USING GAME METRICS TO DRIVE REAL-TIME GAME ADAPTION FOR A MORE ENGAGING AND USEFUL TUTORIAL LEVEL

Michael James Laidlow

MSc Computer Games Technology

University of Abertay Dundee
Institute of Arts, Media and Computer Games
January 2013

University of Abertay Dundee

Permission to copy

Author: Michael James Laidlow

Title: Using Game Metrics to Drive Real-Time Game Adaption for a More Engaging and Useful Tutorial Level

Degree: MSc Computer Games Technology

Year: 2013

(i) I certify that the above mentioned project is my original work

(ii) I agree that this dissertation may be reproduced, stored or transmitted, in any form and by any means without the written consent of the undersigned.

Signature ……………………..

Date …………………………..

Abstract

Designing the tutorial level can be one of the most important aspects of the entire game. To new players, tutorial levels represent the very first experience with the game and can create a lasting impression. Developers are challenged in preparing a game tutorial of the core mechanics appropriate to a game audience with widely varying skills and backgrounds. If the tutorial level is too easy, experienced users will be bored, but beginners will become frustrated if it is made too hard. This dissertation attempts to overcome this problem with the use of real-time game adaption based on in-game metrics. The objective is to develop a more engaging, individualized and specialized learning process.

This goal was achieved by game metrics being chosen to monitor the most important aspects of a game and influence the mechanics associated with them. The evaluation involved players completing two matches that included either both an adaptive enabled match and a static match chosen at random or a control group of only static matches. Game metrics were recorded, analyzed and compared from the players' actions, including player or enemy hits, rewards collected, use of healing and protection and time of play. Players were asked to complete a questionnaire in order to gather their impressions and opinions of the game.

Results support that the tutorial level experience can be improved through adaptive gameplay. Players preferred adaptive matches and achieved higher results in metrics when playing the adaptive match first before playing the static match, when compared to the control group.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1: Tutorial Level Overview

Recent technological advancements in computers, coupled with the exponentially growing popularity of video games, have led to video games to becoming increasingly more complex. Audiences expect every new video game to be better than each of its predecessors. At the price of complexity, there are several ways a developer can improve their video game with advancement in each of the iterations, such as graphics, different game mechanics, artificial intelligence of enemies and others.

In the early days of the video game industry, tutorial levels, to teach the player the mechanics of the game, were relatively uncommon. These early video games usually consisted of using only one or two buttons and the mechanics were much simpler. For example, in the classic Nintendo game Super Mario Bros (1985), the player would simply use the D-Pad to run on a 2D plane and press a button to jump.

On moving fast forward to current games, the complexity is exponentially greater. In the genre first–person shooters, most now require a player to move their character in 3D space, while simultaneously using a different hand to move a camera in 3D space. On top of the basic movement controls, there are buttons to sprint, jump, change gun, fire, secondary fire, throw grenades, melee attack and do other moves. With this rise in complexity, tutorial levels have become much more of a regular occurrence and a requirement in modern video games.

Nowadays, the players’ first impression of a game is often a tutorial in the form of the first level of the game. Many games attempt to show the basic core mechanics with a quick explanation and the display of an icon of the mapped button to press. Once the player presses this button, the learning of that action is usually over and the game moves on to a different concept to be emphasized. There is little testing done to assess if the player has learned the new mechanic or just simply learned to press the button shown on screen with subsequent learning to be based off exploration.

While tutorial levels have helped players to some extent, there remains a problem that potential players for a new game come from a variety of different backgrounds with differing levels of skill. Most games attempt to overcome this by having different game modes, such as easy, medium, and hard. The easy mode would have fewer enemies, less damage taken by the player and less accurate enemies compared to its hard mode counterpart. This attempts to ease in players of different skills so that they can learn the mechanics, be provided with a challenge and eventually work their way up to playing at harder levels. This isn't a perfect remedy, as players have to generalize their skills before choosing a difficulty. A player may be good at one skill such as shooting their gun but bad at another, such as driving a car, making one aspect of the subsequent game too easy and another too hard.

Tutorials can overwhelm a player by bombarding them with new information when the player hasn't grasped an earlier concept, which can cause the player to get confused or simply forget about a past mechanic. In Infinity Ward's game Call of Duty 4: Modern Warfare (2007), players take control of a new recruit, who gets shown the basics at a firing range. The game doesn't move forward until they perform the skill required. Those skills might include shooting from the hip, shooting zoomed in, using the knife and additional moves. Afterwards, the player moves to a small mission where their skill is tested through an obstacle course. Based on the results they achieved at the obstacle course, a game difficulty is recommended to the player and the tutorial ends. While this is a step in the right direction, the obstacle course does not offer any feedback beside the game difficulty recommendation.

Tutorials are often short, so that experienced players don't get bored but if the process is too short, inexperienced players fail to get the instruction that they need to understand, enjoy and succeed in the game. A solution to this problem can be achieved by having a game tutorial that adapts to the skill of the player and tailors itself to the individual player. Not only will this help to teach players all of the mechanics, it will also provide a more entertaining and engaging level that will keep the interest of experienced players over the standard static counterpart.

By using game metrics, developers can monitor almost any action that a player takes, including shooting accuracy, time taken to get between obstacles, gun choice, to which buttons have been pressed and more. This information is vital in determining how a player actually played, as opposed to how the player was intended to play the game.

At the moment, most game metrics are used offline, meaning that developers release the game and record data. From there, they can release updates for the game based on the metric data they gathered. This has proven to be very successful in several games. The approach has limitations, as it creates generalizations and, instead of helping specific players, it primarily monitors the majority of the population using the game, which may not be the ones that you would most want to help in individual tutorial levels in order to grow the general appeal of the game.

Using real-time game adaption based on player metrics is still a relatively new technique. It has gained popularity, such as in creating dynamically controlled enemies that would adjust according to how the player is succeeding in the game. While this is a good idea in theory, it has yet to be seen to be of great usefulness in practice, as there are several problems that a developer must overcome with this approach. Tutorial levels are much simpler, as they only involve a level and not an entire game. Its main focus is in introducing and teaching the mechanics of the game, as opposed to providing the player with the greatest individual challenge through an entire game experience.

1.2: Aim

The aim of this dissertation is to focus on improving the players' skill by identifying weaknesses in their gameplay, adapting gameplay to the player and having the player increase their all-around skill level upon leaving the tutorial level due to a better understanding of the game mechanics. The real-time game adaption will be achieved by the use of in-game metrics and this experience can lead to a more engaging and useful tutorial level for players.

1.3: Dissertation Structure

This dissertation has been divided up into the following structure:

- Chapter 2: Literature Review - More in-depth background about tutorial levels in games, the use of game metrics and real-time game adaption including dynamics difficulties.
- Chapter 3: Methodology - The process of how the experiment was designed, created, and how the experiment will be carried out for testing to achieve the goal of the dissertation.
- Chapter 4: Results – The resulting prototype and data collected from testing.
- Chapter 5: Discussion - Dissecting the results found in Chapter 4 and finding which expectations succeeded and which expectations failed.
- Chapter 6: Conclusion and Future work - Comparison of the expectations from the beginning implementation, the results achieved, and what can be gained through future extensions of this project.
- Chapter 7: Appendices - All research data collected such as questionnaire data and metrics recorded, DVD structure, and website to access the video game.
- Chapter 8: References & Bibliography - A list of all references and sources.

Chapter 2: Literature Review

2.1: Real-Time Adaption in Tutorial Levels

As the popularity of video games has grown, developers have had to take into consideration the wide range of varying players' abilities. One of the ways that developers have attempted to combat this problem is through the use of tutorial levels to teach core mechanics. Unfortunately, it is a difficult balance for developers. One player may understand a mechanic instantly, while others may take some time. Tutorial levels are usually the first experience that a player has with a new game and the first impression is of vital importance. If the tutorial is too hard or confusing, the player may quit. At the same time, if the tutorial is too easy, players will find no challenge and get bored. A tutorial that adapts to the player in real-time, as the player plays the game, can make this task easier for developers and more useful for players.

2.2: Flow in Video Games

The idea of flow, a theory of Mihaly Csikszentmihalyi, is "the feeling of complete and energized focus in an activity, with a high level of enjoyment and fulfilment" (Chen 2008). The goal of any video game designer is to create a perfect flow for their video game. As seen in Figure 2.1, the perfect balance is sought between the challenge of completing the activity and the player's actual abilities.

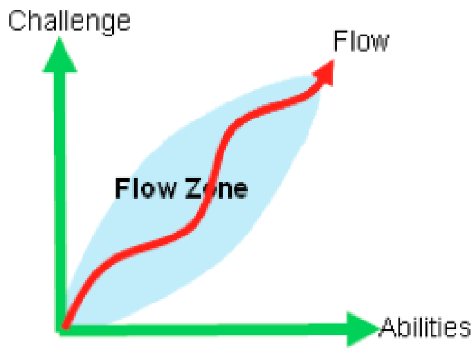

Figure 2.1: Player Flow Experience (Chen 2008).

As shown in Figure 2.2, if the balance is wrong, it can cause players to leave their flow zone. A player can become overwhelmed by a new challenge which may cause anxiety, or alternately lose interest from having no challenge at all (Chen 2008).

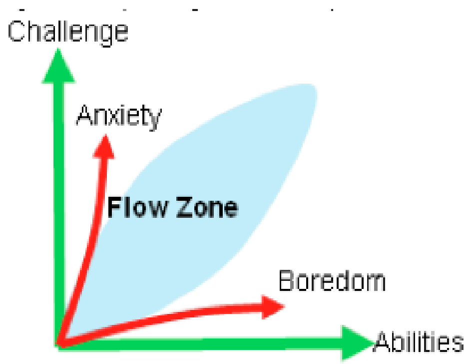

Figure 2.2: Player Psychic Entropies (Chen 2008).

Furthermore, the difficulties of creating a flow-like experience for a single person can be a difficult, making it a daunting task to attempt to create flow for millions of users who are attempting to play a new game. Users that may have played a similar game would have some background knowledge on how the game is to be played. On the other hand, first time players would have no familiarity (Chen 2008).

Figure 2.3 shows the different angles of flow zones for these types of users. The "hardcore" player requires much more of a challenge for their amount of skill and is represented with the steeper angle of their flow zone. The "novice" is the exact opposite and requires the challenge for their skill to be much lower, represented with the flatter angle.

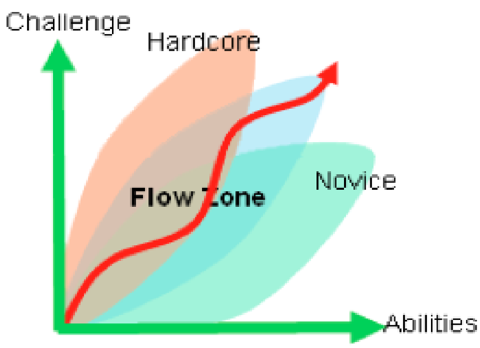

Figure 2.3: Different Skilled Players Flow Zones (Chen 2008).

