The Effect of Dynamic Difficulty Adjustment on User Enjoyment in Games.

Rory Donald McLean

Computer Games Application Development

(BSc Hons), 2017

School of Arts, Media and Computer Games
Abertay University

Table of Contents

Contents

Acknowledgements 3

Abstract 4

Abbreviations, Symbols and Notation 5

1. Introduction 6

2. Literature Review 10

3. Methodology 18

4. Results: 30

5. Discussion 43

6. Conclusion and Future Work: 49

7. Appendices: 52

8. References: 62

Acknowledgements

I would like to thank my Supervisor Karen Meyer for all her help and guidance through this project. I would also like to thank my lecturer Paul Robertson for all of his advice for the gameplay and coding within my project. I would also like to thank all my friends and family as without them I would not be where I am today. I would also like to thank all my participants for their time and feedback.

Abstract

Difficulty settings in games can potentially leave a player bored by the lack of challenge or overwhelmed by a game's difficulty. Through the use of Dynamic Difficulty Adjustment (DDA) a game could better adjust to a player's skill level to increase their enjoyment. This project investigates the uses of DDA to improve a player's enjoyment of a game compared to games with Pre-Set difficulty settings. Different methods of DDA are also utilised to test which method is the most effective at improving enjoyment. Three simple First Person Shooter (FPS) applications were created for this project, one which uses Pre-Set difficulty settings, one which utilises a grading system as its method of DDA and one which adjusts the game's difficulty at run time (during play). The main use of DDA is to change both the player and enemy attributes to make the application easier or harder depending on how well the player is doing. A simple questionnaire was used to gather 20 participants' enjoyment levels and overall experience of the applications. The results found that the majority of participants preferred the application which utilises the grading system version of DDA over any other application. Alongside this it was found that the least enjoyed application was the one which utilised Pre-Set difficulty settings. From the results it was also found that DDA is more effective for "Casual" gamers than those who would be classed as "Avid" gamers. Overall the project was a success, by demonstrating a clear preference for DDA compared to Pre-Set difficulty settings the project's research question has been answered with a positive outcome. Some shortcomings of the project include minor game design faults which made the later levels of the application too difficult. This project demonstrates that future investigation of DDA and its effects on enjoyment is required. This could be done by investigating which attributes are the most effective to modify to increase a player's enjoyment, or possibly the effects of DDA on other genres of games to see if a similar effect on enjoyment is found.

Abbreviations, Symbols and Notation

Dynamic Difficulty Adjustment (DDA) – The process of adjusting a game's difficulty based on player performance.

First Person Shooter (FPS) – A game which is played from a first person perspective.

Heads Up Display (HUD) – The player's UI elements, displayed around the edges of the screen.

User Interface (UI) – Visual elements such as player health and ammunition. Can also include interactive menus.

Non - Playable Character (NPC) – A computer controlled character which cannot be controlled by the player.

1. Introduction

Difficulty settings have been an important aspect of game development for a number of years and can be used to give a player the best suited experience for them to play through a game comfortably, either by giving them a simple and easy game or a complex challenge. This is the case when difficulty in games is done well. However, in some instances difficulty in games can either leave the player bored by the lack of challenge or overwhelmed from the drastic jumps in difficulty e.g. between Normal and Hard.

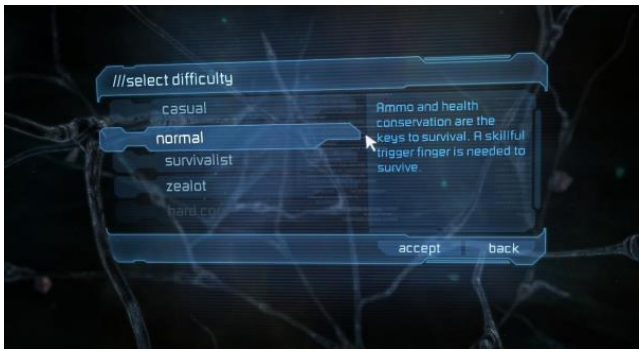


Figure 1- Dead Space 2: Difficulty Select

A relevant example of this is in the *Dead Space* (Dead Space 2008) series of games which were met with good reviews and were highly praised for their mechanics and story which some could argue rejuvenated the horror genre (Andrews 2008). Featuring five difficulty settings (Casual, Normal, Survivalist, Zealot and Hard Core) the player could choose the best setting for them to experience the game, with the latter being unlocked after beating the game on "Zealot" mode. However, the game featured a significant climb in difficulty from its "Normal" setting to "Survivalist". The player could take several attacks from enemies before dying on "Normal" but then be killed in one or two attacks on "Survivalist" (Andrews 2008).

Although it is common practice for many games to ramp up the difficulty between settings, too high an increase in difficulty can leave many players frustrated, and hence discouraged from attempting higher difficulty settings. This is where Dynamic Difficulty Adjustment (DDA) could be used to make the game easier for less experienced players but not lessen the experience for better players.

DDA varies a game's difficulty by adjusting different aspects of the game to make it easier or harder for the player depending on how well they are doing. Examples of this are lowering the game's enemies' maximum health, giving the player more lives and damage output, and spawning less enemies in a given location. The opposite can occur if a player is doing exceptionally well, so the game can make enemies harder to defeat, give the player less resources or make the player more vulnerable to damage. This is what the popular 3rd person shooter game *Resident Evil 4* (Resident Evil 4 2005) made use of by making the game's enemies more aggressive and deal more damage based on how well the player was doing. By tracking the player's accuracy and damage taken the game could adjust how many enemies would appear, how much damage the player could deal and how many resources would spawn. A potential indicator of the success of their implementation of DDA is that this feature was not discovered until the game's official strategy guide was released a year after the game. As such, mention of the game's DDA does not appear in reviews for the game but is discussed and mentioned in Mark Brown's Youtube series *The Game Makers ToolKit* (Brown 2015).

There must be a balance when implementing DDA. On the one hand, the adjustment should not be so obvious as to be noticeable to the player. If the difficulty is increased or decreased too far, this may lead to a negative user experience. Awareness of the DDA system may also allow the player to exploit it. On the other hand, the adjustment should be enough that there is an overall effect on user experience, ideally an improvement.

A key influence for this project has been Robin Hunicke's *The Case for Dynamic Difficulty Adjustment in Games* (Hunicke 2005). By using DDA to manipulate the number of items the player would obtain, they were able to explore how this would affect a player's progress through a game and also monitor the player's view of the game's difficulty and overall experience of enjoyment and frustration with the game. By keeping track of the probability of the player's death when in combat they were able to determine when to intervene and give the player aid to help them win the combat. They stated that;

> "If something this basic can increase a familiar player's enjoyment without ruining their sense of agency or control, imagine what a professionally designed system might contribute to their gameplay experience!" (Hunicke 2005, p.239)

The key focus of this project is to investigate the use of DDA in a simple First Person Shooter Game (FPS) compared to Pre-Set difficulty settings, to determine if its use can make a game more enjoyable, and which method of implementation of DDA would be more effective.

Three different versions of a FPS application have been created to test the uses and enjoyability of DDA and Pre-Set difficulty settings. In one application the player will have no choice as to what difficulty setting they start on but rather start with a grade; set to 3 out of 5. Alongside this as they play through, the game will adjust the difficulty depending on how well they are doing, by increasing or decreasing this grade; this is called the Grading System. The second version of the application will have the user choose which difficulty setting to play on, keeping that difficulty throughout the play session. The final application will adjust the game at run time depending on how well the player is doing rather than adjusting after a player or enemy death.

This game will consist of a simple player controlled character which has to navigate through a level, defeating various Turrets which will shoot at the player. The goal is to defeat all the Turrets in each room and reach the end of the stage to finish the game.

Once all versions of the application have been played, the participant will be asked to fill out a simple questionnaire. The purpose of this will be to determine which version of the application they enjoyed the most and why.

Additional information will also be gathered from the application itself, which keeps track of how many times the difficulty has changed, number of player deaths and what difficulty setting the player ended on. This information will be collected alongside that of the questionnaires to help determine a correlation between DDA and player enjoyment.
To summarise:

“How effective is the use of Dynamic Difficulty Adjustment at improving the user’s enjoyment in games?”

1.2 Aims and Objectives:

- Create a DDA system which utilised a grade system to determine the game's difficulty.
- Create a DDA system which adjusts the game's difficulty as the game is being played rather than adjusting after player or enemy death.
- Create an application with Pre-Set difficulty settings.
- Determine if the use of a DDA system improves a player's enjoyment of a game.
- Compare enjoyment levels of Pre-Set difficulty and DDA.

2. Literature Review

2.1. Dynamic Difficulty Adjustment:

2.1.1. The Case for Dynamic Difficulty Adjustment in Games.

In *The Case for Dynamic Difficulty Adjustment in Games* (Hunicke 2005) the researcher's main goals were to show that DDA can be implemented into a game without disrupting or degrading the core gameplay experience. Hunicke believed that not intruding on the game's overall cycle of action such as the player being able to die, would make for an effective DDA system as disrupting this cycle would break the player's immersion and sense of disbelief when playing a game.

Alongside this they wanted to determine if DDA significantly affects the player's enjoyment of a game's difficulty. One such system they implemented was to predict the probability of the player's death when first engaged in combat, once a likelihood of death had been determined they were able to track when to intervene and supply the player with increments of health during combat. During combat a player is more engaged with the enemies and avoiding damage, hence they may not notice if health is added to their health bar. Hunicke found that this method did not drastically affect the player's suspension of disbelief and immersion in the game. It has been determined that DDA works best when the player is unaware of it being present and thus cannot disrupt the player. A similar mechanic has been implemented into this project.

The results of Hunicke's project, which were gathered via data outputting directly from the application itself, included number of player deaths and time taken to finish. This data was stored to a text file, alongside this a short evaluation form was filled out by participants. The questions were ranged from 1 to 5 to test a player's enjoyment levels and if the players noticed adjustments; for these questions 1 was used for greatly disagree and 5 being greatly agree.

Hunicke found that there was no direct correlation between difficulty adjustment and enjoyment for novice players. However, Hunicke found that more experienced players reported a slightly increased level of enjoyment. Through their studies they were given many suggestions as to how a game could adjust to help the player, which included aim assist for weapons and changing enemy behaviour. These are two aspects of DDA that are being explored in this project.

The results from this study show that:

> *"Even crude adjustment algorithms can improve performance."*

Although only changes to the player's health were being conducted in Hunicke's study, it illustrates that when potentially applied to other game systems such as enemy behaviour or other aspects of the player such as damage output, a DDA system could improve a player's enjoyment of a game.

The player's health was only a small aspect of the game that was affected by DDA in Hunicke's works and does not fully demonstrate what a DDA system could be used for. The study being conducted will build on that of Hunicke by considering DDA modifications to other aspects of gameplay in addition to player health such as enemy behaviour and player damage output.

2.1.2 The AI Systems of Left 4 Dead.

In the web document *The AI Systems of Left 4 Dead* (Booth 2009), they discuss the "AI Director" which was created to manage the game's "dramatic pacing". This pacing determines which paths through the level are accessible to the player, what weapons will spawn and when to spawn enemies and boss monsters. This pacing is what affects a player's "emotional intensity" which is a value given to each player to be used in the game's calculation to determine when to spawn the aforementioned objects. This pacing adjusts the game in real-time to best suit that intensity. If a player's intensity is too high then major threats to the player would be removed. A player's intensity would be affected when being injured, when the player becomes incapacitated or when enemies are

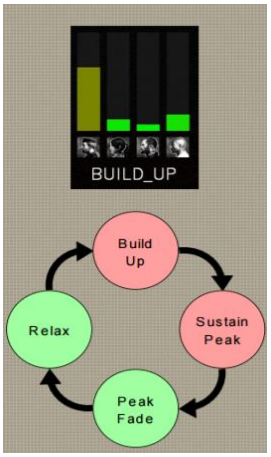


Figure 2- The AI Systems of Left 4 Dead

defeated. The game's enemy population would be determined by this intensity, this takes place in four steps.

Build Up:

Creates a threat population (number of enemies) which will increase the intensity level for the player until it reaches a peak threshold.

Sustain Peak:

Continue the threat towards the player for three to five seconds after player intensity has peaked.

Peak Fade:

Switch the threat towards the player to a minimal threat population and track the player intensity until it has decayed out of the peak range. Intensity is decreased over time to zero when the player is not engaged in combat. This state is needed so current combats can be finished.

Relax:

Minimal threat population is maintained for thirty to forty seconds or until the player has travelled far enough towards the level's end location, once this is done the Build Up phase will resume.

Due to the game's threat population being procedural based on a player intensity, different play sessions will play out differently every time allowing for re-playability and the game being more enjoyable for a longer time. This system does not change the difficulty of the game but rather changes the game's pacing, depending on how well the player is doing. This system of DDA is out the scope of this project but should not be ignored as a means of adjusting the game's difficulty at run time.

This document demonstrates that a game that adjusts and changes every time it is played can lead to a game's life cycle being much longer than other games and be more enjoyable overall. This practice can be applied to games with DDA, by allowing the game to change dynamically to the player, each play session will be different from the last, allowing for the game to be

enjoyable for a longer time. This is supported by Jason Ocampo (Ocampo 2009) in their review of *Left 4 Dead* (Left 4 Dead 2008), there they state;

"*The four scenarios are highly replayable…*(The reason for this) *is the built – in "director" system that manages the action. If you're doing well, it will throw zombies, like a sadistic dungeon master in a game of Dungeons & Dragons. If you're doing badly, it might decide to ease up on you a bit by providing health kits and ammunition.*

2.1.3 All Things Andy Gavin: Making Crash Bandicoot – part 6.

Andy Gavin, co-creator of the game *Crash Bandicoot* (Crash Bandicoot 1996) mentions on his website *All Things Andy Gavin* (Gavin 2011) that they needed a system to help less experienced players to progress through the game but without taking the challenge away from more experienced players. They did this by giving the player a power-up at the start of a round if that player was dying often. Alongside this if the player was dying often because of a specific enemy or obstacle they could lower the speed of that obstacle or enemy each time the player died until they succeeded. Gavin stated that:

"Our mantra became help weaker players without changing the game for the better players… We called all this DDA, Dynamic Difficulty Adjustment… It would lead later Crash games to be the inclusive, perfectly balanced games they became."

By not interfering with the game's overall cycles of action, a concept discussed by Hunicke (2005), mentioned above; the *Crash Bandicoot* game's DDA systems do not interfere with the overall game mechanics and allow the difficulty changes to only apply to weaker players without taking the challenge away from strong players.

A similar approach has been implemented into this project with the Real-Time version of the application. Where minor changes are applied to help weaker players but without taking the challenge away from better players.

One major limitation of this article is that Gavin does not go into great detail as to how they implemented their DDA system but rather gives a brief outline as to what they did for players that were struggling in the game. It is not mentioned how often a player is given help when they are doing poorly at the game or if they are given help after every failed attempt.

