used to improve their performance in terms of achievement-related aspects of the game — considered to be highly important by players.

3.5 Summary

This study surveyed a number of WoW players in order to identify their opinions on game interface customisation through the use of add-ons. The survey results provide insight into motivations for interface customisation, and suggest ways in which players wish to have their game interfaces improved. The findings of this survey provide a basis for the development of a prototype system which aims to improve player experience through interface adaptation. The development of this system is discussed in Chapter 4.

Chapter 4

WatchAndLearn: An Adaptive Feedback Plug-in for World of Warcraft

Following the survey of WoW players (see Chapter 3), a prototype interface adaptation system was developed. In response to survey participants' feedback, the key goal of this system was to provide players with the most relevant information, to allow them to improve their performance in terms of game-play mechanics. As this system is focused on providing players with personalised interface adaptation, the information presented to each player had to be determined by their own behaviour. Also, in accordance with comments from the survey participants, the information presented to players must had to refrain from introducing unnecessary complexity into the game interface.

To achieve these goals, a prototype WoW plug-in called *WatchAndLearn* was developed.

WoW was chosen for the development of the prototype because, as the world's most-subscribed MMORPG, WoW provides a number features which have become standard within the genre. In particular, the general layout of the WoW interface (see Figure 1.3 and Figure 4.14) has become a commonly used template for other modern MMORPGs. Therefore, it was envisages that interface improvements developed and tested within WoW may be applicable to a wide range of other MMORPGs.

The WoW game client allows for interaction with the interface through an Application Programming Interface (API), which provides a set of Lua[1] func-

[1] `http://www.lua.org/`

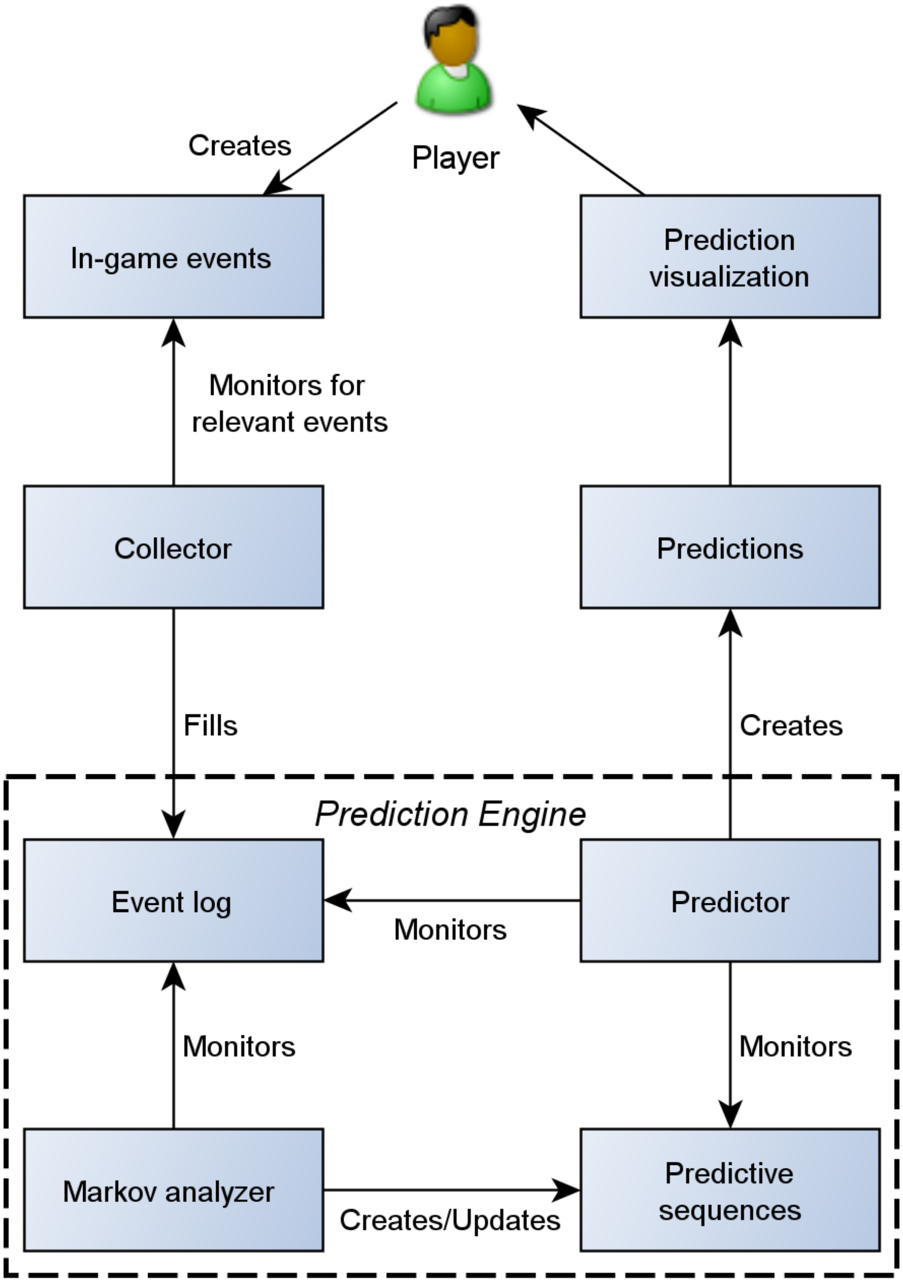


Figure 4.1: An overview of the behaviour of the WatchAndLearn plug-in.

tions and facilities which can be called and utilised by plug-in developers. Additionally, the WoW game client will automatically load correctly configured add-on packages, allowing for community-developed plug-ins to be easily distributed and installed by players. This enables plug-ins to be evaluated by a large number of players without difficulty, making WoW an ideal test-bed for prototype applications.

Figure 4.1 provides an overview of the data flow of the *WatchAndLearn* plug-in, while the following sections in this chapter give a detailed description of the behaviour of each individual system component. In-game actions performed by the player generate various events, which are observed by the plug-in, with relevant events being stored in an event log (see Section 4.1). As this event log is updated, Markov chains are generated or updated, resulting in a number of predictive sequences, with associated outcomes and their estimated probabilities (see Section 4.2). The plug-in compares these generated sequences to the most recent events in the event buffer, in order to estimate the probability of future actions (see Section 4.3). These estimations are presented to the player by displaying predictions within the game interface (see Section 4.4).

Predictions can also be generated using multiple models, as discussed in Section 4.5. Issues involved in determining and utilising model accuracy are discussed in Section 4.6. Finally, the development of the prototype system is summarised in Section 4.7.

4.1 Monitoring Player Behaviour

Characteristics of player behaviour are tracked in two ways: by capturing active ability usage (see Section 4.1.1), and by capturing state information which provides context for player ability usage (see Section 4.1.2).

4.1.1 Event Capturing

The WoW interface is driven entirely by *events*, which are created by the game client and sent to registered interface frames. These events are generated in response to changes within the in-game world, or within the user interface itself. The list of triggered events is continuously growing, with the developers of WoW continually adding additional subsystems and features to the client.

Publicly available API documentation[2] is provided and maintained by various community groups.

Of particular interest for this work is the *UNIT_SPELLCAST_SUCCEEDED* event, triggered when a unit within proximity of the player character successfully uses an in-game ability, capturing the majority of a player's direct input in combat scenarios as well as that of the players and non-player characters within range. Arguments of this event include the name of the spell/ability being cast, and the unique identifier of the unit casting that spell.

A number of events provide information about character interaction within the game. However, while events such as *COMBAT_LOG_EVENT* provide a higher level of detail about combat events (such as spell damage type, amount of damage/healing or amount of damage absorbed), much of this information is superfluous if the intent of data collection is simply to track player behaviour. In short, while it is important to know what action the player elected to take, the outcome of that action is mostly irrelevant. Therefore, the plug-in captures only a subset of available event types, which sufficiently describes player interaction in terms of ability usage. A complete list of all captured events is given in Appendix C.

Upon initial loading, the plug-in registers an event handler for each relevant event type, which capture events as they occur. Each caught event is filtered (removing all events which are not relevant to the player) before being added to an event log, along with time-stamp information. This log persists between play sessions, and is only refreshed or cleared upon explicit instruction, allowing for collection of data over an extended period, which is useful for generating more robust behavioural models (see Section 4.2). Appendix D shows a sample event log, as saved between play sessions.

Additionally, relevant events are added to a buffer which always contains the *n + 1* most recent events, where *n* is the Markov chain pre-sequence length (see Section 4.2.1). The maximum size of this buffer is dependant upon the system configuration — though it is always relatively small (between 1 and 10) — and determines the number of abilities which will be taken into account when generating predictions (see Section 4.2.2).

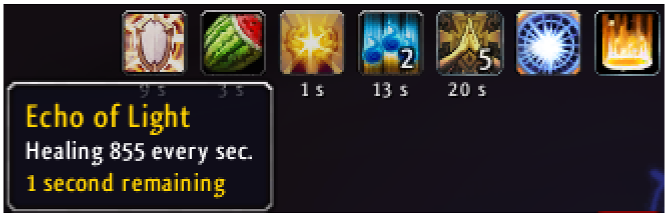
[2] `http://www.wowwiki.com/Event_API`

4.1.2 State Tracking

In addition to combat information, the plug-in collects and logs a selection of information describing the current state of the player's character and the environment around them. The characteristics of this data and the process of logging this information is described below.

Player Buffs

In WoW, player characters may be affected by *buffs* — temporary spells or effects which positively impact attributes and characteristics of the character. These can take the form of statistical boosts, shields (reducing damage taken), boosts for particular player abilities, and a variety of other positive effects. Buffs can be triggered by items or character specialisations, environmental effects, or cast by a player or non-player character. For example, characters of the Priest class may cast the spell *Inner Focus*, granting their character a buff which "Reduces the mana[3] cost of your next Flash Heal, Greater Heal or Prayer of Healing by 25% and increases its critical effect chance by 100%"[4]. By default, a list of icons denoting currently active buffs is displayed in the upper right corner of the interface (see Figure 4.2).



Echo of Light
Healing 855 every sec.
1 second remaining

Figure 4.2: An example of active buffs in WoW.

Buffs can have a profound impact on player behaviour, by drastically increasing damage/healing output of particular abilities, therefore encouraging the usage of those abilities. Additionally, many buffs have a chance to occur randomly (usually with a percentage chance to apply if certain conditions are satisfied). These are generally referred to by players as *procs*, or programmed random occurrences. This results in a need for players to pay close attention to which buffs are currently active, so that they may adjust their current strategy

[3] In WoW, *mana* is a limited resource required in order to use some abilities.

[4] `http://www.wowhead.com/spell=89485`

in order to maximise the benefit of any active effects. This generally manifests as a disruption of otherwise predictable patterns of ability usage. For this reason, it is useful to take currently active buffs into account when estimating likely player actions.

This is achieved by capturing *UNIT_AURA* events, which are triggered by the game whenever a unit has either gained a new buff, lost a buff, or had a buff updated (for example, when the buff is refreshed).

While *UNIT_AURA* events occur regularly, the plug-in only aims to track buffs in relation to player actions. For this reason, the player's active buffs are only recorded when the player has performed an action, indicated by an update to the player's event buffer (see Section 4.1.1). At this point the player's "buff log" — representing all tracked buff data — is updated.

Player buffs are stored in a nested dictionary structure, with ability names as keys and lists of buffs and their observation counts as values. When the player performs an action, the name of that action is used to look up the dictionary and retrieve a list of records, one for each distinct buff which has been previously observed when the key action has occurred. For each currently active buff, the corresponding entry in the record list has its observation count incremented, with a new record created if that buff has not been observed previously. Observation counts are only ever incremented — if a buff is found in the record list but is not currently active, no action is taken.

The resulting buff log provides a running tally of the number of times each buff has been active when a given ability was used. Given the name of an ability, a list of previously observed buffs and observation counts can be retrieved.

Player and Target State

In addition to buffs or active effects, player behaviour can be impacted by the state of the player's character and other characters within the game world. In particular, in combat situations the current state of the player and their target can determine ways in which the player will react. For example, a target with low health may be vulnerable and easily beaten, while a target which is extremely high level or has particularly strong active buffs may be difficult for the player to conquer. Similarly, if a player's character which is dependant upon mana has very little mana left, engaging in combat may be dangerous or unwise.

Figure 4.3 shows player and target unit frames, communicating a num-

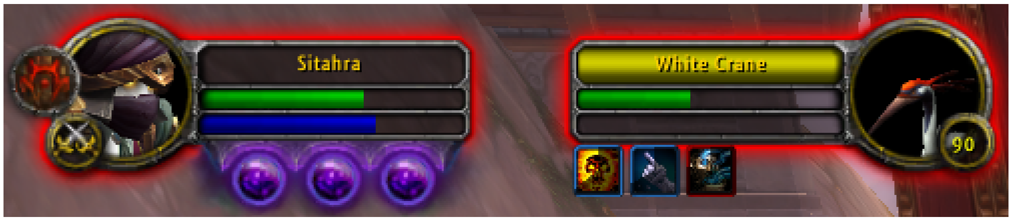


Figure 4.3: Unit frames showing player and target health (left and right green bars, respectively), and player mana (left blue bar).

ber of state factors. State factors can have a significant impact on game-play, and can be used by players to determine optimal courses of action. For example, characters of the Warrior class may use the ability *Execute*[5], which deals a large amount of damage, but is only usable on enemies that have less than 20% health. For this reason it may be worth taking some state factors into account when modelling player behaviour.

WatchAndLearn monitors a small selection of factors which can be used to express the state of the player's character, as well as the player's current target. These include:

- Target health level
- Target reaction (friendly/unfriendly/neutral)
- Player health level
- Player combat status (in combat/out of combat)
- Player group status (solo/small party/large party or raid)

Similarly to active effects, these factors are tracked in terms of their current state whenever an ability is used, by mapping the ability name to observation counts for individual factors. This state dictionary can then be queried by using an ability name as the key, in order to retrieve the total number of observation counts for each particular state factor.

4.2 Player Modelling

This section discusses the modelling of player behaviour for the purpose of prediction generation. Section 4.2.1 discusses the application of Markov chains for

[5] http://www.wowhead.com/spell=5308

generating predictions based upon historical data, while Section 4.2.2 explains how this technique is used to model player behaviour. Section 4.2.3 discusses the application of temporal filtering to improve player model accuracy. Finally, Section 4.2.4 discusses additional considerations which must be taken into account when generating player models.

4.2.1 Modelling With Markov Chains

Markov chains (Markov, 1971) provide a means of predicting the occurrence of events given an observed history. Figure 4.4 shows an event history sequence, where three distinct events (*A*, *B*, and *C*) are observed.

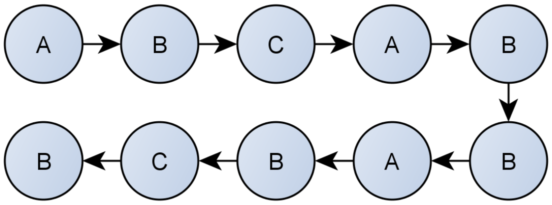


Figure 4.4: An event history sequence, used to generate Markov chains.

These historical sequences can be observed, with a Markov chain created to represent the probabilities that any given event will follow any other event. Figure 4.5 shows a Markov chain representing the probabilities for individual events, based upon the observation of Figure 4.4, and expressed as a directed graph. As *A* was followed by *B* in every case, the probability of *A* suggesting *B* is given as 1.0, while there is no link (expressing a probability of 0.0) between *A* and either itself or *C*. For any event in the chain, the sum of probabilities for all following events will always equal 1.0.

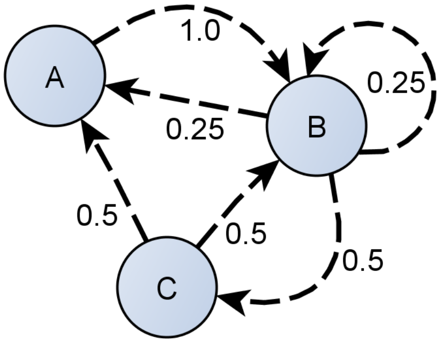


Figure 4.5: A three-state Markov chain.

Markov chains can also be used to model the probability of an event given an ordered sequence of preceding events. Figure 4.6 shows a Markov chain which expresses the probability of possible outcomes following the "pre-sequence" $\{A, B\}$, based upon the event history observed in Figure 4.4. This shows that, given the pre-sequence $\{A, B\}$, based upon observed history, there is a 66% probability that the event *C* will occur, a 33% probability that the event *B* will occur, and a 0% probability that the event *A* will occur. Figure 4.7 shows a similar Markov chain, but for a pre-sequence of three events $\{A, B, C\}$. Again, given a pre-sequence, the sum of probabilities for all following events will always equal 1.0.

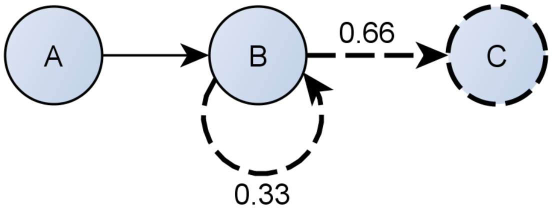


Figure 4.6: A Markov chain showing outcomes for the sequence $\{A, B\}$.

