Dynamic Difficulty Adjustment: Using the Wizard of Oz Method to Investigate Potential Improvements through Biofeedback

by

Stéphane Thomas Horne

A thesis submitted in partial fulfillment

of the requirements for the degree of

Master of Science (MSc) in Computational Sciences

The Faculty of Graduate Studies

Laurentian University

Sudbury, Ontario, Canada

THESIS DEFENCE COMMITTEE/COMITÉ DE SOUTENANCE DE THÈSE
Laurentian Université/Université Laurentienne
Faculty of Graduate Studies/Faculté des études supérieures

Title of Thesis Titre de la thèse	Dynamic Difficulty Adjustment: Using the Wizard of Oz Method to Investigate Potential Improvements through Biofeedback
Name of Candidate Nom du candidat	Horne, Stephane
Degree Diplôme	Master of Science
Department/Program Département/Programme	Computational Sciences
Date of Defence Date de la soutenance	December 19, 2018

APPROVED/APPROUVÉ

Thesis Examiners/Examinateurs de thèse:

Dr. Ratvinder Grewal
(Supervisor/Directeur(trice) de thèse)

Dr. Run-Min Zhou
(Committee member/Membre du comité)

Dr. Kalpdrum Passi
(Committee member/Membre du comité)

Dr. Pradeep Atrey
(External Examiner/Examinateur externe)

Approved for the Faculty of Graduate Studies
Approuvé pour la Faculté des études supérieures
Dr. David Lesbarrères
Monsieur David Lesbarrères
Dean, Faculty of Graduate Studies
Doyen, Faculté des études supérieures

ACCESSIBILITY CLAUSE AND PERMISSION TO USE

Abstract

Engagement is a key factor to consider when developing video games. Flow theory—one of the required components of engagement—dictates that in order to achieve flow, there must be an adequate balance between the player's skill level and the challenge they're faced with. Consequently, game developers have created several AI systems to tailor a game's difficulty to players' skill levels, such as Dynamic Difficulty Adjustment (DDA). This system uses player performance as an indicator of their emotional state. However, this is not the most accurate measure. In this work, the Wizard of Oz method was used to determine if DDAs could elicit higher levels of engagement if given access to information relating to players' emotional states, such as their facial expressions and body language. Results showed that there were no significant differences in engagement between the DDA and the Wizard.

Keywords

Engagement, Flow, Dynamic Difficulty Adjustment, Wizard of Oz

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Dr. Ratvinder Grewal. Not only were you instrumental in the completion of this thesis, with your constant support, patience and dedication to my success, but you were also a major factor in my transition from psychology into computer science. This work would not have been possible without your guidance, and I will forever be grateful for everything you've done for me. To my committee members, Dr. Run-Min Zhou and Dr. Kalpdrum Passi, my sincerest thanks. I would also like to express my gratitude to David Vallieres for allowing me to use his game as the foundation of this thesis. Your generosity saved me months of hard work, and for that, I thank you. Furthermore, I would like to extend my thanks to Amy and Rachelle. As full-time students with part-time jobs, I know how limited your spare time must have been. Yet, you were both willing to sacrifice that spare time to make yourselves available in helping me with my experiment. I can't thank you ladies enough. To all the friends I have made along the way, thank you for your words of encouragement. A special mention to my friend Stefan, for your relentless support. I can't count the number of times you set your own work aside to help me with my own issues; from design quirks to debugging, you were always happy to lend me a helping hand. Thank you so much. Finally, I would like to thank my brother and sister, as well as my mom and dad. You've made so many sacrifices for me, constantly putting my best interest before your own. You've always been there for me, you've always encouraged me to keep pushing forward, and most importantly, you've always believed in me. I would not be the man I am today, nor would I have been able to achieve this milestone without you. Thank you, and I love you guys.

Table of Contents

List of Figures

List of Tables

List of Appendices

1 Introduction

From arcades in the late 70s and early 80s to the high-powered consoles of today—and everything in between—video games have evolved drastically over the past few decades. Not only can we play games in the comfort of our own homes, but we can also play a wide variety of mobile games; this includes smartphones, or portable gaming systems such as the PlayStation Vita®, or the Nintendo Switch®. No matter the medium, video games have become omnipresent; as demonstrated by the 62% of people surveyed in the U.K. in 2017 who reported playing video games on either their computers, dedicated gaming consoles, smartphones, tablets, or handheld consoles [23]. As of 2018, 162 million Americans, representing nearly half of the American population, own a dedicated gaming console [61], further illustrating the prominence of video games in our daily lives.

Unsurprisingly, this has led to an increase in time spent playing video games. According to the Entertainment Software Association (2018), 60% of Americans surveyed play video games daily, while the average video gamer in the US spends 6.44 hours a week playing video games [62]. Another study revealed that in 2018, 43.8% of Canadians surveyed spend at least three hours or more a week playing video games, on average. This same study also showed that only 19.7% of Canadians reported that they never play video games, a significant decrease from the nearly 28% recorded in both 2016 and 2017 [11].

It is clear that video games have become an increasingly popular form of entertainment. Despite this, not all video gamers play the same types of games. A 2018 survey of over 3,000 video game players conducted in six countries has revealed that gamers spend significantly more time playing casual single-player games and single-player RPGs than multiplayer style games, such as massive multiplayer online games and local multiplayer games [62]. Tracy [65] surveyed 500 gamers in the U.S., which revealed that 67% of people preferred to play single player games over multiplayer games, further illustrating the preference for single player games.

While both casual and experienced players may play games for entertainment, the notion of entertainment—and what constitute an entertaining gaming experience—is different for both types of players. Experienced players, for one, tend to play games in order to be challenged. This typically results in a sense of gratification when they've overcome certain obstacles or when they've completed the game. It is said that these players "play to win", and as a result, they are continuously looking to be challenged in order to develop their skill set [1]. Casual players, on the other hand, play games merely as a form of entertainment. Winning is not their primary focus; they prefer to play games as a way to "pass the time", as it were. As a result, these players tend to play games at a difficulty setting that is well within their skill level, because they would rather complete the game without the need to invest too much time and effort.

A preference for single-player games—and the difference in motives between casual and experienced gamers regarding why they play games—present game developers with an intriguing challenge; designing a game that can be both challenging for the experienced players while still being easy enough for casual gamers so that minimal skills are required. This is crucial, as player engagement—discussed in depth in section 2.1—hinges on several factors, one

of which is an appropriate balance between challenge and skill. Clarke & Duimering (2006) have demonstrated that a common factor that negatively impacts a player's level of enjoyment—amongst experienced video gamers—is that the game was too easy, while casual game players cited that games being too difficult had the same effect. As a result, experienced game players are likely to become bored of the game, while casual players are expected to become frustrated with the game. In both instances, player engagement will suffer; thus, leading to game abandonment. This highlights the importance of game balancing; the process of tailoring a game's level of difficulty to an individual's skill level [7].

One of the most primitive methods used for difficulty adjustment comes in the form of static difficulty settings [3, 38]. This involves having the player pick a difficulty setting that they believe is right for them—typically ranging between easy, medium, and hard—before starting the game [3, 54, 74]. However, as the player invests more time, their skill level rises accordingly [3, 38]. Unfortunately, static difficulty settings are just that; static. They are not flexible, and therefore do not respond to an increase in player performance [38, 74]. As such, it is likely that a mismatch between the player's skill level and the level of challenge in the game will ensue [3], thus demonstrating the need for alternate methods of game balancing.

Many techniques have been used to circumvent this issue by tailoring a game's level of difficulty to the player's skill level, such as: procedural content generation, rubber band AIs, and Dynamic Difficulty Adjustment (DDA). In brief, procedural content generation is a method through which algorithms are used to automatically create content for the game [64]. This is typically done with limited or indirect user input [57]. As such, the player's performance has an influence on the content that is generated, thus resulting in more adequate levels of challenge; in

turn maximizing player enjoyment [57]. This technique has been put into practice in commercial games such as Diablo (Blizzard, 1996) and Borderlands (Gearbox, 2009).