2.3: The Need for Tutorial Levels

Tutorial Levels have gained popularity as the complexity of games have increased. Back in the 1980's, video games were much simpler, with only a few buttons or actions available. In Alexey Pajitnov's popular video game

Tetris (1986), players had a limited number of options available to them. A player could move the "Tetriminos" horizontally across the screen or could rotate the "Tetriminos" orientation by 90 degrees. The only aim of the game was to create a horizontal line of 10 blocks without any gaps, which would cause the line to disappear. Once the blocks piled to the top of the screen, the game was over. This whole process could continue for an infinitely long period of time. Tetris (1986) was immensely popular, selling over 70 million copies worldwide (Johnson 2009). This flagship video game was made without a tutorial level, because the game at its core was simple enough that player didn't require to be informed of all the possible actions. Tutorial levels for simpler games, like Tetris (1986), were not needed as the player could figure out the controls as they explored playing the game for the first couple of times (Andersen et al. 2012). However, with the rise of technology and audience expectations, video games are getting increasingly more complex (Chapresto, Mitchell and Seron 2011). With this complexity, tutorial levels, which are often the players' first experience with a new game, are used to teach players the game mechanics that they need to know. There is not much information about tutorials effectiveness on a player's learnability (Andersen et al. 2012). Andersen et al (2012) studied 45,000 people in an attempt to show the usefulness of tutorials. Players were involved in 3 different types of games that varied in complexity and conventionalism. In the most complex game, it was discovered that the play time of the actual game went up by as much as 29%, with player progression through the game increased to a high of 75%. In another game, giving tutorial instructions as the player needed them instead of having them out of context in a manual, increased play time by up to 16% and increased players progression with a high of 40%. Andersen et al (2012) concluded that there was evidence to support that the impact of a tutorial on a game was also related to its complexity, with greater increases in playing time and progression seen with the more complex games.

2.4: Issues with Tutorial Levels

While there is evidence to support the need for tutorial levels in complex video games, there are still several issues that need to be fixed. If solved, they could provide a more useful and efficient tutorial for players, by making them more engaging and as well as improving the learnability of game mechanics.

As Andersen et al (2012) mentioned before, game developers have little advance access to the results of learnability from the tutorials of the games they have designed. With the traditional sales model for most games, especially for console systems, a game's development time continues up until the date of the release of the game in retail stores (Angle 2012). This makes it hard to get feedback from the millions of people who will potentially play the game in advance of the release.

Another issue is that with the rise in popularity of video games, players' skill ranges are now more varied than ever before. Designers have a hard time balancing out the game for all players to perfect the flow of their game. This significant problem is in place for the entirety of the video games and designers have attempted to combat it by optioning different levels of difficulty (Chen, 2008). However, tutorial levels don't usually deal with difficulty, as their main purpose is to teach the mechanics of the game. As a result, the flow is much harder to line up, as some players have had several hours playing games similar to a new game while others would be completely naïve to it.

2.5: Use of Game Metrics

A technique that has been used in the human-computer interaction field, recently gaining popularity in the video game industry, is game metrics (Drachen and Canossa 2009). Game metrics is the measurement of any in-game measurable data. This can range from determining buttons that a user presses, areas of a map explored, weapons used, health, time taken to play a level and other behaviour variables (Drachen and Canossa 2009). Canossa and Cheong (2011), along with several others in the video game industry place great importance on these measurements. They believe that embedded in the measurements is the potential to show the motivations, desires, beliefs and personality of a player. By analyzing this data, games can be tailored in how the game is being played, such as whether the player is using all the game mechanics as designed, or whether players are taking too long in certain aspects of the game and other behaviour.

In recent years, with the popularity of social and mobile games, there has been a shift of focus toward the free-to-play video game sales model over the traditional approach (Angle 2012). Companies like the social game company, Zynga, are using metrics to drive the success of their games using this model (Canossa and Cheong 2011). The free-to-play sales model is based on the fact that anyone can play the game for free. The companies make money by charging for additional content to the game itself, such as extra items, levels, quests and other downloadable content (Angle 2012). In order to encourage players to purchase these non-necessities, they need to entice the player. Players are far more willing to justify a purchase of add-on non-essential material to keep playing something they enjoy (Angle 2012). Free-to-play has proven very successful and according to the Casual Games Sector Report (2012), worldwide free-to-play massively multiplayer online games revenue is over twice as large as the subscription counterpart.

In order to achieve such success, the free-to-play model is in a constant development process for the entire games lifecycle. The initial phase of the game is planned out with several carefully selected and planned game metrics in place that the company can use to maximize the potential of their game (Angle 2012). From here, the game is developed and released to the market. As individual players play the game, the game metrics are recorded and sent to a server controlled by the development team, in a similar fashion to how leader boards or shop data work. Once enough players have played the game, the company can analyze the results and, through the numbers recorded and the statistics found, provide an update to their game in order to make it more enjoyable for the average player (Angle 2012). Tychsen (2008) gives an example of metrics that can be used from a low level such as button presses, to the movement of player-characters within a game world and what weapons are being used.

Game metrics have a distinct advantage over other forms of user research. Opinion based research, such as interviews or surveys, are slower and can contain error. Players may not know why they don't like a certain aspect of the game or the issue identified may not be the root of the problem. Other forms of research, such as video capture can take a significant amount of time and rely on observational input. Game metrics are fast and accurate if designed properly and they can be gathered from a vast number of players easily (Chapresto, Mitchell and Seron 2011).

Planning a game metric must be carefully done. Metrics are a powerful tool and can give you accurate information about a player, but metrics are nothing more than an indicator (Klubeck 2011a). Metrics usefulness only comes from what a developer can derive from the available data. Metrics should be built upon the very root of the question that the developer is trying to answer through the use of testing on the users. Otherwise, developers can gather information on something unexpected. Klubeck (2011a) gives an example of using metrics poorly from the rankings of Amazon book sales. The rankings are supposed to determine how a book’s sales are doing in comparison to other books being sold on Amazon. In the sale ranking, a book is ranked higher if the book has more orders. However, the rankings do not take into consideration some crucial information, such as the quantity of books ordered in each individual sale. A book that has 1000 books ordered in one sale would be ranked lower than a book that has 500 individual single orders.

2.6: Real-Time Game Adaption Using Game Metrics

The idea of adapting a game to an individual player, based on how the player behaves, is a new topic that is interesting many video game designers. Heavily influenced by the theory of flow, game design attempts to reach the ideal ratio of challenge to ability but must take into account that the skill of the players playing a game varies tremendously. Players can have hugely different backgrounds now compared to a couple of decades ago. Developers have to consider the "skills, preferences, and emotional elicitation that can differ widely among prospective players" (Yannakakis and Togelius 2011). Instead of the usual utilization of game metrics, involving the gathering and analyzing information off several players and making general adjustments based off the entire user base, the game responds to controllable parameters in real-time, in an attempt to give the individual player the best possible challenge and enjoyment factor (Yannakakis, and Hallam 2009).

Yannakis and Hallam (2009) did a study on real time game adaption by influencing factors with game metrics on 24 children, 13 boys and 11 girls, from the ages of 8 - 11. The children each played a game specifically designed for this experiment called "Bug Smasher". In this game, the player stood on top of a 6 by 6 square tile topology. In each of the tiles there was a light that would light up to represent a bug. The goal of the player was to step on the corresponding tile to "squash the bug". The bug's position was determined at random, based off a predefined level of spatial diversity on where the previous bugs had been placed. Each child played "Bug Smasher" four times. Two times, the game metrics were disabled and the game remained static. The other two times, the game metrics were enabled and the game adapted to how the child played the game. The adaptive version of “Bug Smasher” took into consideration the average response time of the child and the variances in pressure of the tiles as the child stepped down to squash the bug and adjusted the appearance of the next bug's position accordingly. The adaption occurred 3 times throughout the gameplay at 45, 60 and 75 seconds. Yannakis and Hallam (2009) concluded that the experiment was successful at improving the game with real time adaptions as the children, through a questionnaire, preferred the adaptive game.

2.7: Dynamic Difficulties

One of the more well-known uses of real-time game adaption comes in the form of dynamic difficulties. Many people, especially in the artificial intelligence field, believe that dynamic difficulties are severely underused in most games so far. Most games have static enemies that come in different categories, such as easy, medium, or hard, that the player has selected at the beginning of the game. However, it is difficult to provide the right challenge for everyone if flow is the main concern of the developer (Hunicke 2005). One way dynamic difficulties can be used in game metrics is by having the attributes of enemies change (Hunicke 2005).

However, Adams (2008) has pointed out that there are some significant problems in the utilization of dynamic difficulty that has limited its use in the mainstream of the video game industry for big game releases. Many players enjoy the challenge of a game, even if the initial attempt, results in their character dying, particularly if their character can respawn and attempt the situation again, ultimately becoming successful. To work hard and overcome the obstacle or enemy is likely to be far more rewarding with the difficulty remaining the same as opposed to accomplishing it by having the obstacle made easier or the enemy less formidable in the next attempt, which many believe would just weaken the overall game. It is also possible to take advantage of a system that monitors a player's behaviour in order to get better rewards. Adams (2008) gives the example of a player that could get more ammunition when their health was below a certain level. A player could just injure themselves in order to receive the greater reward, instead of playing the game in the way that it was meant to be played. Lastly, it is hard to adapt certain aspects of the game such as puzzles. It is much easier to change the parameters of enemies for accuracy, damage to the player and overall health.

2.8: Real-Time Adapting Tutorials

Currently, developers do not spend a lot of resources on the tutorial level of a game. They are often static and quickly created. There is usually no attempt to measure the learnability of the players' skills.

Game metrics are predominately used offline. The metrics are gathered and analyzed, with the results used to potentially influence future iterations of the game. It has proven to be successful and is a viable market tool in social and mobile games.

Real-time adaption is a technique that has gained a lot of interest. Using metrics to adapt the gameplay to the individual player, rather than to a generalized audience, is instantaneous compared to the wait time of offline metrics. It has potential flaws, as it can be difficult to fine tune, which is the main reason that it isn't being used mainstream, at the moment.

Using game metrics to drive real-time adaption of tutorials can solve the above issues and provide players a better chance to understand the game at their own pace in an environment that is suited to their needs. A tutorial level is a relatively small piece of the game with a lot of measurable attributes and is the player’s first experience with a game. Real-time game adaption in tutorials could provide a useful tool for developers in showcasing their game and helping potential players to fully experience what the game is capable of providing to them, maximizing the entertainment value and appeal.

Chapter 3: Methodology

3.1: Overview

Through the background research in Chapter 2, it has been established that in order to determine the effectiveness of using game metrics to drive real-time adaption for tutorials, an application will have to be developed that can measure the abilities of players in their utilization of a tutorial. The application measured results must be comparable in order to determine if any improvement occurred in regard to the players skill level.

3.2: Application Requirements

To design a real-time adapting tutorial to test on several players, the application must contain two different versions. It must contain a static version, where variables do not change throughout the course of the game, and an adaptive version, where the variables vary throughout the gameplay. Therefore, in the level design, there must be several key mechanics with controllable parameters that can be changed depending to the player’s actions. Furthermore, the game metrics that these key game mechanics use must be able to be easily measured and monitored in order to be compared after the testing. Finally, a questionnaire must be developed to gather the opinions of the matches that the players just played.