However, this article has helped with the creation of some of the mechanics present in this project and shed light on helping weaker players without taking the challenge away from better players.

2.1.4 A Quick Method for Dynamic Difficulty Adjustment of a Computer Player in Computer Games.

In the work *A Quick Method for Dynamic Difficulty Adjustment of a Computer Player in Computer Games* (Lach 2015) the author makes use of two different DDA systems to test their effectiveness in creating a computer opponent which will match a player's skill level in a top down shooter game (a game which is played from a birds eye perspective).

The first method is dubbed "Full Dynamic Difficulty Adjustment of a Computer Player" (FDDACP). Lach uses an evaluation function to determine a human player's and a Non- Player-Character's (NPC) skill level. Unfortunately Lach does not specify how a skill level is determined for the player or NPC but this is most likely determined by gathering the player and NPC's accuracy and number of deaths and then using those values to determine a skill level. After this the two values are compared with each other, if the difference in skill level is high then the key aspect of the NPC's behaviour is adjusted. Lach states that the attributes of the NPC that are being changed are as follows; damage output, amount of ammunition, view range, time taken to reload and amount of enemy health.

The second method of DDA being implemented is "Single Feature Dynamic Difficulty Adjustment for a Computer Player" (SFDDACP), with this method each aspect of the NPC's behaviour is evaluated and adjusted separately. Each feature of the NPC is obtained and compared with the human player's, if the NPC's is significantly higher than the player's then that value is set to the same as the player's, for example the NPC's maximum health would be set to the same as the player's.

When testing was being conducted, six rates are calculated;

- Opponent's win rate (OWR).
- Player's win rate 1 (PWR1).
- Draw rate (DR).

- Player's win rate 2 (PWR2) this is used to achieve more accurate results.
- Health rate (HR).
- Adjustment rate (AR).

Each of these values gives a percentage of how many times one of three situations occurs: a player wins, an opponent wins and how often a draw occurs. It was determined that the larger the draw rate the better matched the player and opponent are. From this they found that the FDDACP method would adjust better to players who have a mixed skill set, as every aspect is being changed the NPC's attributes are more rounded and not swayed one way or the other too drastically by the human player's skill set. Whereas, the SFDDACP method would be more appropriate for when the player has to accomplish a mixture of tasks to finish the game.

Both of these methods of DDA can be compared to the two methods that are being implemented into this project; the FDDACP method draws similarities to the Grade System Application which changes every aspect of the game such as player and enemy health, damage output and number of enemies in relation to the player's grade. Along with the SFDACP method being compared to the Real Time system which adjusts singular aspects of the game at run time to match the player's skill level.

2.2 Games with DDA:

2.2.1 Resident Evil 4.

Figure 3- What Capcom Didn't Tell You About Resident Evil 4. More Enemies When Doing Well.

Figure 4- What Capcom Didn't Tell You About Resident Evil 4. Less Enemies When Doing Badly.

Resident Evil 4 (Resident Evil 4 2005) makes use of a DDA system to minorly adjust the game's difficulty to better suit a player's skill level. This was done by giving the player a grade which was pre-determined by the difficulty setting the player chose at the beginning of the game i.e. if "Easy" difficulty was selected the player was given the grade "3". However, if the player was doing well at the game, killing enemies, taking little to no damage or being accurate with their weapon, the player's grade could increase. When the player's grade increased, enemies would become more aggressive, take more damage to kill or simply more enemies would spawn than usual. The same thing would happen in reverse if the player was doing badly, their grade would decrease making the game easier. This grade would range from 1 to 10, but depending on the difficulty selected would only scale relative to that difficulty, i.e. if "Easy" was selected the lowest the grade could go to was "1" and the highest it could go was "4".

These features were not made apparent when the game was released but rather came to light when the game's official strategy guide was released and made players aware of the feature. This is testament to how effective the DDA system in *Resident Evil 4* was as it did not impact the game in a significant way but rather subtly helped weaker players to beat the game while giving more experienced players a greater challenge if they were finding the game too easy.

Unfortunately there is no official statement as to how this feature works apart from what is mentioned in the game's strategy guide, Mark Brown creator of the web series *The Game*

Maker's ToolKit (Brown 2015) goes into detail as to how this feature works and what aspects of the game it affects. A very similar aspect of this feature has been created for the project in the means of the Grade System version of the application.

2.3 Significant Points:

Most if not all of these articles demonstrate the need to make DDA systems in games subtle and not overtly known by the player. Each method of DDA mentioned has adjusted their games in minor ways and not overly affected each game's cycles of action, rather each system has been used to help strengthen a weaker player's abilities without taking any challenge away from better players. Alongside this each article mentioned uses their DDA systems to affect their games differently, some use them to change the player's behaviour as discussed in *The Case for Dynamic Difficulty Adjustment in games* whereas others change an enemy's behaviour as mentioned in *A Quick Method for Dynamic Difficulty Adjustment of a Computer Player in Computer Games*. It is believed by the author that an effective DDA system that would significantly improve a player's enjoyment in a game would change the behaviour of both the player and obstacles in a game such as enemies and traps.

2.4 Evaluation Methods:

As stated in the paper *The Case for Dynamic Difficulty Adjustment in Games* a questionnaire was utilised to gather information from the participants, alongside this, information was gathered from the application itself and stored in a text file. These two methods were implemented in this study due to the similarities between the two projects. It is believed that by using a similar style of questions mentioned in Hunicke's work, a clear conclusion can be drawn from the participant's results by using a numbering system similar to Hunicke's 1 to 5 scoring questions. Due to this a Likert Scale (McLeod 2008) is being implemented into the questionnaire, this method of data gathering asks participants how much they agree or disagree with a particular statement on a scale of 1 to 7, 1 being strongly disagree and 7 being strongly agree, with 4 being neutral (McLeod 2008). This method ensures that the participant does not feel pressured into marking down an extreme (strongly agree, strongly disagree) but rather can mark down values such as (somewhat agree or don't know).

3. Methodology

3.1 Game Design

As the main focus of this project was to test the uses of Dynamic Difficulty Adjustment (DDA) compared to Pre-Set difficulty levels, most of the time spent was focused around the implementation of DDA and its corresponding features. Due to this, the use of a game engine was chosen to help streamline the creation of a simple First Person Shooter game (FPS) so more time could be dedicated to the DDA systems of the game. The game engine that was chosen was Unity version 5.3.5f1 (Unity Technologies 2005) and its 3D development tools. A FPS game was created to test the uses of DDA as FPS games are quite commonly played amongst casual and avid gamers. Out of the top 10 games sold in 2016, four of them were First Person Shooter games. (Pereira 2017). It was also chosen due to the number of gameplay features that could be manipulated with DDA such as player accuracy, amount of player health and number of enemies to spawn.

Three different applications have been created for this project. The three applications are:

- Pre-Set: simply asked the user to select a difficulty “Easy”, “Normal” or “Hard”, once this is done the game’s settings are set to match that difficulty and never change during play.
- Grade System: assigns the player a grade and depending on said grade will change the game’s parameters to match that grade. The grade will change depending on how well the player is doing.
- Real Time: adjusts various game settings such as player damage output and aim assist being activated as it is being played rather than after a win or lose state has been reached.

3.2 Gameplay Design

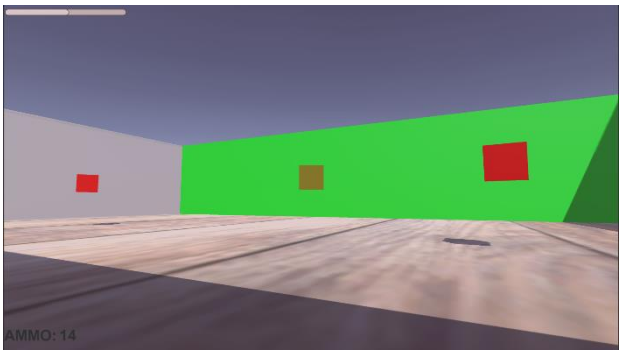


Figure 5- Main Game View, with crosshair and enemies

The type of game that has been created is a simple FPS game. The player can move the camera around with the use of the mouse to aim and can move their character around using the keyboard keys: W, A, S, D. The player can navigate around a simple stage using these controls and can also fire a projectile acting as a bullet from a gun using the left mouse button. As this is a FPS game the player will have a limited number of shots before they have to reload their weapon, this is done by pressing the R key. The reason for implementing a reload mechanic is not just to make the game feel more like a FPS but rather, it is used as a point to gather information from the application to be used for the DDA systems. This is discussed in the section regarding the Grade System application and the Real Time system application.

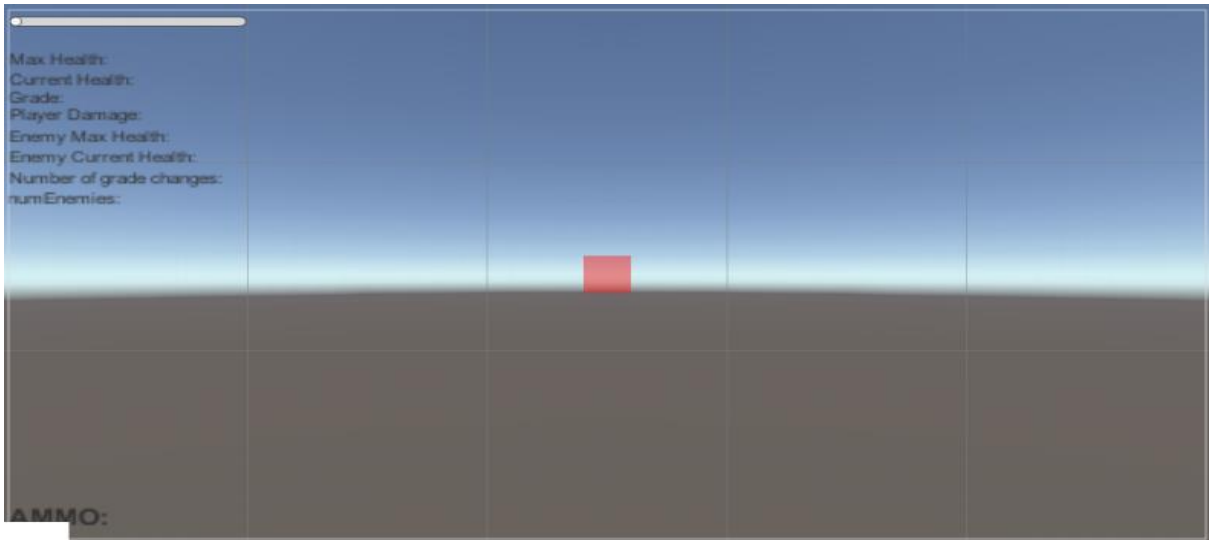


Figure 6- Main Game HUD (Heads Up Display)

Various gameplay features have also been implemented to help make the application feel more like a FPS such as damage indicators for both the player and Turrets. When the player collides with a Turret's bullet projectile the player's screen will flash red for a second. This will help players recognise when they are being damaged and learn to avoid shots. A similar effect happens when a Turret collides with a player's bullet. The Turret will flash when hit to show

a successful hit and will change colour when they are below 50% health to indicate the player has almost defeated them. Alongside this, a simple player Heads Up Display (HUD) has been implemented to display the player's current health and ammunition through a simple slider (Figure 6). This displays how much health the player has left as well as showing the number of shots the player has left before they have to reload. In order to give players a fair chance to acclimatise to the controls and when they spawn into a new room, a brief invincibility is applied to the player which will prevent them from taking damage for a very short time. This is to allow them to get accustomed to the new surroundings when a new round starts.

3.2.1 Enemies

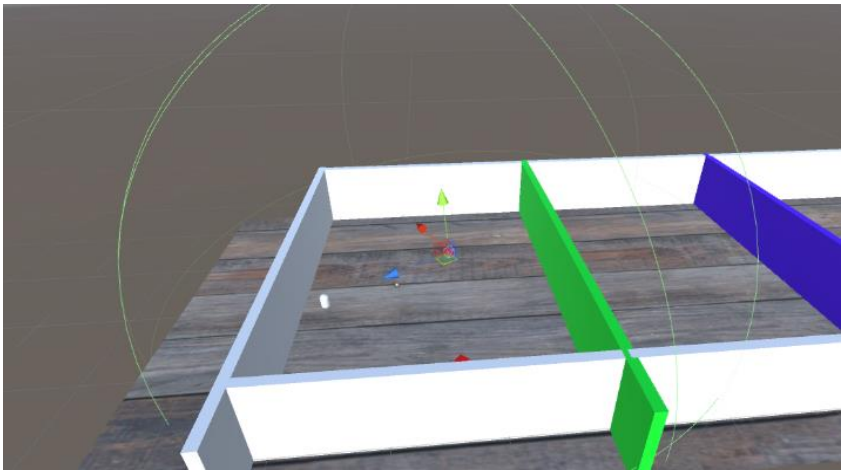


Figure 7- Enemy Field of View

The enemies in the game act as the main obstacle for the player to overcome. Moving along a set track the enemies will move back and forth in a zig-zag pattern acting like sentry Turrets. Each Turret has a collider sphere built around them, and become hostile when the player enters this collider. When this happens, the Turret will turn to face the player and fire a projectile forwards towards the player. Each Turret has a set amount of health, damage output and movement speed depending on the difficulty of the game. Once a Turret's health reaches zero they will be destroyed and no longer endanger the player. Once all Turrets are destroyed in the current room the player will be moved to the next room where more Turrets will appear. Depending on the difficulty setting of the game the number of enemies that will appear in each room will vary to represent that difficulty.

3.2.2 Rooms

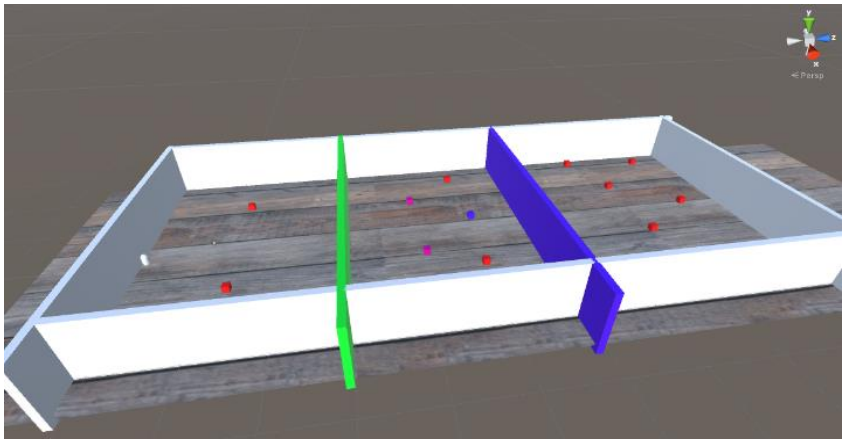


Figure 8- Game Layout, showing enemy starting positions and Rooms

The game is built around three rooms for the player to beat. The player must defeat all the Turrets in the current room in order to progress to the next. Once all three rooms have been cleared the game is over. If a player should be killed by a Turret they will be placed back in the first room. By doing this it is hypothesised that the player will gradually improve at the game after a failed attempt and allow the DDA systems to better adjust to the player's skill level.