Rubber band AIs have been used in racing games such as Mario Kart 8 Deluxe (Nintendo, 2017) and Need for Speed Rivals (2013, Ghost). Simply put, all participants—including the NPCs—are to be thought of as contained within a band. If the human player pulls on either end of the band—meaning if they achieve a sizeable lead or have fallen behind the NPCs—then the AI modifies the parameters of the game accordingly; NPCs are made faster in the event of the former, slower in the event of the latter [37, 38, 46]. As a result, players will not experience frustration with the game, as they are given a chance to catch up if they're in last place. Boredom is avoided as well, as players will always be challenged for first place.

The final technique mentioned above is DDA. While this is discussed in depth in section 2.5, DDA is a method through which game components are modified based on player performance [5, 24, 59]. There are several different ways this can be achieved. One such method is to modify elements in the game environment; if the player is doing poorly, the DDA can increase the number of health pickups and weapons available throughout the level, or vice-versa [38]. A balance between challenge and skill can also be established by the DDA through manipulation of the NPCs' parameters; the DDA could increase the health of all enemies and make them faster—and subsequently more difficult to kill—if the player's skill level exceeds the current level of challenge [3]. Regardless of the method through which the DDA alters the game's level of difficulty, it has been established that they are an effective means for maintaining elevated levels of engagement in video game players [22, 68, 72].

In 2016, David Vallieres—an undergraduate student from Laurentian University—implemented a DDA into a game he designed for his thesis. Following the rules outlined in section 2.5, his DDA made modifications to the NPCs' parameters based on individual participants' performance in order to achieve optimal levels of engagement. His findings supported previous work, showing that DDAs are an effective method for maximizing player engagement [66]. While player performance is typically the indicator used [73] to assess a player's level of frustration/boredom, the authors of this work postulate that it may not be the most accurate measure available. As such, this work attempts to determine if DDAs can be improved upon if given information pertaining to the player's facial expressions and body language. This would allow the DDA to more accurately determine the emotional state of the player, which would subsequently lead to more accurate changes in game difficulty. As it is not yet known if the incorporation of a player's body language and facial expressions into a DDA will improve its performance, not to mention the time-consuming endeavor of implementing such a system, the Wizard of Oz (WoZ) method—discussed in depth in section 2.6—was used to simulate a DDA with these features.

2 Literature Review

2.1 Engagement

While video games have been studied at length over the past few decades, only recently (2007) have researchers focused their attention on the interaction between the player and the

game itself [13]. This is curious, as it has already been acknowledged that video games must have the ability to draw people in for them to be successful [14]. Nonetheless, researchers are still investigating why video games are such a popular form of entertainment.

According to Boyle, Connolly, Hainey and Boyle (2012), theoretical frameworks propose two perspectives that offer answers to that very question: the subjective experiences of the players while playing, as well as their motives for playing video games; this is a notion that is also supported by [53]. As discussed in the introduction, not all players' motives for playing video games are the same. While it's true that games constitute a leisurely activity for both casual and experienced gamers, their motives differ. Gamers that identify as being more casual simply play the games as form of entertainment; they do not have the desire to compete with/vanquish their opponents. This is a trait found within experienced game players.

Subjective player experiences, for their part, revolve around the notion of engagement. Engagement is a term that has different meanings for different disciplines. In the field of human-computer interaction, however, several authors have affirmed that a formal definition of engagement has yet to be agreed upon [9, 49, 50, 51, 53]. One such definition has been proposed by O'Brien & Toms (2018); they have defined engagement as the depth of a person's investment when interacting with a digital system. This definition results from several years' worth of research pertaining to engagement. In 2008, they took it upon themselves to conduct research on the matter, with the primary goal of providing a conceptual definition for engagement.

They first began with a review of the literature published at that time, which led them to conclude that engagement is a complicated principle that can be explained thanks to fragments of

several previously established frameworks. One such framework, known as flow theory, will be elaborated in section 2.2. The remaining frameworks examined shared several similarities, such as the elements that play a role in engagement. These elements include: aesthetics, affect, focused attention, challenge, control, feedback, interest, motivation, novelty and perceived time. Although they may all have an effect on engagement, the elements listed above play their part at different points in time; while some may attract the user to begin an interaction with a device, others signal to the user that it's time to end their interaction. This conclusion allowed the authors to devise a model for the process of engagement, which includes four phases: point of engagement, period of sustained engagement, disengagement and potential reengagement.

The point of engagement occurs when users invest themselves in the interaction. Elements essential to this phase include: aesthetics, novelty, interest, and motivation. In other words, users will become engaged during an interaction if they're motivated to use the medium they're interacting with. They might also become engaged if they're appealed by the medium's aesthetics, if the medium is new to the user or if they have a certain level of personal interest regarding the medium. Regardless, there are several factors which can initiate engagement while interacting with a digital device or medium. However, those factors alone may not be sufficient to keep users engaged for a prolonged period of time. As a result, the next phase in the process of engagement addresses this factor.

According to their proposed process of engagement, O'Brien & Toms (2008) highlight the following elements that are essential in order to keep users in what they've defined as the period of sustained engagement: aesthetics, attention, control, novelty, feedback, challenge, interest and positive affect, amongst others. Based on the elements listed above, we can conclude

that users will continue to be engaged by their interaction with the medium under several circumstances. First, engagement is likely to persist if the graphics keep the user's attention or evoke a sense of realism. Another case where engagement would persist is if their interaction creates a pleasant experience for them. Next, if their interaction with the medium gives them the sense that they are in control, engagement will likely continue. Finally, engagement can persevere if the user's actions give them the sense that they are progressing towards their goal, if the amount of effort they need to put into the interaction is moderate, or if the interaction creates a sense of joy resulting from a new experience found within.

While user engagement may be at its highest point during the period of sustained engagement, it is unlikely that users will continue an interaction indefinitely. As such, the next phase in the process of engagement is, unsurprisingly, disengagement. Disengagement can occur for a multitude of reasons. However, O'Brien & Toms have highlighted the following elements as the most probable causes for this transition from sustained engagement: usability, challenge, positive and negative affect, and perceived time. This is to say that users will become disengaged from the interaction if they feel as though they cannot interact with certain features, or if they feel as though they have to put in too much effort in order to arrive at their goal. Disengagement can also be the results of users experiencing any sort of frustration, boredom, or uncertainty with the medium. Finally, users are likely to become disengaged with the medium if the interaction is too time consuming, or if the they have accomplished their goals, therefore eliminating their need to continue the interaction.

The final phase in the process of engagement is known as reengagement. According to the authors, this is simply a phase that represents the likelihood that users will engage with the

device again. They quote elements such as ease of use, past successes, and positive experience as indicators that users would choose to interact with the device in the future. Factors that reflect the elements mentioned above include: control, challenge, feedback, positive affect, and novelty. In other words, users are more likely to interact with the medium anew if they felt like they were in control of their experience. Furthermore, if they felt like the amount of effort necessary to arrive at their goal was adequate, then it's possible the users would once again engage with the device. Finally, in the event that the device provided enough feedback to inform the user about their progress towards achieve their goal, and if their overall sense upon completing the interaction was satisfying, then it's likely they would have subsequent interactions with the device.

While engagement is an important facet to consider during the development of video games, there are other elements which cannot be ignored. Flow, which is discussed in the following section, is an equally important feature; this ensures that players do not become frustrated with the game, all the while safeguarding against boredom.