3.3: Application Demonstration Metrics

The gameplay metrics will be chosen based on the broken down components of the gameplay itself. There must be a metric for each important part of the gameplay. Each metric chosen must provide valuable information that the controllable parameters can use to influence how the gameplay is working. If a player is using the mechanic well, it should be shown in the metric monitoring and vice versa. Lastly, in the case of adaptive mode being enabled, the game metrics must be simple enough that they can be fed back into the game mechanics themselves so that adjustments can be made.

3.4: Testing Process

The testing process is a vital part of the methodology. In order to get the best results, the application will need to be played by as many people as possible. The easiest way to maximize the number of people playing is by creating an easy delivery method and letting players have access to it at their own convenience. This can be achieved through hosting the application on a website and by giving willing participants around the world a link to access it. After a player plays the application, the game is able to send the game metrics and questionnaire results from each match separately and anonymously by email to a central location for later analysis.

In an attempt to make this a blind study and avoiding having a bias, users will not be notified as to what is specifically being measured or whether they are playing the adaptive version of the game or the static version. Furthermore, in order to gather the most accurate results possible, the study will be broken down into a couple of variations. The first group of participants will play 3 matches. The first match will be the static version of the game and will not be recorded. This is due to provide players an idea of what to expect in the later rounds so that by the second match, users should have an elementary understanding of how the game works. In the second and third matches, players will have a 50% chance of playing the adaptive version of the game first followed by the static game or otherwise, they will play the static version of the game first followed by the adaptive version in the third match. The results from this group can give an unbiased opinion of the preference of the game that they prefer. These results can be used to determine the effectiveness of the game adaption. On average, players should have better results when the adaptive game is enabled. Furthermore, the players should be more familiar with the game by the third round and therefore should perform better regardless of whether the game adaptions are enabled or not. With the adaptive game enabled before the static version, players should be more prepared to handle the challenges of the next game and a greater degree of improvement should be made in the third match. For the group of players that played the static version in the second match, it would be expected that they should not show significantly higher results in the next match. Finally, another group of users will be tested with the 3 matches approach, but these users will only play static versions of the game. This group will be used as a control group. The third game of this match can be compared with the group that played the static match in the third match and the adaptive in the second match to see if there was any significant improvement overall as a result of exposure to the adaptive match.

3.5: Information Analyzing Process

The data collected through all of the tests above will be compared and analyzed. This will be done by comparing game metrics in each of the situations, in order to determine whether there is a significant improvement in the results one way or the other. The mean and the standard deviation for each set of data for the game metrics will be calculated. The average will give an idea of the estimated amount of what the player achieved in the associated game mechanic and the standard deviation will indicate how much variation there is within the data (Weisstein [no date]). Finally, it will be determined if there is a significant difference between the two sets of data by the use of a T-Test. A T-Test, gives the probability that there is no difference between the two observed sets of data and that the difference in distribution is just by chance. If the probability is lower than 5% it supports that the sets of data are independent of each other. The greater the probability from the T-Test, the more likely that the difference in the set of data is due to chance (Weisstein [no date]). Since the number of participants will be relatively small, the variation in this probability will likely not completely prove the result one way or another but will provide an evaluation of the concept and a potential base from which to consider further and more extensive study.

Chapter 4: Results

4.1: Overview

This chapter contains the practical work of the application designed from the requirements, testing methods, and analyzing techniques of chapter 3.

4.2: The Application

The application created is a playable game that can switch between having real-time game adaption enabled or a disabled static game. It was developed using the game engine Unity3D and was exported using Unity3D's web player option, which can be hosted from a website.

The player takes control of a moveable character that is repositioned by the arrow keys on the keyboard and uses the spacebar to attack. The goal of the game is to collect as many coins as possible before falling off the edge of the 13 by 13 terrain. The player collects coins by moving to where the coin is placed on the grid. Once a coin is collected or the lifetime of a coin depletes, in 5 - 10 seconds, the coin will be removed and a new coin will be spawned at a random location in one of the four quadrants in the grid. There is a maximum of 5 coins being available on the grid at one time.

There are a couple of obstacles that the user must be aware of and avoid. The first obstacle is the terrain. As seen in Figure 3.1, the terrain is divided into 5 sections. The 4 quadrants contain an even number of squares for each corner of the map. The last section is a cross that is in the very centre of the map. The cross section is static and will never change. The other

4 quadrants, however, will have pieces that will slowly disappear over time. As the game carries on, if a quadrant is chosen to have one of its squares taken away, that square will turn yellow with a cracked texture as a warning to players that it is going to disappear soon. Then the whole square will be removed and will serve as a permanent hole that the player can fall through.

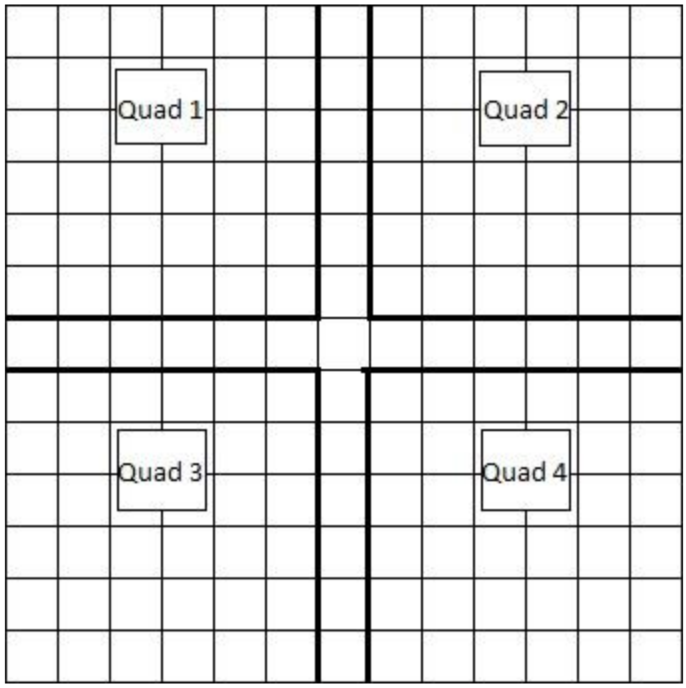

Figure 3.1: Quadrant Layout.

The second obstacle is the enemies, in the form of ghosts. Ghosts appear soon after the beginning of the game and will wander the grid, until the user approaches within a radius of the ghosts, known as the search radius. Once a player is in the search radius, the ghost will shift focus to the player, increase slightly in speed and attempt to hit the player if the ghost is in close enough range to the player. The player can attack the ghosts through the use of the spacebar button, which pushes the ghosts back a set amount of distance. The player can even push ghosts off the terrain edge or into a hole, caused by the terrain being destroyed earlier, which will cause the ghost to respawn after a delay of a couple of seconds. However, if the ghost is able to make contact and hit the player, the player's damage is increased and the player moves a distance that is proportional to the damage that the user has already taken. For example, if a player has never been hit by a ghost and has 0% damage, the player will only be moved one square on the grid. But, if the player has been hit several times from a ghost and has 20% damage, the player will be moved three squares on the grid. Players must become more careful as their damage rises or the risk of falling off the edge or into a hole dramatically increases.

The only source of protection and healing in the game is the light source. There is a small radius of light that moves around the entire terrain. Ghosts actively avoid the light. If they enter the light, they will immediately increase in speed and go outside of the light, where they return back to normal speed. The player can use the light to their advantage, not only for the protection aspect of the light but also for healing. As the player stands in the light, the damage will be lowered. A game screenshot can be seen below in Figure 3.2.

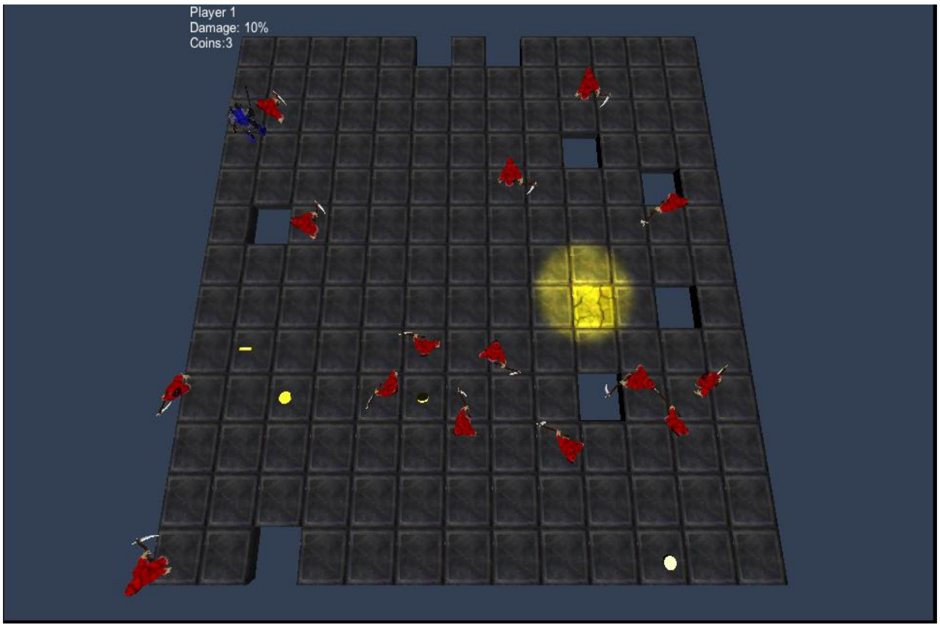

Figure 3.2: Gameplay Screenshot.

4.3: Metrics Monitored

The design of the game was made with clear and concise core mechanics that can be monitored by the game itself. In this game, these metrics are being recorded and saved after each match that a player has played, regardless of whether the player is playing the game with game metrics enabled to drive adaption or disabled for a static level. In this game, the following metrics were monitored:

- Whether having real-time game adaptions are enabled or disabled
- Amount of terrain left in each quadrant.
- The time a player spends in the light.
- The time a player does not spend in the light
- The number of coins collected.
- The number of times a player has been hit by a ghost
- The number of times a ghost has been hit by the player.
- The ghost’s speed while chasing the player
- The ghost’s search radius.
- Time spent in each terrain
- The amount the player has healed
- Time spent since last coin was collected

4.4: Adaption

Using the above monitored game metrics, the game can be adapted to how the player behaves within the game. The following game mechanics can be influenced by the game metrics:

- Light movement and radius size
- Which quadrant the terrain falls from
- Where and when coins spawn in the grid
- Ghost speed and search radius

The light mechanic of the game changes in a couple of aspects when real-time game adaption is enabled. As shown in Figure 3.3, in the static version of the game, the light is moved completely at random with the light radius also fluctuating in size at random. Conversely, in the real-time adaption version, the light mechanic is influenced through the analysis of the player's game metrics. In situations where the metrics determine the player needs the light, the light is moved to the quadrant that the player is in with the light radius increased to the maximum size. In situations where the metrics determine that a player shouldn't need the light, the light is moved to the quadrant that the player is not in and shrinks to a minimum light radius. Otherwise, the light moves at random as occurred in the static game. As seen in Figure 3.4. The light is determined by several factors.