3.3 DDA Grade System

The popular 3rd person shooter game *Resident Evil 4* (Resident Evil 4 2005) makes use of a grading system to act as its DDA system. A similar approach has been taken in the first version of the application. When the game starts the player is given a grade to act as the game's difficulty starting point (Grade: 3). This Grade can increase if the player has done well, decrease if they have done poorly or stay the same if their performance is average. Whether the grade changes depends on a number of rules that have been set up to determine how well the player has done. These rules are as follows:

- If Player Health is Max and all enemies are defeated. Increase Grade.
- If Player Health is between 66% and 100% and all enemies are defeated. Increase Grade.
- If Player Health is between 33% and 66% and all enemies are defeated. Keep Grade the same.
- If Player Health is lower than 33% and all enemies are defeated. Keep Grade the same.
- If Player Dies and has defeated less than 2/3 of the enemies. Decrease Grade.
- If Player Dies and has defeated 2/3 or more of the enemies. Keep Grade the same.

Each time the player dies or kills all the enemies in a given room these rules are checked. When it has been determined which rule has triggered the grade will change and each of the game's variables will change in accordance with that grade. Full detail of the variables set for each grade are given in Figure 9. After this is done the round will begin again, by repositioning the player and spawning a new wave of enemies. (From level 1 if the player dies, or to the next room if the player defeats all the enemies. Ending when the third room is completed.)

This Grade system works from 1 to 5; 1 being the lowest grade (Easy) and 5 being the highest (Hard), these grades are useful for gradually increasing the game's difficulty for stronger players to help give a more enjoyable experience and gradually lower the difficulty for a weaker player if they are finding the game too difficult.

Each of the game's values such as player health and damage output change depending on which grade the player is given and will change depending on how well or poorly they have done. This is done by storing these values in a Switch Statement. A Switch Statement works by storing values and code operations in a list of cases i.e.:

Case 1:

Max Health is 180

Case 2:

Max Health is 160

When a specific condition is met, in this case the grade of the player was set to 1 and should be increased due to doing well, the case that would then be selected is 2. This would then run the code that is within that case which in this example is setting the player's maximum health to be 160.

	Max Health	Max Enemy Health	Max Damage	Max enemy Damage	Enemy move speed	No. of enemies in room 1	No. of enemies in room 2	No. of enemies in room 3
Grade 1	180	100	25	5	3	2	2	3
Grade 2	160	120	20	10	3	2	2	3
Grade 3	140	140	15	10	4	2	3	5
Grade 4	120	160	15	10	4	2	5	7
Grade 5	120	160	15	10	5	2	5	7

Figure 9- Changing values for Each Grade.

However, this system cannot control every aspect of the game. One aspect that does not influence the grade is the player's accuracy. In order to help weaker players with aiming, an aim assist mechanic has been implemented into this application, separate from the grading system. When the player fires a bullet a value is incremented to represent the number of shots fired, alongside this each time a bullet collides with an enemy a separate value is incremented. Each time the player reloads their weapon these values are compared against each other to get a percentage of accuracy. This accuracy is saved into an array and stored until the end of the round (when the player dies or when all the enemies are defeated in the current room). When the round ends each of these accuracies are rounded together to determine a final accuracy percentage, if this percentage is below 50% then the game's aim assist is turned on, or switched off if the accuracy is above 50%.

The aim assist works by projecting a Raycast, which is a built in function of the Unity Engine, out from the player object and in the direction the camera is facing, to where the player is aiming. If the Raycast collides with an enemy's collider sphere then the sensitivity of the mouse is lowered to help the player with hitting the Turrets, when the Raycast is not colliding with a Turrets collider sphere the mouse's sensitivity is set back to normal. By doing this, it is believed that the player will not notice the game adjusting and will not abuse the feature to knowingly make the game easier. Alongside this it is believed that by slightly adjusting the game to help weaker players it will make the game more enjoyable as opposed to a game that does not. A similar mechanic is present in the Real Time version of the application.

This is a rather simplistic version of DDA as every aspect of the game is changing at once to fit around the current difficulty rather than individual aspects changing to better suit the player as a whole, for example if the player was very accurate but was being damaged a lot. This version of DDA is used to help build up the player's skill level as they progress by increasing the game's overall difficulty as they get better and lowering it if they are doing poorly.

3.4 Pre-Set Difficulty

In order to truly test the effects of a DDA system on a player's enjoyment this system must be compared to the enjoyment of an application with Pre-Set difficulty settings. By doing this it is believed a clear comparison can be found between a player's enjoyment of an application where they are able to select their difficulty setting (Easy, Normal and Hard) and one which adjusts to their skill level.

A Pre-Set difficulty version of the application was created. When this application starts the player is given the option to choose which difficulty they wish to play on (Easy, Normal, and Hard). This application was built using the Grade system application as a starting point, which means that when the player selects a difficulty setting, a Grade is hard set in the application and will not change for the duration of the game. Easy difficulty is set to Grade 2 as it was hypothesised that setting "Easy" to Grade 1 would make the game too easy, as a comparison Grade 1 could be considered "Very Easy" with Grade 2 being "Easy". "Normal" is set to Grade 3 and "Hard" is set to Grade 5 as again it was hypothesised that Grade 4 was not difficult enough to be considered "Hard" difficulty. Unlike the Grade system application the player's

grade will not change after the player has died or when all the enemies are defeated, rather the game's values such as maximum health, enemy movement speed and damage output are kept at their original values in relation to the difficulty setting selected at the beginning of the application.

As this application is meant to act as a comparison point, the controls and overall gameplay are identical to the other versions of the application. It should also be noted that similar data is being gathered from this application as the others such as number of player deaths, difficulty played on and time taken to finish. These are being stored in a text file for later evaluation. Like the other applications this data will be used to help determine which application participants enjoyed more.

3.5 Real Time DDA

The focus of this project is to determine if the use of DDA is more enjoyable than Pre-Set difficulty settings. It is also important to evaluate different implementations of DDA and try to determine which could be the most enjoyable. The first version of the application utilises a Grading system to act as the game's DDA, with values and gameplay features changing at the beginning of each round. With the third version of the application, gameplay features and various values will change as the game is being played to adjust better to the player's skill level. With this version of the application, four distinct gameplay changes are being implemented to give the player a greater challenge if they are doing well or make the game easier if they are doing poorly; aim assist, player health, enemy behaviour and damage output.

Similar to the Grade system application an aim assist mechanic has been implemented to help with the player's accuracy. By tracking a player's number of shots fired and comparing those to the number of shots that have collided with a Turret a percentage of accuracy can be determined. If the player's accuracy is below 50% the aim assist is activated. The implementation of the aim assist is exactly the same as it is in the Grade system application, however instead of activating or deactivating at the start of each round this version turns on or off after every reload of the player's weapon. The reason for this is to help the player during the game's round rather than after a failed attempt. It is hypothesised that by helping the

player during play, the overall enjoyment of this application will increase due to potentially less failures.

As discussed in *The Case for Dynamic Difficulty Adjustment in Games* (Hunicke 2005) one way of assisting a player if they are doing poorly at a game is to adjust the player's health when the likelihood of death is high. A similar mechanic has been implemented into this application. When the player first takes damage from a Turret's bullet, a timer is started, to measure how much damage the player takes within a certain timescale. If the player takes more than or equal to 60% of their total health in damage within this time scale, the player is given 30% of their maximum health back. As stated in Hunicke's paper players pay more attention to their current threat and surroundings as opposed to their health bar and thus will potentially not notice if it is being adjusted to help them proceed in the game.

Alongside this, in order to help weaker players with progressing through the game, their damage output is manipulated if they are doing poorly. Similar to the health regeneration mechanic mentioned above, if the player's current health is less than or equal to one third of their maximum health, the player's damage output is increased from 15 to 25.

In order to make the game more challenging for players that are doing well, the game's Turret behaviour is manipulated to give a greater challenge. To do this the player's movement is tracked to determine if the enemy's behaviour should be changed. If the player stops moving around the stage and stands still in a corner, firing on the enemies for a set amount of time the Turrets movement speed will increase. This enables the Turrets and their collider sphere to move a greater distance on their movement path allowing them to see the player from greater distances and hit them, making the game more difficult.

This also makes the Turrets harder to hit and thus forces the player to move around the stage. This mechanic encouraged every play tester to play in a similar play style and thus not potentially change the game's test results due to different play styles.

3.6 Questionnaire/ Data Gathering.

To gather more insightful information from testers on their enjoyment of each application a simple questionnaire was created and was given to each tester after they had finished playing the three applications. Although information is being gathered directly from the applications themselves, the information they give can only be used for determining how well a participant did at each application but not how much they enjoyed them. The information that is being gathered from the applications are as follows:

- Number of times Grade changes.
- Current health (amount of health the player had at the end of the last round played).
- Number of deaths (total).
- Number of deaths in each room.
- Total time played.
- Damage taken in each room (at the end of last round played).
- Player accuracy.

This is why a questionnaire is being utilised as it is believed that the information gathered from both the applications and questionnaire can be used together to draw more conclusions between the player's enjoyment of an application and how well they did.

The types of questions being asked on the questionnaire are as follows, with a copy of the questionnaire being included in the Appendix section:

- How often the participant plays games per week. (Used for player metrics, to possibly establish a link between casual/ avid gamers and enjoyment with DDA).
- If the idea of a game adjusting its difficulty to the player's skill level appeals to them.
- Which application the participant enjoyed the most.
- Which application they found the most difficult.
- If they noticed which applications had the DDA systems in place.

- What features of the game they believed to be affected by DDA.
- And any other comment on gameplay/ game design they may have.

Once a participant finished all applications they were handed the questionnaire to fill out, they were not given any results from their play session until after the questionnaire was finished. This was to avoid any bias between applications or distort test data. A Likert scale (McLeod 2008) was used for the questionnaire to help give a numerical value to some of the statements. For example:

> "On a scale of one to seven, how much did you enjoy application 1? (Please Circle)
> 1------2------3------4------5------6------7"

Once all the testing was completed an average was taken from all the values given for each application. Using this average, a clear indicator of enjoyment can be given to each application and compared to each other to determine which version of the application was the most enjoyable. A higher average indicates a more enjoyable application.

At the end of the questionnaire there is also a small section for participants to comment on the game's design and how well it functioned, this is used to help determine any bugs or faults with the applications so they may be addressed and amended, as well as any other useful information that may not have been anticipated.

3.7 Data Evaluation:

For this project's testing stages, 20 participants were chosen at random to play through the applications and take part in the questionnaire. One question that is being asked in the questionnaire is as follows:

> "How often do you play video games? (Please tick)
> 0 to 5 hours a week.
> 5 to 10 hours a week.
> Over 10 hours a week."

This is being used to help determine if the effects of DDA could potentially be more enjoyable for “Casual” gamers as it could make an overbearing challenge easier to overcome or more “Avid” gamers who enjoy an added layer of challenge to their games.

Both quantitative data and qualitative data are being gathered for this project as a clear metric for a player’s enjoyment of an application cannot clearly be found from one form of data alone. By using the Likert scale (McLeod 2008) questions which ask a player how much they enjoyed an application, alongside data gathered from the applications, it is believed that this information can be used in conjunction with the section of the questionnaire which asks for comments on the application as a whole. This is to try and find a clear connection between a player’s enjoyment and how difficult they found the application along with which parts of the application they enjoyed the most.

3.8 First Iteration Testing.

Once all three applications, questionnaire and bug testing was completed, the first testing process was conducted. The first applications that were being tested were quite different from the final iteration of the project. At the time, testers had suggested implementing some cover for the player character to hide behind in order to protect themselves from damage. This small feature was soon implemented and testing was continued, with the former test results being discarded from the final evaluation. However this small feature led to a drastic fault with the application as a whole. Players were now able to prevent themselves from taking damage entirely and due to being in cover were able to shoot and destroy the Turrets without them being able to fire back. This unfortunately would distort the desired test results as different play styles were being utilised by players. It is believed that with a more complex and robust DDA system the application would be able to adapt to different play styles, however this is out of the scope of this project. To resolve this issue, all cover was removed from the game and all test results gathered with this iteration was removed from the end results. By doing this, it is likely that all players will play in a similar play style which in turn would help keep test results in the desired state.

4. Results:

4.1 Questionnaire Results:

For the testing of this project, 20 participants were selected with four groups of 5 playing the applications in different orders:

Five participants would play: PreSet – Grade – RealTime.

Five participants would play: Grade – PreSet – RealTime.

Five participants would play: RealTime – PreSet – Grade.

Five participants would play: RealTime – Grade – PreSet.

This was to avoid participants having any preference of application due to the order they were played in and would generate more accurate results towards enjoyment.

Each participant was asked to fill out a short questionnaire after all three applications were completed, the first question asked the participant how often they would play video games a week. This was used to help form a demographic for “Casual” and “Avid” gamers to try and find a link between Dynamic Difficulty Adjustment (DDA) and how often a tester would play games.

“How often do you play video games?”

Number of Hours	0 to 5 hours:	5 to 10 hours:	Over 10 hours:
Number of participants	3	6	11

Figure 10 – Number of hours a week video games are played.

Although participants were chosen at random, the majority of participants can be described as “Avid” gamers.

The second statement of the questionnaire asked which difficulty the participant would typically play a game on. This was used as a means to gather information on the preferred difficulty setting of participants and would be used to help draw a connection between which applications they would prefer due to how difficult they found each application.

“What difficulty setting do you normally play on when playing games?”

Difficulty Setting	Easy	Normal	Hard
Number of participants	0	16	4

Figure 11 – Preferred Difficulty.

It can be seen that the vast majority of participants preferred to play games on “Normal” difficulty settings, with only 4 preferring harder settings and none out of the 20 participants choosing to play games on “Easy”.

The third statement asked participants if the idea of a game adjusting its difficulty settings to better suit their skill level appealed to them. It was hypothesised that the outcome of this question would reflect the overall outcome of the project with the applications which featured DDA being the most enjoyed by participants.

“Does the idea of a game changing its difficulty to suit your skill level appeal to you?”