2.2 Flow

Earlier in this section, it was noted that flow theory was one of several theoretical frameworks consulted when O'Brien & Toms (2008) developed their model for the process of engagement. Here, flow theory is elaborated upon, as it is an important facet relating to engagement. As first described by Csikszentmihalyi [17]—which was later refined [18, 47]—

flow is the state in which "attentional resources are fully invested in the task at hand, so that objects beyond the immediate interaction generally fail to enter awareness". It occurs when a person is void of any threats or distractions that require them to split their focus; instead, they can devote all of their attention to the task at hand [53]. This is commonly referred to as being "in the zone" [53]. Flow theory, similar to engagement, hinges on a variety of elements. In this case, there are a total of six: an appropriate balance between challenge and skill, a clear set of goals to be achieved by that individual, feedback relating to the progress towards those goals, the notion that we are in control of the situation, the loss of self-consciousness, and the distortion of time [47]. It should be noted that these elements rely upon one-another [53]. The figure below illustrates this concept:

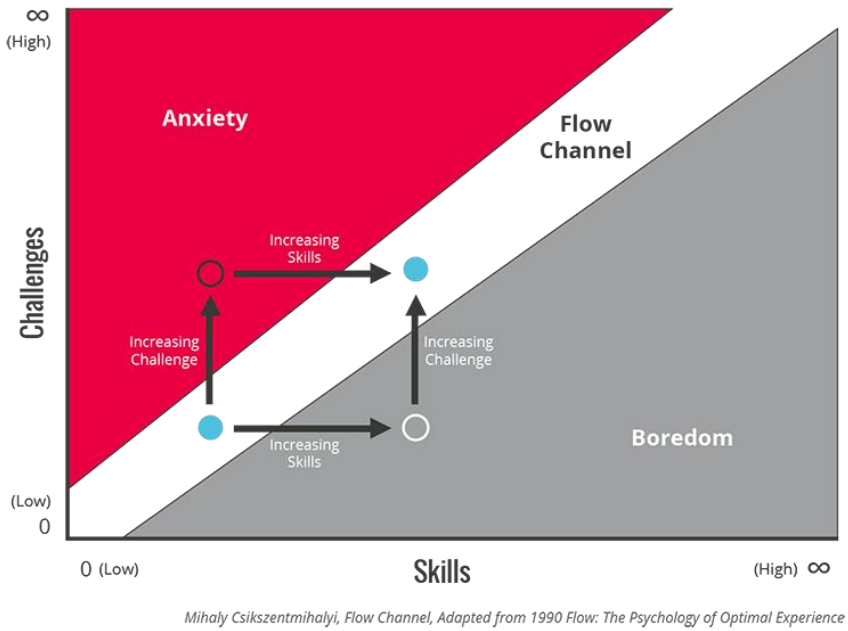


Figure 1: Visual representation of the Flow concept

As made apparent in the figure, an individual must find themselves in the flow channel in order to experience flow. Although the individual may perform very well in instances where

their skill level greatly exceeds the level of challenge, they will also become bored. However, should the level of challenge exceed the individual's skill level, their performance will suffer, resulting in feelings of frustration. This graph clearly illustrates the need for a balance between challenge and skill; one of the eight elements required to achieve flow. However, the seven other elements aren't as apparent. In order to better explain this principle, as well as to highlight the seven remaining elements required for flow, consider that the individual is playing a game of Mario Kart.

One of the elements required to achieve flow is the distortion of time. This is similar to the adverbial phrase "Time flies when you're having fun". Previous research [49, 51, 53] has established that it is common for video-game players to lose track of time. Therefore, if the individual is playing Mario Kart, they are likely to experience this phenomenon. Another required element is that the individual must have clear goals. In the case of playing Mario Kart, the goal is evident: to finish the race as quickly as possible. The individual's progress towards this goal is also one of the required elements of flow, and it is made obvious through the display of not only the current lap—of which there are typically three—but also their current position relative to the other racers. Because the outcome of the race is entirely dependent on the individual's performance, they know that they are in control of the situation; a fifth requirement for experiencing flow. For beginners, a typical distraction encountered is the ability to play without thinking about which buttons accomplish particular functions. As a result, there is an inability to experience the loss of self-consciousness; the final component required to experience flow. Csikszentmihalyi [47] explains that during the loss of self-consciousness "[there is a lack of] the required attentional resources, [and as such], the self-reflective processes that often

intrude into awareness and cause attention to be diverted from what needs to be done are silenced". Therefore, if the individual is thinking about the button mapping, they have not devoted all of their attentional resources into the race itself; thus demonstrating that they have not experienced a loss of self-consciousness. With both engagement and flow having been discussed, this leads to the scale that was used to measure engagement during this study.

2.3 User Engagement Scale

As mentioned above, authors in this field have differing opinions on what engagement entails as it pertains to interactions with digital mediums. However, for the purpose of this thesis, engagement is regarded as it has been defined above due to the high citation frequency of the article from which it was retrieved [2, 15, 32, 36], validating it as an acceptable definition of engagement. Following their work on the process of engagement, O'Brien and Toms (2009) developed a questionnaire for the purpose of quantifying engagement: The User Engagement Scale (UES). This questionnaire was composed of questions aimed to address some of the elements cited above that they believe influence engagement, which were then grouped to form the six following subscales: aesthetic appeal, focused attention, perceived usability, endurability, novelty, and felt involvement.

The questions used for the focused attention were designed to gauge an individual's sense of being absorbed by the interaction and whether or not they lost track of time. The perceived usability subscale included questions designed to address the negative affect experienced as a

result of an interaction, as well as the individual's perceived degree of control and effort expanded throughout the interaction. Questions are included to quantify an individual's thoughts regarding the visual appeal of the interface, as well as their interpretation of the attractiveness of the interface, hence establishing the aesthetic appeal subscale, while the felt involvement subscale included questions revolving around the individual's sense of being "drawn in" and being entertained. In order to determine an individual's sense of novelty, the authors included questions relating to the former's curiosity and interest throughout the interaction. Finally, the endurability subscale was composed of questions designed to determine an individual's feelings regarding the overall success of the interaction and their willingness to interact with—or to recommend—the interface to others in the future.

After its inception, Wiebe, Lamb, Hardy & Sharek (2014) examined the UES to determine its feasibility as a tool for measuring engagement during video game play. While they suggested the questionnaire could benefit from reducing the number of subscales, their results demonstrated that it was indeed an appropriate tool for measuring engagement while playing video games. O'Brien and Toms (2016) then conducted a synthesis of over forty published works that have used it to investigate user engagement in a range of digital domains—such as information search, online news, online video, education, consumer applications, haptic technologies, social networking systems and video games—thus demonstrating its effectiveness as a tool for measuring engagement. O'Brien, Cairns & Hall (2018) have since revised the UES, grouping the endurability, novelty and felt involvement subscales into one, which they've named the reward subscale. The new version of the scale has been named the revised User Engagement

Scale (UESz), and it has been found to be an effective measure of engagement [49, 53]. As such, this questionnaire was used to gauge player engagement over the course of this study.

2.4 AI in Games: Finite State Machines

Several models can be used when designing AIs that control the NPCs in a video game. For games that involve elaborate behaviours, more complex models are required, such as fuzzy logic and behaviour trees [41, 44, 71]. Fuzzy logic determines the behaviour of a character based on their degrees of membership to certain traits. For example, a character can be a member of several different traits, such as “Run”, “Jog”, and “Walk”. Based on their degree of membership to each of these traits, as well as what is called a defuzzification method, the AI would be able to determine the speed at which the character moves around [56, 71]. This is not to say that the speed associated with one of the three traits would be selected, but rather that a speed somewhere in the range between two of the traits would be computed [45]. A behaviour tree, on the other hand, represents a high level behaviour in the form of a tree [52, 71]. In order for this behaviour to be carried out by the NPCs, the actions comprised within the children of the root must be successfully accomplished. There are different methods that dictate how the actions in the child nodes are carried out, however they will not be covered in this work as they do not fall within the scope of this experiment.

While these two models are often used for complex behaviours, there are others that exist that can be used when designing more simplistic NPCs. One such model, which was used to

design the NPCs in Dunjions, is known as a finite state machine. One of the most popular methods for developing AIs, this method was used to create non-playable characters within games like Pac-Man as early as 1979 [45], and continued to be a popular method into the mid-2000s [71] with games such as Doom and Quake [40].