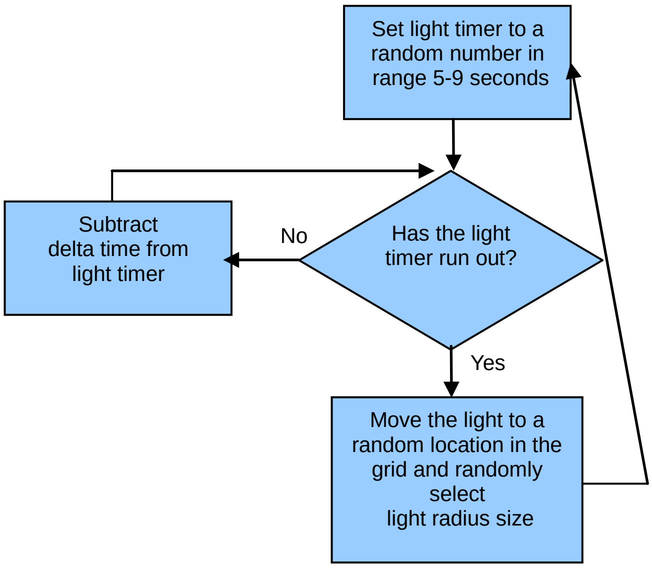

Figure 3.3: Static Light Mechanic Flow Chart.

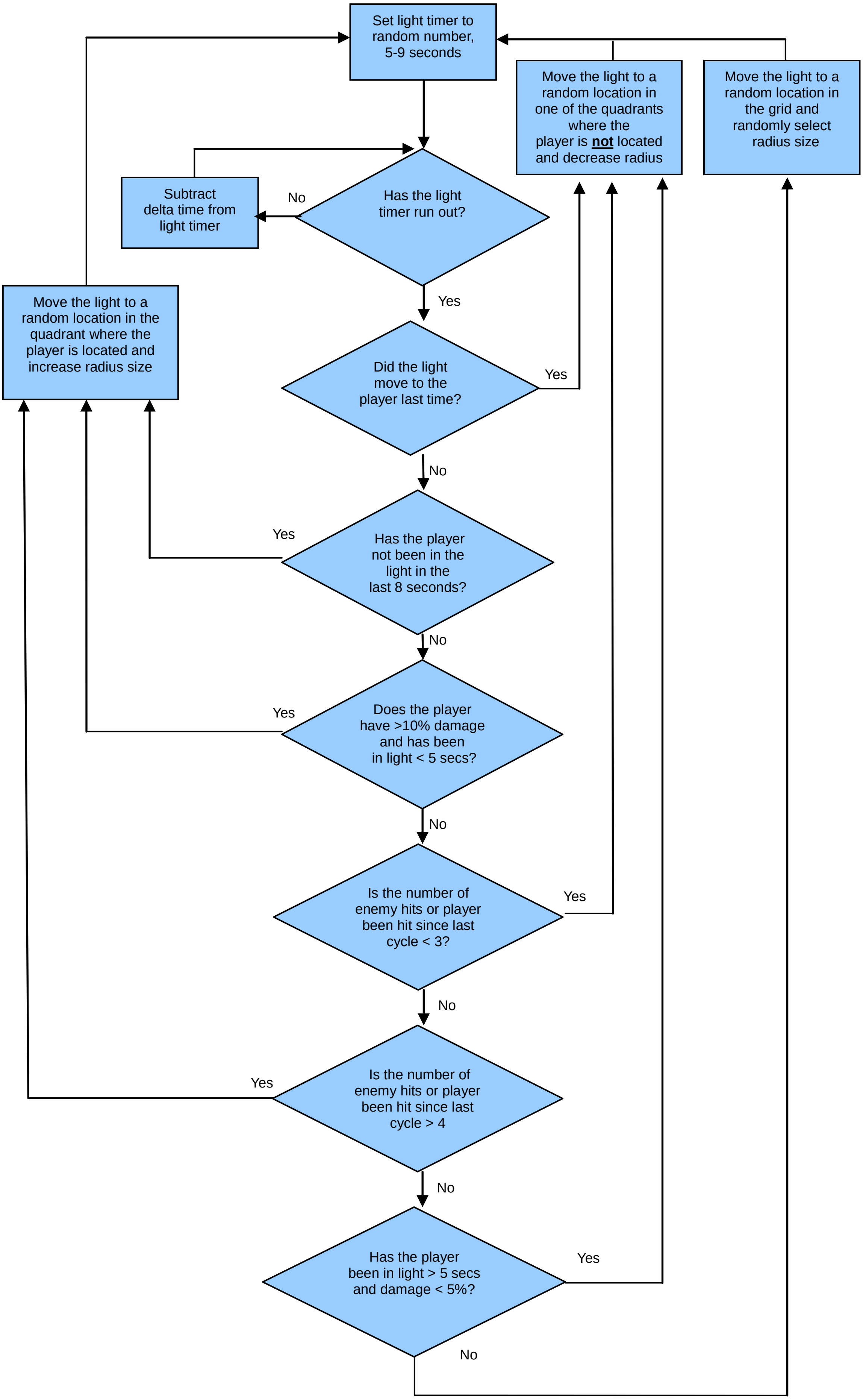

Figure 3.4: Real-Time Adaptive Light Mechanic Flow Chart.

Each enemy has two main variables, the enemy speed and the enemy search radius. The enemy speed controls how fast the enemy chases the player. The enemy search radius controls how close a player can be to the enemy before it will start chasing. In the static version, the variables are constant at 3.4 and 10.4 respectively. In the real-time adaption version these variables are influenced by a couple different factors. When there are a lot of enemy hits or player hits, the enemy speed will decrease. When there are few enemy hits or player hits, the enemy speed will increase. In addition, as a coin is collected, the enemy search radius is increased but if the player goes for a while without collecting a coin, the search radius decreases. The enemy search radius flow chart can be seen in Figure 3.5 and the enemy speed flow chart can be seen in Figure 3.6.

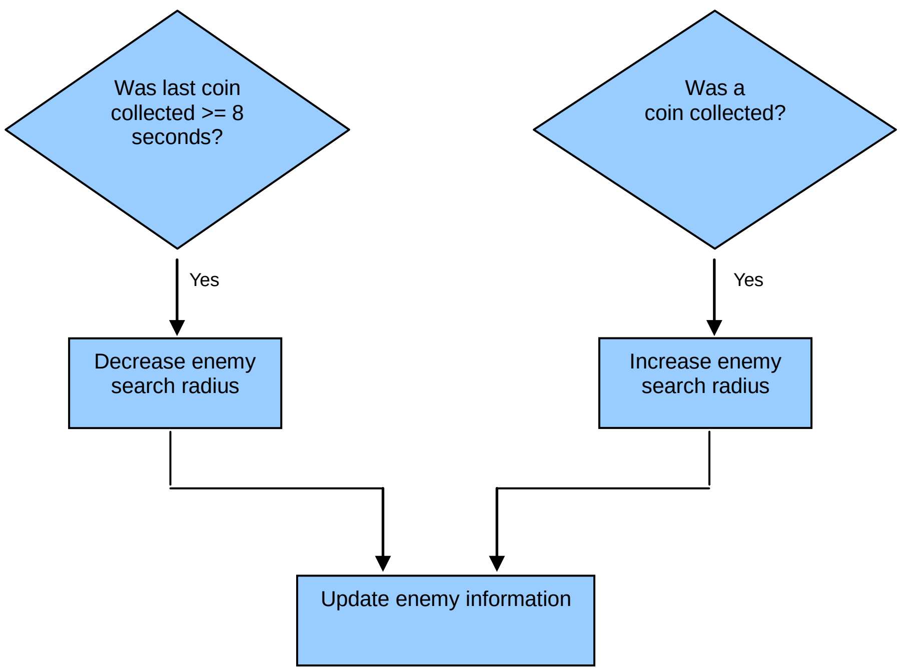

Figure 3.5: Adaptive Enemy Search Radius Flow Chart.

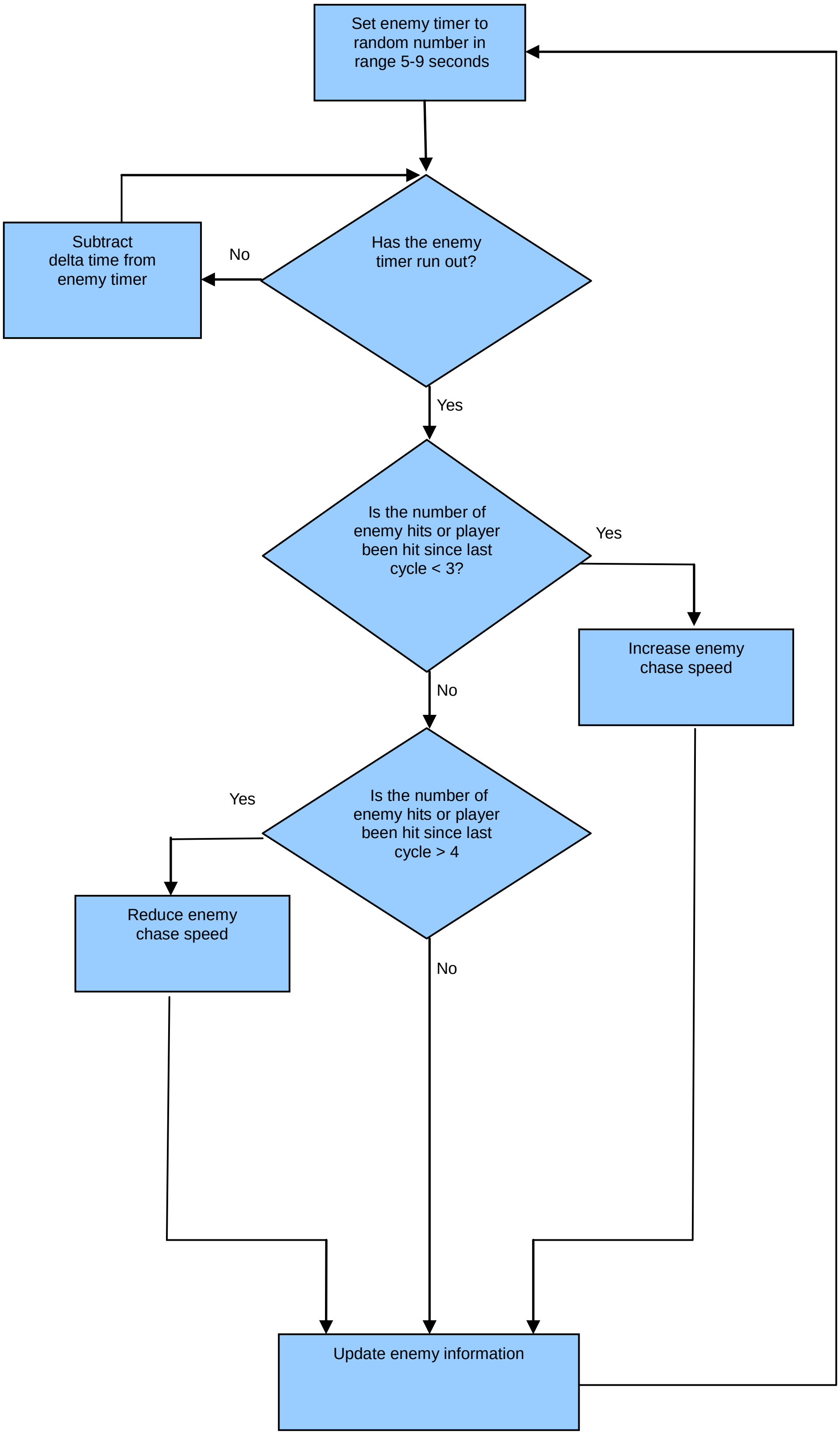

Figure 3.6: Adaptive Enemy Speed Flow Chart

The coin spawn mechanic is in control of where and when the coins are spawned on the grid. In static game, it is at complete random and can be anywhere on the grid that doesn't already have a coin. The flow chart for the static version can be seen in Figure 3.7 below. In the real-time game adaption enabled version, as shown in Figure 3.8, in order for the player to use all of the terrain, the coins are spread out. However, if a player hasn't picked up a coin recently, the coin will spawn in the quadrant where the player is located.