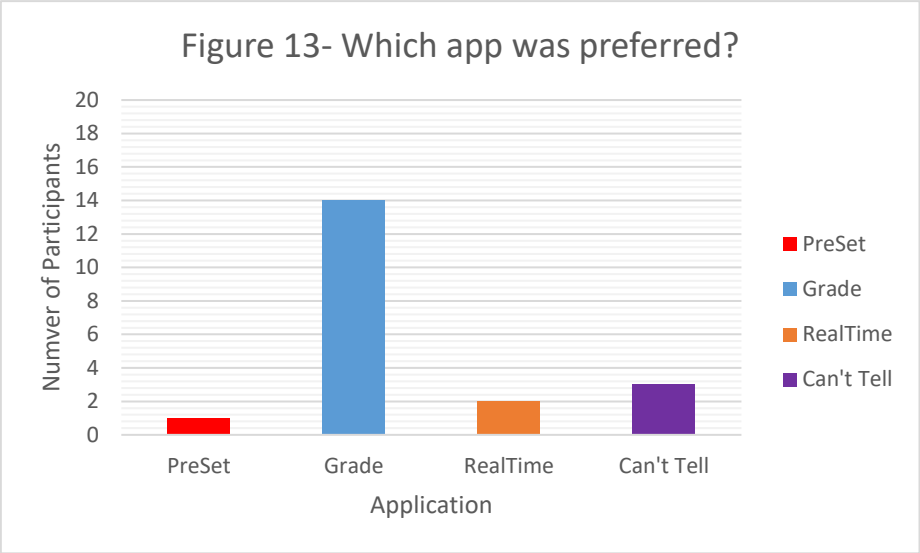
Answer	Yes	No
Number of participants	18	2

Figure 12 – Idea of a game adjusting its difficulty.

As shown almost all of the participants stated that they would like the idea of an application changing how difficult it was to better suit their skill level, with only 2 out of the 20 stating that they would not. However, out of the 2 participants that stated “no”; one of them said they preferred the “PreSet” application and the other stated they preferred the “Grade System” application. Although there was one participant that stated they did not like the idea of a game adjusting its difficulty they seemed to still prefer the application which featured DDA.

The fourth statement asked the participants which application they preferred over the rest. The answer to this question would reflect the entire outcome of the project, if the preferred application was one of the applications with DDA then the research question: *“How effective is the use of Dynamic Difficulty Adjustment at improving the user’s enjoyment in games?”* would be answered with a positive outcome towards DDA’s use in games.

“If any, which application did you prefer 1, 2, 3 or can’t tell?”



From this chart, it is clearly demonstrated that overall the preferred application was the “Grade System” application with 14 out of the 20 participants choosing it as their preferred application. Alongside this it is shown that both applications featuring DDA were considered more enjoyable than the application which featured Pre- Set difficulty settings. However the “RealTime” application only has one more participant compared to the “PreSet” application, showing only a small improvement in enjoyment.

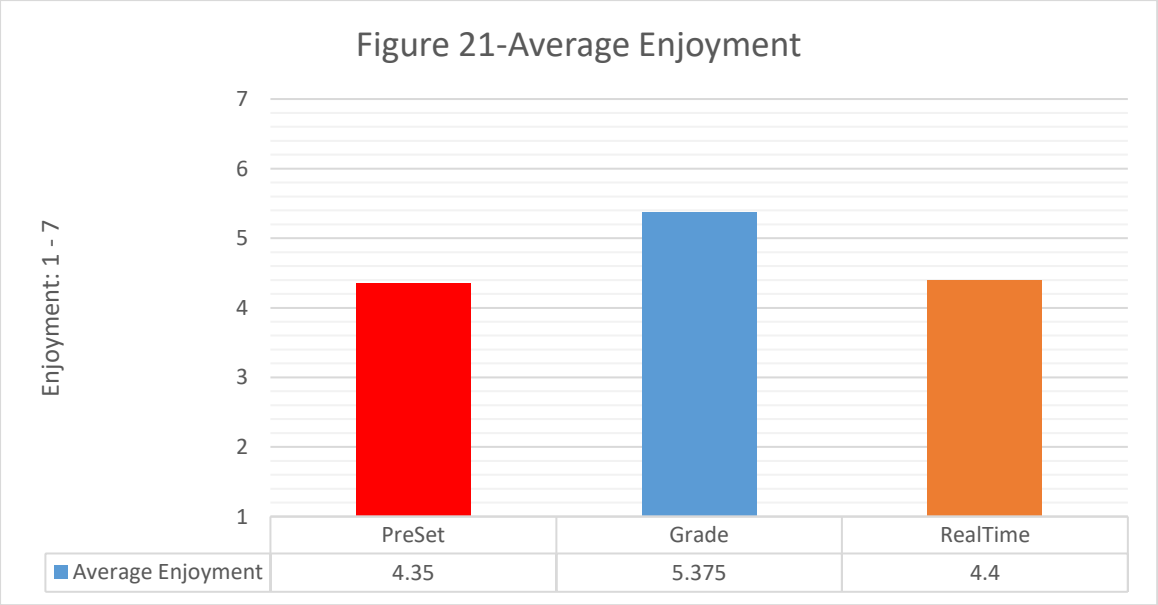
Taking the number of hours played as a means to determine if a participant is a “Casual” or “Avid” gamer, these figures were then used in correlation with the level of enjoyment question to help find a connection between which application was most enjoyed amongst “Casual” and “Avid” gamers.

It can be seen that both “Avid” and “Casual” gamers overall preferred the “Grade System” application. It must be noted that three out of the 20 participants chose 0 to 5 hours, six chose the 5 to 10 hours and 11 chose the 10 plus hours. See Appendix E for graphs on average enjoyment of “Casual”, “Normal” and “Avid” gamers.

Statements five to seven asked participants how much they enjoyed each application on a range from 1 to 7. The following tables show the level of enjoyment for each application, each table represents the different order the applications were played in and how much a participant enjoyed each application.

“On a scale of one to seven, how much did you enjoy application 1/2/3?
1------2------3------4------4------5------6------7”

The results shown in Appendix F show that throughout the four groups the “Grade System” application is consistently the preferred application with the “PreSet” and “RealTime” applications being fairly close to each other in terms of enjoyment. In order to demonstrate a clear answer as to which application was enjoyed the most overall, an average of all outcomes was taken and compared against each other.



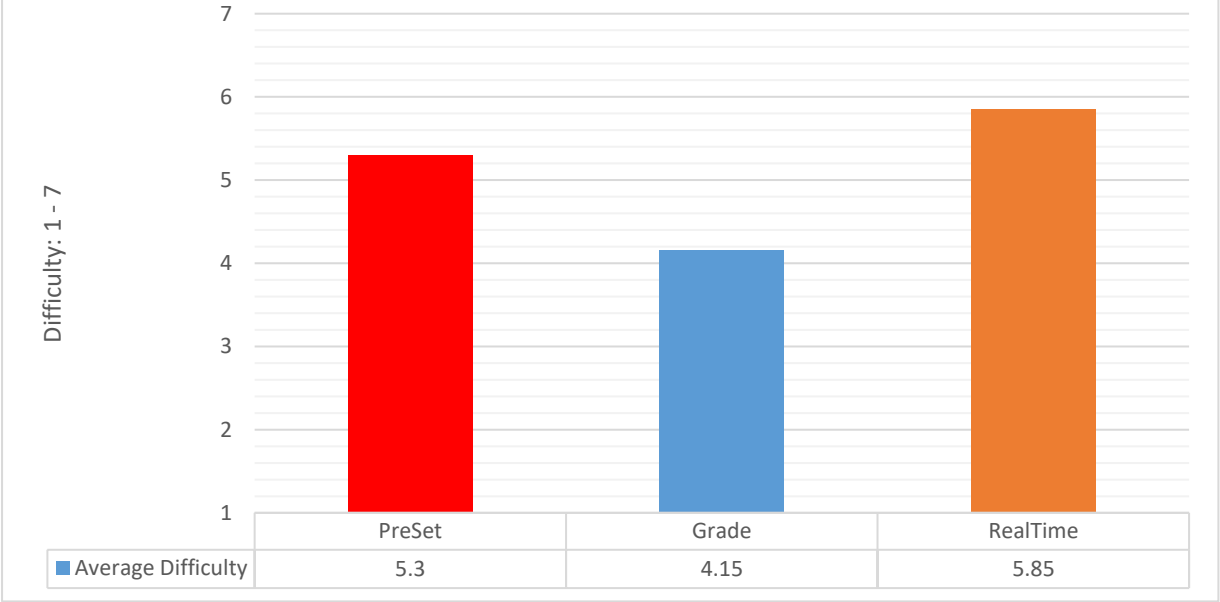
From this chart, it can be seen that applications “PreSet” and “RealTime” are the less enjoyed applications but only have a difference of 0.05 between each other. This means testers only had a minor improvement of enjoyment with a DDA system which adjusts to them at run time compared to an application where the tester chooses their difficulty level at the beginning.

Statements eight to ten asked each participant how difficult they found each application. This statement was used to help find a connection between how difficult a tester found an application and how much they enjoyed it. It was hypothesised that the more difficult a tester found an application the less they would enjoy it.

“On a scale of one to seven, how challenging did you find application 1/2/3?
1------2------3------4------5------6------7”

As shown in the graphs in appendix G, applications “PreSet” and “RealTime” are considered the most difficult throughout the four groups. However, it should be noted that the measures of difficulty towards the “PreSet” application will vary significantly due to the difficulty setting each tester was able to play the application on. As the majority of testers stated that they prefer to play games on “Normal” or “Hard” difficulties, this is most likely the difficulty setting they chose to play the application on.
An average was also taken from all the results and compared against each other to determine which application participants found the most difficult overall.

Figure 26-Average Difficulty

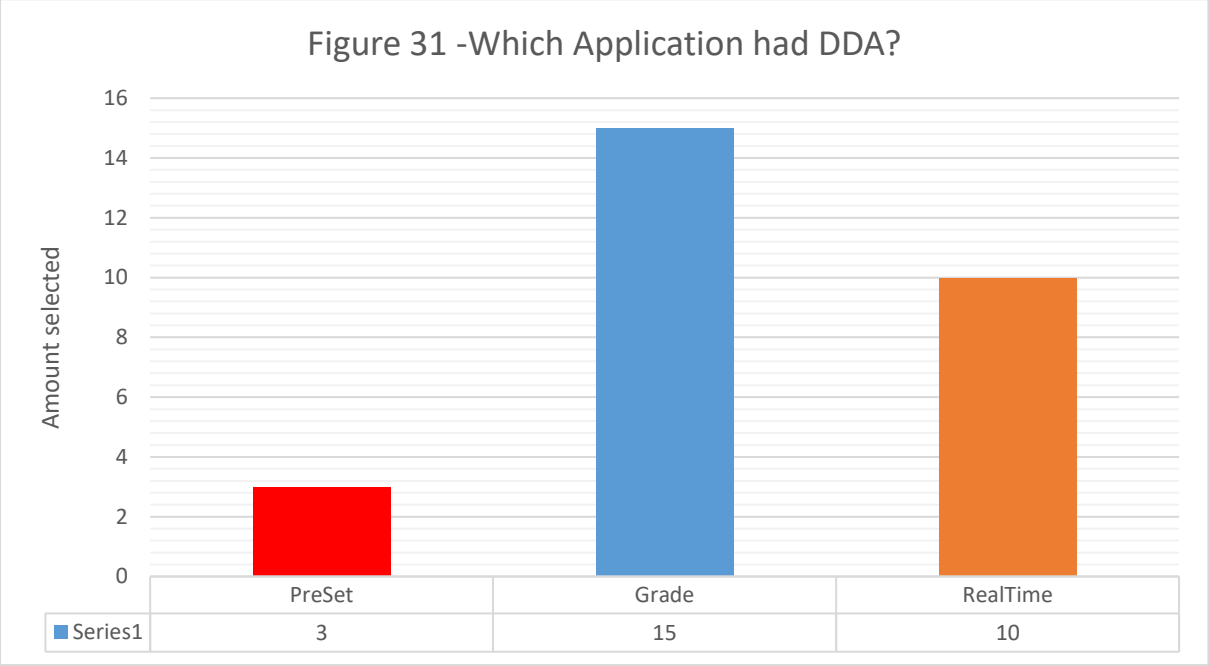


Overall participants found the “Grade System” application to be the least difficult, it should be noted that it was this application which was considered to be the most enjoyable. Much

like the average level of enjoyment, the “RealTime” and “PreSet” applications share a similar trend with having a difference of only 0.55.

The eleventh statement asked each participant which application they believed had DDA. Again, to avoid bias towards any one application, four sets of data was taken, with a fifth being the average of the outputs. See Appendix H for individual graphs.

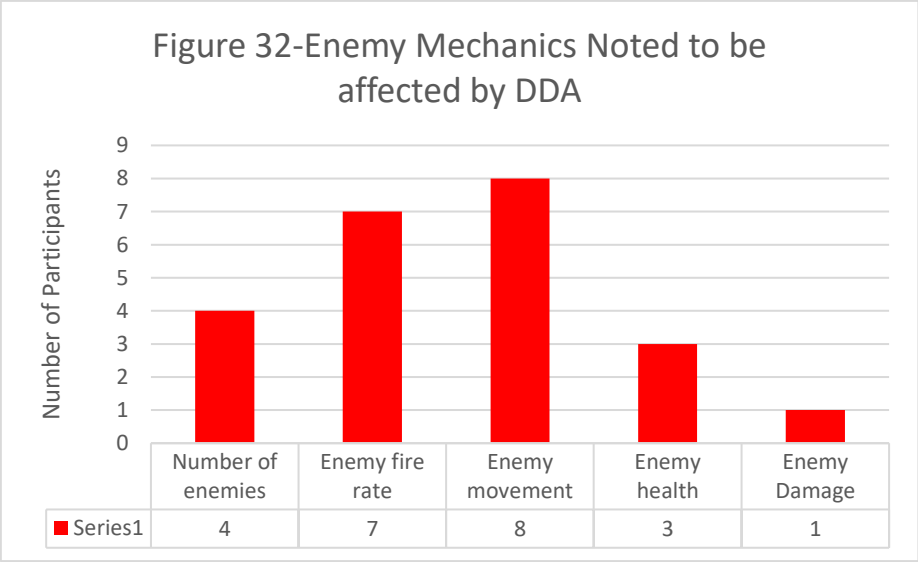
“Which application(s) do you think had Dynamic Difficulty Adjustment?”

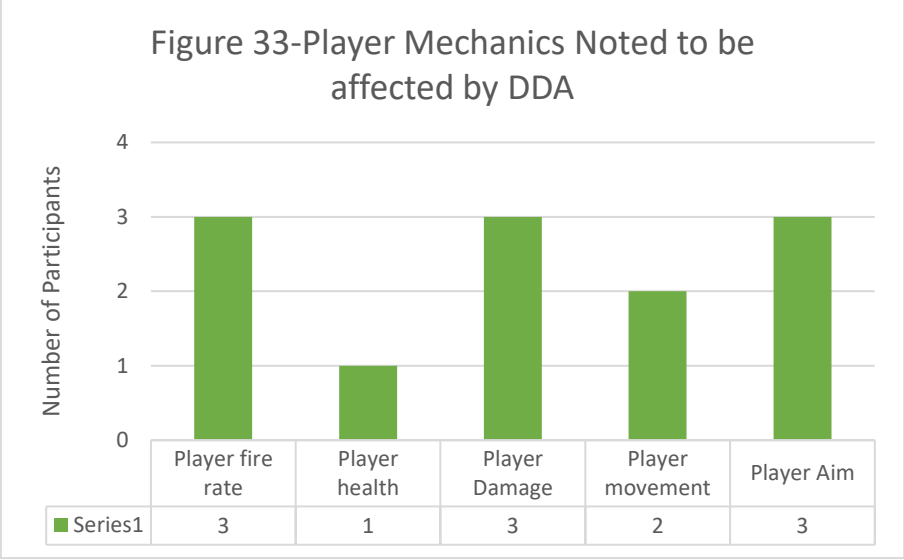


As shown, it is clear that the majority of testers were able to notice DDA mechanics being present in the “Grade System” application, with half of the participants noticing DDA mechanics in the “RealTime” application. It should be noted that participants could select more than one option for this question on the questionnaire. It should also be noted that some participants believed that DDA mechanics were present in the “PreSet” application, which did not include DDA.