Finite state machines are expert systems which can be represented using connected graphs [40, 44]. The nodes in the graph represent the potential states for the AI, while the edges represent the transition from one state to another. These transitions are described by a condition that needs to be fulfilled in order to move from one state to another [12]. A series of actions are comprised within each state and subsequently performed by the AI [71]. In the Figure below, we can see the example used by Yannakakis (2018) to summarize how a FSM would operate in Ms. Pac-Man if she was controlled by an AI rather than by a human:

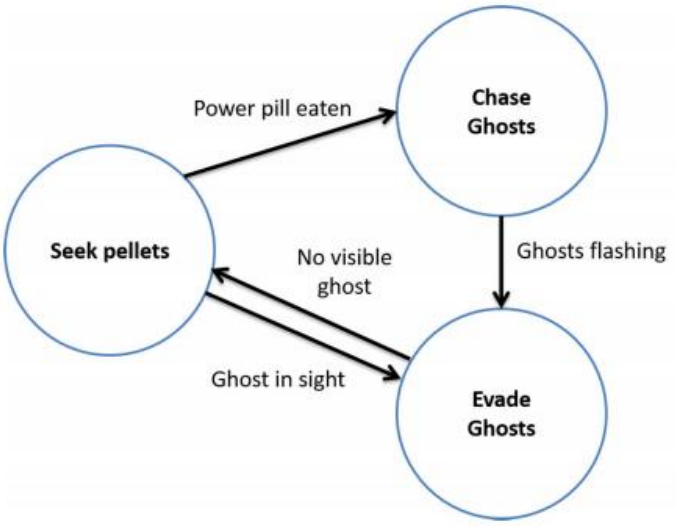


Figure 2: Directed graph illustrating a FSM for Ms. Pac-Man

The three nodes of this directed graph demonstrate the three potential states of the main character: either she is chasing the ghosts, she is seeking pellets, or she is evading the ghosts. The conditions that dictate the transitions from one state to another are listed along the edges connecting two nodes. Therefore, Ms. Pac-Man can only chase the ghosts if she is currently seeking pellets and consumes a power pill. She will evade the ghosts from either of the two other states; once they become visible if she is seeking pellets, or once they begin to flash if she is chasing them. Lastly, she can only seek pellets if she is evading the ghosts, and this occurs when there are no ghosts in sight. The NPCs in Dunjions are designed in a similar fashion. They have two states: wander and attack. If they cannot see the player, they wander around the map. Once the player is in their line of sight, they switch into the attack state. They'll remain in that state until they've been killed, until they've killed the player, or until the player flees and is no longer in their line of sight, at which time they switch back into the wander state.

While FSMs have several limitations, the most notable includes their inability to deal with more complex behaviours. Furthermore, as games that use FSMs grow in complexity, the number of states required grows exponentially, making them much more complex and difficult to manage [41, 42, 44]. They've also been criticized for their lack of modularity and reusability [44, 52, 71]. However, a FSM was the appropriate model for the NPCs found in Dunjions, as they are not required to perform complex behaviours. Also, issues relating to a FSM's modularity, reusability, and increase in complexity do not apply in this context, as there is no need to expand on Dunjions or reuse the FSM in any way.

2.5 AI in Games: Dynamic Difficulty Adjustment

As mentioned in section 1, static difficulty settings are not an ideal method for game balancing, as they do not respond to increases in player skill levels. As such, several dynamic game balancing mechanisms were introduced, including Dynamic Difficulty Adjustment systems (DDA). DDAs were demonstrated to be an effective method for tailoring the level of challenge within a game to each individual player's level of skill. This is achieved either by manipulating elements in the game environment—such as the amount of health and armor pick-ups, or the number of weapons made available—by altering the map itself [38], or by manipulating the NPCs' parameters. The DDA implemented in Dunjions makes use of the latter method, modifying the parameters for the enemy NPCs found throughout the game. However, these manipulations must follow a strict set of guidelines; if the player becomes aware that the difficulty is being altered, it leaves them with the sense that they are being "cheated', which can lead to feelings of frustration [4, 37]. As such, Andrade, Ramalho, Gomes & Corruble (2006) have highlighted three key components required in all DDAs for the purpose of avoiding that very issue.

The first such guideline is that the DDA must have the ability to quickly identify the player's initial skill level. Once this has been established, the DDA must then perform the necessary changes to match the level of challenge presented to the player with their aforementioned skill level. It stands to reason that there exists a sizeable gap in skill level between the most novice players and their experienced counterparts—otherwise there would be no need for game balancing—thus illustrating the importance of this guideline. The DDA

employed in Dunjions achieves this through the use of multipliers, of which there are two: one that is applied in the event of a checkpoint—making the game more difficult—and another, used to make the game easier, that is applied in the event of a death. In the case of an unskilled player, the multiplier to be applied after they have died is increased with each consecutive death; thus, tailoring the level of challenge to their skill level. The same logic is applied for skilled players; the more frequently they arrive at a checkpoint without dying, the higher the value for the corresponding multiplier, subsequently raising the level of challenge in order to mirror their skill level.

Next, it is important that DDAs are capable of monitoring a player's performance. Consequently, they must also be able to identify both increases and decreases in their performance; this allows the DDA to evaluate how the changes are affecting the player, and whether or not further adjustments need to be carried out. Once again, this is an equally important element to be incorporated into DDAs; as mentioned in section 2.2, when there is a mismatch between player skill and the level of challenge those players are faced with, they will either experience frustration—if the game is too hard—or in the event the game is too easy, boredom. If the DDA fails to react to changes in the player's performance, then they cannot present the player with an adequate level of challenge. In Dunjions, the DDA is capable of monitoring—and appropriately reacting to—changes in player performance. This facet is monitored over the course of each checkpoint by computing the player's progress—equivalent to the percentage of monsters vanquished before a death—and comparing it to the amount of progress made the life prior. Performance is also gauged by evaluating the amount of health the player has remaining upon arriving at a checkpoint; the more health remaining, the better the

player is deemed to be performing. This information is used by the DDA to make suitable changes.

The final component mentioned as a guideline to be followed when developing a DDA is that the game's behavior must remain believable despite the changes being performed [3]. Therefore, safeguards must be used to avoid scenarios where the player would become aware that the difficulty is being altered; such as a monster's attack power being set to 0. In Dunjions, the DDA makes use of base values in order to adhere to this guideline; each monster parameter is assigned a minimal base value that cannot be altered by the DDA's adjustments. Instead, the adjustments are performed on integers that are later added to these base values. In cases where the game needs to be as easy as possible for the player, those integers are set to 0; rather than the base values themselves. As such, the NPCs' behaviours remain believable despite the changes being performed by the DDA. Furthermore, the degree of changes carried out by the DDA are designed to be gradual. As a result, drastic changes to the game's difficulty will not happen from one life—or checkpoint, for that matter—to the next, thus making it significantly more difficult for players to perceive these changes, subsequently making the NPCs' behaviours believable throughout the course of the experiment.

2.6 Wizard of Oz Method

First used in 1985, the WoZ technique was created as a tool to be used for the development of speech interfaces [31, 33, 41, 55]. Thanks to this method, people can interact

with components of an interface that have yet to be implemented. In order for this to occur, two computers, which are located in separate rooms, are connected to one another. People that are testing the interface sit at the first computer, while a confederate (Wizard) sits at the other [41]. This is illustrated in the diagram below:

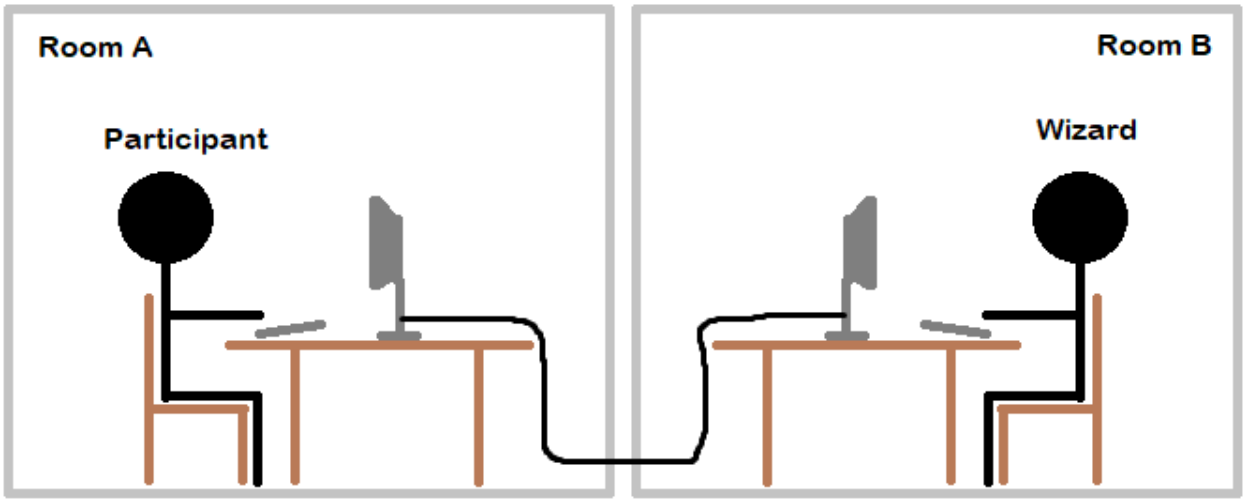


Figure 3: Setup for the WoZ method

It's important to note that the participants are unaware that the interface that they are testing contains components that have yet to be implemented, nor are they aware that their computer is connected to another computer in an adjacent room. They are simply told that the developers are seeking feedback about the interface as a whole. Before the "experiment" commences, the confederate undergoes training with regards to how the unimplemented components are intended to work. This is essential; because the confederates are expected to act as the unimplemented portions of the interface, they need to react to any user input in the same manner that the developers envisioned that the interface would react in the same scenario [31]. Therefore, during the testing phase, participants interact with the interface, and the confederate steps in and responds to the user input when appropriate. The participants are under the

impression they are only interacting with the interface, and are unaware that there is, in fact, a "man behind the curtain".

This method raises questions regarding the necessity of involving confederates instead of implementing the elements of the interface for which designers are seeking feedback. Simply put, implementation takes time. While this may seem trivial, if participants don't respond well to those elements, designers have saved themselves time that otherwise would have been wasted. As such, the WoZ method allows developers to test those aspects without having to implement them [8]. If they're well received, then developers can move forward with their implementation. Otherwise, the time they saved using the WoZ method can be repurposed into creating an alternate solution.

2.7 Design Principle for User Interfaces

In order to use the WoZ method described in the previous section, an interface must be created for the Wizard to allow them to act as the unimplemented portions of the software. While this may seem like a simple process, there are several elements that significantly affect the ease of use—and ultimately the performance of the Wizard—when using the interface. A number of these elements are discussed in *The design of Everyday Things*, and while the author refers to simple objects in his examples, they also apply to more complex items, such as the user interface that was designed for this study. Each of these aspects—along with their influence in the design process—will be discussed briefly.

First and foremost, there is affordance. The notion of affordance simply means that hints pertaining to how an object is intended to be interacted with should be made evident to the user. Some of the examples listed in the book include door knobs, balls, and chairs. Door knobs are round objects that “invite” people to turn them, chairs are flat surfaces on which users are inclined to sit down, and balls hint that they should be thrown or bounced. No matter the object, in order for users to grasp how it is intended to be used, there must be a certain level of affordance associated with it. According to Norman, if the object requires a picture, a set of instructions or a label to show the users how it is intended to be used, the design has failed, as this is supposed to be accomplished thanks to affordance. In the figure below, the interface provides the user with an adequate amount of affordance; the knobs on the sliders invite the user to “click and drag” them, users associate the checkboxes with their primary function of being clicked, and the buttons give the user the impression that they can be pushed. While some initial guidance as to the purpose of the sliders and buttons may be necessary, users should only require little amounts of practice before they able to operate the UI quickly and efficiently.

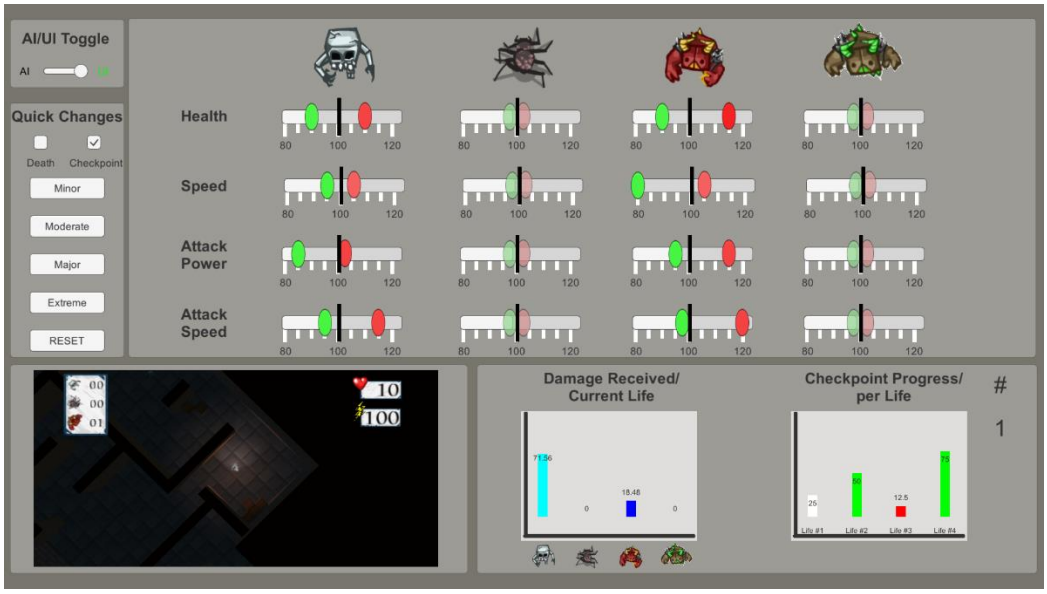


Figure 4: Interface used during this experiment

The next principle that must be considered when designing an object is visibility. In this context, visibility means that it is evident to the user how the device should be used. This does not mean that the user must be made conscious of how the object operates, however. Visibility can be achieved through the use of subconscious clues. With the interface displayed in the Figure above, the sliders offer visibility as their handles are placed in the middle of a horizontal plane, while tick marks that lead towards either extremity subtly hint to the user that the handles should be pulled in those directions.

A third principle that is outlined by Norman [48] is that of mapping. Mapping is a term used to define the relationship between an action and its subsequent reaction. For example, when typing on a keyboard, each and every keystroke is met with the appearance of that key on the screen (assuming of course the keys being pressed are letters). This relationship is important, because it allows the users to make the connection between their actions and the effects that those actions have on the world, thus allowing them to learn how to use the device. This

relationship is established in the interface used in this study when the users manipulate the sliders. The handles on the sliders move in the direction they are dragged after the user has clicked on them, and they remain in the position they were in when the user lets go of the left mouse button. Furthermore, when users select one of the quick settings, the handles immediately move to the desired values, once again establishing the relationship between the user's actions and the effects that ensue.

Next, Norman discusses the principle of feedback, one of the simplest, yet surprisingly important aspects of design. When the user performs some sort of action, it's essential that this action be met with some form of feedback, be it auditory, visual or tactile. Examples of feedback include buttons lighting up when they've been pressed (elevators/visual), buttons that "click" when they've been pressed (computer mouse/auditory), or the buttons that vibrate when they're being pressed (gaming controllers/tactile). The dashboard designed for this experiment provides feedback in response to a user's actions; the colour of the handles that are being dragged light up, the buttons turn grey to give the illusion of being physically pressed when clicked, and checkboxes are filled with a checkmark when they've been selected.

When trying to close a window on a computer, common knowledge dictates that there is only one way to accomplish this goal: by positioning the cursor over the X in the top right corner (or the red circle in the top left corner for Mac), and making a left-click on it with the mouse. Other actions—such as hovering over the button, right-clicking on the button, or using the scroll wheel—do not have the same effect. This is another design principle, that of constraints. According to this notion, it is essential that the number of actions that can be performed be restricted. Although this may seem counterintuitive, by reducing the number of actions that can

be performed, the likelihood that a device is used improperly is reduced. This was incorporated into the interface by limiting all elements with which the user could interact with to a single method of interaction. The mouse is used to interact with every element in the interface; the checkboxes are selected by clicking the left button on the mouse, the slider handles are clicked and dragged, and the buttons are pressed using a left mouse click. There are no keyboard shortcuts that can accomplish the same actions, and there are no other methods of using the mouse for these interactions.