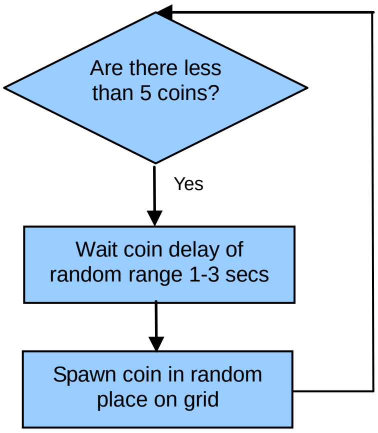

Figure 3.7: Static Light Mechanic Flow Chart.

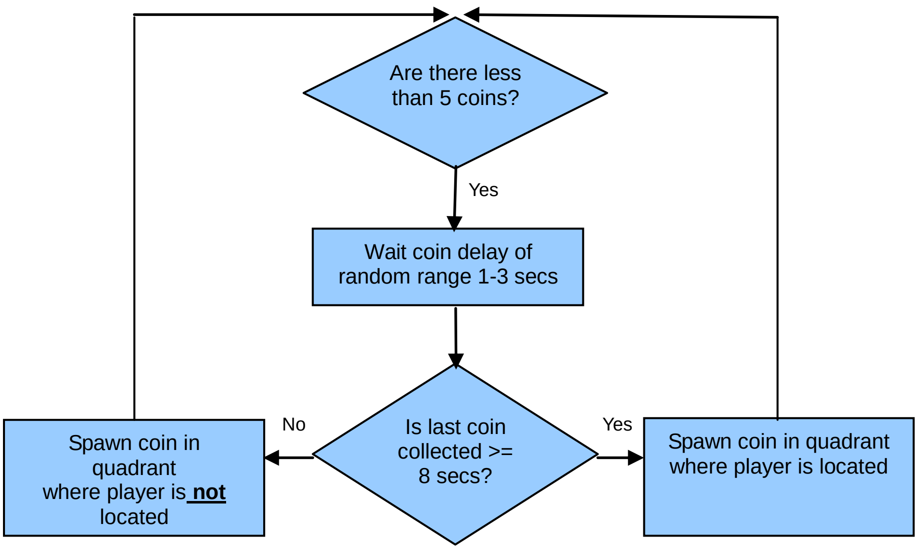

Figure 3.8: Real-Time Adaptive Coin Mechanic Flow Chart.

Over time, the terrain slowly starts to disappear. If the square on the grid is selected, it will turn yellow with cracks in it. After a couple of seconds, the square will disappear and become a hole in the terrain. The player will now have to avoid the hole for the rest of the game. In the static version of the game, a square is chosen at complete random as shown in Figure 3.9. In the real-time adaptive, the square is selected randomly within the quadrant that the player has spent the most time in. This drives the player to leave the quadrant and move around more.

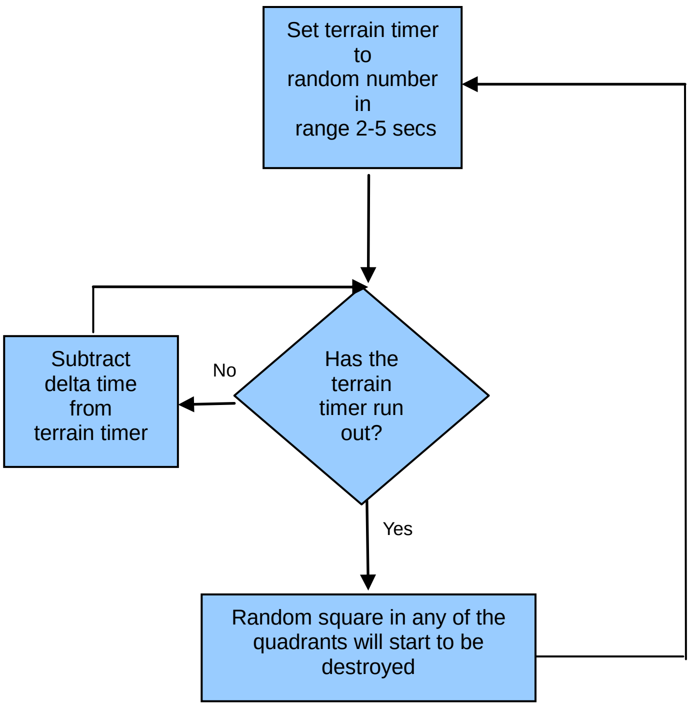

Figure 3.9: Static Terrain Falling Flow Chart.

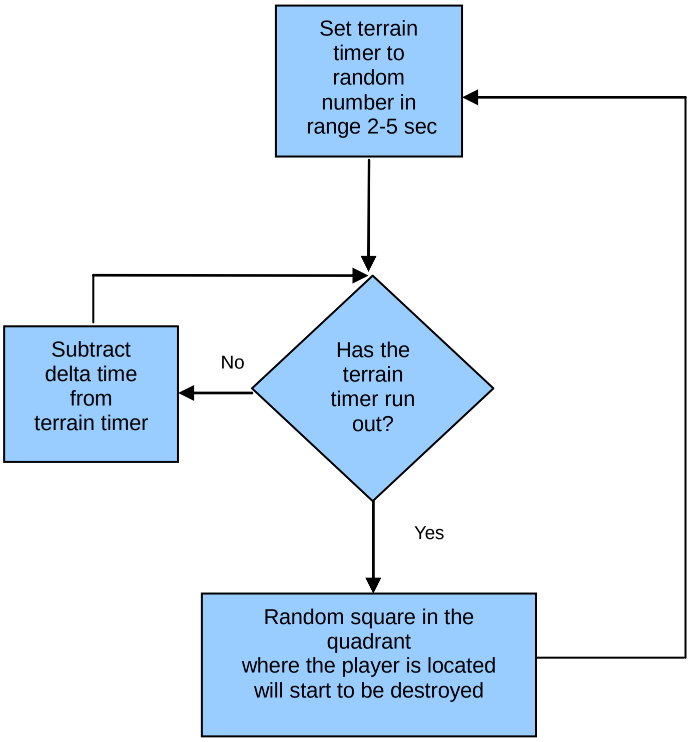

Figure 3.10: Real-Time Adaptive Terrain Falling Flow Chart.

4.5: Data Collected

After designing and deploying this game to a website, several people played the game and their metrics were recorded. If any match, 2 or 3, in any game played by a player lasted less than 5 seconds, all of those players' matches were removed from the database. This removal was done since a time less than 5 seconds would be too short for any game metrics to take effect. In addition, a game lasting less than 5 seconds is usually due to a mistake being made and the results would be skewed to the other match, significantly affecting the results. In the end, there were 33 unique players that had their data stored and analyzed further. Of those players, 23 played both a real-time adaptive enabled match and a static match in a random order, leaving 10 players to serve as the control group playing only the static version of the game. In order to review the full set of data and statistics done, refer to the appendices.

Based on the data that was recorded from the players, on average, when the game metrics driven adaption was enabled, the player did better all the way across for the different game mechanics when compared to the disabled game metrics adaption counterpart. As shown in Figure 4.1, the player spends more time in the light, collects more coins and has a higher number of hits both to the enemy and to themselves. The standard deviations are a little higher in the adaption enabled matches but the difference did not appear to be anything significant. As shown in Figure 4.2, using the T-Test to calculate the probability of independence from each other, all the values are fairly low but only time in light, and coins collected were under the 5% value used to determine independence. With this evidence, it appears that there is a correlation between the adaption being enabled and these higher results.

Table 4.1: Comparison of Real-Time Adapted and Static Matches.

	Adaption Enabled Match		Static Match	
	Average	Standard Deviation	Average	Standard Deviation
Time In Light	14.82 secs	9.48	8.73 secs	6.28
Total Play Time	91.90 secs	52.36	69.24 secs	37.86
Coins Collected	14.13 coins	7.73	10.61 coins	6.53
Number of Times Player was Hit	22.91 hits	12.95	19.57 hits	8.55
Number of Enemies Hit	23.17 hits	15.99	17.087	10.69

Table 4.2: T-Test Results between Adapted and Static Matches.

	T Test Results
Time In Light	0.0026
Total Play Time	0.0766
Coins Collected	0.0471
Number of Times Player was Hit	0.2978
Number of Enemies Hit	0.1209

More analysis was completed by further in-depth examination of the differences between matches 2 and 3 for both versions of the game. It was found, after separating the data that in the static match, that the averages for Match 3 were significantly higher than Match 2. While at the same time, the averages found when game metrics were enabled were not significantly different between the two matches and, in fact, were slightly higher in the second match.

Table 4.3: Adaption Match 2 and Match 3 Comparisons.

	Match 2		Match 3	
	Average	Standard Deviation	Average	Standard Deviation
Time In Light	16.01 secs	10.12	13.73 secs	9.16
Total Play Time	98.55 secs	54.83	85.81 secs	51.64
Coins Collected	15.18 coins	8.55	13.17 coins	7.13
Number of Times Player was Hit	25.72 hits	15.61	20.33 hits	9.95
Number of Enemies Hit	25.45 hits	20.07	21.08 hits	20.07

Table 4.4: T-Test Results between Adaption Match 2 and Match 3.

	T Test Results
Time In Light	0.6153
Total Play Time	0.4990
Coins Collected	0.4860
Number of Times Player was Hit	0.1663
Number of Enemies Hit	0.3331

Table 4.5: Static Match 2 and Match 3 Comparisons.

	Match 2		Match 3	
	Average	Standard Deviation	Average	Standard Deviation
Time In Light	4.21 secs	2.05	13.67 secs	5.55
Total Play Time	50.01 secs	22.09	90.22 secs	41.09
Coins Collected	7.42 coins	5.92	13.27 coins	6.56
Number of Times Player was Hit	16.42 hits	8.55	23.00	7.46
Number of Enemies Hit	14.58 hits	10.67	19.81	10.51

Table 4.6: T-Test Results between Static Match 2 and Match 3.

	T Test Results
Time In Light	0.0001
Total Play Time	0.0152
Coins Collected	0.0221
Number of Times Player was Hit	0.0952
Number of Enemies Hit	0.3791

After the first group of players was analyzed, the control group was analyzed. In this group it was determined that, on average, the results were better for Match 3. Though according to the T-Test, the differences in the matches were not significant enough indicate independence.

Table 4.7: Control Match 2 and Match 3 Comparisons.