4.2 Mechanics Affected by DDA:

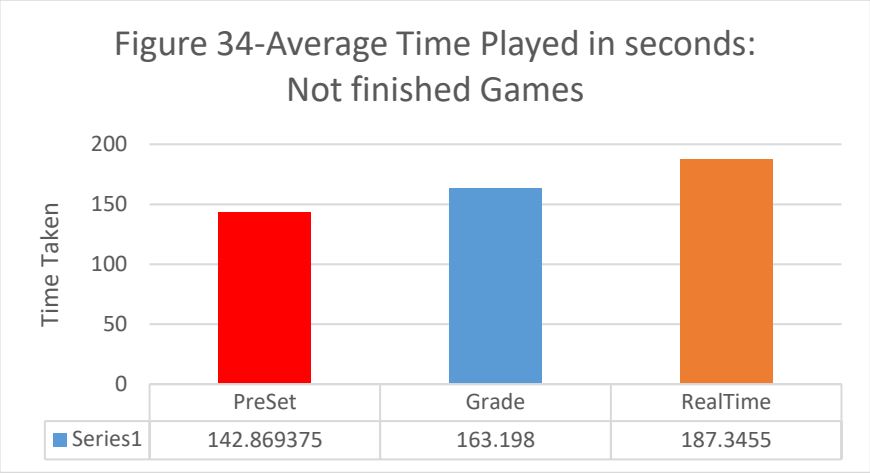
As part of the questionnaire participants were asked to note which parts of the application they believed to be affected by DDA. The following tables demonstrate which parts of the application participants noticed had DDA mechanics the most. Some of the noted mechanics that participants mentioned to be affect by DDA are in fact not affected at all, such as the enemy fire rate and player movement.



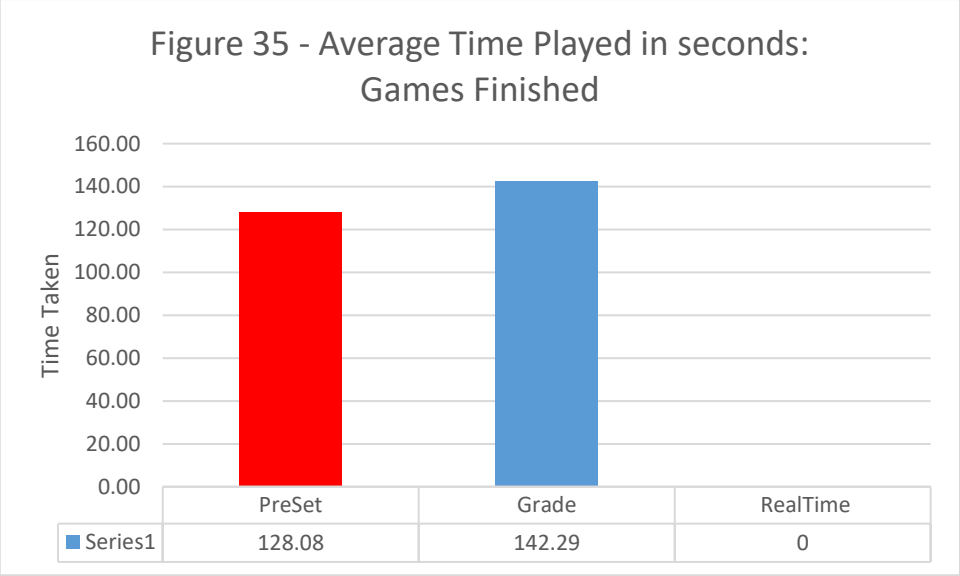


4.3 Application Data Results:

At the beginning of each test session, the participant was told to play each application for as long as they were enjoying it or until they reached the end of the application. The following chart is the average time spent on each application.

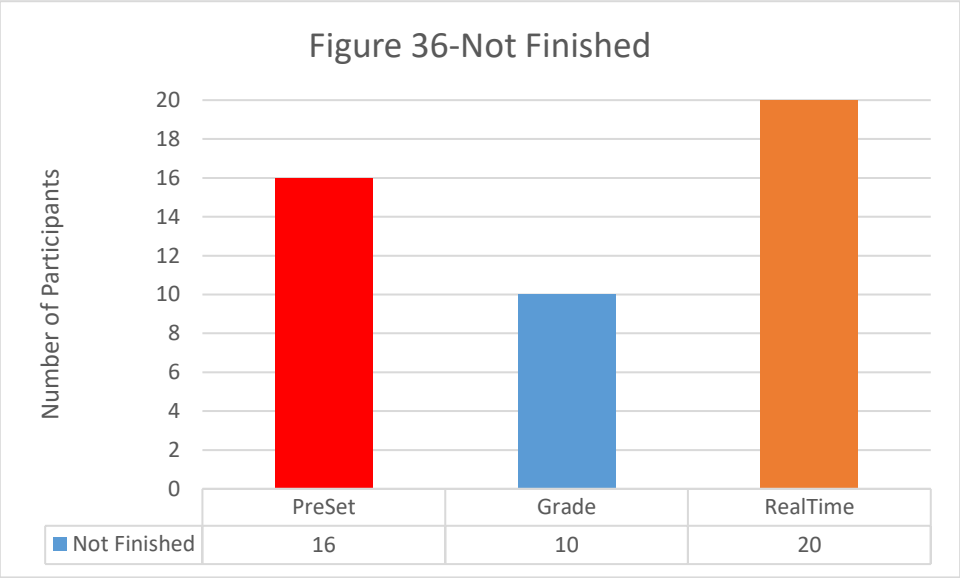


It should be noted that the majority of participants did not finish each application. Although participants were told to play each application for as long as they were enjoying it, it is unclear if more time was taken on an application due to a desire to reach the end or an enjoyment of the game.



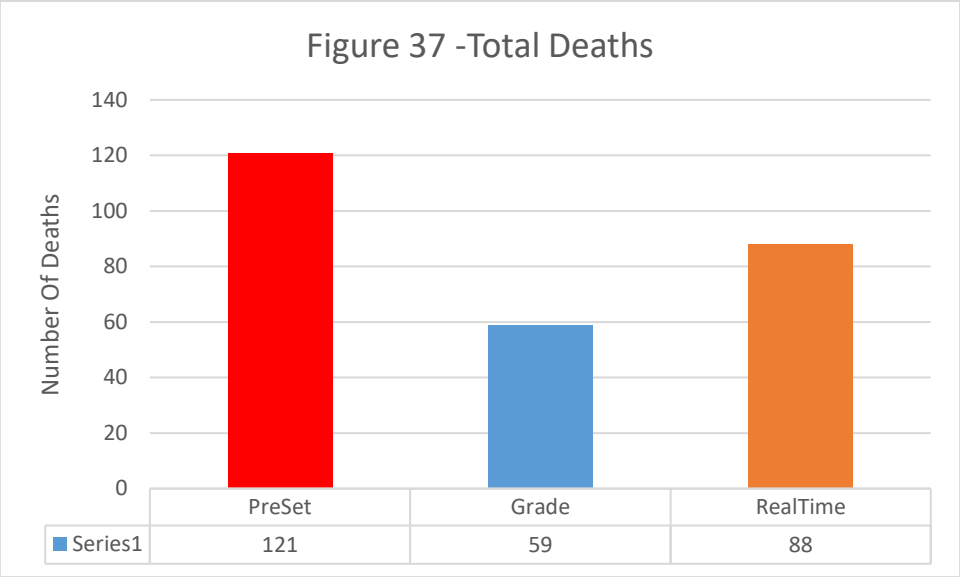
From the results taken from the average seconds played for games that were finished, the “Grade System” application was played for the longest time. As the majority of participants were able to finish the “Grade System” application the average time spent will be significantly higher than the other applications.

The following data is the number of participants who were unable to reach the end of the application or chose to stop before reaching the end.



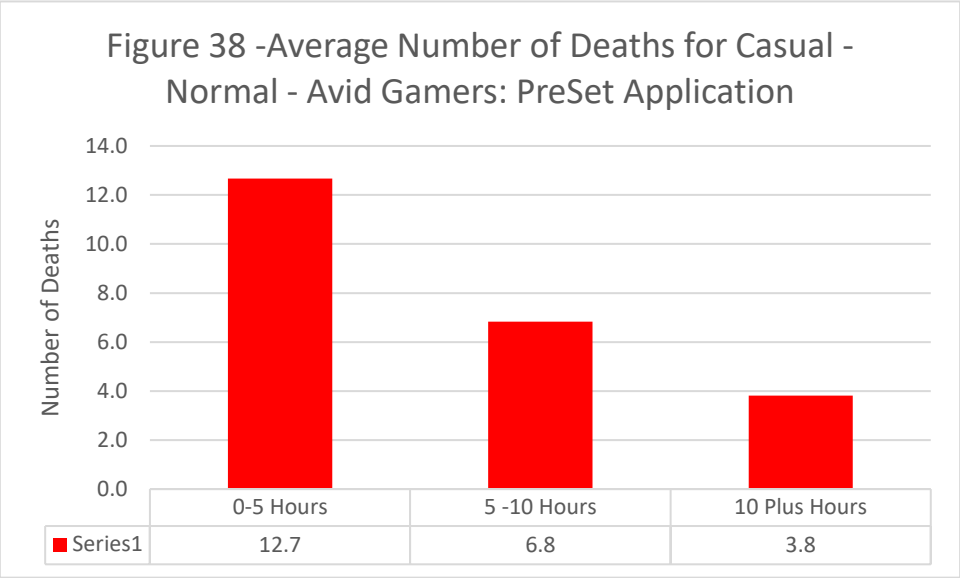
As shown, all 20 participants were unable to reach the end of the “RealTime” application, it is hypothesised that the application was made too difficult during implementation and thus has affected the number of participants that were able to complete the application and their overall enjoyment.

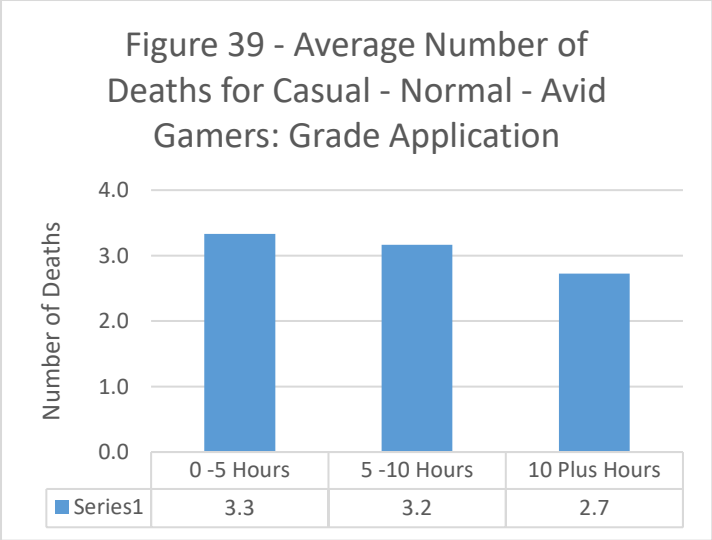
In order to draw more of a connection between how difficult a player found an application and how much they have enjoyed it, the total number of deaths have been gathered from each application.

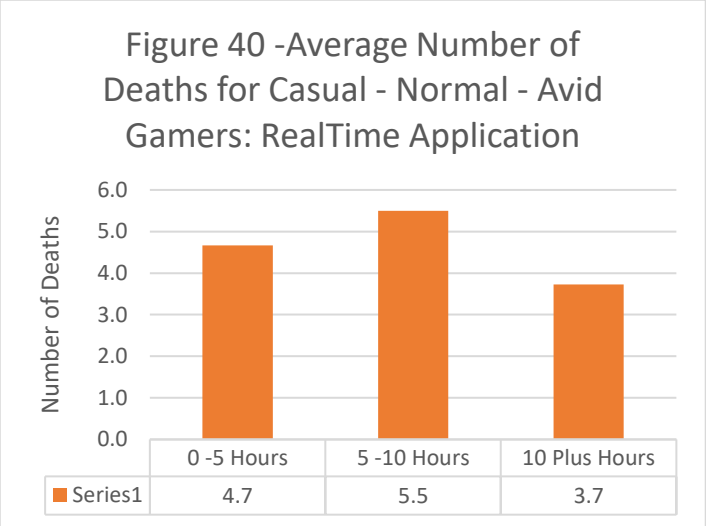


As shown, the results from this chart are very similar to that of the average difficulty of each application. This indicates a link between the number of deaths that occurred, overall enjoyment level and how difficult the participants considered each application.

To draw more of a conclusion towards difficulty and enjoyment for “Casual” and “Avid” gamers, the number of deaths for “Casual” and “Avid” gamers have been gathered from each application alongside the total number of deaths.

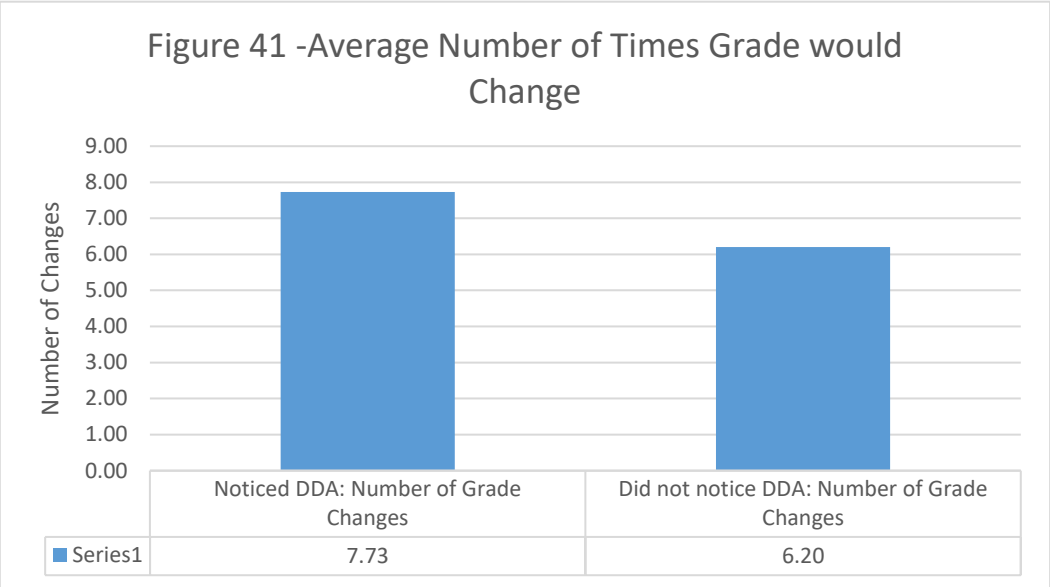






From these results it can be seen that on average the “Casual” participants died more often in both the “PreSet” application and the “Grade System” application. Again it must be noted that there were significantly less “Casual” participants than there were “Avid”.