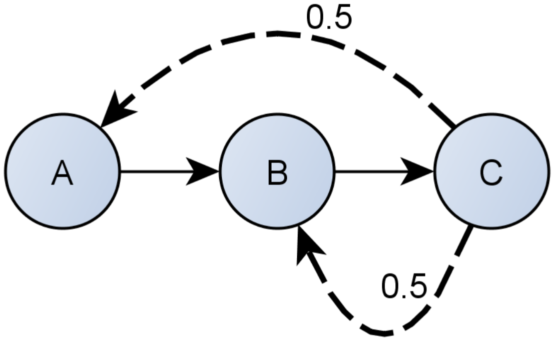
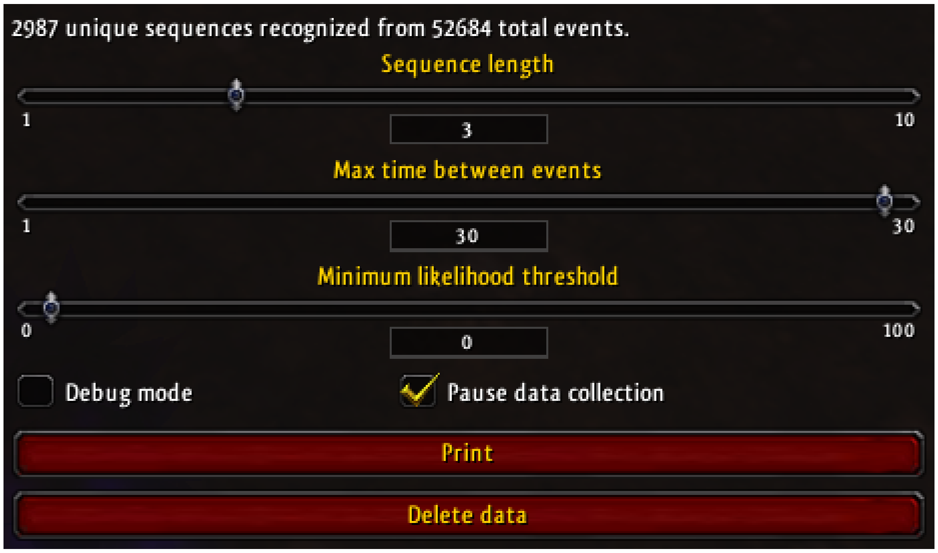


Figure 4.7: A Markov chain showing outcomes for the sequence $\{A, B, C\}$.

This only expresses observed outcomes for a single pre-sequence permutation, ignoring other permutations which are either present within the event history, or potentially possible, but unobserved. As more event types are included, the number of possible permutations increases. For *n* distinct events $(E_1, E_2, E_3, ...E_n)$, there are n^2 possible pre-sequence permutations, though the actual permutations requiring modelling are dependant upon observed history.


2987 unique sequences recognized from 52684 total events.

Sequence length: 3

Max time between events: 30

Minimum likelihood threshold: 0

□ Debug mode

☑ Pause data collection

Figure 4.8: The *WatchAndLearn* in-game configuration interface for general configuration settings.

4.2.2 Modelling Player Actions

Player ability usage is modelled by generating Markov chains of observed event history. The process is automated, allowing for chains to be updated regularly without requiring any additional player action. The key data source for this process is the event buffer, which contains the $n + 1$ most recent relevant events, where n is the pre-defined length of Markov chain pre-sequences. Players can configure the Markov chain pre-sequence length, as well as a variety of other settings affecting prediction using the in-game configuration interface (see Figure 4.8).

On every interface update cycle (once per frame drawn to the screen), the plug-in checks whether the contents of the event buffer have changed. If a change is observed, the most recent event is taken as the "outcome", with all n preceding events in the buffer being taken as the pre-sequence chain resulting in that outcome. This process is shown in Figure 4.9.

With a pre-sequence chain now identified, the model dictionary is queried to determine whether that chain has been previously observed. The queried dictionary contains a collection of pre-sequences as keys, and lists of outcome-count pairs as values (where the count is the number of times this outcome has been observed following the key pre-sequence), as shown in Figure 4.10.

If the key chain is not present in the dictionary, it is inserted, with an initial value containing only the observed outcome, with a count of 1. If the key

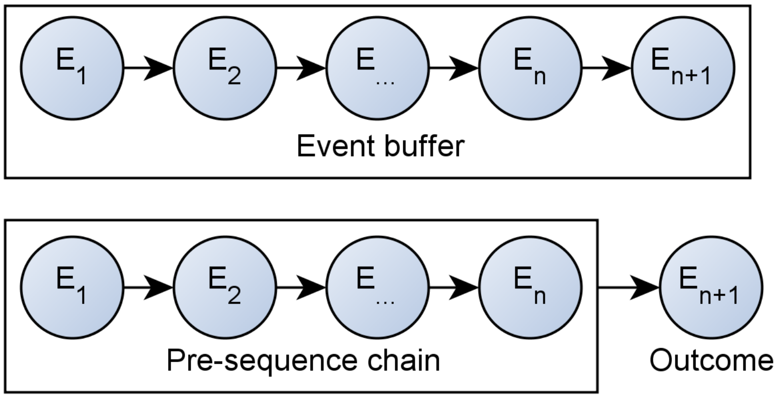


Figure 4.9: An event buffer containing $n + 1$ events, rearranged into a Markov chain pre-sequence of length n and outcome.

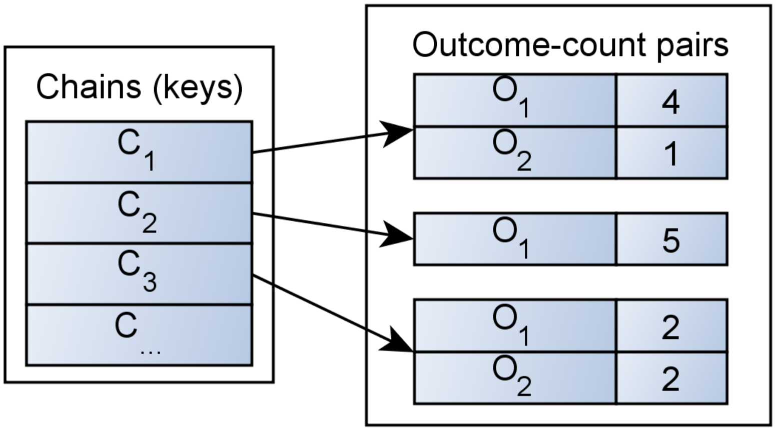


Figure 4.10: Chains and outcomes are stored in a dictionary, where the presequence chain is the key, and the outcome and frequency count of that outcome is the value.

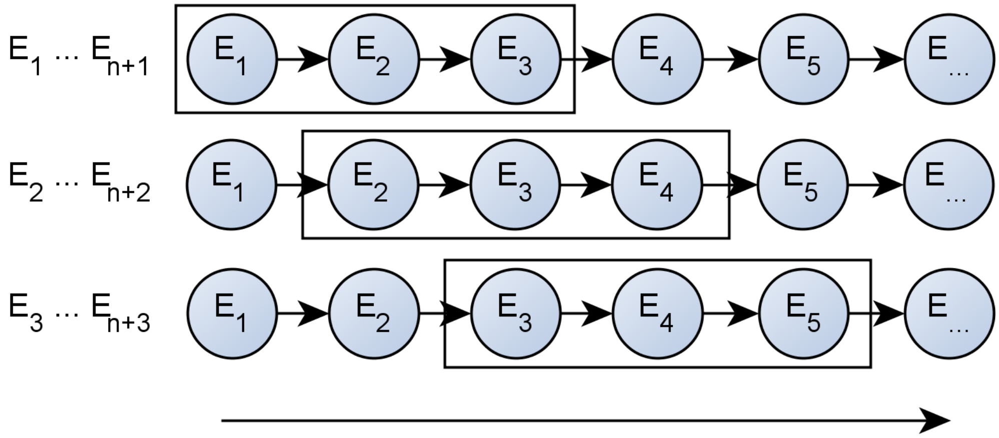


Figure 4.11: A sliding selection of $n + 1$ events from a complete event log. An example of selections for $n = 2$ is shown.

is found, the value list is checked to see whether the outcome has been previously observed. If a matching value is found, the count for that outcome is incremented. If no matching value is found, the outcome is inserted into the outcome list, with a count initialised to 1.

The modelling process runs continuously, monitoring the event buffer for changes and updating Markov chains as necessary. Additionally, the player model can be completely refreshed whenever a significant change to configuration settings occurs. For example, modifying the pre-sequence chain length triggers a full refresh. This process is similar to the on-going modelling process, but uses the complete event history saved to the event log (containing all observed events, persisting between play sessions) as a data source, instead of the event buffer which contains only the most recent events.

When generating a model using the entire event log, sub-sequences of length $n + 1$ (where n is the pre-sequence chain length) within the event log are analysed individually, by making a "sliding selection" of events from the event log, beginning with events E_1 through E_{n+1}, followed by events E_2 through E_{n+2}, and so forth. This selection process is visualised in Figure 4.11. After this, the standard process of identifying chains and outcomes is performed.

4.2.3 Temporal Filtering

As shown in Figure 4.8, players can elect to set a maximum time period between events that will be considered for analysis. This allows for sequences of events which occur over extended periods of time — with significant periods of inactivity in between — to be ignored, preventing patterns from being incorrectly identified. In particular, long periods of inaction — where the player may be taking a break, or otherwise occupied — will also result in a break in terms of model generation.

Where the event buffer is found to contain two consecutive events with timestamps which differ by a value exceeding the specified period, all events which occurred prior to the latter event are flushed from the buffer. The result is that no chain pre-sequences will ever be created where there is a delay of more than the allowed time period between two consecutive events. This setting only applies to consecutive events.

4.2.4 Additional Considerations

Two key issues impacting accuracy arise when using Markov chains to model player behaviour: total number of observations, and Markov chain pre-sequence length.

Total Number of Observations

As more events are observed, the robustness of any generated model increases as uncommon outliers are suppressed by more commonly occurring outcomes. Because of this, models that are generated based upon a small number of actions will generally be less accurate than models generated using a more substantial event history. At early stages in model generation (with few source events), it is impossible to establish which outcomes are potential outliers. Events which occur rarely within limited event logs may in fact be common outcomes. Because of this, total number of event observations should be taken into consideration when estimating the potential accuracy of a generated model.

Markov Chain Pre-sequence Length

As chain pre-sequence length increases, the total number of unique pre-sequence chains will increase, and the total number of predicted outcomes for each of those chains will — in general — decrease. Because of this, it is expected that longer pre-sequence chains will result in fewer but more accurate predictions. While increased accuracy is desirable, for shorter event logs increasing the pre-sequence length may reduce the number of predictions to the point that the model fails to predict any outcome at all.

The trade-off in this case is between fewer and more accurate predictions (with the risk of having no predictions made at all), and more (less accurate) predictions. In cases where a large number of events have been observed, longer pre-sequence chains may be used to generate more accurate predictions without running the risk of failing to predict any outcome. The ideal chain pre-sequence length will ultimately be determined by the size of the source event log.

4.3 Prediction Generation

With a player's game-play habits observed and recorded, the player model can then be used to generate predictions of future behaviour.

The prediction generation process is automatically triggered whenever the contents of the event buffer have changed (indicating that the player has recently performed a relevant action). The contents of the event buffer are inspected, with the most recent n events (where n is the Markov chain pre-sequence length) are selected. These events are used to form a chain pre-sequence, which is then used as the key for a look-up in the model dictionary (see Section 4.2). The retrieved value is a list of all abilities that have been previously observed following the key chain sequence, along with their observation counts.

In order to calculate the estimated likelihood for each prediction, the number of observed occurrences (for all outcomes) is summed. The estimated likelihood of each outcome is then calculated as the observation count for that outcome, as a proportion of the sum of all observation counts. The result is a set of predicted outcomes with predicted likelihoods ranging between 0.0 and 1.0, with the sum of all likelihood estimations being 1.0. This is shown in Figure 4.12.

With predicted likelihood values calculated, outcomes can also be ranked according to the order of estimated likelihood. Methods of displaying estimations to the player are discussed in Section 4.4.

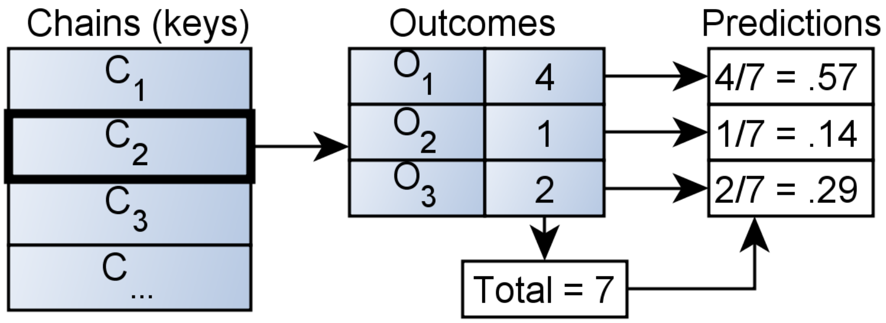


Figure 4.12: The process of identifying a chain and all outcomes, then calculating predicted likelihood.

4.3.1 Augmenting Prediction Likelihood Estimations

While Markov chains are used to identify likely outcomes and base-line estimations for the likelihood of those outcomes, these estimations can be augmented by considering characteristics of other collected data, such as currently active buffs and state information. The goal is to utilise this additional information in order to improve the accuracy of estimations of predicted likelihood.

Historically observed buff and state data are applied to likelihood estimations in a similar manner.

Estimation Using Buff History

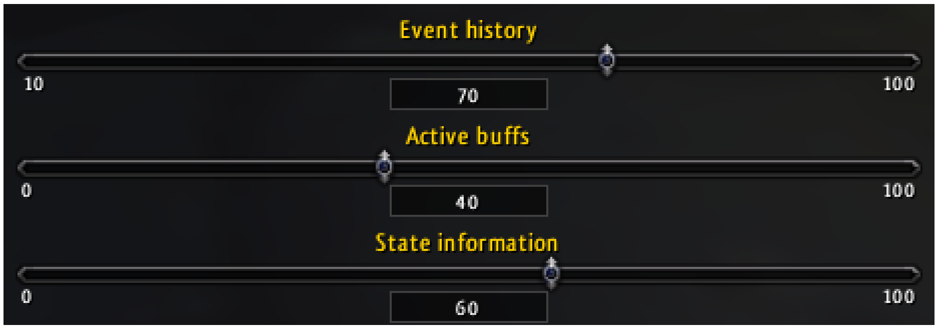
Using the name of the predicted ability as a key, the buff history dictionary is queried in order to retrieve a record containing all buffs that had been previously observed (and their observation counts) when the key ability was used. This list is compared with the currently active player buffs, with the total observation counts for all buffs which are also currently active being summed up. The number of observed occurrences for all buffs is also summed. The raw outcome estimation is then calculated as the sum of observation counts for currently active buffs as a proportion of the sum of observation counts for all buffs.

Estimation Using State History

Similarly to buff history estimation, the name of an ability predicted according to event history is used to query the state history dictionary, retrieving a record which contains occurrence counts for pre-determined state factors. The similarity between this retrieved record and the current player and target state is calculated, resulting in a value between 0.0 and 1.0, where 0.0 indicates no similarity between the current state and any historically observed states for the key ability, and 1.0 indicates complete parity between the current state and all historically observed states for the key ability.

Applying Augmented Likelihood Estimation

WatchAndLearn allows the player to configure the relative weightings of event history, active buff history, and state history for prediction likelihood estimations. Figure 4.13 shows the in-game configuration interface allowing for adjustment of these weightings. As buff and state-based history is used to modify likelihood estimations only (and not predict outcomes), the minimum weighting for event history is greater than zero.



Event history

10

70

100

Active buffs

0

40

100

State information

0

60

100

Figure 4.13: The *WatchAndLearn* in-game configuration interface for relative weightings of event history, active buffs and state information.

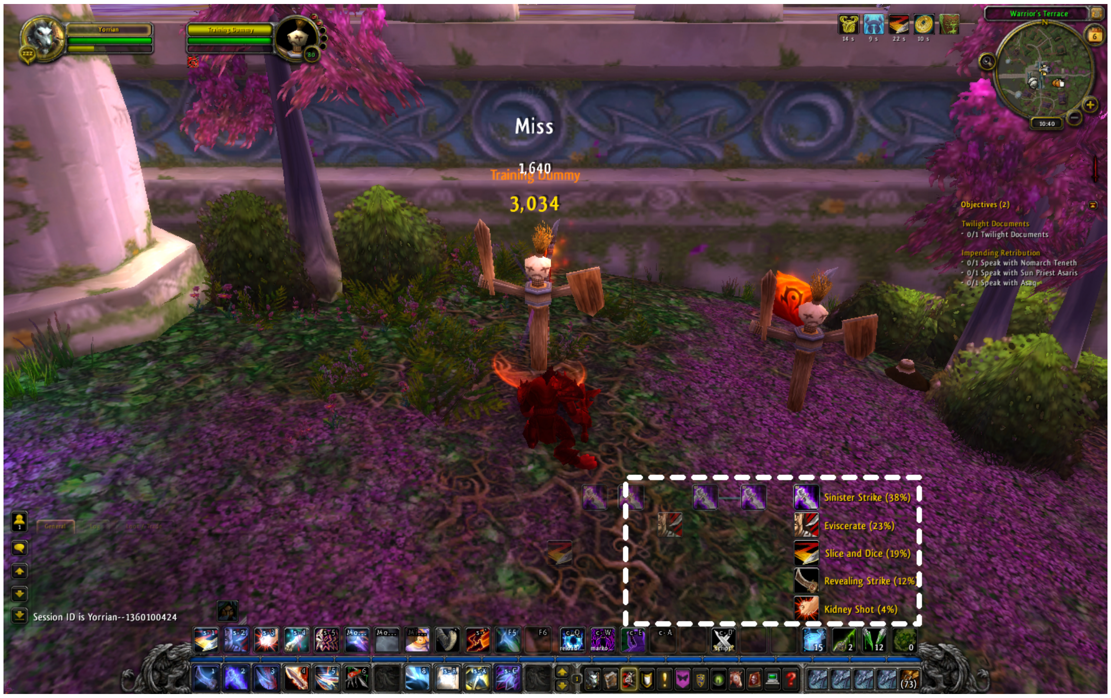
The following equation shows the process of calculating an predicted likelihood P for an outcome O, where E is the initial estimation based upon event history, B is estimation based upon buff history, S is estimation based upon state history, and W is the weighting for each factor:

$$P_O = \frac{E_O \times W_E + B_O \times W_B + S_O \times W_S}{W_E + W_B + W_S}$$

4.4 Prediction Visualisation

In order to provide players with an unobtrusive and intuitive visualisation of ability predictions, the decision was made to augment the existing interface by adding an additional prediction visualisation panel, rather than by modifying existing interface elements. This allows players to immediately access information as needed, without needlessly complicating existing interface elements by adding information which may not always be needed. The visualisation can be seen in context in Figure 4.14, with the highlighted area shown in more detail in Figure 4.15.