	Match 2		Match 3	
	Average	Standard Deviation	Average	Standard Deviation
Time In Light	5.55 secs	2.89	08.50 secs	4.86
Total Play Time	60.99 secs	26.09	70.67 secs	23.71
Coins Collected	7.30 coins	4.35	09.60 coins	2.80
Number of Times Player was Hit	19.30 hits	9.23	14.60 hits	9.13
Number of Enemies Hit	12.90 hits	7.94	13.40 hits	9.49

Table 4.8: T-Test Results Between Control Matches.

	T Test Results
Time In Light	0.0607
Total Play Time	0.4213
Coins Collected	0.1207
Number of Times Player was Hit	0.2064
Number of Enemies Hit	0.9090

Finally, the most important aspect to be analyzed was the comparison between the control group of Match 3 and the Match 3 of the static version of the game, where players played the adaptive version first. In this, we see a higher average total in all measures, with a similar standard deviation in favour of the static match following the adaptive version over the control group. However, only the players' time spent in the light was indicated as independent according to the T-Test.

Table 4.9: Control Match 3 and Static Match 3 Comparisons.

	Control Match 3		Static Match 3	
	Average	Standard Deviation	Average	Standard Deviation
Time In Light	8.50 secs	4.86	13.67 secs	5.55
Total Play Time	70.67 secs	23.71	90.22 secs	41.09
Coins Collected	9.60 coins	2.80	13.27 coins	6.56
Number of Times Player was Hit	14.60 hits	9.13	23.00 hits	7.46
Number of Enemies Hit	13.40 hits	9.49	19.82 hits	10.51

Table 4.10: T-Test Results between Control Match 3 & Static Match 3.

	T Test Results
Time In Light	0.0226
Total Play Time	0.2117
Coins Collected	0.1125
Number of Times Player was Hit	0.0665
Number of Enemies Hit	0.1383

After the three matches, players were given a questionnaire to fill out. The results of the questionnaire can be seen below in Table 4.11 and table 4.12. While players tended to notice a difference between matches, players tended to prefer the game metrics driven adaption match regardless of which game number it was played in. As well, the majority of the players tested were non-experienced game users.

Table 4.11: Hours Played a Week Results.

	Percent
0 – 2 hours	60.87%
3 – 5 hours	30.43%
6 – 8 hours	08.70%
9 + hours	00.00%

Table 4.12: Results of Player Preferences between Matches.

	Percent
Adaption	60.87%
Static	13.04%
No Preference	26.09%

Chapter 5: Discussion

5.1: Overview

Analysis of the data based on the results from the 33 people who tested the application has shown correlation to support that tutorials can be improved from the use of real-time adaption, driven by game metrics.

5.2: Adaptive Versus Static Matches

In the comparison between the adaptive and the static matches, the results show that the main mechanics of the game were received better in the adaption mode, as was expected. The use of light almost doubles in the adaptive version, likely due to the fact that the light will move to the same quadrant as the player in several situations and players seemed to take advantage of the proximity to the light. Coin collection, on average, is higher when the game adapts and has a low T-Test result, which could be a result of the player lasting longer on average but, as the coins are spread out more, it still requires the player to walk around the entire grid. Lastly, the enemy and player hits each increased slightly when the game is adapting to the metrics in place. This could be simply due to the fact the player is lasting longer in those matches. A more interesting factor is the closeness between the number of enemy hits and player hits. Achieving this type of balance was the goal of the metrics adaption. Based on the differences between metrics enabled and disabled, it appears that when the metrics were enabled, they worked as they were designed to and balanced out the game more.

5.3: Comparison of Adaptive Match 2 and Match 3

Upon further examination of the results for the real-time adaption matches by differentiating the results between Match 2 and Match 3, while the metrics are slightly higher in Match 2, according the T-Test results they are not statistically significant enough to show evidence for independence. The results seen could just be distributed through chance. The way that the metrics work, it is possible the game plays within an elastic band and does not allow that much variation. This, as Adams (2008) pointed out earlier, is one of the issues in dynamic difficulties. Another possibility is that players did not learn much from their static game in Match 2.

5.4: Comparison of Static Match 2 and Match 3

Upon a more in depth comparison of the static matches, two of the major mechanics, coin collecting and the use of the light have risen significantly in the third match as shown with the T-Test result. This implies that there is a correlation between the results being higher in third match compared the results in the second match. The large increase in the metrics for Match 3 could be a result of having a better understanding of the game mechanics by that group playing the game metrics enabled version first in Match 2. However, the results for the enemy and player hits were not significantly changed. There could be several reasons for this. The players could be learning to avoid the enemies since they remain at a steady level of difficulty. The slight increase in number could be due to the extended length of the game or be simply due chance.

5.5: Comparison of Control Match 2 and Match 3

As players play any game, there will be improvement with each round due to familiarity. The control match is used as a baseline and will be used later against the other group. In comparing the control group against itself for Match 2 and Match 3, there were slight improvements on average. However, in the T-Test results, there was no value with a low enough probability to show independence.

5.6: Comparison of Control Match 3 and Static Match 3

In the final comparison, one of the main purposes of this dissertation was to evaluate the results between the control group Match 3 and the group that played the real-time adaptive match first before playing the static version in Match 3. The only T-Test probability that showed independence was the time in light. However, the average for time in light, overall game time, coin collection, enemy hits and player hits were higher in static Match 3. While the increase in time could explain the increase in coin collection, enemy hits, and player hits, in order for the player to increase the time they had to improve at moving the character around and using the light for protection.

5.7: Questionnaire Results

The majority of the results of this study were taken from people who do not play video games very often, in the 0 – 2 hour range per week. It would be expected that these are the users that would be most in need of a tutorial level. Most users that played both the static version and the adaptive version preferred the adaptive version. This could lead to a better first impression of the game as players learn the metrics and potentially interest a greater number of people in playing the game more.

Chapter 6: Conclusion & Future Work

6.1: Conclusion

Overall, the findings in this study have suggested that tutorials can be improved with the use of real time game adaption driven by game metrics. Players have shown significant improvement in the static version of the game after playing the adapting game first. This could be a way of showing users how to play by showing the use of mechanics over the traditional approach of prompting or exploration. As the tutorial level is often the first experience players have with a game, it is vital that the players get the best experience right at the beginning. More players preferred having a real-time adaption enabled version of the game over a static level. As well as having a tutorial level that adapts to the player to ensure that mechanics are learned, player engagement is vital for the any first impression. As each player plays differently, the experience should maximize their individual engagement in the game, instead of trying to fit every player into a generalized frame and format.

6.2: Future Work

This dissertation has only touched upon the potential benefits for tutorials based on real-time game adaption using in game metrics to monitor players as they progress. Further study on this type of tutorial system could be taken in several directions. Since the study is still rather limited in the number of people and complexity of games, expanding the current scope to permit the collection of more information is required before developers spend more time and money on their tutorial levels. Additional options for tutorial levels could include creating more content through procedural generation based on a player's weakness. For example, in first-person shooter games, the accuracy of a player could be determined through the metrics. Depending how the player does, targets could be adjusted to move or alter in size as appropriate. The length of the tutorial could also be shortened if the players skill is ranked high enough, as experienced players would rather play the actual game content than remain in a tutorial mode.

Real-time game adaption can be used in other places besides the tutorial level. Already dynamic difficulties are gaining popularity, but real-time game adaption can be much more than that. Player metrics can be constantly monitored in a game. If a player isn't using a certain game mechanic or has chosen a path that the designer may not have expected, new game content could appear such as prompts to let the player know of something they may have forgotten or lowering the price of other weapons or items if the stores aren't being used to their fullest extent. In a one against another game, adaptive metrics might provide an opportunity to more evenly handicap each player to provide a more even contest. A metrics "score" could be provided within a game to permit another form of competition for players within the game. Older game might be capable of being modified into more complex and challenging games for people that have already completed them.

APPENDICES

Appendix A – Game Metrics Results

In the tables below are all the game metric data collected for the players' Match 2 and Match 3.

Table A.1: Game Metric Data Collected from Adaptive Tests.

Game Metric	Player Results					
Adaptive Version						
Match #	3	2	3	2	2	3
Quad 1 Time	12.4	12.56844	44.92	14.37294	63.89702	39.7528
Quad 2 Time	18.12994	12.16073	38.028	15.80566	68.2758	48.0881
Quad 3 Time	13.119	10.839	23.14	30.16864	45.1311	36.33902
Quad 4 Time	9.37058	13.2904	69.69	45.63266	42.47216	46.1713
Cross	2.17	0	5.93	0	3.32	0
In Light	20.97	10.857393	12	18.85962	34.45	33.94761
Not in Light	34.22974	38.00147	169.1178	87.12	188.6772	136.40389
Num Enemies	15	15	15	15	15	15
Coins Collected	6	10	25	14	35	23
Enemy Hits	9	29	46	58	62	28
Player Hits	8	27	36	48	51	23
Times Fallen	0	0	0	0	0	0
Damage Taken	8	27	36	48	51	23
Enemies Speed	3.7	3.3	3.7	3.1	3.4	3.7
Enemies Search	10.4	10.9	8.6999	10.8	11.7	11.3
Quad 1 T Left	33	32	23	32	15	24
Quad 2 T Left	29	33	28	33	14	23
Quad 3 T Left	34	33	29	27	19	27
Quad 4 T Left	34	33	17	25	23	22
Health Healed	3	8	19	33	26	19
Static Version						
Match #	2	3	2	3	3	2
Quad 1 Time	36.03	24.58321	4.519	2.039	37.28	28.607
Quad 2 Time	17.1785	8.646	3.57	3.12	22.149	15.6789
Quad 3 Time	19.8529	12.82	9.939	22.8	33.32	16.93
Quad 4 Time	5.77015	5.03	9.6	7.76	15.24	29.97133
Cross	0	2.32	0	0	0.32	0
In Light	4.959945	11.35	4.3507	6.788	22.5704	8.86
Not in Light	73.87188	42.05	23.63844	29.811	85.75	82.329
Num Enemies	15	15	15	15	15	15
Coins Collected	16	12	6	3	20	18
Enemy Hits	19	33	0	23	14	29
Player Hits	14	25	3	20	17	28
Times Fallen	0	0	0	0	0	0
Damage Taken	14	15	3	20	17	28
Enemies Speed	3.4	3.4	3.4	3.4	3.4	3.4
Enemies Search	10	10	10	10	10	10
Quad 1 T Left	34	34	35	33	25	28
Quad 2 T Left	30	32	34	33	31	30
Quad 3 T Left	28	33	34	35	26	27
Quad 4 T Left	28	32	34	34	30	32
Health Healed	5	7	1	6	8	13

Table A.1 (cont'd).