Consistency is another one of the design principles covered in Norman's text. In essence, consistency means that different devices that are used to perform identical tasks should operate in a similar fashion. Computer mice, for instance, are all different. Some may be ergonomically shaped, while others may have different sized buttons. However, what they all share in common is the functionality of the left and right mouse clicks, as well as the scroll wheel. This is why consistency is an important design principle; it helps eliminate any confusion when using different interfaces that have the same end goal. An example where there still seems to be a lack of consistency can be seen with the number pads on computer keyboards or cellphones. On some devices, number one is the top-left most key, with two and three located on adjacent keys on the same row, before moving down a row for numbers four through six. Seven through nine are then on the second to last row, with zero having a row all to itself. However, this pattern is not consistent across all devices. Some will have number one in the bottom-left most corner, with the remaining numbers ascending from left to right and from the bottom row to the top row. While this may seem trivial, certain professions rely on a user's ability to quickly input information using these number pads, such as nurses. If the layout changes from one device to the next, it

creates the potential for nurses to input improper dosages, which could result in a patient's death. As such, the interactive components of our interface are consistent with similar components on different devices.

Steven Few's textbook titled *Information Dashboard Design; The Effective Communication of Data* was another important reference used during the design of the prototypes. According to this text, along with several newsletters [27, 28, 29, 30] that he has published, there are five components that are key when creating a dashboard for rich and rapid monitoring. First and foremost, the dashboard must fit on a single screen. An element that was also highlighted in other works [34, 58], this requirement is stressed in large part due to a human's working memory, which is defined as the temporary storage and processing of information that is rapidly forgotten unless rehearsed [6, 10, 35, 60]. If users must navigate between several different tabs to gather the information they're seeking, not only is it time consuming, but it also requires that the users make mental notes for each piece of data that was found on a different tab; presenting the possibility that some of the numbers are forgotten or mixed up. This component was incorporated into the final design of this interface; it does not have any tabs that the user would need to cycle through, and all of the information is displayed on a single screen.

Secondly, the display media must communicate information quickly and efficiently [26, 34]. Although this may seem as simple as placing as much information as possible into a table or a graph, designers must take the time to give context to those mediums, as the data contained

within may lose all meaning without it [39]. As Few puts it, “Measures of what’s going on in the business rarely do well as a solo act; they need a good supporting cast to succeed” [26]. This means supplying data to which users can compare the values being presented in order to establish what they mean in the bigger picture. Furthermore, it takes time to choose the proper medium for displaying the data. Some forms of data representation, such as graphs and charts, are better suited for data from which trends or patterns are expected. Others forms of data representation—such as tables—are more appropriate when the data contained within is not used to determine patterns, but rather to have the data expressed in the most precise manner possible or to facilitate the extraction of individual values [30]. Displaying data using improper mediums will ultimately hinder its significance; trends are harder to detect when data is displayed in a table, while exact values are difficult to extract from a bar graph. This aspect was taken into account during the final implementation of the dashboard used in this work. Different types of graph were carefully considered in order to properly communicate important data to the user, with the final decision resulting in the use of bar graphs over line graphs. Furthermore, the graphs give the user an appropriate amount of context, as they have titles, the values are displayed above the bars, and comparisons can be made by the user because all the relevant data is contained within the same graphs.

Another one of the pitfalls in dashboard design is that most dashboards fail to create a visual emphasis on information requiring the user’s immediate attention [26, 27]. While all the information on a dashboard may be important, this does not imply that all the information is of equivalent importance. As such, it’s imperative that dashboards draw the user’s attention to the areas of the screen that contain the most important information. This is best achieved when

ONLY important information is highlighted, and the use of decorative features must be kept to a minimum [34]. From this point forward, the user can take the time to examine the rest of the dashboard at their leisure. If too many areas are highlighted, the user won't know where they should be focusing their attention, similar to when there is no highlighting at all:

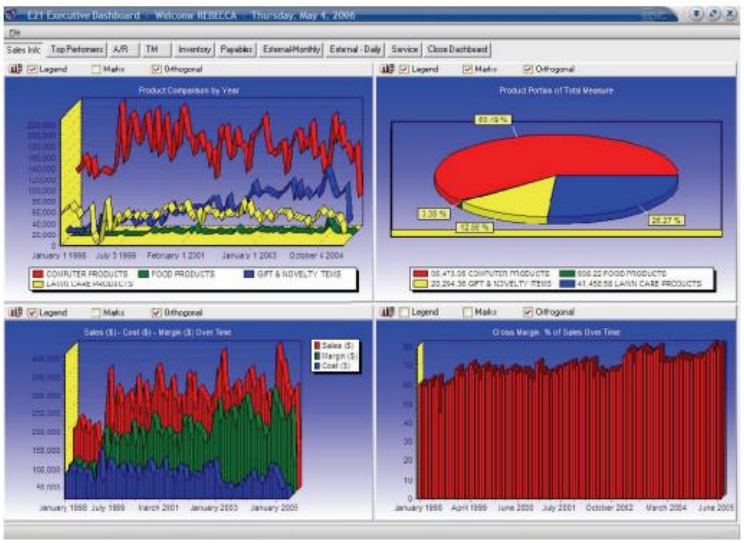


Figure 5: Dashboard that does not use colours sparingly

This is a dashboard that Few [26] uses to stress that very issue. Notice that the information contained within all four quadrants is visually salient; as a result, it's impossible to determine what is most important. If, however, only the information in one of the four quadrants was made visually salient, then the eye would be drawn to that quadrant first. This could have been achieved by using neutral colours for the information in the other three quadrants. This desirable effect was replicated in the final design of the interface used in this study by using neutral tones for the background and panels; saving warm colours for the handles on the sliders and for the bars within the graphs. As a result, the Wizard's eyes are drawn first to the graphs,

from which they can extrapolate the types of changes that they should make. Their eyes should then be drawn to the sliders, where they perform those changes.

Another important aspect of dashboard design is that the information to be arranged in the most logical manner possible [26]. This means that not only should similar information be grouped together, but also that the location chosen to display the information on the dashboard is of great significance. The need for similar information to be grouped together is fairly straightforward; it allows for quick comparisons, and it provides the user with a logical sequence in which to view the information. If done correctly, it will allow the user to navigate efficiently through the dashboard. With regards to the placement of the information, Few [26] states that certain areas of the screen draw the user's attention more than others. This notion appears to have been derived from what is known as the Z pattern of processing. Although this phenomenon is said to apply only to text-heavy content, such as novels or newspapers, it has certain aspects that apply to dashboard design. In order to better illustrate the Z pattern of processing, the Gutenberg Diagram was created, as seen in the figure below:

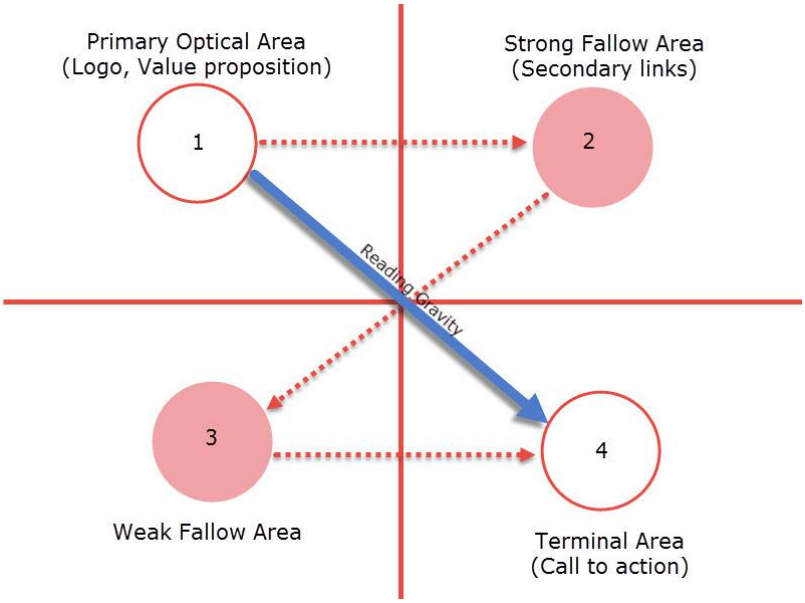


Figure 6: Gutenberg diagram

Traditionally, texts that are meant for Western readers are written from left to right and from top to bottom. As such, the Gutenberg diagram divides any display medium into four quadrants, with the top left quadrant being denoted as the primary optical area. Instinctively, this is where readers focus their attention when first presented with the medium, and is therefore the area that should contain the most salient information [43]. Readers then scan the display medium in a series of sweeps, from left to right and from top to bottom. This is tendency that has been developed from reading, and is denoted in the diagram as reading gravity [43]. As a result, the top right quadrant (strong fallow area) should contain any important information that was not included in the primary optical area. The bottom left quadrant, known as the weak fallow area, should consequently only contain information of minimal importance, as reading gravity dictates that a reader's eyes will be guided away from this area and towards the terminal area (bottom right quadrant). Below is an image created by Few [26] that is intended to guide users when designing their own dashboards:

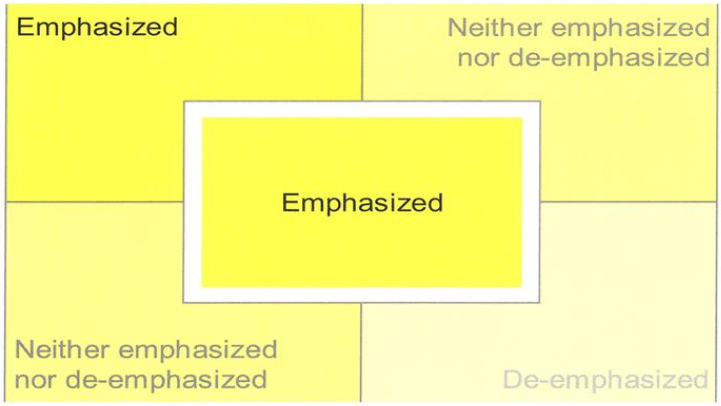


Figure 7: Areas of visual emphasis on a dashboard

Closer inspection of this image reveals that the top left corner of the dashboard is one of only two areas that are emphasized, with the center of the dashboard being the other. It would seem that this image follows the Gutenberg diagram, as the primary optical area—the quadrant which should contain the most important information—is also located in the top left corner of the display medium. Furthermore, the emphasized area located in the center of the dashboard seems to be adopted from the idea that reading gravity will pull the user's focus from the top left corner, through the middle of the dashboard, towards the bottom right of the display. As a matter of fact, Few mentions that this is most effective when surrounded by a white border, something that has also been said to reinforce reading gravity [43]. Although these two diagrams may not assign the same significance to the top right and bottom left quadrants, it's clear that Few's interpretation shares several key elements with the Gutenberg diagram, and should therefore be consulted when designing a dashboard. The spatial arrangement of the implemented design for this work puts this notion into practice. In the top left corner, a switch was placed that allows a user to toggle between the two conditions of the experiment: DDA making changes and Wizard making changes. While this is not essential to the Wizard, the experimenter must quickly make

this change when launching the program. Directly below this switch is the quick changes panel. This was included in the most salient location of the display, as it affords the Wizard with the opportunity to alter as many parameters as possible as quickly as possible. The graphs were placed in the bottom right corner of the display due to their importance regarding potential changes to be made. Because reading gravity pulls the users attention to this area of the interface, information contained within is examined quickly by users, hence the decision to place the graphs here. The mirror image of the participant's display was placed in the least salient area of the interface, as it was deemed to be the least essential component with regards to providing the Wizard with information essential for modifying the parameters within the game. Finally, the sliders were placed in the center/top-right corner of the interface because reading gravity pulls the users attention through this area.

The final aspect mentioned that is essential to dashboard design is that it must be aesthetically pleasing [26, 34]. This last point can be viewed as a double-edged sword. Although it's important for the dashboard to be easy on the eyes, it's just as important to avoid adding any decorative features that distract the user and take away from the significance of the information that the dashboard is trying to convey [34]. An example of this can be seen in the dashboard below:

Figure 8: Dashboard using glare as a decorative feature

Clearly, the gauges were designed to look like those that might be found on an actual dashboard. Glare was even added in order to create the illusion that the gauges are protected by a glass surface. Despite this attempt at making the dashboard more visually appealing, Few puts this effect into perspective by referencing everyday encounters with glare: “Notice how much care has been put into making this display look like an actual dashboard, down to the glare of sunshine on the surface of the gauges [...] [but] when we encounter glare in the real world, we find it annoying” [29]. Whenever glare reflects off a surface, it makes objects harder to see, in which case most people are forced to squint. Therefore, it’s easy to understand why this attempt at making the dashboard more visually appealing is seen as a distraction. Instead, developers should follow this guideline when designing their dashboards: “Effective dashboards lack eye candy. They are not designed to wow people upon first sight, but rather to inform people with precisely what they need when the need it, day in and day out.” [27]. This comment is made in reference to a well-designed dashboard which can be seen below:

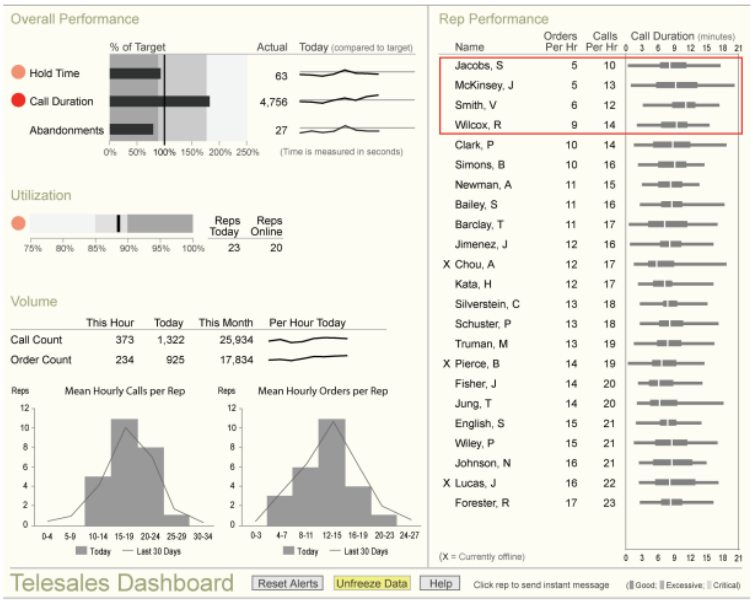


Figure 9: Example of a well-designed dashboard

This particular dashboard does not possess any distracting decorations. While it may seem plain, the use of neutral tones is both easy on the eyes, and allows for important data to stand out from the rest. Furthermore, by limiting the use of colour, the overwhelming effect that was found in the dashboard in Figure 5 is eliminated. As mentioned above, the final implementation of the dashboard used in this study makes use of neutral colours for the background and panels. This makes the interface easy on the user's eyes. Distracting decorations were also neglected. Finally, the use of bright, vivid colours was reserved for information that requires their immediate attention, as it stands out from less salient information.

2.8 Prototyping Methods

In section 3.2, the procedure of designing the interface is discussed. This was not as simple as creating and implementing a single design, however, but rather the process of developing several different prototypes and then evaluating the pros and cons of each one. Using prototypes in this manner is useful when trying to determine requirements for the final product, yet different prototyping methods exist that can accomplish that very goal; such as throwaway prototypes and evolutionary prototypes. While they may accomplish a similar goal, these methods cannot necessarily be used interchangeably. In this section, throwaway prototypes and evolutionary prototyping are discussed, as well as the similarities and differences between these two methods.