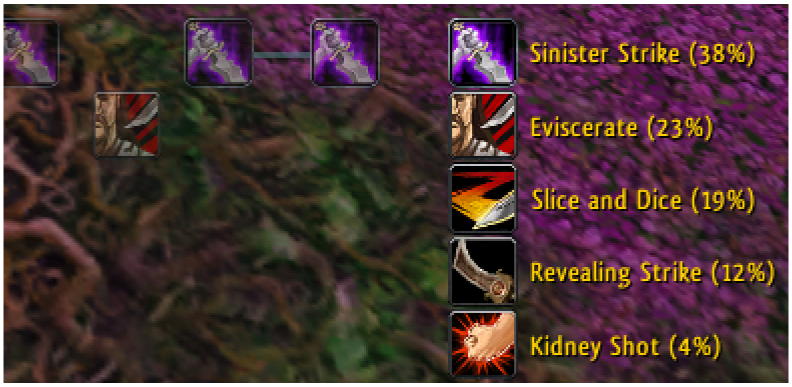
Predictions are displayed as a ranked list of abilities, with the ability icon, ability name, and predicted likelihood listed. The predicted abilities are ranked in descending order of estimated likelihood. Since it is possible that a large



Session ID is Yorrian--1360100424

Sinister Strike (38%)
Eviscerate (23%)
Slice and Dice (19%)
Revealing Strike (12%)
Kidney Shot (4%)

Figure 4.14: The default WoW interface, with the *WatchAndLearn* prediction visualisation activated (highlighted in the lower right of the screen).



Sinister Strike (38%)
Eviscerate (23%)
Slice and Dice (19%)
Revealing Strike (12%)
Kidney Shot (4%)

Figure 4.15: The *WatchAndLearn* prediction visualisation.

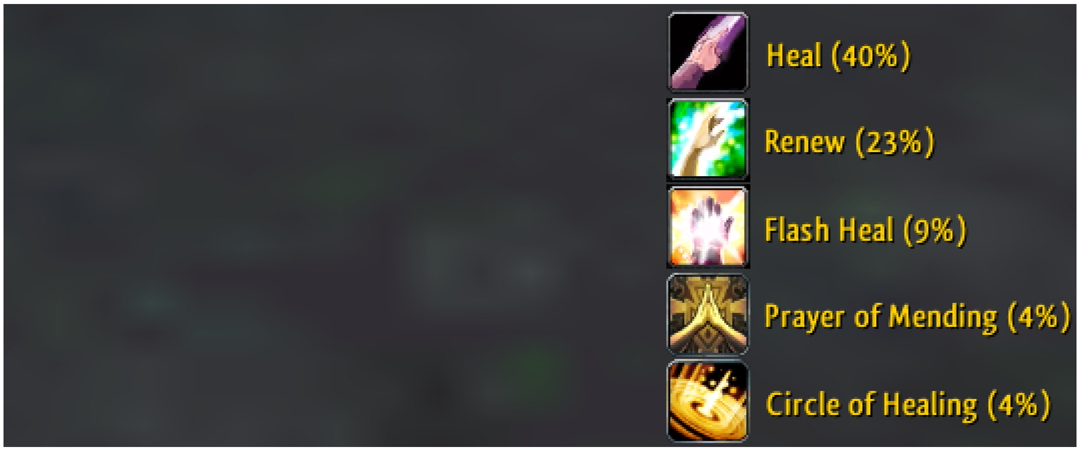


Figure 4.16: A list of predicted abilities and their estimated likelihoods.

number of different abilities may be predicted at any time, the total number of predictions shown to the player at a given time is capped, with the default cap value being set to 5 predicted abilities. This value can be adjusted in the in-game settings configuration interface (see Figure 4.21). If too many abilities are predicted, the lowest ranked abilities will be removed from the displayed list.

The prediction list is populated and updated as predictions occur (see Section 4.3). As a result, by default the prediction list will be empty, with the list only being populated once a number of actions have been observed (as determined by the pre-configured length of Markov chain pre-sequences). Once enough events have occurred, the list will be populated with any predicted abilities. Figure 4.16 shows the prediction list after initially being populated with five predicted abilities.

Since the number of displayed abilities is capped, it is possible that some predictions will not be shown to the player. Figure 4.16 shows that the total predicted likelihood for all abilities (shown in parentheses) is less than 100%, indicating that other abilities have been predicted, but are not displayed in the prediction visualisation.

In addition to showing ability predictions, the visualisation aims to illustrate to the player what ability pattern resulted in the current set of predictions. Once the player selects a displayed ability, the icon for that ability begins to slowly 'float' leftward across the screen. This is shown in Figure 4.17, where the player has used the highest ranked ability, *Heal*. In this case, the usage of this ability has not significantly altered predicted outcomes, so the predicted

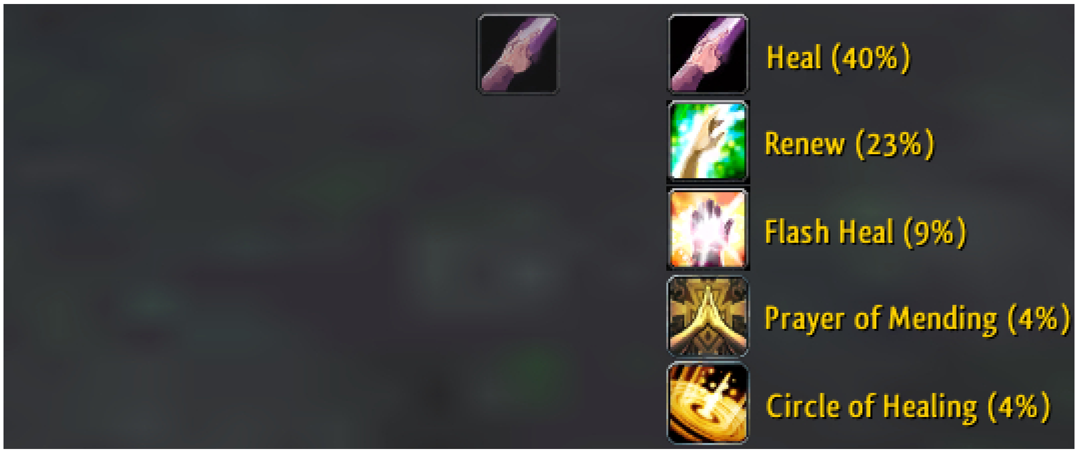


Figure 4.17: A list of predicted abilities, and the most recently selected ability leading to these predictions.

list remains similar to the initial predicted list shown in Figure 4.16.

As the player continues to use abilities, the visualisation continues to 'float' the appropriate icon across the screen, creating a short history visualisation allowing for identification of the patterns recognised by the plug-in. In the case that the player uses an ability which is not contained in the predicted list, all predictions temporarily disappear (until the prediction model begins to recognise familiar sequences again).

While all historically used abilities (that were previously shown in the prediction list) are shown for a short amount of time, only the most recent abilities are used for prediction. The actual number of abilities used depends on the pre-configured Markov chain pre-sequence size. Sequences of abilities which were used as pre-sequences for Markov chain prediction are shown with lines drawn between those abilities (see Figure 4.18).

Since player ability usage also results in constant re-analysis of Markov sequences, the predicted abilities and their likelihoods are also subject to change. For example, Figure 4.18 shows that repeated usage of the *Heal* ability has resulted in a higher predicted likelihood for that ability than was previously shown in Figure 4.17 (an increase from 40% to 41%).

As the icons of abilities used previously continue to float further away from the visualisation, their opacity is gradually reduced until they are eventually removed from the interface (see Figure 4.19).

Unlike previous figures, Figure 4.19 shows a case where the player has selected the ability *Renew* instead of *Heal*. Since *Renew* was ranked second in the

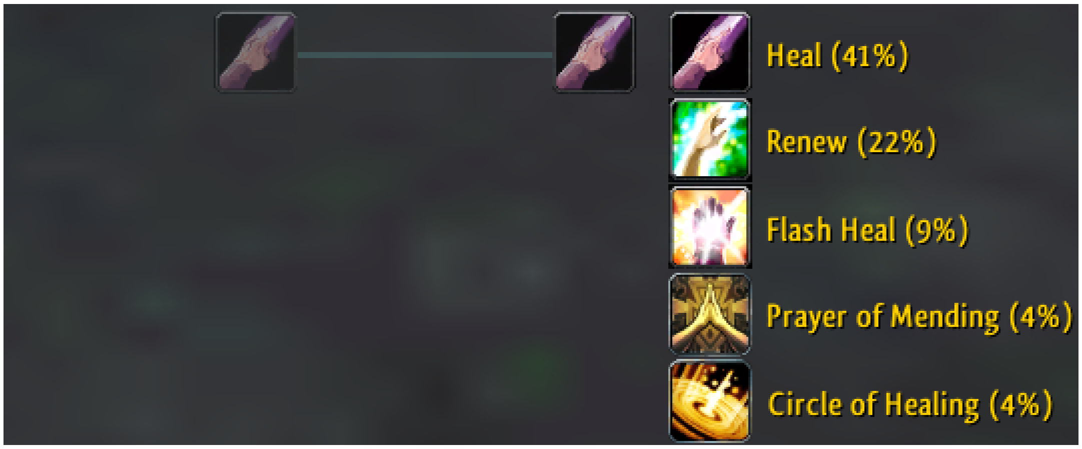

Heal (41%)
Renew (22%)
Flash Heal (9%)
Prayer of Mending (4%)
Circle of Healing (4%)

Figure 4.18: Links are drawn between historically used abilities which have resulted in the current prediction list.

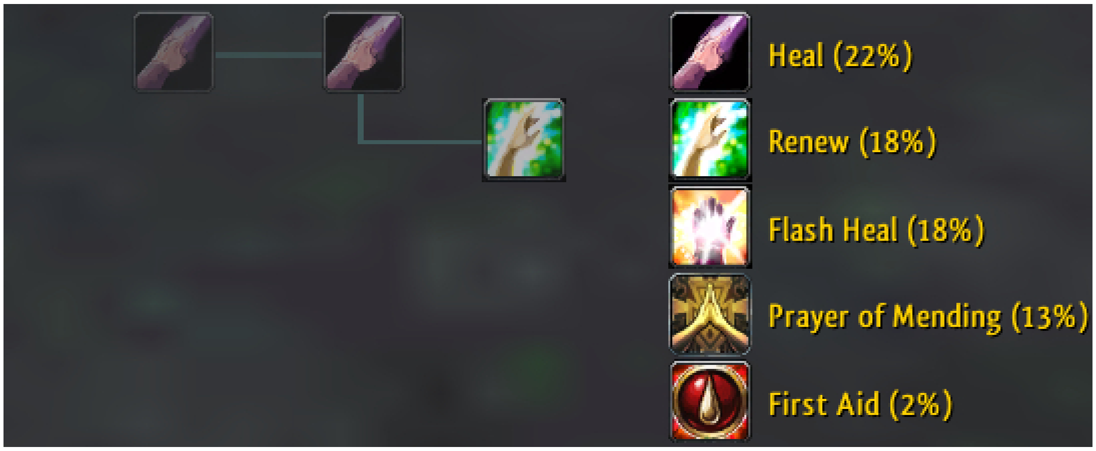

Heal (22%)
Renew (18%)
Flash Heal (18%)
Prayer of Mending (13%)
First Aid (2%)

Figure 4.19: Historically used abilities drift leftward and fade as the time since they were used increases.

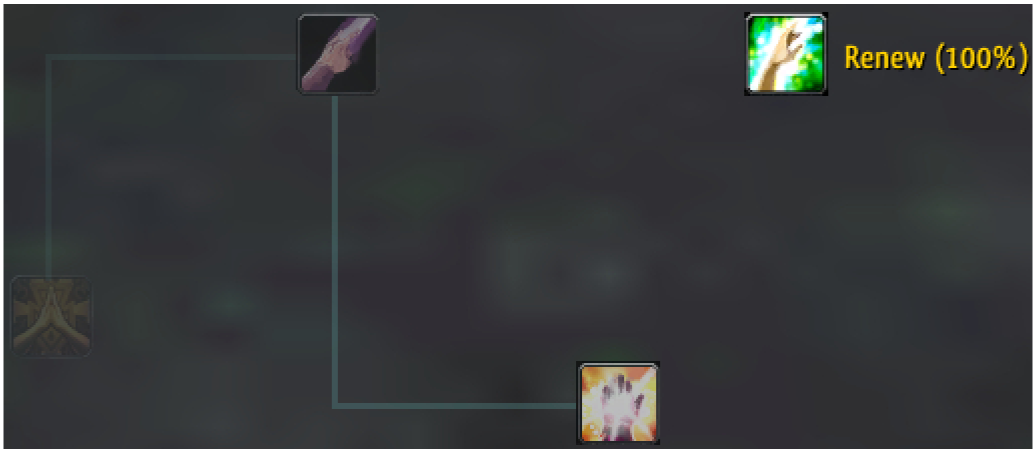


Figure 4.20: Some pre-sequence combinations may result in very few predicted outcomes.

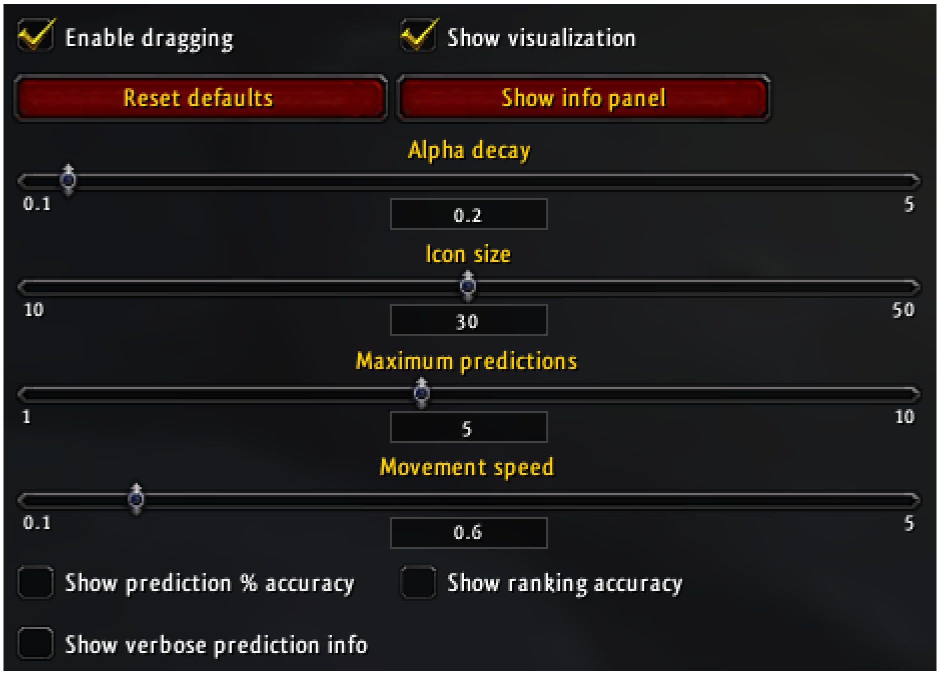
previous prediction list, this is reflected in the floating history icons. Additionally, this change in ability pattern (from {*Heal, Heal, Heal*} to {*Heal, Heal, Renew*}), has resulted in a different prediction list, with different estimations for predicted abilities.

Figure 4.20 shows an example of a pre-sequence involving abilities with varied predicted likelihoods (with the first and third abilities in the sequence being ranked fourth and fifth in the predicted list, respectively). In this case, the predictive model has only suggested a single outcome, with a predicted likelihood of 100% — indicating that following this pre-sequence, the only outcome that has been previously observed is *Renew*.

A number of configuration options are provided to users, allowing players to modify characteristics of the visualisation including: the maximum number of displayed predictions, ability icon size, and the rate at which selected abilities move and fade. The in-game configuration interface for these settings is shown in Figure 4.21.

4.4.1 Model Summary

In addition to providing players with a dynamic visualisation of predicted abilities, a tool allowing for visualisation of the current predictive model is provided. This shows all known pre-sequences, and allows the viewer to see all predicted outcomes, their predicted likelihood, and the total number of times those outcomes have been observed. Players are able to access this visualisation — and all other plug-in configuration options — from within WoW's standard



☑ Enable dragging ☑ Show visualization

Alpha decay
0.1 5
0.2

Icon size
10 50
30

Maximum predictions
1 10
5

Movement speed
0.1 5
0.6

☐ Show prediction % accuracy ☐ Show ranking accuracy

☐ Show verbose prediction info

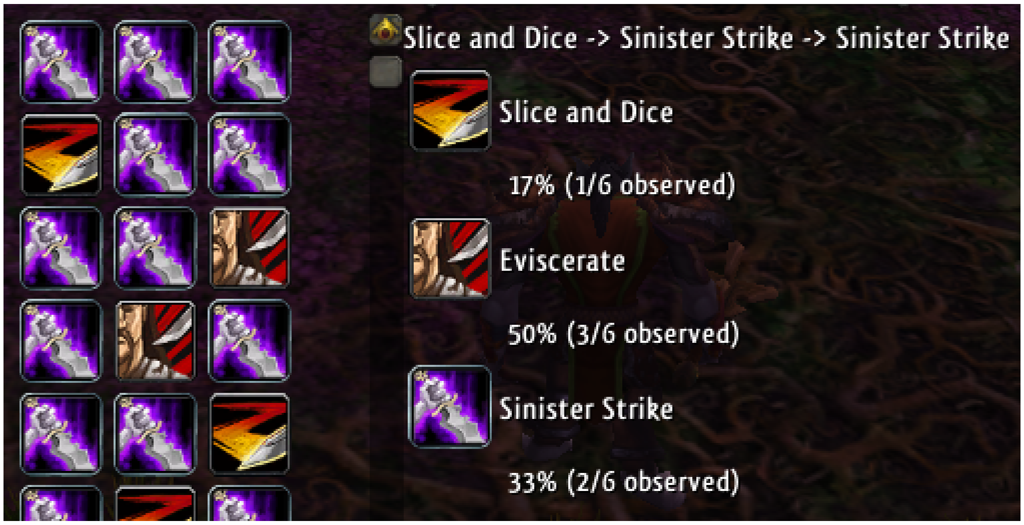
Figure 4.21: The *WatchAndLearn* in-game configuration interface box for visualisation options.

plug-in options interface. A snapshot of the model visualisation is shown in Figure 4.22.

4.5 Model Subscriptions

WatchAndLearn also provides a model subscription component, allowing players to generate predictive models based upon other players behaviour. As well as supporting single-model prediction based upon an external source, models can be generated from a combination of an arbitrary number of sources, including external sources as well as the players own historical data. This allows the player to augment their own predictive data in different ways, by applying different model combinations, or by completely replacing predictions with those based upon other players' behaviour.