Game Metrics	Player Results				
Adaptive Version					
Match #	2	2	2	2	2	3
Quad 1 Time	21.92	18.7108	23.333	8.14	37.098	6.6
Quad 2 Time	24.91	7.138	24.744	9.73	23.55695	11.02
Quad 3 Time	35.606	13.35997	35.09969	11.36	20.8876	1.22
Quad 4 Time	33.27199	6.4415	24.8822	12.038	15.712	5.79
Cross	1.8839	7.37	2.68	7.29	0.658	0
In Light	33.65	6.439976	11.95	6.98	12.203	2.31
Not in Light	83.95	46.58	98.78674	41.59	85.707	22.343
Num Enemies	15	15	15	15	15	15
Coins Collected	24	8	17	10	16	4
Enemy Hits	7	17	35	3	19	16
Player Hits	8	25	36	8	18	12
Times Fallen	0	0	0	0	0	0
Damage Taken	8	25	36	8	18	12
Enemies Speed	3.4	3.4	3.7	3.5	3.3	3.5
Enemies Search	11.1	3.4	11.3	10.9	11.3	10.3
Quad 1 T Left	33	30	29	32	26	34
Quad 2 T Left	30	34	30	33	30	33
Quad 3 T Left	31	33	28	33	31	36
Quad 4 T Left	21	34	30	34	31	34
Health Healed	18	6	18	6	6	1
Static Version						
Match #	3	3	3	3	3	2
Quad 1 Time	27.406	26.55272	9.229	2.06644	25.29	16.69
Quad 2 Time	18.629	21.29708	16.29	20.4	17.94809	0
Quad 3 Time	47.302	14.5	18.066	18.83	35.075	23.95
Quad 4 Time	71.147	29.609	8.78	16.412	25.69	3.42
Cross	5.1	0.5603	2.84	1.1	0	0
In Light	22.48	15.408889	6.109	11.29	17.29	3.89
Not in Light	147.1027	77.111	49.10084	47.119	86.88	40.17
Num Enemies	15	15	15	15	15	15
Coins Collected	25	11	7	8	17	10
Enemy Hits	15	28	36	0	24	28
Player Hits	33	34	31	14	27	20
Times Fallen	0	0	0	0	0	0
Damage Taken	33	34	31	14	27	20
Enemies Speed	3.4	3.4	3.4	3.4	3.4	3.4
Enemies Search	10	10	10	10	10	10
Quad 1 T Left	28	31	30	32	31	33
Quad 2 T Left	24	31	35	33	26	35
Quad 3 T Left	30	26	30	34	31	32
Quad 4 T Left	20	31	35	29	29	31
Health Healed	29	16	6	9	20	7

Table A.1 (cont'd).

Game Metrics	Player Results					
Adaptive Version						
Match #	2	3	3	2	3	3
Quad 1 Time	15.9415	34.32	6.299	34.84	29.33	7.83
Quad 2 Time	18.64732	32.01	6.8702	27.78899	23.94	11.60153
Quad 3 Time	15.9032	12.838	13.96	57.71	31.37	10.7272
Quad 4 Time	15	39.6476	12.163	34.96	33.117	13.7205
Cross	2.0446	0	0.88	4.26	1.259957	13.72
In Light	11.577	25.1717	14.88947	21.95127	14.07	10.539
Not in Light	55.96189	93.65	25.28371	137.6285	104.94	47.06
Num Enemies	15	15	15	15	15	15
Coins Collected	11	19	12	18	11	11
Enemy Hits	2	36	11	27	18	12
Player Hits	5	33	10	34	24	11
Times Fallen	0	0	0	0	0	0
Damage Taken	5	33	10	34	24	11
Enemies Speed	3.7	3.2	3.6	3.7	3.5	3.4
Enemies Search	10.8	11.2	11.2	10.8	10.5	10.1
Quad 1 T Left	31	27	34	28	28	32
Quad 2 T Left	30	29	33	27	28	34
Quad 3 T Left	32	32	31	19	26	32
Quad 4 T Left	32	23	33	24	26	33
Health Healed	4	28	9	23	17	8
Static Version						
Match #	3	2	2	3	2	2
Quad 1 Time	41.57399	8.39	6.86	14.10991	6.81	15.60968
Quad 2 Time	32.76686	17.19	2.6	15.47	4.38	9.918
Quad 3 Time	29.3157	17.888	5.2	20.0123	10.9	3.74
Quad 4 Time	38.42613	17.655	4.25	19.01112	4.87	9.37
Cross	4.348	0	3.689	1.999	0	2.53
In Light	15.44582	5.389	1.82	11.21	4.6	3.3
Not in Light	130.986	55.73	21	59.384	21.43317	37.84
Num Enemies	15	15	15	15	15	15
Coins Collected	19	10	1	9	3	2
Enemy Hits	12	15	1	12	6	12
Player Hits	13	22	4	20	7	16
Times Fallen	0	0	0	0	0	0
Damage Taken	16	22	4	20	7	16
Enemies Speed	3.4	3.4	3.4	3.4	3.4	3.4
Enemies Search	10	10	10	10	10	10
Quad 1 T Left	23	32	34	31	35	32
Quad 2 T Left	27	31	35	29	35	32
Quad 3 T Left	26	34	36	31	33	35
Quad 4 T Left	29	30	34	36	35	35
Health Healed	11	6	1	5	0	3

Table A.1 (cont'd).

Game Metrics	Player Results				
Adaptive Version					
Match #	3	3	3	2	3
Quad 1 Time	15.6	20.58	0.24	8.46	14.91
Quad 2 Time	9.9188	8.45	5.52	8.8	20.17
Quad 3 Time	3.74	21.78	23.84	14.887	17.17
Quad 4 Time	9.3711	33.53	18.51	16.1	34.777
Cross	2.5	0	0.96	2.8602	0.98
In Light	3.32	10.46	5.97756	7.16	11.15
Not in Light	37.842	73.8997	43.11129	43.999	77
Num Enemies	15	15	15	15	15
Coins Collected	3	19	11	4	14
Enemy Hits	12	14	20	21	31
Player Hits	16	19	17	23	35
Times Fallen	0	0	0	0	0
Damage Taken	16	19	17	23	35
Enemies Speed	3.4	3.7	3.2	3.2	3.4
Enemies Search	10	11.9	10.9	10.2	11.2
Quad 1 T Left	32	29	36	34	33
Quad 2 T Left	32	33	35	34	32
Quad 3 T Left	35	30	30	31	30
Quad 4 T Left	35	26	31	31	25
Health Healed	3	14	6	11	18
Static Version					
Match #	2	2	2	3	2
Quad 1 Time	5.78	9.63	7.5	19.08991	14.146
Quad 2 Time	2.54	4.73	13.55	15.92288	13.817
Quad 3 Time	12.6	30.69	12.89	27.33931	9.732
Quad 4 Time	4.35	15.39	24.819	30.157	13.865
Cross	3.4	2.7	2.56	4.6201	0
In Light	1.1	4.6	5.52	10.441837	2.126
Not in Light	27.68283	58.57	57.81	86.68729	49.43
Num Enemies	15	15	15	15	15
Coins Collected	1	11	12	15	8
Enemy Hits	15	32	8	21	10
Player Hits	17	30	20	19	16
Times Fallen	0	0	0	0	0
Damage Taken	17	30	20	19	16
Enemies Speed	3.4	3.4	3.4	3.5	3.4
Enemies Search	10	10	10	10.9	10
Quad 1 T Left	34	29	31	31	33
Quad 2 T Left	34	33	33	31	34
Quad 3 T Left	35	35	31	30	31
Quad 4 T Left	34	30	32	26	33
Health Healed	0	0	6	10	0

Table A.2: Game Metric Data Collected from Static Tests.

Game Metrics	Player Results				
Match 2					
Quad 1 Time	10.72	8.24	16.79808	7.771179	2.92
Quad 2 Time	14.05	11.89551	12.25936	0.568	3.89
Quad 3 Time	12.98	8.148	18.99524	8.2557	7.9011
Quad 4 Time	6.6654	10.37	17.03885	7.93826	13.8023
Cross	3.24	18.34	3.021852	0.384	0
In Light	7.5	1.988	6.78983	2.69	8.75
Not in Light	40.163	55	61.54651	22.227	19.76
Num Enemies	15	15	15	15	15
Coins Collected	10	6	7	2	0
Enemy Hits	9	12	15	3	0
Player Hits	27	22	23	5	10
Times Fallen	0	0	0	0	0
Damage Taken	27	22	23	5	10
Enemies Speed	3.4	3.4	3.4	3.4	3.4
Enemies Search	10	10	10	10	10
Quad 1 T Left	30	32	31	36	35
Quad 2 T Left	31	33	31	35	35
Quad 3 T Left	32	31	32	33	33
Quad 4 T Left	30	29	32	34	33
Health Healed	6	3	7	5	0
Match 3					
Quad 1 Time	18.359	5.76	20.6473	14.35548	84.4683
Quad 2 Time	18.06	5.78	18.965776	8.139	0.4196
Quad 3 Time	27.32	12.04	22.2222	9.893	20.531
Quad 4 Time	25.73	17.3	25.182876	23.38	9.396
Cross	0.56	0	2.293066	4.51	0
In Light	4.05892	4	13.8377	3.3425	18.629
Not in Light	85.98	36.9	75.19052	56.93	95.6541
Num Enemies	15	15	15	15	15
Coins Collected	13	6	14	8	10
Enemy Hits	29	26	22	10	8
Player Hits	32	21	23	15	7
Times Fallen	0	0	0	0	0
Damage Taken	32	21	23	15	7
Enemies Speed	3.4	3.4	3.4	3.4	3.4
Enemies Search	10.4	10.4	10.4	10.4	10.4
Quad 1 T Left	31	34	30	32	27
Quad 2 T Left	34	35	31	34	36
Quad 3 T Left	30	32	30	33	31
Quad 4 T Left	30	32	29	28	33
Health Healed	7	4	12	1	7

Table A.2 (cont'd).

Game Metrics	Player Results				
Match 2					
Quad 1 Time	23.57499	4.88728	29.4185	8.93937	18.17422
Quad 2 Time	16.76686	4.3035	18.61	16.789	28.49
Quad 3 Time	19.3157	5.037	25.805	32.38	21.19903
Quad 4 Time	16.426	20.06652	24.595	29.1838	8.668
Cross	4.348	2.234955	1.079932	0.92	2.319
In Light	9.44582	1.78486	3.19264	7.42789	5.908
Not in Light	70.986	34.74473	96.3169	80.782336	72.94
Num Enemies	15	15	15	15	15
Coins Collected	12	4	11	12	8
Enemy Hits	12	18	26	12	22
Player Hits	13	19	38	17	19
Times Fallen	0	0	0	0	0
Damage Taken	13	19	38	17	19
Enemies Speed	3.4	3.4	3.4	3.4	3.4
Enemies Search	10	10	10	10	10
Quad 1 T Left	27	31	25	30	31
Quad 2 T Left	29	34	33	27	34
Quad 3 T Left	30	34	29	31	29
Quad 4 T Left	31	35	28	32	31
Health Healed	11	3	3	15	2
Match 3					
Quad 1 Time	15.9415	12.61	11.81973	16.4788	8.29
Quad 2 Time	18.647	0	12.1199	15.28992	10.91
Quad 3 Time	22.903	20.735	20.97	27.2838	17.4
Quad 4 Time	15	3.919	16.78	15.25681	17.5
Cross	2.044614	0	6.23	3.58	0.5
In Light	8.577	8.59264	9.77835	9.3269	4.9
Not in Light	65.96189	28.67257	58.14178	68.56264	49.7
Num Enemies	15	15	15	15	15
Coins Collected	11	7	11	10	6
Enemy Hits	6	0	6	15	12
Player Hits	4	4	10	19	11
Times Fallen	0	0	0	0	0
Damage Taken	4	4	10	21	11
Enemies Speed	3.4	3.4	3.4	3.4	3.4
Enemies Search	10.4	10.4	10.4	10.4	10.4
Quad 1 T Left	31	31	33	31	31
Quad 2 T Left	30	36	34	32	33
Quad 3 T Left	32	30	30	30	32
Quad 4 T Left	31	35	30	31	33
Health Healed	6	0	9	8	4

Appendix B – Questionnaire

In the tables below are all the questionnaire data collected from the players after completing all 3 matches.