Out of the 20 participants that took part in the project, 15 noted that the “Grade System” application featured DDA mechanics with only 5 not noticing any DDA mechanics present.



From this chart, we can see that on average the participants that noted that the “Grade System” application had DDA mechanics, had their grade change more frequently than those that did not. Again it should be noted that only 5 participants did not note the “Grade System” application to have DDA. It is hypothesised that with further testing with greater numbers of participants a similar trend will occur, with the majority of participants noticing DDA in the “Grade System” application with only a small number not noticing the features.

“Casual” 0 -5 Hours Played	“Normal” 5 – 10 Hours Played	“Avid” 10 Plus Hours Played
3	4	8

Figure 42 – Number of “Casual”, “Normal” and “Avid” games who noticed DDA, in the “Grade System” application.

“Casual” 0 -5 Hours Played	“Normal” 5 – 10 Hours Played	“Avid” 10 Plus Hours Played
0	2	3

Figure 43 – Number of “Casual”, “Normal” and “Avid” games who did not noticed DDA, in the “Grade System” application.

The results show that, the majority of “Avid” gamers were able to notice the DDA mechanics present in the “Grade System” application. Alongside this all 3 “Casual” gamers were also able to notice the DDA mechanics present in the application. It is unclear whether playing games for a longer time enables participants to notice changes more frequently than those that do not. However, it is hypothesised that with further testing and with larger numbers of participants a clearer outcome on whether playing games for longer affects the player’s ability to notice changes in a game.

5. Discussion

5.1 Preferred App Overall:

From the results of this project it would appear that applications which feature Dynamic Difficulty Adjustment (DDA) mechanics were preferred over an application with Pre-Set difficulty settings. Overall the “Grade System” application was significantly enjoyed more compared to the other applications.

It was hypothesised that due to the “Grade System” application adjusting every aspect of the game rather than individual aspects to better suit the player, is why it was considered the most enjoyable application.

For example the “Grade System” application will de-spawn enemies from the game if the player is doing poorly whereas the “RealTime” application consistently has the same number of enemies throughout the game. It was believed that the number of enemies present in the game is a major factor of the participant’s enjoyment; if the player was faced against a large number of enemies it could make the game seem unfair and too difficult. During implementation the “RealTime” application was given the same number of enemies that would be present on the “Hard” setting of the “PreSet” application but with less health and less damage output. This at its core is a gameplay fault and has unfortunately affected the overall enjoyment of the “RealTime” application.

This is supported by participants stating that the “Grade System” application was the least difficult application overall but also the most enjoyable. As stated it is hypothesised that an application which will adjust both the enemy behaviour and player behaviour is the main contribution for a player enjoying an application compared to one which will only adjust the player character’s behaviour; as in the “RealTime” application.

However, although the information gathered from the questionnaires demonstrated a preference towards the “Grade System” application, the information gathered from the applications themselves appears to contradict this information. As stated, participants were told to play the applications for as long as they found them enjoyable or until they finished them. Taking the times from the participants that were unable to finish the applications as a measure of enjoyment, it is shown that the “RealTime” application on average was played for

the longest time with the “PreSet” application played for the shortest time. It was theorised that this is due to the participants not wanting to be “beaten” by the application and a need to beat the game rather than continuing to play it out of enjoyment. This idea is also supported by the results of this project showing that the “Grade System” application was the most enjoyed while having the highest completion rate out of the three applications.

As discussed earlier in the Literature review section of this project Lach’s work *A Quick Method for Dynamic Difficulty Adjustment of a Computer Player in* Games (Lach 2015) involved two different methods of DDA, one method which adjusts every aspect of the enemy Non – Playable Character (NPC) and one method which adjusts individual elements to better match against the player. As stated these two methods drew similarities to the “Grade System” application and “RealTime” application respectively.

Similar to the results of Lach, the “Grade System” application which changes every aspect of the game such as enemy behaviour and player behaviour was considered to be the preferred application. This demonstrates that an application which features DDA is best suited when every aspect of a game is adjusted to better suit the player’s needs. This is in contrast to adjusting individual aspects of a game as in the “RealTime” application. However, it must again be stated that the enjoyment of the “RealTime” application might have been negatively affected by the game design and the number of enemies present in the application.

5.2 Avid and Casual Gamers:

In the case of DDA having more of an effect on “Casual” (played games for 0 to 5 hours a week) gamers, it appears that all 3 participants stated they preferred the applications which featured DDA, with the majority preferring the “Grade System” application. It was believed that due to the application changing the difficulty to levels comparable to the “Easy” setting in the “PreSet” application, it allowed the player to progress through the game without any overwhelming challenges. This allowed them to finish the game, as it was also thought that players have a greater level of enjoyment towards a game and this enjoyment incentivises them to finish. Unfortunately there were very few “Casual” participants for this project so it is unclear if the preferred application being those that featured DDA is by chance or a common trend with “Casual” gamers, this unfortunately is a limitation of this study.

Similar to the “Casual” gamers, the majority of “Avid” (played games for 10 plus hours a week) gamers stated they preferred the applications which featured DDA with only 1 stating they preferred the “PreSet” application and 2 stating they could not tell which application they preferred the most.

Using the average enjoyment of each application taken from “Casual” and “Avid” gamers it would appear that DDA has more of an effect on “Casual” gamers than those who would be classed as “Avid” gamers. With “Casual” gamers there was a difference of 1.67 in the average enjoyment rate using the Likert Scale (McLeod 2008) values between the “PreSet” application and “Grade System” application where there was only a difference of 0.37 for the “Avid” gamers.

Players who stated they played games between 5 to 10 hours a week, also stated that the “Grade System” application was their preferred application, demonstrating a similar gap of enjoyment between the “PreSet” application and the “Grade System” with a difference of 1.66 in the rate of enjoyment. It would appear that DDA has less of an effect on enjoyment, for players who play video games more often.

This however indicates that the use of DDA is an important factor for helping newer and weaker players enjoy a game, compared to starting a game and becoming frustrated with the difficulty and giving up. This supports Andy Gavin’s views on DDA demonstrated on his website *All Things Andy Gavin* (Gavin 2011) where he mentions helping weaker players without changing the game for better players.

It is hypothesised that by adjusting the player to make them “stronger” and adjusting the enemies to make them “weaker” the player will feel like they are doing well at the game and thus potentially enjoy the game more. This is supported by the results of the average enjoyment of each application and the number of deaths that occurred. The “Grade System” application overall had the least amount of deaths recorded and was considered the most enjoyable and preferred application. This could be due to the player becoming “stronger” when doing poorly at the game and then being able to do better and potentially beat the game. Alongside this, both “Casual” and “Avid” gamers performed significantly better in the “Grade System” application compared to that of the other two applications. This is supported

by the number of deaths for “Casual” gamers in the “PreSet” application being almost four times the amount that occurred in the “Grade System” application.

Alongside this, the results gathered from this project seem to contradict the results gathered from Hunicke’s work *The Case for Dynamic Difficulty Adjustment in Games* (Hunicke 2005) where they found that more experienced gamers showed a slight increase in enjoyment and more novice players showed no direct connection between difficulty adjustment and enjoyment on a game.

In order to judge what aspects of a DDA application a player enjoys further, more research and study should be conducted in this area. This would involve more detailed questionnaires and focus group studies with participants to further understand which parts specifically the participant enjoyed and which parts they did not. In order to gain firmer conclusions on whether or not DDA affects a player’s enjoyment, larger numbers of participants would be needed. As there were only 20 participants present for this project, a more appropriate number ranging in the hundreds would be needed to fully gauge how important the need for DDA would be if the majority of those participants showed signs of improved enjoyment.

5.3 Noticing DDA Mechanics:

It was hypothesised that DDA features work best when they are not noticed by the player and if they were, would negatively affect the player’s enjoyment due to the player’s sense of disbelief being broken. As demonstrated by the results of this project, a large majority of participants noticed DDA features being present in the “Grade System” application and the “RealTime” application, with some even noting to have seen DDA features in the “PreSet” application where none were present. It would appear that the hypothesised consensus of noticeable DDA mechanics would negatively affect a player’s enjoyment is rejected. This is demonstrated by the “Grade System” application being the preferred application and most enjoyed but was noted by participants to have featured DDA mechanics the most.

From the results, participants noticed more DDA mechanics involved with the game’s enemies rather than the player character. It would appear that noticing changes involved with the game’s enemies does not negatively affect a player’s overall enjoyment of a game as demonstrated more noticeably with the “Grade System” application. As with this application

the number of enemies would change depending on how well the player was doing along with how much damage it takes for the enemies to be defeated. These two DDA mechanics can be argued to be the most noticeable, but appears not to have negatively affected the enjoyment level of the “Grade System” application.

Out of the 20 participants only 1 noticed a change in the player’s health and 3 noticed changes to the player’s aim. These results are similar to that of Hunicke with their work *The Case for Dynamic Difficulty Adjustment in Games* (Hunike 2005). Hunicke found that participants did not notice changes being applied to the player’s health during combat as the participants are more focused on the enemies and current obstacles, paying little attention to their “Health Bar”.

This appears to be a common recurrence with First Person Shooter (FPS) games. It is theorised that in order to make effective DDA mechanics which are not noticed easily by the player and thus not potentially obstruct the player’s sense of disbelief, they must be featured in mechanics which are not overly noticeable. This would include a player’s health bar, chances of critical hits and possibly the maximum health of enemies.

During the testing stages of the project, none of the participants after noticing a DDA mechanic tried to manipulate the mechanic in their favour and thus potentially break the game and negatively affect the results of this project. If a player were to notice and manipulate a DDA mechanic in a professional game they could potentially break the games cycle of actions and overall flow of the game. Due to this it is important to take note of how noticeable a DDA mechanic might be and how much of an impact it has on a game as a whole.

5.4 Gameplay/ Game Design and Approach:

The chosen game that was created for this project was a FPS game. This was due to the belief that both “Casual” and “Avid” gamers will likely have played a FPS before and the skills required to play and understand a FPS are interchangeable between different FPS games. This appears to have been the correct approach as each participant mentioned they had played a FPS before and were relatively satisfied with the controls and understandings of the game. Alongside this the different mechanics and features in a FPS game worked well in conjunction with the DDA features implemented into this project.

With the majority of FPS games the mechanics that are changed to make a game more difficult or easier are normally based on how much health the player and enemies have along with how much damage each of them can do. With this in mind, changing the enemies' health and damage output in this project suited very well with DDA and helped determine what would be considered "Easy" and "Hard" for the participants to beat.

However, there were some problems with the overall construction and reception of the applications from the participants. For example some participants noted that the game's crosshair affected their overall accuracy and made the firing inconsistent. This at its core is a gameplay fault of the project. If more time was spent on the appearance of the application such as User Interface (UI) design to make the player's crosshair more accurate to those featured in other FPS games along with a more visible health bar; it is believed these minor issues would be resolved. Alongside this some participants stated they would have liked a practice area at the beginning of the game to get accustomed to the controls which would have improved their overall experience of the project. This issue could have been avoided by allowing the participants a "practice run" of the application before actual testing would begin. Despite these minor issues the majority of participants stated they enjoyed taking part in the project and enjoyed playing each of the applications as a whole.

Although the questionnaire design served very well in gathering information on the preferred application and how much each participant enjoyed the applications, it was unfortunately not used as a means to gather which parts of the application participants enjoyed in particular. If any future work on this topic was to be conducted a focus group would be utilised to gather this information as without it a lot of speculation as to why a participant enjoyed an application is required.

6. Conclusion and Future Work:

6.1 Conclusion:

From this project many conclusions related to the use of Dynamic Difficulty Adjustment (DDA) in relation to a player's enjoyment were found.

Throughout the research stage of this project it was found that a need to keep DDA from obstructing a game's cycle of actions was a critical part of making a game more enjoyable, as taking away the challenge from a more experienced gamer would only lead to frustration. Alongside this DDA appears to have more of an effect towards a player's enjoyment when the DDA affects many aspects of a game as present in the "Grade System" application rather than individual changes like the ones used in the "RealTime" application. However, through the use of DDA a weaker player can progress through a game they would have otherwise given up on or resorted to lowering the difficulty which could potentially frustrate them. At the same time, this can give a greater challenge to more experienced players who could grow bored by the lack of challenge. From this project's implementation and testing stages almost all the players, both "Casual" and "Avid", showed an increased level of enjoyment towards DDA compared to games with Pre-Set difficulty levels; this is further testament to its inclusion in games as it is clear that its use does increase a player's enjoyment significantly. The shortcomings of this project were the minor faults in regard to game design which has had an effect on player's enjoyment, however overall the outcome of this project has been positive towards the use of DDA to effectively improve a player's enjoyment in games.

6.2 Future Work:

6.2.1 DDA Used With Pre-Set Difficulty Levels:

Although it has been demonstrated that DDA is more enjoyable than Pre-Set difficulty levels in this project, it is theorised that using DDA in conjunction with Pre-Set difficulty levels could be even more effective at improving a player enjoyment. By allowing a player to select their difficulty level in a game, this encourages them to set how difficult a challenge they wish to play on. This difficulty setting is then used as a measurement of how much the game should be manipulated by DDA. For example if a player selects “Easy” difficulty then the game could apply further help through the use of DDA if the player is doing badly; giving them more health or damage output. While potentially supplying greater challenges if they are doing well; make the enemies slightly tougher than they normally are on “Easy” difficulty settings. This in turn could help prepare them for the harder difficulty settings. Alongside this, if a player selected the “Hard” difficulty and were still finding the game relatively simple, DDA could be used to give an even greater challenge and potentially keep the player engaged and enjoying the game for longer.

6.2.2 Different Game Genres:

From this project DDA has only been implemented into a simple First Person Shooter (FPS) game. It was believed that a FPS was the simplest game for a variety of participants to understand and easily control. The use of DDA with a FPS has proven to be quite effective due to the number of factors in the game that can be changed such as the player’s aiming, player and enemy health and damage output. To further research the uses of DDA on a player’s enjoyment it would be invaluable to test its uses on other genres of games to see if a similar effect of enjoyment is replicated. Such genres could include platformer games where the player has to make precise jumps to progress through a level, DDA could be used to give the player more power ups and check points if they are doing poorly. This again would help weaker players without disrupting the games cycles of actions for better players.