Model subscriptions can be used as a training tool, providing novice players with suggestions based upon expert player behaviour. Similarly, more advanced players that are inexperienced with specific play-styles, or are curious about other advanced players' approaches to game-play may benefit from being able to practice ability rotations while using predictions based upon model



Slice and Dice -> Sinister Strike -> Sinister Strike

Slice and Dice

17% (1/6 observed)

Eviscerate

50% (3/6 observed)

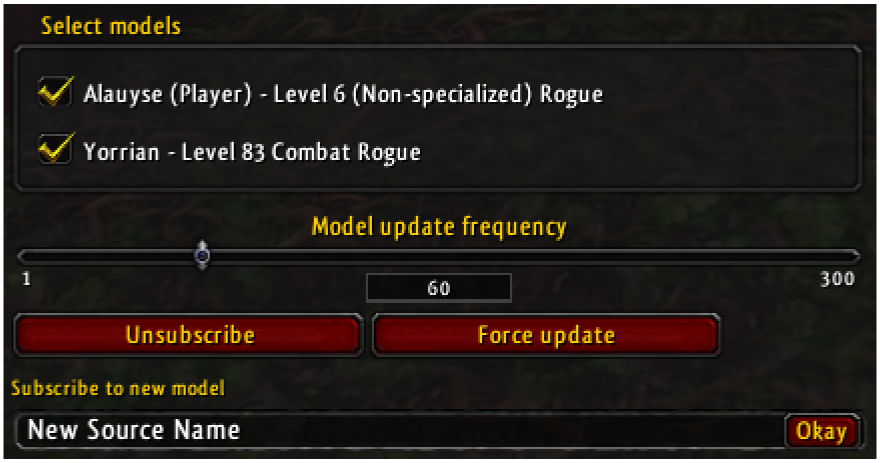
Sinister Strike

33% (2/6 observed)

Figure 4.22: A snapshot of the model summary interface. Pre-sequence patterns are listed to the left, with observed outcomes for the selected pattern listed to the right.

subscriptions for suggestions.

The in-game configuration interface for model subscriptions is shown in Figure 4.23. In this example, the current player, *Alauyse*, has subscribed to another player, *Yorrian*, and will receive event log updates every 60 seconds. With both models activated, predictions will be made based upon a combination of both characters' event history.



Select models

☑ Alauyse (Player) - Level 6 (Non-specialized) Rogue

☑ Yorrian - Level 83 Combat Rogue

Model update frequency

1 — 60 — 300

Figure 4.23: The model subscription in-game configuration interface.

4.5.1 Subscription and Update Process

Subscriptions are managed and maintained using the add-on communication channels of WoW, which allow for communication between different game clients running the same game plug-in. Messages are sent directly between the subscriber and the subscription source, with no communication data being accessible to other players.

Establishing a Subscription

A subscription is initiated when a player uses the in-game plug-in configuration interface (see Figure 4.23) to request a subscription to a specific player. If the specified player has the *WatchAndLearn* plug-in activated, they respond by sending the subscriber a packet containing identifying characteristics of the player's current character, including their character class, level, and specialisation information. The subscriber logs this information, and creates an empty entry in the event log under the name of the subscription source, which will be used to store future updates from the source.

Once a subscription is established, updates are continually requested, regardless of whether the subscription is currently active. This allows for a subscriber to freely alternate between different model subscription combinations, which will always be up-to-date.

The subscription source itself does not keep track of its currently active subscribers, making it impossible for a source to 'drop' a subscriber. However, players may elect to prevent any responses being made to update requests from subscribers.

Update Requests and Responses

Once a subscription has been established, updates are requested by the subscriber at a regular interval determined by configuration settings (see Figure 4.23), ranging between 1 and 300 seconds. When the update interval has elapsed, the plug-in cycles through all currently active (on-line) subscription sources, and requests an update from each.

Reducing the update interval to low values (closer to 1 second) will approximate real-time updating, allowing for the subscriber to receive event updates shortly after those events occur. Increasing the update interval will result in a longer delay between source events and reception of those events by a subscriber, but will also reduce the amount of processing required due to a reduced overhead of update requests and responses.

In order to minimise the amount of processing required when sending an update request, the subscriber must submit the time-stamp of the last received update (with a default of 0 in the case of a subscriber source which has not yet sent an update). Upon reception of an update request, the subscription source creates a result list containing all event log entries that have occurred since the last update. This is achieved by beginning with the latest event, and iterating backward through the event log until the last update time-stamp exceeds event log entry time-stamps. The resulting list of events is then sent to the subscriber as a serialised text stream.

After receiving and de-serialising the text stream containing the latest events from the subscription source, the subscriber appends these to the previously created event log for the subscription source model. If the subscription source being updated is also currently being used for ability prediction, the prediction process is immediately triggered, in order to force a full update of predictive sequences for the currently active model (see Section 4.5.2).

The process of subscription update requests and responses is shown in Figure 4.24.

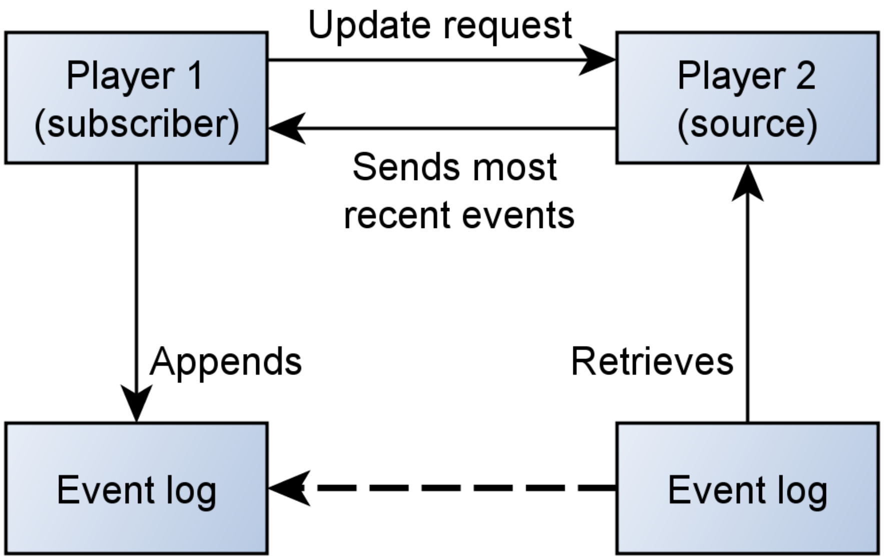


Figure 4.24: The automated process of updating a subscribers event log by requesting the contents of a source event log.

4.5.2 Prediction Using Multiple Sources

Prediction with multiple active subscriptions is handled very similarly to standard prediction using a single player event log (see Section 4.3). When the player's event buffer is updated, the prediction cycle is triggered, resulting in the creation or update of Markov chains based upon event log data. However, in the case of multiple source prediction, the event log used is a composite log containing events from all currently active sources. Depending upon the configuration, this could be a single external source, multiple external sources, or a combination of external sources and the player's own event log. With this composite event log created, analysis for prediction generation is identical to the standard single-source prediction process.

Predictions are specific to particular source combinations, meaning that predictions must be entirely regenerated whenever the active model selection is changed. For example, if a player has three currently active sources, and deactivates one of those sources, all predictions must be regenerated in order to prevent the deactivated source from being used for prediction.

4.6 Model Accuracy

The accuracy of the generated predictive model can be calculated retroactively by comparing predicted outcomes with actual observed outcomes (i.e. actions taken by the player). This provides a means of evaluating the appropriateness of the applied model, and possibly identifying potential improvements to the modelling approach. Sections 4.6.1 and 4.6.2 discuss two methods of estimating model accuracy based upon observed behaviour.

4.6.1 Ranking Accuracy

Ranking accuracy describes the accuracy of the ordering of predicted outcomes. This can be determined using a simple calculation where a set number n is divided by the summed ranked predicted likelihood for the previous n abilities used. The result of this calculation is a number between 0.0 and 1.0, with 0.0 indicating total inaccuracy, and 1.0 indicating total accuracy for the last n predictions. This metric is simplistic in that it does not take into account the actual predicted likelihood of any event — rather, it simply measures the accuracy of the ranked ordering.

For example, consider the case where two abilities *A* and *B* are predicted to occur, with 92% and 8% likelihood respectively. Three situations may arise: neither event occurs, event *A* occurs, or event *B* occurs. However, should neither event occur, the accuracy of the prediction for this single event is 0. Should event *A* or *B* occur, the accuracy of the prediction is 1.0 or 0.5, respectively.

This accuracy estimation only describes the correctness of the outcome ordering, and does not reflect the accuracy of the predicted likelihood of an outcome.

4.6.2 Accuracy of Predicted Likelihood

It is impossible to assert that any prediction could be 100% accurate, unless only a single event or no events were predicted (and that event occurred). If events *A* and *B* were predicted, and event *A* did occur, the predictive model still was not entirely correct, in that it also predicted that event *B* may occur, when it did not.

In response to these issues, a second approach to evaluating accuracy in real-time was explored. Every predicted event has an associated likelihood, which is proportional to the number of times that this event has occurred previously given the pre-sequence, relative to the number of times any other event has occurred given the same pre-sequence. This is presented to the player as a percentage, where the summed likelihood of all predicted events always equals 100%.

In order to evaluate the accuracy of predictions for the last *n* events, the likelihood of all events is summed, with the total result divided by *n*, resulting in a value between 0 and 100, with 0 indicating total inaccuracy and 100 indicating complete accuracy. We may consider the earlier example, where two events *A* and *B* are predicted, with likelihood of 92% and 8% respectively. Should event *A* occur, the accuracy for this single prediction is evaluated to be equal to the likelihood of the event which occurred (92%).

One side-effect of this approach is that, except for cases in which a single event is predicted (and that event occurs), the reported accuracy of the model will never be 100% - even in cases where the top-ranked event occurred in every case. This reflects the ambiguity that is inevitable in any predictive system where multiple possible outcomes are possible.

4.7 Summary

This chapter has described the development of *WatchAndLearn*[6,7], a prototype WoW plug-in that provides adaptive feedback to players based upon observed behavioural patterns. The implemented system aims to model player actions and use this data to generate predictions of upcoming actions. In order to determine the effectiveness of this approach, the plug-in must be made available for usage by WoW players, with player feedback and measurements of prediction accuracy collected. Chapter 5 discusses the evaluation of *WatchAndLearn*.

[6] Video demonstration of *WatchAndLearn*: `http://bit.ly/Y7BhoA`.

[7] Development site for *WatchAndLearn* (including source code): `http://wow.curseforge.com/addons/watchandlearn/`

Chapter 5

Prototype Evaluations

In order to determine the effectiveness of the techniques developed during the implementation of *WatchAndLearn*, discussed in Chapter 4, the prototype system needed to be evaluated by WoW players. This chapter discusses the procedure and outcome of this evaluation.

The user evaluation was intended to serve three key goals:

1. Evaluate the efficacy of the prototyped modelling approach.

2. Evaluate the appropriateness of the prototyped prediction visualisation.

3. Evaluate the user reaction to the prototype plug-in.

To achieve these goals, an in-game user trial was conducted, where participants were asked to download, install, and play the game with the prototype plug-in. This chapter discusses the procedure and outcome of this evaluation. Firstly, the evaluation procedure is discussed, including an outline of the evaluation methodology (Section 5.1) and key characteristics of the participants recruited (Section 5.2). Section 5.3 describes the questions participants were asked during the course of the evaluation and the methods used to elicit feedback, while Section 5.4 details what data was automatically collected by the plug-in for later analysis. Section 5.5 outlines the results of the evaluation, with implications of these results discussed in Section 5.6. Finally, the overall outcome of the evaluation is summarised in Section 5.7.

5.1 Methodology

Upon agreeing to take part in the evaluation, participants were directed to a link where they could download an adapted version of the *WatchAndLearn* plug-in[1], which was augmented to automatically run a series of trials with different configuration settings, as well as administering an in-game questionnaire with the participants to collect demographic data and feedback. After completing the survey, participants were asked to upload a log file which included all collected predictive data, the predictive sequences used, accuracy measurements for all predictions, and the feedback and answers provided by the participants.

The plug-in used for evaluation did not utilise state or buff data when estimating prediction likelihood (as discussed in Section 4.3.1). This was ignored when generating estimations for the evaluation, in order to reduce potential irregularity of model accuracy, as — at the time of the evaluation — prediction augmentation using state and buff history was not reliably accurate.

The evaluation itself consisted of a 'warm-up' period, where the system was trained as players generated predictive sequences through play. During this warm-up, prediction visualisations were active — with the intention of allowing the participant to become accustomed to the behaviour of predictions. Following the warm-up, three trials were conducted. The only differing factor between the three trials was the pre-sequence length of the Markov chains used to generate predictions (see Section 4.2.1), with this variable set at 2, 3, and 4 for trials 1, 2, and 3 respectively. Each trial continued until the participant had performed 50 actions, with the exception of the warm-up trial which consisted of 100 actions (more actions were required during training in order to generate a more robust predictive model). While the predictive model continued to develop during each trial as actions were observed, it was reset at the end of each trial to the state that was recorded at the end of the warm-up period. This was intended to ensure that later trials were not favoured by more robust predictive models than earlier trials.

Participants were asked to play in their normal (typical) manner while performing the evaluation, and while it was suggested that the evaluation would be completed faster if they placed themselves in a heavily combat-oriented scenario, no specific requirements were made regarding player behaviour.

[1] Available for download at `http://bit.ly/WZtp8b`

Table 5.1: Summary of the responses for participant's demographic data.

Question	Average (S.D.)
Years playing WoW	5.4 (2.2)
Average hours played per week	31.9 (16.8)
Self-rating of mechanical knowledge (7-point Likert scale)	6 (1.2)

The complete trial process was expected to take 20-45 minutes. Participants were compensated for their time with a 30-day World of Warcraft game-time code[2].

Approval for this evaluation was obtained from the Ethics Committee of the Faculty of Computing and Mathematical Sciences at The University of Waikato (see Appendix E).

5.2 Survey Participants

Requests for participation in the evaluation were posted on a number of WoW-related internet discussion forums[3]. In total, 20 participants responded and took part in the evaluation, 3 of whom were female (15%), and 17 of whom were male (85 %). Additional demographic data is shown in Table 5.1. On average, the participants reported having played the game for 5.4 years (standard deviation of 2.2 years), with an average weekly play-time of 31.9 hours (standard deviation of 16.8 hours). On a 7-point Likert scale, the average self-rating for mechanical knowledge was 6 (standard deviation of 1.2). These statistics suggest that the participants were, in general, expert players with a high level of understanding of the game.

Character Roles and Play-style

WoW characters typically fall into one of three roles, depending on character configuration and class constraints: damage-dealing (DPS), damage-mitigating (tanking), and healing. These three roles are most obvious in group play, where players must play to their role's strengths in order to assist their group. A five-

[2] World of Warcraft Game Time — Blizzard Store. `http://us.blizzard.com/store/details.xml?id=110000004`

[3] Forums included: *MMO Extravaganza — Penny Arcade Forums* (`http://forums.penny-arcade.com/categories/massively-multiplayer-online-e`)
World of Warcraft — GP Forums (`http://www.gpforums.co.nz/forum/WOW/?s=`)

man group will typically consist of 1 tank, 1 healer, and 3 DPS characters. A ten-man group will typically consist of 1-2 tanks, 3-4 healers, and 4-6 DPS characters. While a twenty-five man group (the largest size in dungeon raiding) will typically consist of 2-3 tanks, 6-8 healers, and 14-17 DPS characters (Spencer, 2007). The result is that in general, typical role usage in competitive play will be approximately 15% tanks, 30% healers, and around 55% DPS characters, with actual numbers depending upon the content.

Of the 20 participants that took part in the survey, 3 (12%) played tanks, 7 (28%) played healers, and 13 (52%) played DPS characters during the evaluation, providing a fairly representative spread of character roles (albeit from a small sample size).

5.3 Questionnaires

After logging into the game with the *WatchAndLearn* plug-in activated, participants were asked to answer a number of questions providing basic demographic information, as well as some information regarding their game-play habits in WoW. These questions were presented to the player using an in-game dialogue box (an example is shown in Figure 5.1). The questions asked, with answers in brackets, were:

1. Age *[Text]*

2. Gender *[Text]*

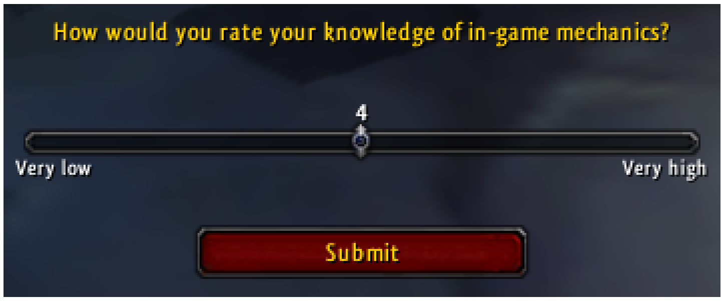
3. How many years have you played WoW? *[0-9]*

4. On average, how many hours per week do you play WoW? *[0-100]*

5. How would you rate your knowledge of in-game mechanics? *[7-point Likert scale with anchors 1: Very low, 7: Very high]*

After each trial (excluding the warm-up trial), participants were asked to rate how accurate they felt the predictions were, on a 7-point Likert scale with the anchors 1 being *Not accurate* and 7 being *Very accurate*.

After the three trials were completed, the participants were asked a series of follow-up questions, all of which were rated on a 7-point Likert scale (Likert scale anchors are given in brackets):



How would you rate your knowledge of in-game mechanics?

Figure 5.1: A 7-point Likert-scale question, as presented in-game to study participants.

1. Overall, how accurate did you feel the predictions were (for all trials)? *[Not accurate–Very accurate]*

2. Do you feel that the system correctly identified patterns of ability usage? *[Not at all–Very much]*

3. How clear was the visualisation in terms of communicating which abilities were predicted to occur next (regardless of the accuracy of the prediction)? *[Not clear at all–Very clear]*

4. Did you select your abilities based upon the predictions provided? *[Never–Always]*

5. Did you check all predicted abilities before selecting an ability to use? *[Never–Always]*

6. Did you feel that you had enough time to inspect all predicted abilities before selecting an ability to use? *[Never–Always]*

Finally, participants were asked to provide any other feedback or comments regarding the plug-in.