Table B.1: Questionnaire Data Collected from Adaptive Tests.

Questionnaire	Player Results								
First Time?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Match Favourite?	M3	M2	M2	M2	M2	M2	M2	M2	None
Notice Difference?	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes
Hours	3 – 5	3 – 5	0 – 2	0 – 2	6 – 8	0 – 2	6 – 8	3 – 5	0 – 2

Table B.1 (cont'd).

Questionnaire	Player Results								
First Time?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Match Favourite?	M2	M2	None	None	M2	M2	None	M2	M2
Notice Difference?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Hours	0 – 2	3 – 5	0 – 2	3 – 5	0 – 2	3 – 5	0 – 2	0 – 2	0 – 2

Table B.1 (cont'd).

Questionnaire	Player Results				
First Time?	Yes	Yes	Yes	Yes	Yes
Match Favourite?	None	M3	None	M3	M2
Notice Difference?	No	Yes	No	Yes	Yes
Hours	0 – 2	0 – 2	0 – 2	0 – 2	3 – 5

Table B.2: Questionnaire Data Collected from Static Tests.

Questionnaire	Player Results									
First Time?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Match Favourite?	M2	None	M2	M2	M3	M2	M3	None	None	M3
Notice Difference?	No	Yes	No	No	Yes	No	No	No	No	No
Hours Played	3 – 5	3 – 5	0 – 2	0 – 2	0 – 2	6 – 8	0 – 2	0 – 2	0 – 2	0 – 2

Appendix C – DVD Structure

Enclosed with this dissertation is an accompanying DVD containing an electronic version of this paper, project source code, and data collected excel files. Below is a list of the DVD's structure and organization of the disc.

- Documents
    - Laidlow-Dissertation.pdf – An electronic copy of the dissertation paper.
    - Laidlow-AdaptiveResults.xls – A file containing all data gathered during the adaptive tests.
    - Laidlow-StaticResults.xls – A file containing all data gathered during the static tests.
    - Laidlow-Comparisons.xls – A file containing all the comparisons made in the analysis.
- Source Code
    - Project Metrics – A Unity Project Folder containing all source code.
    - App-Demo.exe – An executable file for playing the application.

Appendix D – Access to Game

The application game is being hosted at the following web address:

- http://www.micklaidlow.com/WebPlayer.html

REFERENCES

References

Adams, E. 2008. *The Designers Notebook: Difficulty Modes and Dynamic Difficulty Adjustment.* [online]. Gamasutra.com. Available from: http://www.gamasutra.com/view/feature/3660/the_designers_notebook_.php [Accessed 25 September 2012].

Andersen, E., et al. 2012. *The Impact of Tutorials on Games of Varying Complexity*. Center for Game Science. [online]. Available from: http://grail.cs.washington.edu/projects/game-abtesting/chi2012/chi2012.pdf [Accessed 25 September 2012].

Angle, R. 2012. *Game Industry Lessons Learned*. Hoppsbusch. [online]. Available from: http://hoppsbusch.com/blog/?p=264 [Accessed 24 September 2012].

Canossa, A. and Cheong, Y. 2011. *Limits of Gameplay Metrics Analysis and Phenomenological Debugging*. Siren. [online]. Available from: http://sirenproject.eu/sites/default/files/Digra2011_intention.pdf [Accessed 25 September 2012].

Casual Games Sector Report. 2012. *Freemium Gaming Metrics*. Casual Games Association. [online]. Available from: http://casualconnect.org/mag/summer2012/CGA_F2PGames_Report.pdf [Accessed 25 November 2012].

Chapresto, E., Mitchell, K. and Seron, F. 2011. Capture and Analysis of Racing Gameplay Metrics. IEEE. [online]. Available from: http://ieeexplore.ieee.org.libproxy.abertay.ac.uk/xpls/abs_all.jsp?arnumber=5887312&tag=1 [Accessed 25 September 2012].

Chen, J. 2008. *Flow in Games*. ACM. [online]. Available from: http://www.jenovachen.com/flowingames/Flow_in_games_final.pdf [Accessed 20 September 2012].

Drachen, A. and Canossa, A. 2009. *Analyzing User Behavior via Gameplay* Metric. ACM. [online]. Available from: http://dl.acm.org.libproxy.abertay.ac.uk/citation.cfm?id=1639613 [Accessed 15 October 2012].

Hunicke, R. 2005. *The Case for Dynamic Difficulty Adjustment in* Games. ACM. [online]. Available from: http://dl.acm.org.libproxy.abertay.ac.uk/citation.cfm?id=1178573 [Accessed 24 September 2012].

Infinity Ward. 2007. *Call of Duty 4: Modern Warfare*. [disc]. Microsoft Xbox 360. Activision.

Johnson, B. 2009. *How Tetris conquered the world, block by block*. The Guardian. [online]. Available from: http://www.guardian.co.uk/technology/gamesblog/2009/jun/02/tetris-25anniversary-alexey-pajitnov [Accessed 22 September 2012].

Klubeck, M. 2011. Using Metrics as Indicators. *Metrics.* Apress. 2011, pp. 83-95 [online]. Available from: http://link.springer.com.libproxy.abertay.ac.uk / content/pdf/10.1007%2F978-1-4302-3727-3_4 [Accessed 15 October 2012].

Nintendo Creative Department. 1985. *Super Mario Bros*. [cartridge]. Nintendo Entertainment System. Nintendo.

Pajitnov, A. 1986. *Tetris*. [cartidge]. Nintendo Entertainment System. Nintendo.

Tychsen, A. 2008. *Crafting User Experience via Game Metrics Analysis.* Center for Computer Games Research. [online]. Available from: http://www.cs.uta.fi/~ux-emotion/submissions/Tychsen.pdf [Accessed 25 September 2012].

Weisstein, E.W. [no date]. *Hypothesis Testing*. MathWorld - A Wolfram Web Resource. [online]. Available from: http://mathworld.wolfram.com/HypothesisTesting.html [Accessed 25 November 2012].

Yannakakis, G.N. and Hallam, J. 2009. *Real-Time Game Adaption for Optimizing Player Satisfaction*. IEEE. [online]. Available from: http://ieeexplore.ieee.org.libproxy.abertay.ac.uk/stamp/stamp.jsp?tp=&arnumber=5067382 [Accessed 25 September 2012].

BIBLIOGRAPHY

Bibliography

Adams, E. and Rollings, A. 2007. *Fundamentals of Game Design.* 2nd ed. New Jersey: Pearson.

Booth, M. 2009. *The AI Systems of Left 4 Dead.* Valve. [online]. Available from: http://www.valvesoftware.com/publications/2009/ai_systems_of_l4d_mike_booth.pdf [Accessed 26 September 2012].

Cheong, Y. et al . 2011. *A Computational Approach Towards Conflict Resolution for Serious Games.* ACM. [online]. Available from: http://dl.acm.org.libproxy.abertay.ac.uk/citation.cfm?id=2159368&bnc=1 [Accessed 26 September 2012].

De Alfaro, L., et al. 2007. *Game Refinement Relations and Metrics.* Logical Methods in Computer Science. [online]. Available from: http://wwwhome.cs.utwente.nl/~marielle/papers/dAMRS08.pdf/LMCS-final-4.pdf [Accessed 13 October 2012].

Denieffe, D., et al. 2007. *A Game Assessment Metric for the Online Gamer.* Advances in Electrical and Computer Engineering. [online]. Available from: https://el.trc.gov.om/htmlroot/ENGG/tcolon/e_references/Consolidated/Computer%20Science/Journals/A%20Game%20Assessment%20Metric%20for%20the%20Online%20Gamer.pdf [Accessed 13 October 2012].

Hunicke, R. 2004. *MDA: A Formal Approach to Game Design and Game Research.* North Western. [online]. Available from: http://www.cs.northwestern.edu/~hunicke/MDA.pdf [Accessed 25 September].

Hussain, T. 2012. *Serious Game Design Tutorial.* Game Tech. [online]. Available from: http://www.gametechconference.com/sites/default/files/presentations/2012_Gametech__InteractiveTutorial_final.pdf [Accessed 25 September 2012].

Klubeck, M. 2011. Planning a Good Metric. *Metrics.* Apress. 2011, pp. 57-81 [online]. Available from: http://link.springer.com.libproxy.abertay.ac.uk/content/pdf/10.1007%2F978-1-4302-3727-3_3 [Accessed 15 October 2012].

Klubeck, M. 2011. Designing Metrics. *Metrics.* Apress. 2011, pp. 25-56 [online]. Available from: http://link.springer.com.libproxy.abertay.ac.uk/chapter/10.1007%2F978-1-4302-3727-3_2 [Accessed 15 October 2012].

Marczak, R., et al. 2012. *Feedback-Based Gameplay Metrics.* ACM. [online]. Available from: http://dl.acm.org.libproxy.abertay.ac.uk/citation.cfm?id=2336733 [Accessed 13 October 2012].

Overmars, M. 2009. *Designing Good Games*. YoYo Games Ltd. [online]. Available from: http://www.cs.umd.edu/class/spring2011/cmsc498m/Resources/GM-Tutorial-Designing-Games.pdf [Accessed 25 September 2012].

Pedersen, C., Togelius, J. and Yannakakis, G.N. 2009. *Modeling Player Experience in Super Mario Bros.* IEEE. [online]. Available from: http://julian.togelius.com/Pedersen2009Modeling.pdf [Accessed 25 September 2012].

Shaker, N., Yannakakis, G.N. and Togelius, J. 2012. *Towards Player-Driven Procedural Content Generation.* ACM. Available from: http://dl.acm.org.libproxy.abertay.ac.uk/citation.cfm?id=2212942&bnc=1 [Accessed 26 September 2012].

Valve Corporation. 2008. *Left 4 Dead*. [disc]. Microsoft Xbox 360. Valve Corporation.

Yannakakis, G.N. and Hallam, J. 2004. *Evolving Opponents for Interesting Interactive Computer Games.* [online]. Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.60.6629&rep=rep1&type=pdf [Accessed 25 September 2012].

Yannakakis, G.N. and Hallam, J. 2004. *Interactive Opponents Generate Interesting Games.* The Maersk Institute for Production Technology. [online]. Available from: http://yannakakis.net/wp-content/uploads/2012/02/cgaidepaper.pdf [Accessed 25 September 2012].

Yannakakis, G.N. and Togelius, J. 2011. *Experience-Driven Procedural Content Generation.* IEEE. [online]. Available from: http://ieeexplore.ieee.org.libproxy.abertay.ac.uk/stamp/stamp.jsp?tp=&arnumber=5740836 [Accessed 13 October 2012].