However, in some genres of games DDA has shown to be more of a disadvantage than a potential helping hand. For example the rubber – band effect (Adams 2008) which is a term used when a player's car in a racing game is either far behind or far ahead of their opponents. If the player is far behind and losing a race, the AI controlled cars will slow down to allow the player to catch up. The opposite occurs if the player is far ahead, the AI controlled cars will very quickly catch up to the player acting as if the player's and AI controlled car are connected by a rubber band. If this type of DDA is used in a serious racing game simulator the player's immersion can be shattered and potentially leave them frustrated with the game by not accurately simulating a real race. However, if the type of racing game it is used in is more casual and party orientated then this use of DDA might still be useful as it could potentially allow younger and less experienced players a chance to win the game.

6.2.3 DDA Used With Level Design:

Although it has been demonstrated that DDA can be used effectively in regards to manipulating a player's behaviour and an NPC's behaviour, DDA could also be used effectively when applied to level design. Using a FPS as an example, DDA could be used to give players additional cover to hide behind if they were taking a lot of damage over a short period of time, this would help the player progress through the game without any major intrusion in the games mechanics. Alongside this if a player was doing well, an additional path through the game's level could be opened, this could be determined by how many enemies have been defeated, how much damage the player has taken and how quickly the player had performed specific tasks such as defeating enemies or clearing levels. This path could give the player a more difficult challenge and potentially greater rewards, which could encourage players to repeat the level to try and gain access to this extra path and thus increase the play time of the game and potentially increase the player's enjoyment.

7. Appendices:

7.1 Appendix A – Grade System Rules:

```
//rule 0 if player health is max, and kills enemy, increase grade
Rules[0] = ((CurrentHealth_ == MaxHealth_) && (aliveEnemy == false));

//rule 1 if player health is between 66% and 100% and kills enemy, increase grade
Rules[1] = (CurrentHealth_ >= (MaxHealth_ * 2/3) && aliveEnemy_ == false);

//rule 2 if player health is between 33% and 66% and kills enemy, keep grade the same
Rules[2] = (Currenthealth_ > (MaxHealth_ * 1/3) && (CurrentHealth_ < (MaxHealth_ *2/3)) &&
aliveEnemy_ == false);

//rule 3 if player health is lower than 33% and kills enemy, keep grade the same
Rules[3] = (CurrentHealth_ < (MaxHealth_ * 1/3) && aliveEnemy_ == false);

//rule 4 if the player kills less than 2/3 of enemies then decrease grade
Rules[4] = (numEnemies >= (total_enemies * 2/3) && alivePlayer_ == false);

//rule 5 if the kills 2/3 or more enemies and dies, keep the grade the same
Rules[5] = (numEnemies <= (total_enemies * 2/3) && alivePlayer_ == false);
```

7.2 Appendix B – Grade System – Reload/Accuracy Check Code:

FPS_Shooting Script:

```
//reload pistol
If(Input.GetKeyDown(KeyCode.R))
{
	reloadText.SetActive(false);
	pistolAmmo_ = 15;
	if(fire != 0 || bullet_script.enemyHit != 0)
	{
	//accuracy is based on number of times enemy is hit / number of times fired then rounded
		Accuracy = Mathf.Round((bullet_script.enemyHit / fire) * 100);
	}
	//store accuracy in final accuracy array
	DDA.AllAccuracy[numberOfReloads] = accuracy;
	numberOfReloads++;

	//reset
	Bullet_script.enemyHit = 0;
	Fire = 0;
}
```

DDA Script:

ResetVariables() Function:

```
//keep track of all the accuracy over the course of a round
For(int I =0; I < FPS_Shooting.numberOfReloads; I++)
{
	FinalAccuracy += AllAccuracy[i];
}
FinalAccuracy = Mathf.Round((FinalAccuracy / FPS_Shooting.numberOfReloads));

//when player dies or kills all enemies, if accuracy is lower than 50% turn on aim assist
If(FinalAccuracy < 50.0f)
{
	badAim = true;
}
Else
{
	badAim_ = false;
}
```

7.3 Appendix C – RealTime– Reload/Accuracy Check Code:

FPS_Shooting Script:

```
//reload pistol
If(Input.GetKeyDown(KeyCode.R))
{
	reloadText.SetActive(false);
	pistolAmmo_ = 15;
	if(fire != 0 || bullet_script.enemyHit != 0)
	{
	//accuracy is based on number of times enemy is hit / number of times fired then rounded
		Accuracy = Mathf.Round((bullet_script.enemyHit / fire) * 100);
	}
	//aim assist will turn on and off depending on player accuracy when they reload
	If(accuracy < 50.0f)
	{
		badAim_ = true;
	}
	Else // if accuracy is > 50.0f
	{
		badAim_ = false;
	}
	//reset
	Bullet_script.enemyHit = 0;
	Fire = 0;
}
```

CharacterControler Script:

```
If(FPS_Shooting.badAim == true)
{	// send out raycast from player camera
	If(Physics.Raycasy(ray, out hit, 100.0f))
	{	//if ray cast hits an enemy
		If(hit.transform.tag == “Enemy”)
		{
			CamMouseLook.sensitivity = 2.5f;
		}
		Else
		{
			CamMouseLook.sensitivity = 5.0f;
		}
	}
}
Else
{
	CamMouseLook.sensitivity = 5.0f;
}
```

7.4 Appendix D – Questionnaire:

Date:	Name:

E-mail:

Disclaimer: you will not be asked for any personal information; you are free to leave at any time if you do not want to continue the experiment. If you wish to know the results of the experiment after all research has been completed feel free to leave your e-mail address and you will be send the results.

Purpose: To test whether users enjoy a game that adjusts its difficulty settings to better suit their skill level (DDA, Dynamic Difficulty Adjustment) or a game with pre-set difficulty levels such as easy/normal/ hard. The questions below ask for the opinions of the tester on the two applications they have just played; the data gathered will be used to find a connection between DDA and a player's enjoyment of a game. It should be noted that the focus of this project is the difficulty level in this application and not other factors such as art design or sound design.

How often do you play video games? (Please Tick)

0 to 5 hours a week. ☐

5 to 10 hours a week. ☐

Over 10 hours a week. ☐

What difficulty setting do you normally play on when playing games? (Please Circle)

Easy/ Normal / Difficult / Other (Please Specify)

Does the idea of a game changing its difficulty to suit your skill level appeal to you? (Please Tick)

Yes ☐

No ☐

If any, which application did you prefer 1, 2, 3 or can't tell? (Please Tick)

1. ☐

2. ☐

3. ☐

Can't Tell. ☐

On a scale of one to seven, how much did you enjoy application 1? (Please Circle)

1------2------3------4------5------6------7

On a scale of one to seven, how much did you enjoy application 2? (Please Circle)

1------2------3------4------5------6------7

On a scale of one to seven, how much did you enjoy application 3? (Please Circle)

1------2------3------4------5------6------7

On a scale of one to seven, how challenging did you find application 1? (Please Circle)

1------2------3------4------5------6------7

On a scale of one to seven, how challenging did you find application 2? (Please Circle)

1------2------3------4------5------6------7

On a scale of one to seven, how challenging did you find application 3? (Please Circle)

1------2------3------4------5------6------7

Which application(s) do you think had Dynamic Difficulty Adjustment? (Please Tick all that apply)

1. ☐
2. ☐
3. ☐

Which parts of the application do you think were affected by Dynamic Difficulty Adjustment?

Please specify:

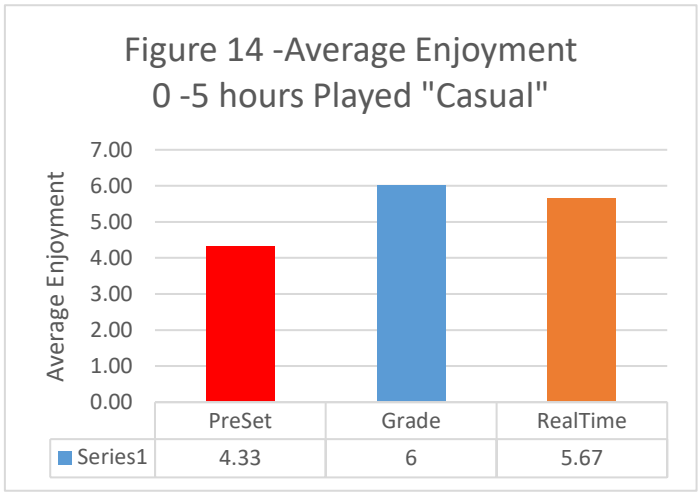
If you found any issue with the applications, please specify below:

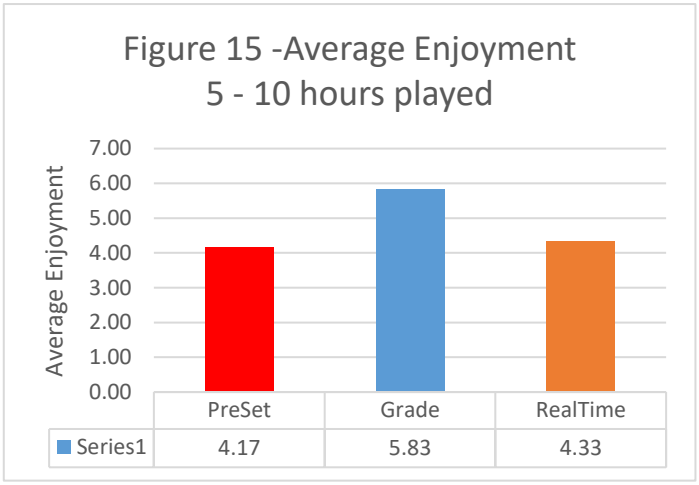
Any other Comments:

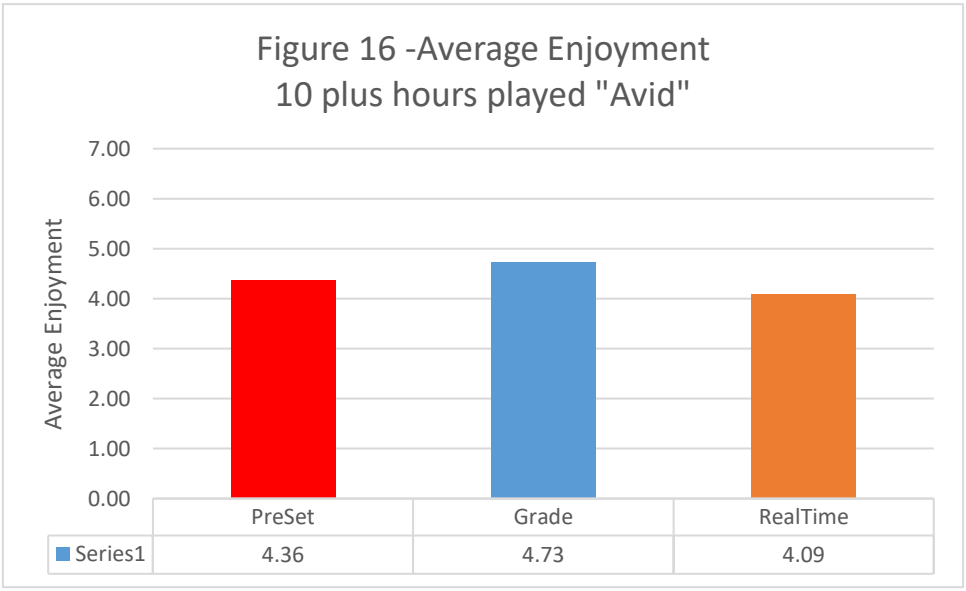
Please confirm that you acknowledge the information you put on this form will be used it statistics and information that will be used in a research paper and you are willing to allow this information to be used.

Signature:

7.5 Appendix E – Average Enjoyment for Casual – Normal - Avid:



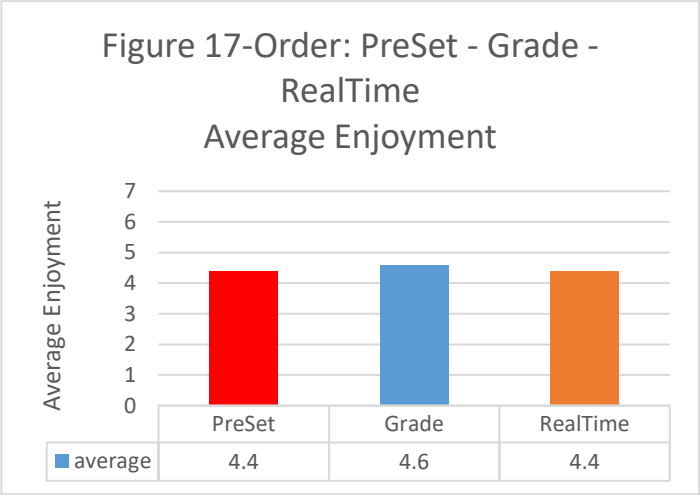


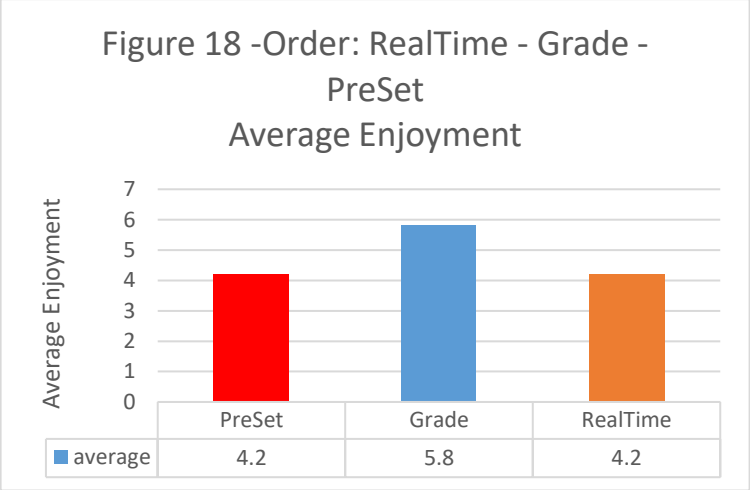


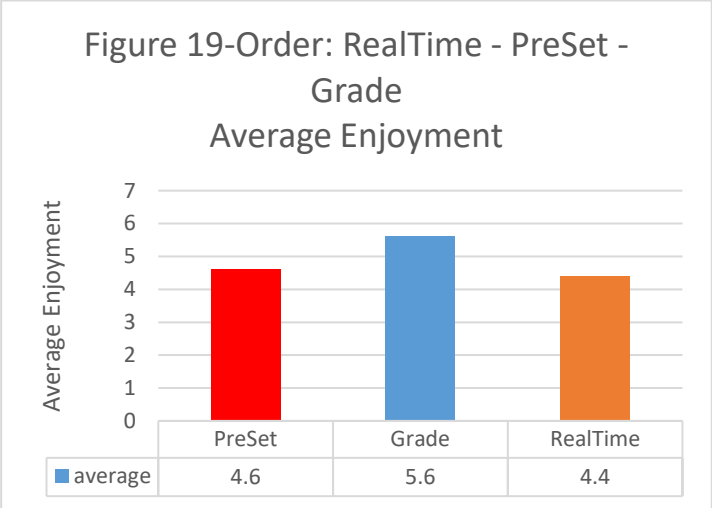
7.6 Appendix F – Average Enjoyment of Applications:

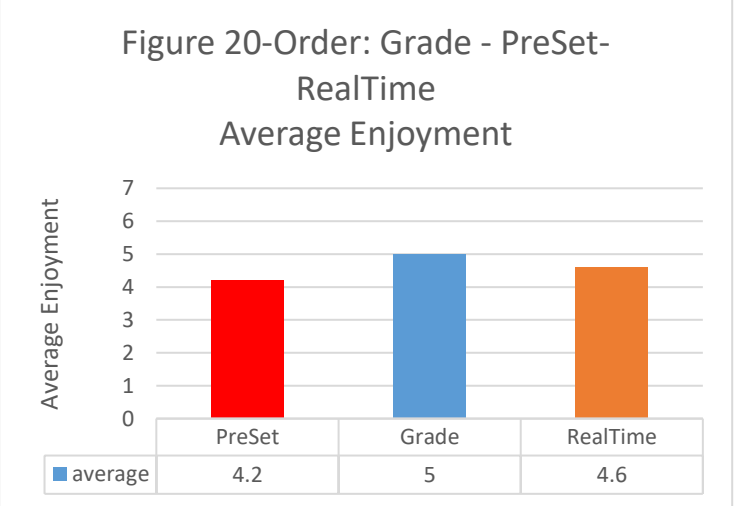
"On a Scale of one to seven, how much did you enjoy application 1/2/3?