5.4 Data Collection

A number of data elements describing the evaluation session were captured, including the complete predictive model (which included both predictive sequences and historical data used to generate those sequences), player ability usage (time-stamped), and a list of abilities predicted by the plug-in each time the participant used any ability. The 'actual' accuracy of predictions was also measured, where the likelihood estimations (ranging from 0 to 100) and the ranked ordering (from 1 to 5, where 1 indicates the ability with the highest predicted likelihood) of whichever ability the player selected was recorded. In cases where the selected ability was not predicted at all (including when no predictions were made), the accuracy and rank was recorded as 0.

5.5 Results

This section discusses the results of the analysis carried out on the game-play data collected by the plug-in, and the feedback and comments elicited from the participants. Firstly, results are given for the accuracy of the system's modelling approach, both in terms of the perceived accuracy and the measured accuracy (Section 5.5.1). Following this, participants' responses for a variety of questions about their reaction to the system are presented (Section 5.5.2). Finally, participants' feedback and comments are discussed (Section 5.5.3).

5.5.1 Prediction Accuracy

The accuracy of the applied model can be described in terms of three aspects: accuracy as perceived by the participants, accuracy of the likelihood estimations, and accuracy of the ranked ordering of the predicted abilities.

Perceived Accuracy

Participant responses included ratings (on a 7-point Likert scale) of perceived accuracy for each trial, and overall for the three trials.

Table 5.2 shows the Likert-scale rating counts of the perceived accuracy for each trial, as well as the perceived overall accuracy. Table 5.3 shows a more

Table 5.2: Rating counts of perceived accuracy for each trial, and overall.

	Rating *1: not accurate, 7: very accurate*						
	1	**2**	**3**	**4**	**5**	**6**	**7**
Trial 1	0	2	5	5	5	3	0
Trial 2	0	1	3	4	5	6	1
Trial 3	0	1	4	2	7	5	1
Overall	0	3	1	5	6	4	1

Table 5.3: Ratings counts of perceived accuracy for character roles for each of the trials, and overall.

		Rating *1: not accurate, 7: very accurate*						
	Role	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Trial 1	**Damage**	0	2	3	1	5	2	0
	Healing	0	0	1	1	0	1	0
	Tanking	0	0	1	3	0	0	0
Trial 2	**Damage**	0	1	2	3	4	3	0
	Healing	0	0	1	0	0	1	1
	Tanking	0	0	0	1	1	2	0
Trial 3	**Damage**	0	1	3	2	3	3	1
	Healing	0	1	0	0	1	1	0
	Tanking	0	0	0	2	1	0	1
Overall	**Damage**	0	2	1	3	4	3	0
	Healing	0	1	0	0	1	1	0
	Tanking	0	0	0	2	1	0	1

Table 5.4: Average ratings of perceived accuracy, by character role.

	Average Rating (S.D.) *1: not accurate, 7: very accurate*			
	Damage	**Healing**	**Tanking**	**All Roles**
Trial 1	4.2 (1.1)	4.3 (1.5)	3.8 (0.5)	4.1 (1.3)
Trial 2	4.5 (1.3)	5.3 (2.1)	5.3 (1.0)	4.8 (1.3)
Trial 3	4.5 (1.5)	4.7 (1.5)	5.3 (0.5)	4.7 (1.3)
Overall	4.4 (1.4)	4.3 (2.1)	5.0 (1.4)	4.5 (1.4)

detailed break-down of the same data, separated according to the role (damage/healing/tanking) of the character being used to perform the evaluation.

Table 5.4 shows the average perceived accuracy ratings for each trial and overall, for each of the three character roles.

A one-way repeated measures analysis of variance was carried out to test for differences between participant ratings of perceived accuracy for each trial. No significant difference between ratings for each trial was found, $F_{2,57} = 1.53$, $p = 0.23$. Figure 5.2 shows mean ratings for each trial at a 95% confidence interval.

Figure 5.3 visualises the data from Table 5.4. Ratings range from 1 (Not accurate) to 7 (Very accurate), with the default level of 4 representing a moderate level of measured accuracy. While there is no significant statistical difference between class roles and average perceived accuracy between trials, nearly all ratings show an overall positive level of perceived accuracy, with the single

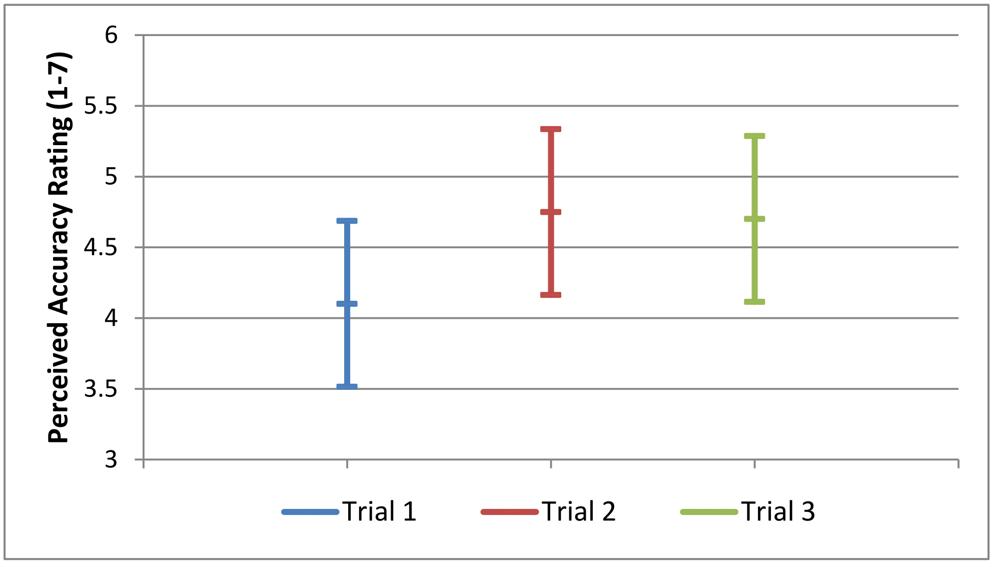


Figure 5.2: Perceived accuracy ratings over three trials, at 95% confidence interval.

exception being the average rating of trial 1 accuracy by participants playing tanking characters (averaging 3.8 with a standard deviation of 0.5). This group also reported the most significant improvement between trials 1 and 2 (an increase from an average rating of 3.8 to 5.3). While the differences between individual trials are not significant, these results show a generally positive rating of the plug-in amongst all class-roles.

Accuracy of Likelihood Estimation

The "actual" accuracy of predictions was also measured, where the predicted likelihood (ranging from 0 to 100) of whichever ability the player selected was recorded. In cases where the selected ability was not predicted at all (including when no predictions were made), the accuracy was recorded as 0. With this method of representing accuracy, any non-zero recording indicated some positive level of accuracy. Extremely high average accuracy values (approaching 100) were unlikely, as the highest possible value for each selection was equal to the likelihood of the highest ranked predicted ability. For example, where two abilities were predicted, each with an estimated likelihood of 50%, the highest possible recorded value would be 50%. For these reasons, this measure of accuracy was intended to primarily serve as a means of comparing estimated likelihood accuracy between trials. Table 5.5 shows measurements of average prediction accuracy for all three trials, and overall.

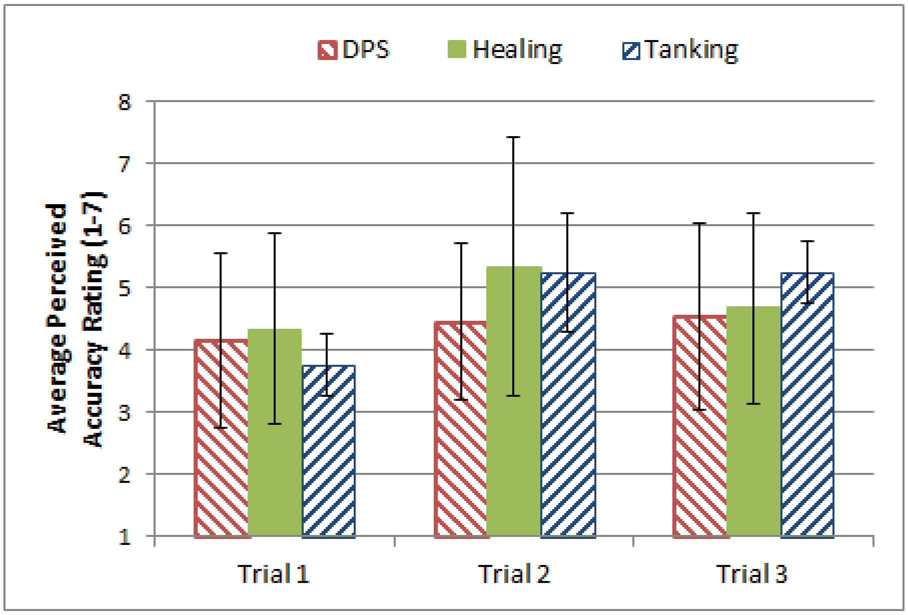


Figure 5.3: Average perceived accuracy ratings over three trials, separated by character role.

Table 5.5: Average measured accuracy for all trials.

	Average Accuracy % (S.D.)
Trial 1	31.18 (13.40)
Trial 2	36.46 (15.09)
Trial 3	40.26 (20.49)
Average	35.97 (16.33)

Figure 5.4 shows the perceived and measured accuracy across all three trials. There was a moderate correlation between the ratings for perceived accuracy and the measured accuracy over the three trials ($r^2 = 0.77$).

A one-way repeated measures analysis of variance was carried out to test for differences between the measured prediction accuracy for each trial. No significant difference between accuracy for each trial was found, $F_{2,57} = 1.51$, $p = 0.23$. Figure 5.5 shows the mean measured accuracy of each trial at a 95% confidence interval.

It was expected that as Markov chain pre-sequence length is increased, fewer and more accurate predictions would be made. Collected data supports this assumption. Figure 5.6 shows a strong inverse correlation between the number of predictions made per trial and the average accuracy of those predictions ($r^2 = 1.0$), with accuracy increasing and the total number of predictions made decreasing as the pre-sequence length increased between trials.

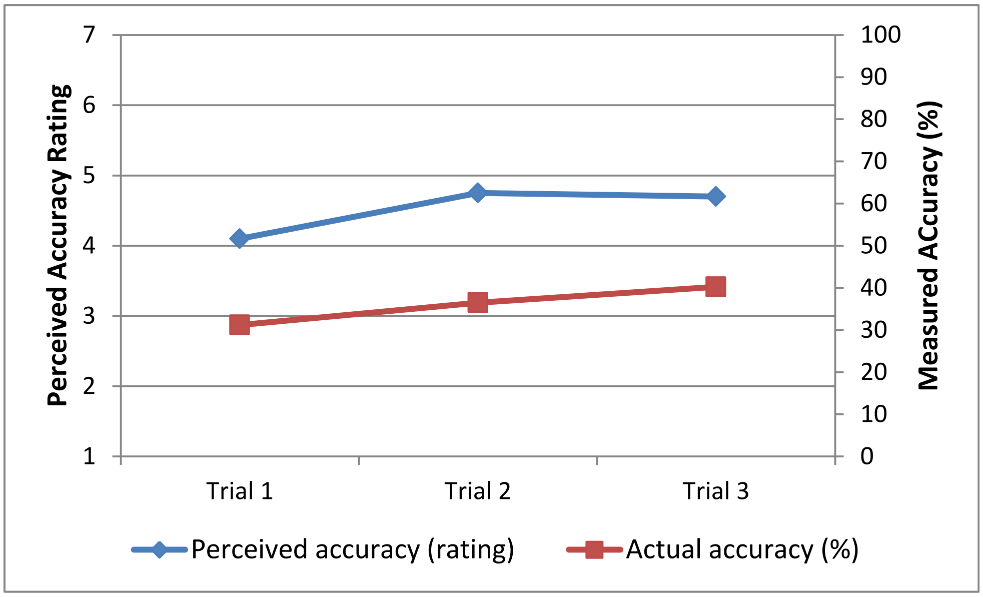


Figure 5.4: Perceived accuracy ratings compared to measured accuracy.

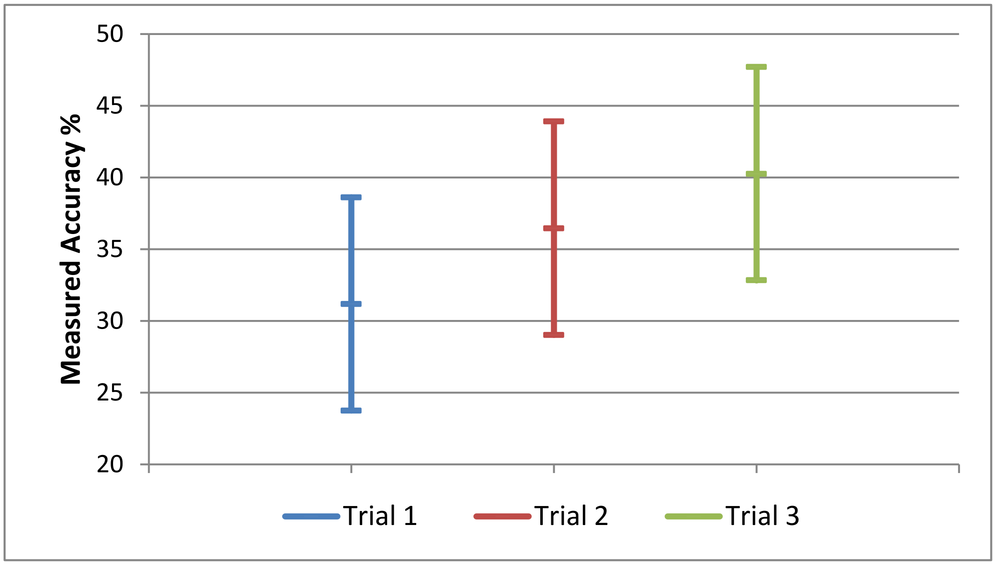


Figure 5.5: Measured accuracy over three trials, at 95% confidence interval.

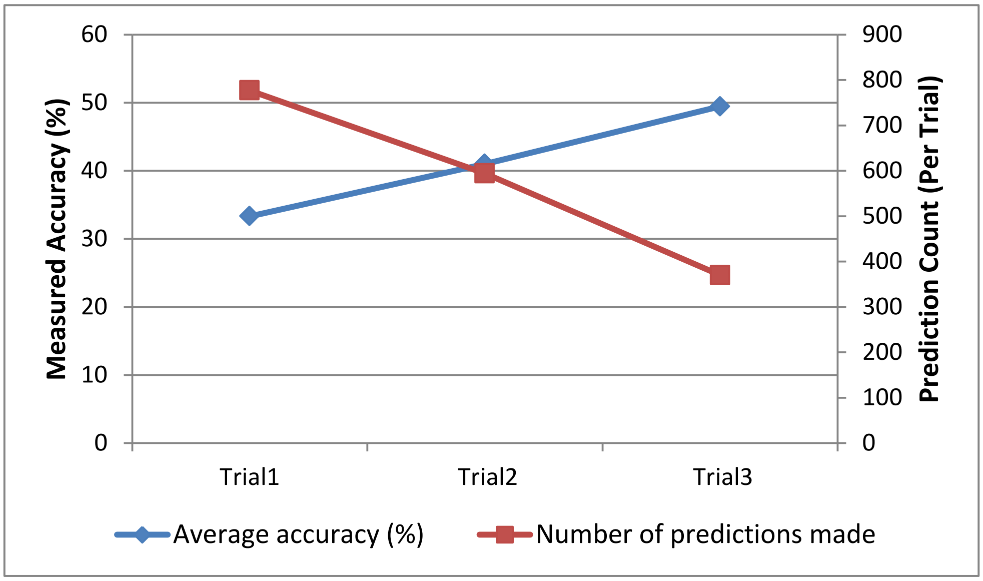


Figure 5.6: Measured accuracy compared to the number of predictions made for each trial.

Table 5.6: Rank of selected ability Vs. number of predicted abilities (%). Highest proportion emphasized.

Abilities	**Rank of selected ability**					
predicted	**Unranked**	**1**	**2**	**3**	**4**	**5**
0	*100.0*	0	0	0	0	0
1	49.8	*50.2*	0	0	0	0
2	33.2	*38.6*	28.2	0	0	0
3	20.7	*45.6*	19.7	13.9	0	0
4	17.0	*42.5*	20.1	12.0	8.5	0
5	12.1	*36.8*	17.8	15.5	10.3	7.5

Ranking Accuracy

In addition to explicit percentage values expressing estimated predicted likelihood, predicted abilities were displayed as a ranked list, ordered from the most likely to the least likely. The intention of this was to more clearly express which abilities were estimated to be more likely, allowing players to recognise this at a glance, without the need to parse more specific estimation values.

The rank of selected abilities was recorded, as was the list of predicted abilities. Table 5.6 shows a breakdown of the number of predicted abilities against the actual rank of the selected ability, expressed as a percentage of observations.

This data is also visualised in Figure 5.7, showing that in all cases where one or more predictions were made, the first ranked ability was most likely to

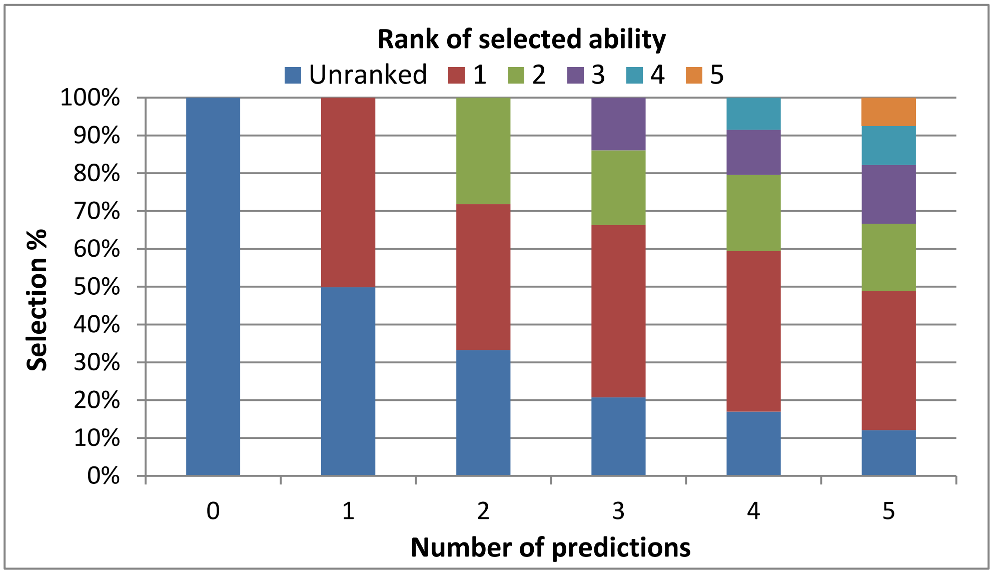


Figure 5.7: Proportion of ability rankings Vs. rank of selected ability.

be chosen. This also takes situations into account where the selected ability was unranked (not predicted). Further, this shows that as more abilities are predicted, the chances of an unranked ability being selected reduce substantially. For example, where only one ability was predicted, 50.2% of the time that ability was selected (with some unranked ability being selected 48.2% of the time). Where five abilities were predicted, an unranked ability was only selected 12.1% of the time. The accuracy of ability rankings was significantly more reliable than the accuracy of likelihood estimates (see Figure 5.5), indicating that while the applied model was capable of providing a rough estimation of which ability would be used, estimating the actual likelihood of that ability being used was more difficult.

5.5.2 Final Questionnaire Ratings

Following the completion of all three trials, participants were asked a number of follow-up questions, intended to gauge their general reaction to various aspects of *WatchAndLearn*. The 7-point Likert scale ratings provided by participants for these questions are shown in Table 5.7 (full descriptions of answer ranges for each question are provided in Section 5.3).

While ratings were not low, and standard deviation was high, feedback did indicate that participants generally did not select abilities based upon the predictions provided, choosing instead to use their own judgement when selecting

Table 5.7: Final Questionnaire Average Ratings

#	Question	Average Rating (S.D.)
1	Did you select your abilities based upon the predictions provided?	3.2 (1.7)
2	Do you feel that the system correctly identified patterns of ability usage?	4.8 (1.7)
3	Did you check all predicted abilities before selecting an ability to use?	3.3 (1.8)
4	Did you feel that you had enough time to inspect all predicted abilities before selecting an ability to use?	4.1 (1.8)
5	How clear was the visualisation in communicating which abilities were predicted to occur next (regardless of the accuracy of the prediction)?	5.8 (1.3)

abilities (question 1: rating of 3.2, standard deviation of 1.7). This is not surprising, since participants were asked to play as they normally would, rather than being asked to play according to the provided suggestions. Additionally, the high level of player experience and self-rated knowledge of game mechanics suggests that players have the skills necessary to select abilities without assistance, and therefore may have found the visualisation of predictions superfluous.

The fact that ratings imply that the participants at least occasionally used the predictions to inform their decisions may suggest a positive reaction to the predictions that were provided, somewhat confirmed by the higher-than-moderate ratings for perceived correctness of pattern usage recognition (question 2: rating of 4.8, standard deviation of 1.7). Similarly, participants indicated that they did not often check all of the predicted abilities before selecting an ability to use (question 3: rating of 3.3, standard deviation of 1.8), which may have been affected by the fact that the participants did not feel that they always had enough time to inspect all of the predicted abilities (question 4: rating of 4.1, standard deviation of 1.8). Again, the fact that the participants occasionally elected to check all of the abilities may be a positive outcome.

Participants also responded very positively to the visualisation used to display predictions, indicating that regardless of the prediction accuracy, the visualisation was effective at conveying which abilities were predicted to occur next (question 5: rating of 5.8, standard deviation of 1.3).

5.5.3 Participant Feedback

In addition to accuracy ratings, participants were asked to provide any additional comments they had regarding *WatchAndLearn*. Participant comments were mixed, but generally positive. Most participants remarked that the system worked to a degree, though a number of specific drawbacks and suggested improvements were provided. The comments discussed in this section are representative of the feedback received.

Positive Feedback

Most of the positive feedback was in regard to the overall accuracy of the system. While accuracy levels were not high, the overall level of accuracy was high enough that most participants were satisfied that *WatchAndLearn* was able to model their behaviour to some degree.

> "Very accurate, even when I went away from what would be the norm based on ability procs[4]. I tried to throw it some curves by selecting odd skills during the initial warm-up, and it even went so far as to slot those into down times in what would've been a normal rotation."

> "The system works very well for a base rotation set in stone, it seems to get confused by procs though, seeing as how I'm a frost DK, it often suggested I howling blasted randomly because it found patterns in my Howling Blasts, even though it's only to be used when a random proc pops that gives you a free one. Other than that it picked up on my Frost Strike/Obliterate weaving pretty accurately, and even seemed to pick up on the cool-down on my Pillar of Frost because of the one minute interval before each use."

> "[...] for the most part it recognized a pattern and displayed pretty accurate results."

> "Was surprised at how well it picked up on what i was doing. Obviously my rotation isn't complicated because of my low level but it was as soon as the first run that i saw it figuring it out. Well done."

[4] Programmed random occurrences or *procs* are used in WoW to simulate random ability effects, and tend to introduce unpredictability into otherwise repetitive ability rotations. See also Section 4.1.2.

> “Really neat, it actually helped me concentrate and try to focus on my rotation more.”

Some participant’s comments indicated that there was perceived potential for a more fully developed implementation of this approach:

> “It was interesting, and I look forward to seeing what it can do with much longer trials, and data gathering periods. (And when it can take cool-downs into account).”

> “Could be turned in a useful add-on for boss moves predictions in case of conflicting timer!”

> “This was a good add-on. I hope to see it soon.”

Negative Feedback

Most of the negative feedback was in regard to the model’s inability to provide accurate predictions for abilities which were used rarely. The more powerful abilities in WoW tend to have a long “cool-down” period associated with their usage, discouraging players from using those abilities unless absolutely necessary. The rarity of the usage of these abilities resulted in very low predicted likely usage, regardless of whether other factors (such as extremely low player health) increased the likelihood of those abilities being used (as these factors were not taken into account when estimating likelihood).

> “It had issues with using big cool-downs, but given how randomly you’d be using them in a real situation (ie - a big damage spike), I can’t see how it would know unless it monitored the conditions around the player too.”

> “It’s good, but unless it takes cool-downs into account (including procs allowing spells to be cast before the CD is up) it won’t be that accurate.”

> “The predictions for the first 2 trials were way off, the third was improved by getting some predictions correct, although some were still incorrect like telling me to use CD’s that still have 1 minute cd timer left or abilities that still have many seconds left to tick.”

Character role also factored into the perceived accuracy and usefulness of *WatchAndLearn*. In particular, participants that played healers — where action is typically only required when allies have recently taken damage — noted that predictions were generally unreliable:

> "I guess because healer ability use isn't predetermined its more reactive to party damage, unless you can monitor who is hurt and by how much, with what's on cool-down etc. it's hard to predict what ability I or any healer will use."

> "I think the main problem with it as a healer I found was that my healing is based on damage and encounter rather than like a 'rotation' as such."

5.6 Discussion

This section discusses the various findings of the prototype evaluation. Firstly, limitations impacting the results of the evaluation are discussed in Section 5.6.1. Following this, evaluation findings are discussed in terms of the original three evaluation goals:

1. Evaluate the efficacy of the prototyped modelling approach (see Section 5.6.2).

2. Evaluate the appropriateness of the prototyped prediction visualisation (see Section 5.6.3).

3. Evaluate the reaction to the prototype plug-in, from a variety of players (see Section 5.6.4).

5.6.1 Limitations of the Evaluation

A number of factors impacted the results of the evaluation. The participant sample was reasonably representative of the general WoW population in terms of the character role spread. However, the sample size of twenty participants may have may the statistical significance of the results. The high reported level of experience and knowledge of the game indicates that the participant group was predominantly comprised of expert players, meaning that findings based upon this evaluation may not be representative of different groups of WoW players. Novice players, in particular, may benefit more from assistance when selecting abilities, and might therefore express a significantly different reaction to *WatchAndLearn*

While practical limitations necessitated relatively short trial periods (of fifty actions per trial), longer trials may have provided more data points allowing for stronger statistical analysis. Finally, the short evaluation period limited the aspects of *WatchAndLearn* which were tested during the evaluation. A number of features could not be evaluated, some of which may have improved model accuracy (see Chapter 4).

For these reasons the results of the evaluation provide an overview of the behaviour of the core aspects of this approach to modelling player behaviour in order to provide tailored ability predictions, and should serve as a basis for further work in this area.

5.6.2 Modelling

Accuracy was measured and evaluated in two ways: in terms of the accuracy of predicted likelihood (represented as a proportion of the summed likelihood of all predicted abilities), and in terms of the accuracy of the ranked ordering of predicted abilities.

Predicted Likelihood

Predicted likelihood increased as Markov chain sequence length increased, though the difference between trials was not statistically significant. As anticipated, increasing the Markov chain sequence length also decreased the total number of abilities predicted, showing that a longer sequence length ultimately results in fewer and more accurate predictions. The side-effect of this is that in cases where sequence lengths are longer and there is a lack of historical data for model training, the model may repeatedly fail to make any prediction at all. There are two ways that this can be counteracted: by shortening the chain sequence length, or by increasing the amount of historical data available for model training.

As discussed earlier, shortening chain sequence length results in more predictions, but negatively impacts prediction accuracy. This trade-off may only be worthwhile where the model is struggling to create predictions, though players may prefer to have no predictions over having unreliable predictions. Alternatively, a longer chain length (enabling more accurate predictions) can be maintained if more source data is made available for modelling. The simplest method of increasing the amount of available source data is to increase the amount of model training time. It may be beneficial to suspend predictions until the player has played long enough for the model to be sufficiently developed. Once the model is deemed appropriately fit for usage, prediction can be enabled. A composite approach may also be beneficial, where chain sequence length is shortened while the model is in early stages of development, and lengthened as the player is observed and their model becomes increasingly robust. The result should be a moderately constant number of predictions which increase in accuracy as the model is developed.

Finally, modelling data can be augmented using data from external sources. As discussed in Chapter 4, *WatchAndLearn* currently supports subscriptions to other players, which allows the subscriber to utilize historical data generated by those players. Additionally, *WatchAndLearn* supports usage of composite models consisting of an arbitrary number of source models. Players can augment their own model by subscribing to another player and combining that model with their own data. As the player's own model develops, the external model can be disabled. This approach would allow the player to benefit from immediately available, and reasonably accurate predictions, with the ability to later transition to more personally tailored predictions by disabling the external model once their own model is sufficiently robust. The drawback of this approach is that any external model will not provide personalised predictions and may be more suitable in cases where the goal of the player is to learn from the source model of an expert player.

Prediction Rankings

In general, prediction rankings were successful in estimating which ability would be used, with the highest ranked ability being selected more than any other ability. In cases where multiple abilities were predicted, the proportion of selections matched the estimated rankings (with the second ranked ability being selected less often than the first, the third less often than the second, and so forth). Furthermore, the higher the total number of predicted abilities, the more likely that one of the abilities on the predicted list would be selected (in 87.9% of cases where five abilities were predicted, one of those abilities was selected by the player).

Prediction rankings provided players with a more general overview of the predictions made by the model, without requiring players to read and compare specific likelihood values. In terms of usefulness to the player, actual likelihood values may be extraneous, except for in cases where the discrepancy between predictions is extreme (for example, if two abilities are predicted, one estimated at 90% likelihood and one estimated at 10% likelihood). In these situations it may in fact be more appropriate to ignore the extremely unlikely predictions, where a clear favourite is obvious.

The appropriateness of such an approach would depend upon the requirements of users. Some players may prefer to be exposed only to predictions which have a likelihood exceeding some minimum threshold, while others may prefer to see all predictions regardless of accuracy. This may also impact the perceived accuracy of the model — players who select an ability with an extremely low predicted likelihood, but which is still in the prediction list may not care that the ability was considered unlikely to occur, as long as it was presented as an option at all (whereas failing to display an ability which is ultimately selected gives the impression that the model failed to consider that the outcome was a possibility at all). Setting a minimum likelihood threshold is included in *WatchAndLearn* as a configurable option, though this was set to zero during the evaluation.

5.6.3 Visualisation

In terms of participant feedback, the visualisation used to convey predictions to players was very well-received (see Section 5.5.2). Combined with feedback regarding the accuracy of the plug-in, feedback received showed that participants were able to correctly identify which abilities were predicted, as well as the predicted likelihood and ranking of those abilities. This included cases where *WatchAndLearn* failed to predict the correct ability, which resulted in lower perceived accuracy. In terms of the visualisation only, players correctly identifying these faults is a positive outcome, indicating that the visualisation succeeded in conveying the model's predictions.

While participants indicated that they did not usually check all the predicted abilities before selecting an ability to use, this may be explained by the evaluation conditions — where participants were not asked to base their behaviour on the predicted abilities. However, other feedback indicated that players may not have always been able to check all suggested abilities, which may be due to the limited amount of time between ability usage in some cases (less than half a second) combined with up to five abilities being predicted at once. The positive feedback regarding the visualisation indicates that this was not considered to be a significant problem.

5.6.4 Participant Responses

While the prototype was developed to test the application of predictive modelling and interface adaptation, the user evaluation was also intended to gauge user reaction to *WatchAndLearn* itself, and to determine whether a similar implementation might be useful to players in some capacity.

A significant issue noted by a number of participants was that the system did not take ability cool-downs into account. This resulted in abilities which were on cool-down and thus unavailable for usage being predicted. Abilities which are extremely effective tend to have longer cool-downs, acting as a trade-off, ensuring players only use those abilities when absolutely needed. As these abilities were always used rarely, their predicted likelihood was always extremely low, despite the fact that other conditions (for example, extremely low health) may increase the likelihood of that ability being used.

Participants playing certain character roles tended to respond differently. In particular, participants that played healers explicitly commented on the inability of *WatchAndLearn* to accurately model ability patterns which were not part of a set rotation. Healers generally play more reactively, using abilities only when allies are missing enough health to justify the ability cost. Generating accurate models of characters with more reactive roles may necessitate an alternative approach, where external conditions such as health levels of allies and estimated threat level are taken into account. Alternatively, the current approach may be successfully applied as an alternative system intended for use by damage-dealing characters only.

Despite these reported issues, participant response to *WatchAndLearn* was mostly positive. A number of the participants indicated that they were surprised or impressed by the model's accuracy, despite reported drawbacks. Others reported that *WatchAndLearn* was successful in assisting them by allowing them to concentrate more fully on their ability rotation. Finally, a number of the participants commented that they felt there was potential for the system, indicating that they would be interested in seeing a fully featured release and that they “look forward to seeing what it can do with much longer trials, and data gathering periods” and “hope to see it soon.”

5.7 Summary

The prototype evaluation discussed in this chapter was successful in identifying issues with the modelling approach, and gauging player reaction to *WatchAndLearn*.

Overall, the applied approach was successful in modelling and predicting player behaviour. While the accuracy of likelihood estimations was not high, ordered ranking was generally accurate, with accuracy levels increasing as more abilities were predicted. Given the short model training period, this indicates that the model is able to accurately predict *which* abilities will be used with very little training (though exact likelihood values are lacking in accuracy), with higher levels of accuracy possible given a longer training period.

The evaluation also highlighted the potential value of augmenting the modelling technique by considering additional game-play factors in order to take abilities with long cool-downs into account, and to better support play-styles which are not heavily based upon consistent ability rotations.

Chapter 6 discusses thesis contributions and possible improvements to the modelling approach in more detail.

Chapter 6

Conclusions and Future Work

This thesis aimed to determine whether player-modelled adaptive interfaces can improve players' game experience in MMORPGs

A survey of WoW players (see Chapter 3) identified a number of different motivations for interface customisation and showed that players want their game interfaces to provide them with more simplified but highly relevant information. In particular, the study found that interface customisation was most important for achievement-related motivational factors, such as advancement, competition and game-play mechanics.

The results of this survey informed the development of a prototype system that utilises models of player behaviour in order to provide players with suggestions for potential future actions (see Chapter 4). A user study of this system showed that the modelling approach was able to successfully generate predictions of future actions, given a short training period (see Chapter 5). The visualisation used to display predictions was positively received, with some participants indicating that the prediction system improved their game experience by allowing them to better concentrate on their patterns of action.

The following sections present the conclusions of this thesis (Section 6.1), and potential areas of future work (Section 6.2).

6.1 Thesis Conclusions

The findings of this thesis show that adaptive user interfaces have the potential to improve players' game experience in MMORPGs.

The contributions of this work can be discussed in terms of the three key research questions, presented in Section 1.2:

Which aspects of MMORPG interfaces can be modified in order to improve the player's game experience? A survey of WoW players (see Chapter 3) showed that the player's game experience can be improved by maximising the availability of highly relevant game-play information, while reducing the visibility of less relevant information. More specifically, findings indicated that most players prefer to have their interfaces modified in ways which would provide them with a mechanical game-play advantage.

How can player actions be modelled and utilised in order to improve the player's game experience? As is shown in Chapter 4, Markov chains can be used to model players' patterns of ability usage, allowing for the prediction of upcoming actions. The generated behavioural model can be used to determine appropriate adaptations to the player's game interface.

Do interface modifications based on modelling player actions improve the player's game experience? Feedback received during the evaluation of the prototype system (see Chapter 5) shows that player-modelled interface adaptation can be used to provide players with important and relevant information, allowing the player to focus on their patterns of action usage. This can help to improve mechanical performance, an aspect of game-play which is important to MMORPG players.

6.2 Future Work

The work accomplished in this thesis also resulted in the identification of a number of potential avenues for future research. Section 6.2.1 discusses ways in which the accuracy of generated predictions can be improved for the existing system, while Section 6.2.2 provides suggestions for possible applications of accuracy estimations. Section 6.2.3 describes a number of alternative uses for predictions generated using observed player behaviour. Finally, Section 6.2.4 discusses potential applications of adaptive feedback in other computer games.

6.2.1 Improving Prediction Accuracy

The user study of the *WatchAndLearn* prototype highlighted two potential improvements to prediction accuracy: considering character role when and play-style when generating predictions, and taking action availability into account when estimating likelihoods.

Considering Character Roles

The prototype evaluation showed that participants who played certain character roles experienced different levels of prediction accuracy. In particular, participants who played healing characters tended to perform actions more reactively, with the actual action performed often being dependant upon the state of the player's target (for example, the health level of the target) rather than being determined by which abilities had been used previously.