1------2------3------4------4------5------6------7"







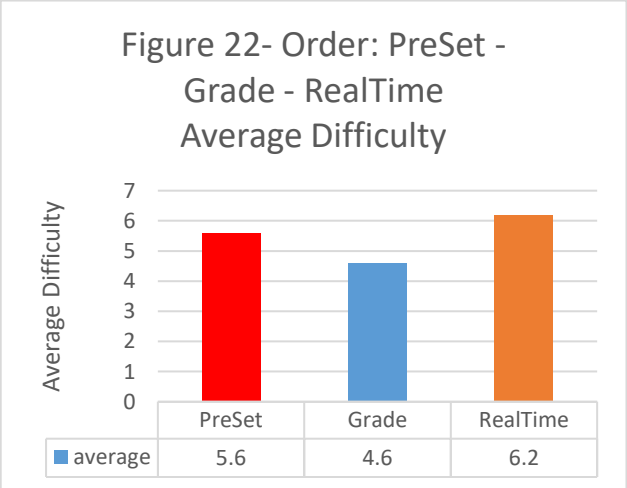


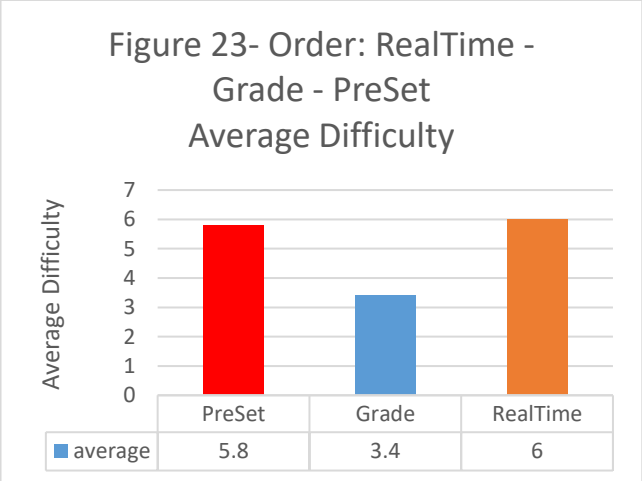
From these results there is no clear difference in enjoyment based on which order a participant played each application.

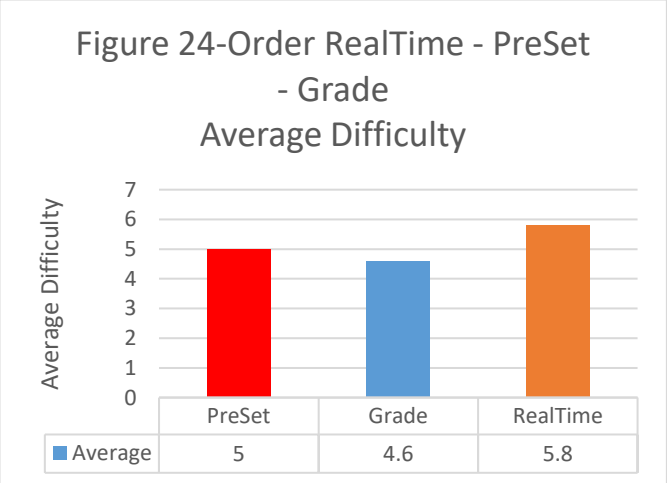
7.8 Appendix G – Average Difficulty of Applications:

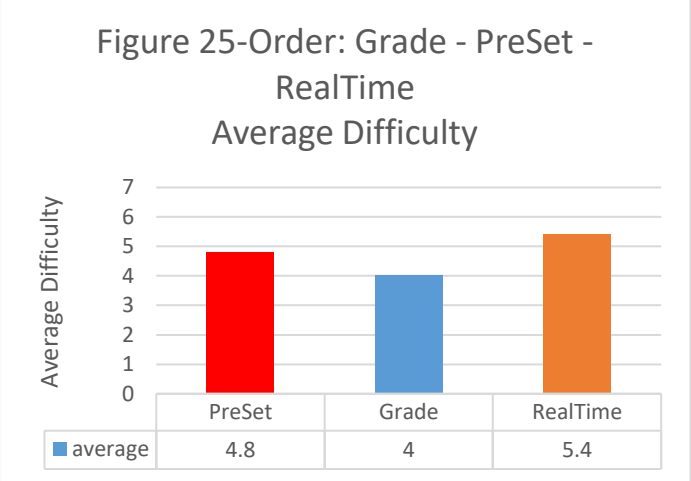
"On a scale of one to seven, how challenging did you find application 1/2/3?

1------2------3------4------5------6------7"



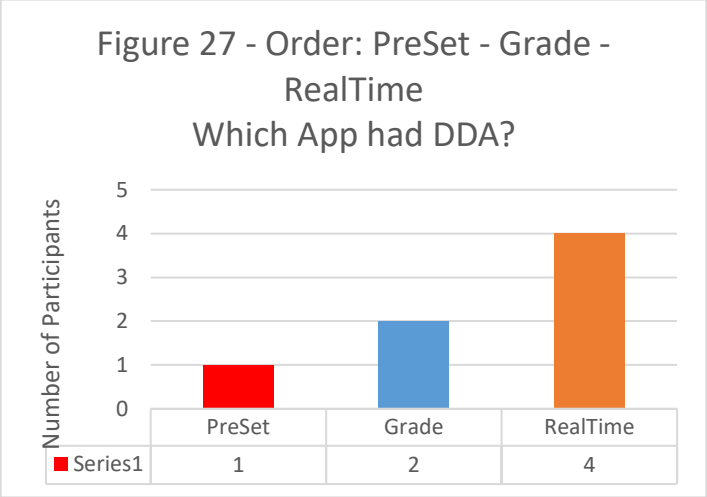


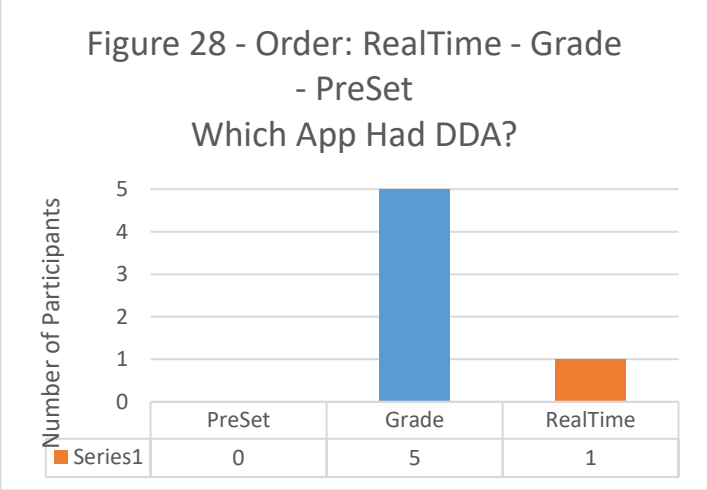


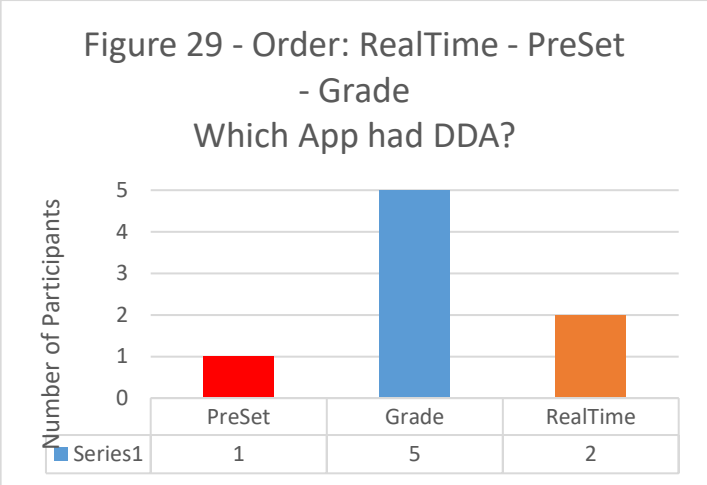


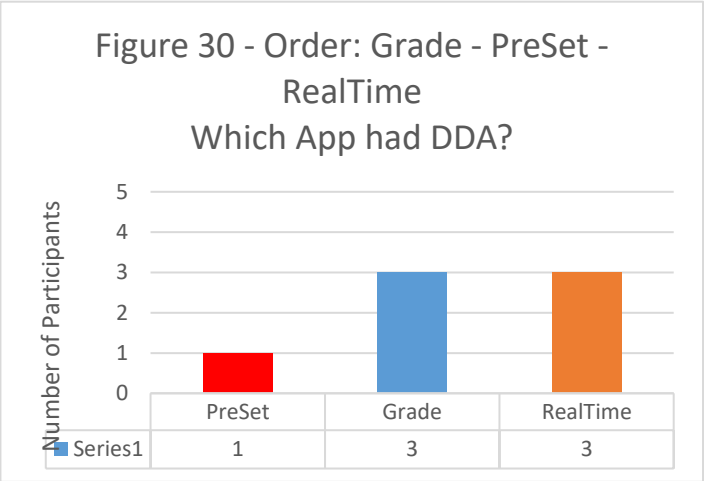
7.9 Appendix H – Which App Had DDA:

"Which application(s) do you think had Dynamic Difficulty Adjustment?"









From these results, regardless of the order the applications were played in, DDA mechanics were most commonly noticed in the "Grade System" application, except in order: PreSet – Grade- RealTime (Figure 27).

8. References:

Adams, E. 2008. *The Designer's Notebook: Difficulty Modes and Dynamic Difficulty Adjustment*. Available from: http://www.gamasutra.com/view/feature/132061/the_designers_notebook_.php?print=1. [Accessed 11 April 2017].

Andrews. 2008. TrustedReviews. *Dead Space review.* Available from: http://www.trustedreviews.com/Dead-Space-review [Accessed 7 February 2017].

All things Andy Gavin. 2016. Available from: http://all-things-andy-gavin.com/2011/02/07/making-crash-bandicoot-part-6/ [Accessed 20 October 2016].

Booth, M. 2009. *The AI Systems of Left 4 Dead*. In: Artificial Intelligence and Interactive Digital Entertainment Conference, October 14-16, Standford, California. Available from: http://www.valvesoftware.com/publications/2009/ai_systems_of_l4d_mike_booth.pdf [Accessed 7 February 2017].

Booth, M. 2009. *The AI Systems of Left 4 Dead.* [online image]. Available from: http://www.valvesoftware.com/publications/2009/ai_systems_of_l4d_mike_booth.pdf. [Accessed 7 April 2017].

Brown, M. 2015. *What Capcom Didn't Tell You About Resident Evil 4*. [online image]. Available from: https://www.youtube.com/watch?v=zFv6KAdQ5SE&t=8s. [Accessed 7 April 2017].

Crash Bandicoot. 1996. [Computer Game]. PlayStation. Naughty Dog.

GameSpot. 2017. *2016's Best-Selling Games in the US Revealed.* Available from: http://www.gamespot.com/articles/2016s-best-selling-games-in-the-us-revealed/1100-6447090/ [Accessed 16 March 2017].

Hunicke, R. 2005. *The Case for Dynamic Difficulty Adjustment in Games*. In: ACM SIGCHI International Conference on Advances in computer entertainment technology, June 15 -17, Valencia, Spain. New York: AMC Press. pp. 429 – 433.

Lach E. 2015. *A Quick Method for Dynamic Difficulty Adjustment of a Computer Player in Computer Games.* In: Rutkowski L., Korytkowski M., Scherer R., Tadeusiewicz R., Zadeh L., Zurada J. (eds) Artificial Intelligence and Soft Computing. Part II. Lecture Notes in Computer Science, vol 9120. Springer, Cham. pp. 669 – 678.

Left 4 Dead. 2008. [Computer Game]. Xbox 360, Microsoft Windows, OS X. Valve Corporation.

Mark Brown. 2015. *What Campcom Didn't Tell You About Resident Evil 4 | Game Maker's ToolKit.* Available from: https://www.youtube.com/watch?v=zFv6KAdQ5SE&t=1s [Accessed 01 February 2017].

McLeod, S. 2008. SimplyPsychology. *Likert Scale*: pp.1. Available from: http://www.simplypsychology.org/likert-scale.html [Accessed 7 February 2017].

Ocampo. 2009. *Left 4 Dead (Game of the Year Edition) Review.* Avaliable from: http://uk.ign.com/articles/2009/05/21/left-4-dead-game-of-the-year-edition-review [Accessed 19 April 2017].

Resident Evil 4. 2005. [Computer Game]. GameCube, PlayStation 2, Microsoft Windows, Wii, Zeebo, iOS, PlayStation 3, Xbox 360, Android, PlayStation 4, Xbox One. Capcom.

Unity Technologies. 2005. *Unity Engine*. [software]. Version 5.3.5f1. Available from: https://unity3d.com/get-unity/download/archive

Williams, M. 2013. *Dead Space 2 Difficulty Select.* [online image]. Available from: https://gamedevelopment.tutsplus.com/articles/new-to-tuts-game-development-start-here--gamedev-4296. [Accessed 7 April 2017].