This suggests that current state and historical state data should be factored into prediction generation, as discussed in Section 4.3.1. However, the relevance of state data varies depending upon the nature of the character being played. For example, the prototype evaluation showed that prediction accuracy was higher for characters which were classified as damage dealers than it was for healers, when using event-based prediction (without any consideration given to state characteristics). The variation between prediction accuracy for different character roles indicates that while event-based prediction may be sufficient for damage dealers, other character roles (such as healers) may benefit from prediction improvement using state data.

Ideal weightings for state-based predictions compared to event-based prediction are not obvious, and may be dependant upon factors other than character role (such as personal play-style). For example, certain players may follow more prescriptive patterns, regardless of their characters primary role.

Determining appropriate weightings may be accomplished automatically, by providing initial predictions based entirely upon event history, and gradually increasing the relative weighting of state-based estimation until a maximum level of accuracy is found.

Alternatively (as is currently implemented in the prototype system), the player may determine weighting preferences and set these themselves. In these cases, players may benefit from having default pre-configured weighting configurations determined by their character role, rather than having a single default weighting configuration applied to all characters.

Modelling Action Availability

Feedback received during prototype evaluations indicated that a number of participants felt that prediction accuracy was adversely impacted by the inability of the system to properly consider rarely used but highly effective abilities (see Section 5.6.4). The high effectiveness of these abilities tend to be balanced by their high 'cool-down' timer periods which prevent players from repeatedly using a powerful ability within a short period of time. Because of this, these abilities are rarely used, resulting in low predicted likelihood in most circumstances. This does not take into account the possibility that a limited number of observations of a particular action may be caused by the limited availability of that action, and that in cases where the action is available for usage, the likelihood of that action be performed may be drastically increased.

Abilities which have high cool-downs tend to also be used rarely due to their high value, which encourages players to refrain from performing those actions until absolutely required. In these cases, the current predictive model is incapable of providing accurate likelihood estimations, as there is no established measurement of action effectiveness.

In order to improve accuracy when predicting actions which feature long cool-down's, factors such as the current availability and relative effectiveness of all actions should be taken into account.

6.2.2 Utilising Accuracy Estimations

Methods of estimating model accuracy, as discussed in Section 4.6, may only be applied retroactively, meaning that they cannot be calculated until after a prediction has been made, and the player has then performed an action. This limits the usefulness of such an estimation, as any harm caused by an inaccurate estimation has already been caused. However, this data can be used to improve prediction accuracy by altering system behaviour based upon past accuracy.

One way of achieving this is to simply hide all predictions from the player until accuracy levels reach an appropriate threshold. This approach is particularly relevant in the early stages of model development, where outlier data can have a significantly detrimental impact on prediction accuracy. As observed accuracy begins to reliably exceed a pre-determined threshold, prediction visualisations can be activated.

Another possible approach, which may be used in combination with the above approach, is to utilise prediction accuracy ratings for the purpose of selecting an appropriate Markov chain pre-sequence length. As discussed in Section 4.2.4, longer chain pre-sequence lengths can result in more accurate predictions, but will also reduce the number of total predictions, potentially preventing the system from making any predictions at all. While predictions are presented to the player based upon a single chain pre-sequence size, models using a variety of pre-sequence lengths may be generated in the background, with the accuracy of all models measured as the player performs actions. If accuracy measurements show that an alternative chain pre-sequence length is consistently resulting in more accurate predictions, those predictions can be displayed to the player instead. As the player model continues to develop, the currently active chain pre-sequence length can continue to be dynamically changed, providing the player with more accurate predictions.

6.2.3 Alternative Uses for Predictions

The prototype system that was developed utilised behavioural predictions to display estimated upcoming actions to the player. This allowed for the prediction system to be evaluated by having users provide feedback about their perceived prediction accuracy. While this application of predictions was useful for the purpose of the user evaluation, behavioural predictions can be applied in order to improve player's game experience in a number of ways.

Suggesting Alternative Play-styles

The prototype system was successful in identifying different behavioural patterns of players, allowing for recognition of common sequences of abilities. This data can be used to describe the play-style of the individual player in terms of their ability usage. For example, some players will usually perform action C after the sequence $\{A, B\}$, while other players will more often perform action D following the same sequence.

With expected patterns of action identified, it is possible to provide suggestions for alternative valid behaviour, where the outcome is either equally or more favourable in terms of player goals. This can be used as a tutoring tool, assisting novice players by encouraging them to utilise action patterns which may not have been apparent. This approach could also be applied to more advanced players, by suggesting patterns of ability usage which are similarly effective, but will introduce variety into the player experience.

Supporting Dynamic Difficulty Adjustment

As discussed in Section 2.1.3, auto-dynamic difficulty adjustment can be leveraged in order to keep players within a continual zone of flow (Bailey and Katchabaw, 2005; Chen, 2006a). This is achieved by monitoring player performance and increasing difficulty (in cases where the player is not being challenged) or decreasing difficulty (in cases where the player is struggling) accordingly. While monitoring player performance was not a consideration of this thesis, behavioural models can be used to determine appropriate methods of applying dynamic difficulty adjustment.

As an example application of this, typical player behaviour can be inspected and analysed in order to modify the abilities and traits of enemies which the player must face. A number of role-playing games (of which MMORPGs are a subset) such as Diablo 3[1], Torchlight 2[2] and Path of Exile[3] increase difficulty by providing enemy characters with resistances to certain types of damage (see Figure 6.1). In order to remain effective, the player must react by adjusting the abilities that they are using in order to maximise their damage output.

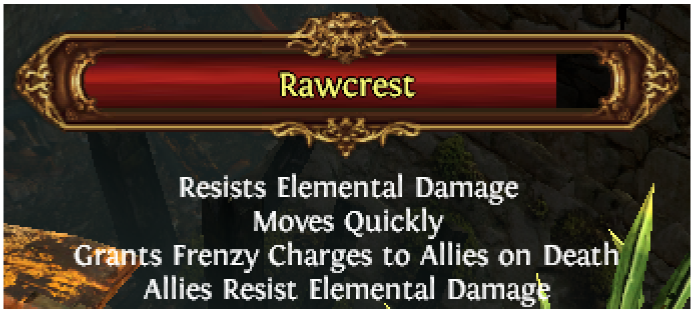

Rawcrest
Resists Elemental Damage
Moves Quickly
Grants Frenzy Charges to Allies on Death
Allies Resist Elemental Damage

Figure 6.1: Health bar and active effects for a target in Path of Exile.

Models of player behaviour can be used to identify what actions the player is likely to perform, and to determine whether the likely action will be effective given a certain enemy type. In cases where dynamic difficulty adjustment is being used, enemy effects (such as resistances to certain damage types) can be

[1] http://us.battle.net/d3/en/
[2] http://www.torchlight2game.com/
[3] https://www.pathofexile.com/

increased or decreased accordingly, resulting in either an increase or decrease in the relative difficulty for the player.

For example, behavioural models may suggest that the player continues to use abilities which deal fire damage to enemies. If performance indicators suggest that the player may be bored or under-challenged, new enemy characters can be created with heightened resistance to fire damage, thereby increasing the level of challenge for the player.

Creating Believable Game Agents

Bryant and Miikkulainen (2007) discuss the development of 'visibly intelligent' game agents, which behave in a human-like manner, rather than simply performing optimally (see Section 2.2.4). This allows for game agents to appear human-like, increasing the sensation of immersion for the player. Additionally, human-like agents allow for more engaging encounters between the player and the agent, as human-like behaviour tends to introduce an element of unpredictability.

For example, while the optimal course of action may involve an agent using a highly-damaging ability whenever possible, more realistic behaviour may involve the agent occasionally 'forgetting' to follow the optimal course of action, resulting in a less predictable experience for the opposing player.

The prediction system presented in this thesis can be applied directly to game agent behaviour, by using models generated by previously observed human players to determine upcoming actions from the game agent. Models can either be created by the game's developers, or by observing actual player behaviour. In the latter case, models can either be static (where an agent's behavioural model does not change throughout the lifetime of the agent) or dynamic (where the agent continues to learn while observing player behaviour).

In online games (such as MMORPGs), models can be generated by using multiple players as sources, even if the game agent is not in proximity of those players. This can be achieved using the approach to model subscriptions described in Section 4.5. This allows for the behaviour of game agents to continue to change, according to emerging patterns in the behaviour of actual players. Additionally, different sources can be activated and deactivated in order to generate differing behavioural models, allowing individual game agents which may share common ability sets to exhibit more individualistic behaviour.

6.2.4 Other Applications of Adaptive Feedback

While the prototype system was developed specifically for WoW, the implemented approach to player modelling and behavioural prediction is intended to be applicable to other MMORPGs, as well as to a more diverse variety of computer game genres.

Applying Adaptive Feedback to Other MMORPGs

MMORPGs allow the player to take control of the actions of an in-game character. Because of this core similarity, most MMORPGs tend to allow for players to interact with the game in similar ways. Generally, player interaction is performed by selecting character abilities from an action bar in order to deal damage to enemy characters, with different ability combinations resulting in varied types and levels of damage output. In short, player interaction in most popular MMORPGs is generally similar to player interaction in WoW.

Additionally, a number of popular MMORPGs, including RIFT[4], Warhammer Online: Age of Reckoning[5], and Age of Conan[6] provide the means for development and usage of third-party game plug-ins. This allows for development of plug-ins which will provide similar functionality to the prototype plug-in developed for this thesis.

However, other popular MMORPGs such as Guild Wars 2[7], Dungeons and Dragons Online[8], and Lord of the Rings Online[9] do not provide the means for advanced interface customisation by players. In these cases, an adaptive feedback system may only be implemented by the game's developers.

While the details of implementation may differ from game to game, the approach to providing adaptive feedback that is presented in this thesis will be applicable to any MMORPG which is inherently similar to WoW. A list of core requirements for the application of this approach are provided in the following section.

[4] http://www.riftgame.come/en/

[5] http://warhammeronline.com

[6] http://www.ageofconan.com

[7] https://www.guildwars2.com/

[8] http://ddo.com

[9] http://lotro.com

Applying Adaptive Feedback to Other Game Types

While the system presented in this thesis was able to successfully model sequences of discrete player actions, there are limitations to the appropriateness of this approach to other types of games. The modelling approach used in this work is applicable for any game which meets the following criteria:

1. **The player interacts with the game through a series of discrete actions.** Any series of discrete actions can be modelled using Markov chains, allowing for the generation of predictive sequences. Any game where the primary source of player action is discrete — such as button presses or ability selections — may be modelled in this way. Continuous actions such as analogue movement will require different modelling techniques.

2. **The player is able to perform a variety of actions at any given time.** Predictive models of player behaviour can be useful in those circumstance where players may perform a variety of different actions. In cases where only one possible (or logical) action exists, predictions of likely outcomes are redundant.

3. **Different action combinations result in different outcomes.** The relative effects of different action selections determines the usefulness of action prediction. In cases where all possible actions will result in an identical outcome, the path chosen by the player to reach that outcome is generally irrelevant. Behavioural prediction increases in value as the variety of potential outcomes (and therefore, the uncertainty with which each outcome will occur) increases.

4. **(Optional) Not all actions will result in equally favourable outcomes.** In cases where certain outcomes are considered more favourable than others (for example, having a player character inflict more or less damage upon an enemy), behavioural prediction can be used to determine whether likely outcomes are also favourable for the player. Alternately, this can allow for the system to be augmented so that outcomes which are highly favourable but also highly unlikely (given player history) can be suggested to the player. This is considered an optional criterion as the usefulness of this information is dependant upon the goals of the game.

5. **(Optional) Action selection is impacted by contextual information.** The proposed system also allows for consideration of contextual information

such as currently active player effects (buffs) and player and target state information. This can be used to augment predictions based upon action history by modifying estimations of predicted likelihood. Similar measurable effects can be taken into account for any other game. For example, in a car racing game, the current speed and current lap may have an impact on possible action likelihood. While this may result in more robust predictive likelihood estimations, it is not mandatory, and may be ignored in cases where contextual information does not impact player actions.

References

S. Adinolf and S. Turkay. Controlling your game controls: Interface and customization. In C. Steinkuehler, C. Martin, and A. Ochsner, editors, *7th Annual Games+Learning+Society (GLS) Conference*, pages 13–22. ETC Press, 2007.

A. Alcorn. Pong. *Published by: Atari Inc.*, 1972.

C. Ang, P. Zaphiris, and S. Mahmood. A model of cognitive loads in massively multiplayer online role playing games. *Interacting with Computers*, 19(2):167–179, 2007.

C. Bailey and M. Katchabaw. An experimental testbed to enable auto-dynamic difficulty in modern video games. In *Proceedings of the 2005 GameOn North America Conference*, pages 18–22, 2005.

W.S. Bainbridge. The scientific research potential of virtual worlds. *science*, 317 (5837):472–476, 2007.

L. Baker. Factbox: A look at the $65 billion video games industry, 2011. URL `http://uk.reuters.com/article/2011/06/06/us-videogames-factbox-idUKTRE75552I20110606`. Last accessed: February 19, 2013.

L. Baker and M. Nayak. Factbox: A look at the $78 billion video games industry, 2012. URL `http://www.reuters.com/article/2012/06/01/us-videogameshow-e3-show-factbox-idUSBRE8501IN20120601`. Last accessed: February 19, 2013.

R. Bartle. Hearts, clubs, diamonds, spades: Players who suit muds. *The Journal of Virtual Environments*, 1(1), 1996.

C. Beal, J. Beck, D. Westbrook, M. Atkin, and P. Cohen. Intelligent modeling of the user in interactive entertainment. In *AAAI Stanford Spring Symposium, Stanford CA*, 2002.

J.E. Beck, B.P. Woolf, C.R. Beal, et al. *ADVISOR: a machine-learning architecture for intelligent tutor construction*. University of Massachusetts at Amherst, 2001.

Blizzard Entertainment. World of Warcraft. *Published by: Blizzard Entertainment, Inc*, 2013.

R. Bridle and E. McCreath. Inducing shortcuts on a mobile phone interface. In *Proceedings of the 11th international conference on Intelligent user interfaces*, pages 327–329. ACM, 2006.

B.D. Bryant and R. Miikkulainen. Acquiring visibly intelligent behavior with example-guided neuroevolution. In *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, volume 22, page 801. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007.

E. Castronova. *Synthetic Worlds: The Business and Culture of Online Games*. University of Chicago Press, 2005. ISBN 9780226096278.

D. Charles, A. Kerr, M. McNeill, M. McAlister, M. Black, J. Kcklich, A. Moore, and K. Stringer. Player-centred game design: Player modelling and adaptive digital games. In *Proceedings of the Digital Games Research Conference*, volume 285. Citeseer, 2005.

J. Chen. Flow in games, 2006a. URL `http://jenovachen.com/flowingames/thesis.htm`.

J. Chen. flow, 2006b. URL `http://www.jenovachen.com/flowingames/flowing.htm`. Last accessed: February 19, 2013.

A. Cockburn, C. Gutwin, and S. Greenberg. A predictive model of menu performance. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 627–636. ACM, 2007.

A.T. Corbett, K.R. Koedinger, and J.R. Anderson. Intelligent tutoring systems. *Handbook of humancomputer interaction*, pages 849–874, 1997.

B. Cowley, D. Charles, M. Black, and R. Hickey. Toward an understanding of flow in video games. *Computers in Entertainment (CIE)*, 6(2):20, 2008.

Crytek. Crysis 2. *Published by: Electronic Arts*, 2011.

M. Csikszentmihalyi. *Flow: The Psychology of Optimal Experience*. Harper Perennial, New York, 1991.

M. Csikszentmihalyi and I.S. Csikszentmihalyi. *Optimal experience: Psychological studies of flow in consciousness*. Cambridge University Press, 1992.

C. Deaker, M. Masoodian, and B. Rogers. A survey of players opinions on interface customization in world of warcraft. *Advances in Computer Entertainment*, pages 198–213, 2012.

J.Y. Douglas and A. Hargadon. The pleasures of immersion and engagement: Schemas, scripts and the fifth business. *Digital Creativity*, 12(3):153–166, 2001.

A.E. Elo. *The rating of chessplayers, past and present*, volume 3. Batsford London, 1978.

L. Findlater and K.Z. Gajos. Design space and evaluation challenges of adaptive graphical user interfaces. *AI Magazine*, 30(4):68, 2009.

K.Z. Gajos, K. Everitt, D.S. Tan, M. Czerwinski, and D.S. Weld. Predictability and accuracy in adaptive user interfaces. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1271–1274. ACM, 2008.

I. Geel. Mmodata.net, 2012. URL `http://www.mmodata.net`. Last accessed: February 19, 2013.

J. Holopainen and S. Meyers. Neuropsychology and game design. *Consciousness Reframed III*, 2000.

M.G. Jones. Creating electronic learning envnironments: Games, flow, and the user interface. In *Selected Research and Development Presentations at the National Convention of the Association for Educational Communications and Technology (AECT)*, pages 2803–2806, 1998.

P. Langley and M. Fehling. The experimental study of adaptive user interfaces. *Institute for the Study of Learning and Expertise*, 1998.

T. Lavie and J. Meyer. Benefits and costs of adaptive user interfaces. *International Journal of Human-Computer Studies*, 68(8):508–524, 2010.

C.A. Lindley and C. Sennersten. A cognitive framework for the analysis of game play: tasks, schemas and attention theory. In *Workshop on the cognitive science of games and game play, the 28th annual conference of the cognitive science society*, 2006.

A. Markov. Extension of the limit theorems of probability theory to a sum of variables connected in a chain. 1971.

Microsoft. Kinect — xbox.com, 2013. URL `http://www.xbox.com/en-US/kinect`.

Riot Games. League of legends. *Published by: Riot Games*, 2009.

P. Ross. Intelligent tutoring systems. *Journal of Computer Assisted Learning*, 3(4): 194–203, 1987.

R.C. Schank and R.P. Abelson. Scripts, plans, goals and understanding: An inquiry into human knowledge structures. 1977.

D. Sleeman, J.S. Brown, et al. Intelligent tutoring systems. 1982.

K. Spencer. Planning raid group composition 101, 2007. URL `http://priestlyendeavors.wordpress.com/2007/08/29/planning-raid-group-composition-101/`. Last accessed: February 19, 2013.

P. Sweetser and P. Wyeth. Gameflow: a model for evaluating player enjoyment in games. *Computers in Entertainment (CIE)*, 3(3):3–3, 2005.

J. Togelius, R. De Nardi, and S.M. Lucas. Making racing fun through player modeling and track evolution. 2006.

J. Togelius, R. De Nardi, and S.M. Lucas. Towards automatic personalised content creation for racing games. In *Computational Intelligence and Games, 2007. CIG 2007. IEEE Symposium on*, pages 252–259. IEEE, 2007.

T. Tsandilas et al. An empirical assessment of adaptation techniques. In *CHI'05 extended abstracts on Human factors in computing systems*, pages 2009–2012. ACM, 2005.

N. Van Hoorn, J. Togelius, D. Wierstra, and J. Schmidhuber. Robust player imitation using multiobjective evolution. In *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, pages 652–659. IEEE, 2009.

A.H. Watt and F. Policarpo. *3D games: real-time rendering and software technology*, volume 1. Addison-Wesley, 2001.

N. Yee. Motivations of play in MMORPGs: Results from a factor analytic approach. *The Daedalus Project, available at http://www. nickyee. com/daedalus/motivations. pdf*, 2005.

N. Yee. The demographics, motivations, and derived experiences of users of massively multi-user online graphical environments. *PRESENCE: Teleoperators and Virtual Environments*, 15:309–329, 2006.

N. Yee. Motivations of play in online games. *Journal of CyberPsychology and Behavior*, 9:772–775, 2007.

Appendix A

Ethical Approval for Survey of World of Warcraft Players

Computing and Mathematical Sciences
Rorohiko me ngā Pūtaiao Pāngarau
The University of Waikato
Private Bag 3105
Hamilton
New Zealand

Phone +64 7 838 4021
www.fcms.waikato.ac.nz

THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

3 April 2012

Chris Deaker
C/- Department of Computer Science
THE UNIVERSITY OF WAIKATO

Dear Chris

Request for approval to conduct a user study with human participants

I have considered your request to conduct an on line evaluation study for your research project *Adaptive Interfaces for Video Games* for your COMP594 course.

The purpose of the evaluation is to specifically identify the ways in which players of Massively Multiplayer online (MMO) games utilize customizable interfaces in a currently popular game (World of Warcraft). Reported results will not detail participant's names or any other identifying characteristics.

The procedure described in your request is acceptable.

The research participants' information sheet and consent form, meet the requirements of the University's human research ethics policies and procedures.

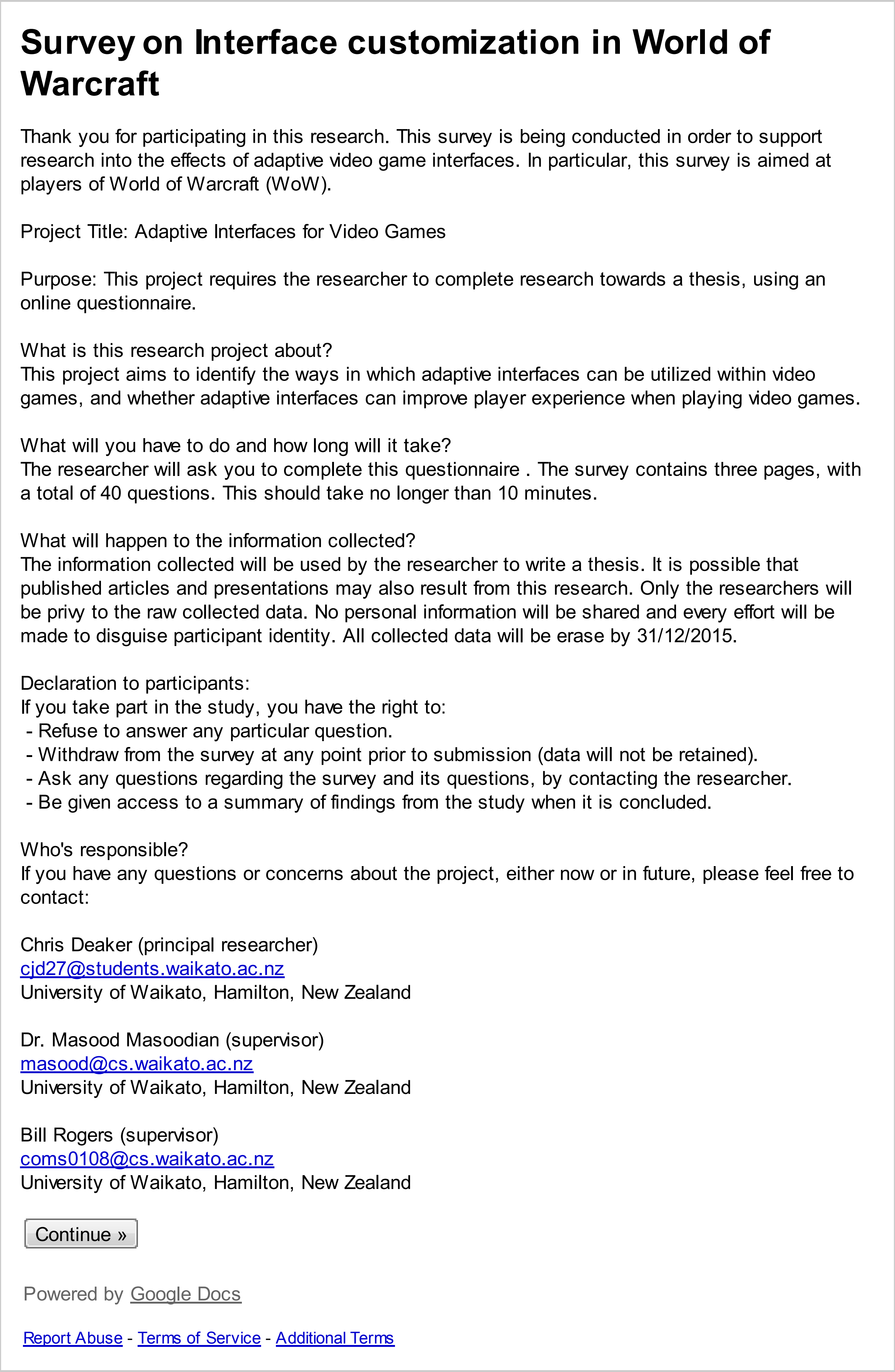
Yours sincerely,

Hunt.

Lyn Hunt
Human Research Ethics Committee
School of Computing and Mathematical Sciences

Appendix B

Survey Questions



Survey on Interface customization in World of Warcraft

Thank you for participating in this research. This survey is being conducted in order to support research into the effects of adaptive video game interfaces. In particular, this survey is aimed at players of World of Warcraft (WoW).

Project Title: Adaptive Interfaces for Video Games

Purpose: This project requires the researcher to complete research towards a thesis, using an online questionnaire.

What is this research project about?
This project aims to identify the ways in which adaptive interfaces can be utilized within video games, and whether adaptive interfaces can improve player experience when playing video games.

What will you have to do and how long will it take?
The researcher will ask you to complete this questionnaire . The survey contains three pages, with a total of 40 questions. This should take no longer than 10 minutes.

What will happen to the information collected?
The information collected will be used by the researcher to write a thesis. It is possible that published articles and presentations may also result from this research. Only the researchers will be privy to the raw collected data. No personal information will be shared and every effort will be made to disguise participant identity. All collected data will be erase by 31/12/2015.

Declaration to participants:
If you take part in the study, you have the right to:
- Refuse to answer any particular question.
- Withdraw from the survey at any point prior to submission (data will not be retained).
- Ask any questions regarding the survey and its questions, by contacting the researcher.
- Be given access to a summary of findings from the study when it is concluded.

Who's responsible?
If you have any questions or concerns about the project, either now or in future, please feel free to contact:

Chris Deaker (principal researcher)
cjd27@students.waikato.ac.nz
University of Waikato, Hamilton, New Zealand

Dr. Masood Masoodian (supervisor)
masood@cs.waikato.ac.nz
University of Waikato, Hamilton, New Zealand

Bill Rogers (supervisor)
coms0108@cs.waikato.ac.nz
University of Waikato, Hamilton, New Zealand

Survey on Interface customization in World of Warcraft

Participant information

Age:

Gender:

(Optional) If you would like to be contacted with information about the findings of this research, please provide your email address.

Contact details will remain confidential.

« Back   Continue »

Powered by Google Docs

Report Abuse - Terms of Service - Additional Terms

Survey on Interface customization in World of Warcraft

General gameplay information

This section contains questions relating to your play-style.

How many years have you been playing WoW for?

e.g. "6", "0.5"

On average, how many hours do you play WoW per week?

0 if you no longer play WoW.

Are any of your WoW characters currently at maximum level (level 85)?

○ Yes

○ No

How would you rank your level of knowledge of in-game mechanics?

	1	2	3	4	5	
Very low	○	○	○	○	○	Very high

Please rate the following aspects of WoW, according to their importance to you when playing:

	Very low	Low	Moderate	High	Very high
Advancement	○	○	○	○	○
Mechanics	○	○	○	○	○
Competition	○	○	○	○	○
Socializing	○	○	○	○	○
Relationships	○	○	○	○	○
Teamwork	○	○	○	○	○
Discovery	○	○	○	○	○
Role-playing	○	○	○	○	○
Character customization	○	○	○	○	○
Escapism	○	○	○	○	○

Please describe any other aspects of WoW which you feel are important (if not listed above).

« Back Continue »

Powered by Google Docs

Report Abuse - Terms of Service - Additional Terms

Survey on Interface customization in World of Warcraft

Interface modification

This section contains questions specific to interface modification in WoW. While some questions are specific to third-party interface add-ons, this also includes adjustment of settings provided by the default interface.

Do you modify your in-game interface using third-party 'mods' or add-ons?

○ Yes

○ No

How important do you feel the ability to modify your in-game interface is?

	1	2	3	4	5	
Not important	○	○	○	○	○	Very important

Do you enjoy the process of modifying your in-game interface?

	1	2	3	4	5	
Not at all	○	○	○	○	○	Very much

Please rate the following effects of interface customization, according to their importance to you when playing WoW:

	Very low	Low	Moderate	High	Very high
Removing unecessary information from the default interface	○	○	○	○	○
Providing additional information not available in the default interface	○	○	○	○	○
Improving the look and feel of the default interface	○	○	○	○	○
Providing easier access to important game functions or features	○	○	○	○	○

Please list any other effects of interface customization, which you feel are important to your gameplay experience (If not listed above):

How important do you feel interface customization is in relation to the following aspects of WoW?

	Not important	Slightly important	Moderately important	Quite important	Very important
Advancement	○	○	○	○	○
Mechanics	○	○	○	○	○
Competition	○	○	○	○	○
Socializing	○	○	○	○	○
Relationships	○	○	○	○	○
Teamwork	○	○	○	○	○
Discovery	○	○	○	○	○
Role-playing	○	○	○	○	○
Character customization	○	○	○	○	○
Escapism	○	○	○	○	○

Do you feel that some interface add-ons give players an unfair game advantage?

○ Yes

○ No

If yes, please elaborate:

Please list any third-party interface add-ons that you feel add value to your gameplay experience (either add-on descriptions or names are appropriate). For each add-on, please describe why you feel it is important to your gameplay experience.

Please describe any functionality that you feel would add value to your gameplay experience, not currently provided by the in-game interface or any third-party add-on.

« Back Submit

Powered by Google Docs

Report Abuse - Terms of Service - Additional Terms

Appendix C

List of Captured Game Events

The following WoW API events captured for processing. Not all captured events are used for modelling. Current API details are available from `http://wowprogramming.com/docs/events`.

Table C.1: Captured API events.

Event	**Fired when**
PLAYER_ENTER_COMBAT	The player begins melee auto-attack mode
PLAYER_LEAVE_COMBAT	The player stops melee auto-attack mode
PLAYER_TARGET_CHANGED	The player changes targets
UNIT_SPELLCAST_CHANNEL_START	A unit starts channeling a spell
UNIT_SPELLCAST_CHANNEL_STOP	A unit stops or cancels a channeled spell
UNIT_SPELLCAST_CHANNEL_UPDATE	A unit's channeled spell is interrupted or delayed
UNIT_SPELLCAST_DELAYED	A unit's spell cast is delayed
UNIT_SPELLCAST_INTERRUPTED	A unit's spell cast is interrupted
UNIT_SPELLCAST_START	A unit begins casting a spell
UNIT_SPELLCAST_STOP	A unit stops or cancels casting a spell
UNIT_SPELLCAST_SUCCEEDED	A unit's spell cast succeeds
UNIT_ENTERED_VEHICLE	A unit has entered a vehicle
UNIT_EXITED_VEHICLE	A unit has exited a vehicle
UNIT_AURA	A unit loses or gains a buff or debuff

Appendix D

Sample Event Log

The following is a short de-serialised excerpt from a player event log, saved into a Lua array. Formatting, automatic commenting and indentation is performed by the WoW client.

```
{
        "UNIT_SPELLCAST_SUCCEEDED", -- [1]
        {
                "player", -- [1]
                "Jade Raccoon Despawn Aura - HW", -- [2]
                "", -- [3]
                0, -- [4]
                122732, -- [5]
        }, -- [2]
        1350523131, -- [3]
}, -- [1]
{
        "UNIT_SPELLCAST_SUCCEEDED", -- [1]
        {
                "player", -- [1]
                "Has Tabard", -- [2]
                "", -- [3]
                0, -- [4]
                57818, -- [5]
        }, -- [2]
        1350523131, -- [3]
}, -- [2]
```

```
{
        "UNIT_SPELLCAST_SUCCEEDED", -- [1]
        {
                "player", -- [1]
                "Orgrimmar Champion", -- [2]
                "", -- [3]
                0, -- [4]
                93825, -- [5]
        }, -- [2]
        1350523131, -- [3]
}, -- [3]
{
        "UNIT_SPELLCAST_SUCCEEDED", -- [1]
        {
                "player", -- [1]
                "LOGINEFFECT", -- [2]
                "", -- [3]
                0, -- [4]
                836, -- [5]
        }, -- [2]
        1350523131, -- [3]
}, -- [4]
{
        "UNIT_SPELLCAST_SUCCEEDED", -- [1]
        {
                "player", -- [1]
                "Shadowform", -- [2]
                "", -- [3]
                1, -- [4]
                15473, -- [5]
        }, -- [2]
        1350523148, -- [3]
}, -- [5]
{
        "PLAYER_TARGET_CHANGED", -- [1]
        {
        }, -- [2]
        1350523149, -- [3]
```

```
}, -- [6]
{
        "UNIT_SPELLCAST_START", -- [1]
        {
                "player", -- [1]
                "Vampiric Touch", -- [2]
                "", -- [3]
                2, -- [4]
                34914, -- [5]
        }, -- [2]
        1350523151, -- [3]
}, -- [7]
{
        "UNIT_SPELLCAST_SUCCEEDED", -- [1]
        {
                "player", -- [1]
                "Vampiric Touch", -- [2]
                "", -- [3]
                2, -- [4]
                34914, -- [5]
        }, -- [2]
        1350523152, -- [3]
}, -- [8]
{
        "UNIT_SPELLCAST_STOP", -- [1]
        {
                "player", -- [1]
                "Vampiric Touch", -- [2]
                "", -- [3]
                2, -- [4]
                34914, -- [5]
        }, -- [2]
        1350523152, -- [3]
}, -- [9]
{
        "UNIT_SPELLCAST_START", -- [1]
        {
                "player", -- [1]
```

```
                        "Mind Blast", -- [2]
                        "", -- [3]
                        3, -- [4]
                        8092, -- [5]
                }, -- [2]
                1350523152, -- [3]
        }, -- [10]
        {
                "UNIT_SPELLCAST_SUCCEEDED", -- [1]
                {
                        "player", -- [1]
                        "Mind Blast", -- [2]
                        "", -- [3]
                        3, -- [4]
                        8092, -- [5]
                }, -- [2]
                1350523154, -- [3]
        }, -- [11]
        {
                "UNIT_SPELLCAST_STOP", -- [1]
                {
                        "player", -- [1]
                        "Mind Blast", -- [2]
                        "", -- [3]
                        3, -- [4]
                        8092, -- [5]
                }, -- [2]
                1350523154, -- [3]
        }, -- [12]
        {
                "UNIT_SPELLCAST_SUCCEEDED", -- [1]
                {
                        "player", -- [1]
                        "Shadow Word: Pain", -- [2]
                        "", -- [3]
                        4, -- [4]
                        589, -- [5]
                }, -- [2]
        1350523154, -- [3]
}, -- [13]
{
        "UNIT_SPELLCAST_CHANNEL_START", -- [1]
        {
                "player", -- [1]
                "Mind Flay", -- [2]
                "", -- [3]
                0, -- [4]
                15407, -- [5]
        }, -- [2]
        1350523156, -- [3]
}, -- [14]
{
        "UNIT_SPELLCAST_SUCCEEDED", -- [1]
        {
                "player", -- [1]
                "Mind Flay", -- [2]
                "", -- [3]
                5, -- [4]
                15407, -- [5]
        }, -- [2]
        1350523156, -- [3]
}, -- [15]
{
        "UNIT_SPELLCAST_SUCCEEDED", -- [1]
        {
                "target", -- [1]
                "Shadow Claw", -- [2]
                "", -- [3]
                0, -- [4]
                116128, -- [5]
        }, -- [2]
        1350523157, -- [3]
}, -- [16]
```

Appendix E

Ethical Approval for Prototype Evaluations

Computing and Mathematical Sciences
Rorohiko me ngā Pūtaiao Pāngarau
The University of Waikato
Private Bag 3105
Hamilton
New Zealand

Phone +64 7 838 4021
www.fcms.waikato.ac.nz

THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

15 November 2012

Chris Deaker
C/- Department of Computer Science
THE UNIVERSITY OF WAIKATO

Dear Chris

Request for approval to conduct a user study with human participants

I have considered your request to conduct a further on line evaluation study for your research project *Adaptive Interfaces for Video Games* for your COMP594 course.

The purpose of the evaluation is to investigate the benefits and possible applications of adaptive user interfaces within video games, specifically it aims to collect feedback ona prototype application which augments the default World of Warcraft interface with adaptive feedback. Personal information collected will be limited to the participants gender and age.

The procedure described in your request is acceptable.

The research participants' information sheet and consent form, meet the requirements of the University's human research ethics policies and procedures.

Yours sincerely,

LA Hunt.

Lyn Hunt
Human Research Ethics Committee
School of Computing and Mathematical Sciences