Throwaway prototyping is a quick and dirty prototyping method [19] wherein a prototype—usually implemented through pen and paper—is created [19, 67]. Generally used to helps clarify the user's requirements [19, 70], this method only implements the poorly understood requirements for a product in order to receive feedback [19, 63]. There are several benefits to throwaway prototypes, most notably that they are quick to design, thus allowing users to provide clarifications about their requirements early in the design process [63]. Once users have seen how their requirements may take shape and have provided feedback about the design, the prototypes are discarded, and developers begin creating the final product [19, 67].

While throwaway prototypes are used to clarify requirements that are already known to the developers, evolutionary prototyping is used to uncover requirements for an interface that were not thought of from the outset [58]. As a result, this procedure is known to yield robust final products [58]. This is achieved through the use of rigorously developed prototypes [19].

These prototypes should only implement the confirmed requirements, and they are only to be used experimentally; they are not to be released as final products. Once developed, the prototypes are given to potential users for their feedback. After using the prototypes, users report problems they encountered throughout their interaction, as well as enhancements they believe would improve the final product. Feedback from this step in the evolutionary prototyping process is then addressed by the developer, at which time a new prototype is developed, and a new iteration of the process can commence [19].

In summation, developers must carefully consider which prototyping method to use before proceeding in the design and implementation of a product. Throwaway prototypes are essential when the user's initial requirements are unclear; a prototype—which is eventually discarded—is created rapidly to give the user a sense of how their requirements would take shape, thus allowing them to refine those requirements as necessary. Evolutionary prototypes, for their part, are more appropriate when attempting to discover requirements for a system that were not initially identified by the user. Unlike the former prototyping method, one of the prototypes created using the evolutionary prototyping method will continue to be refined, eventually leading to the design that is implemented as the final product.

3 Design

3.1 Artificial Intelligence

This experiment made use of a game that was created in [66]. The Artificial Intelligence (AI) that was designed for the game was modeled after the Dynamic Difficulty Adjustment (DDA) system. Such systems, as discussed previously, can be used to alter the difficulty of a game for the purpose of maximizing player engagement. However, not all DDAs make the same adjustments. As mentioned above, certain systems will change the configuration of the map, while other systems may increase/decrease the number of health or inventory pickups made available to the player. The DDA that was implemented in Dunjions, however, makes adjustments to the NPC enemies found throughout the game. More specifically, parameters such as enemy health, attack power, attack speed and movement speed can be modified to alter the game's difficulty level. Those parameters are modified following either one of two potential events: when the player arrives at a checkpoint, or when the player has died, as seen in the Figure below:

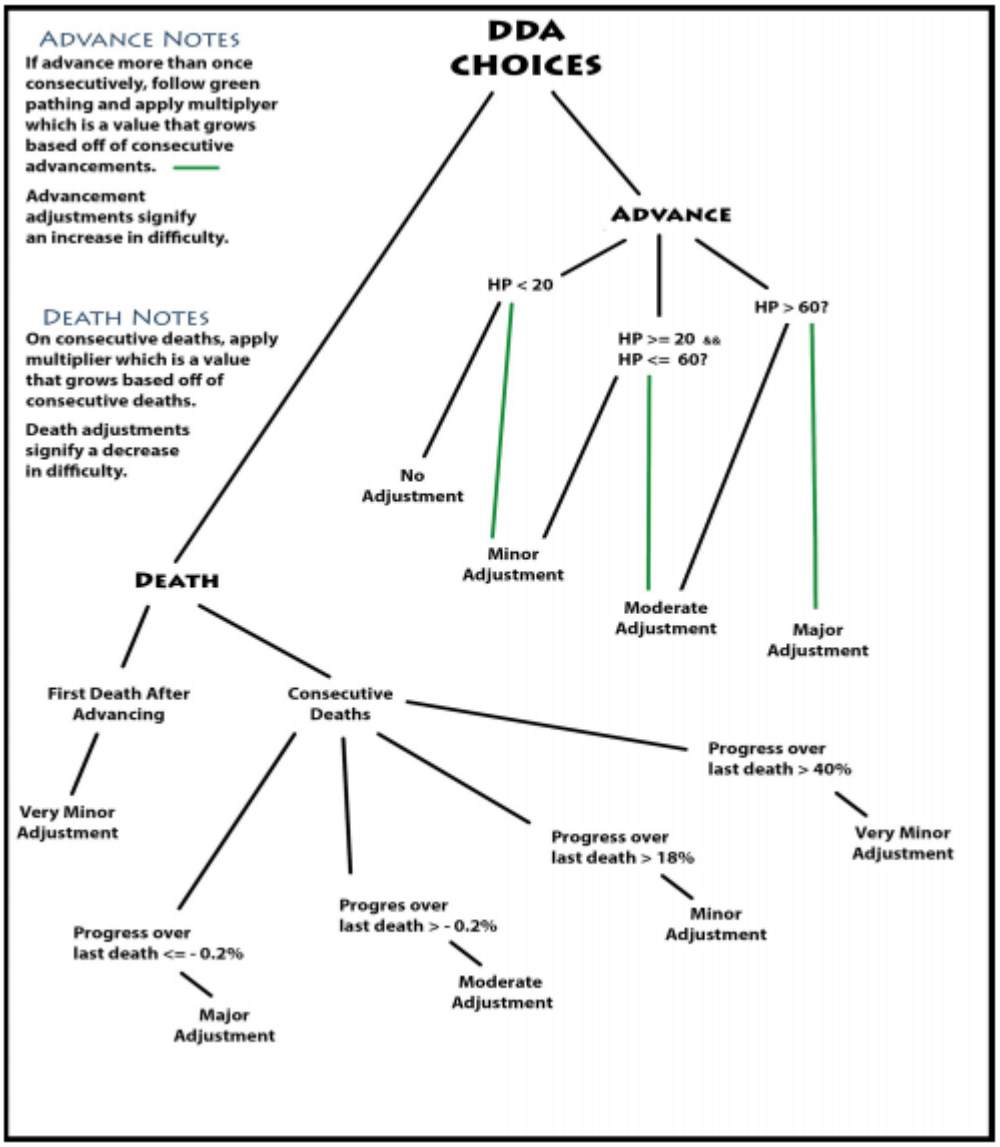


Figure 10: DDA decision tree for Dunjions, retrieved from [66]

Inspection of this decision tree reveals that changes are only made following one of these two events in order to keep the player unaware that the game's difficulty is being altered. This is paramount, as it has already been demonstrated that player's feel "cheated" when they become aware that the difficulty was adjusted while they are playing [4, 37]. Therefore, by making these

changes between lives or between checkpoints, the likelihood that these changes will be noticed by the player is reduced.

In the case where the player arrives at a checkpoint, the AI first checks to see if this constitutes a consecutive checkpoint without a death. Then, the AI verifies the amount of health the player has remaining. With this information, the AI adjusts each one of the four monsters' parameters in order to increase the difficulty of the game. The degree of the adjustments made is relative to the amount of health the player had when they arrived at the checkpoint. The most significant adjustments are made if the health of the player is above 60. Moderate adjustments are made if the health is between 20 and 60, while minor adjustments are made if the health is lower than 20. This formula is used for both checkpoint scenarios. Therefore, in order for the modifications to have more of an impact when the user makes it to a second consecutive checkpoint—because the game is clearly still too easy—an additional multiplier is used for that scenario only. Therefore, if they've made it to consecutive checkpoints without dying, the additional multiplier makes those changes more significant.

In the event that the player has died, the DDA has been designed to alter the monsters' parameters to make the game easier. However, unlike the modifications made upon arriving at a checkpoint, not all monsters will receive the same adjustments after a death. First, the AI calculates the amount of progress made by the player before their death. As discussed in Section 4.1.3, players must kill a specific number of each type of monster in order to gain access to the next room; progress is therefore computed by comparing the number of enemies killed against the required number of enemies that need to be killed. In the event that the player has died more than once on the same checkpoint, the AI compares the amount of progress made this life to the

amount of progress made the last life. If the user's progress was at least 40% better than the previous life, minimal (0.96) changes are made, which can be seen in the figure above. If, on the other hand, the user's progress was at least 18% better than the previous life, mild (0.94) changes are made. In the event that the user's progress increases by at least 2%, the AI makes moderate (0.91) changes to the monsters, while the most significant changes (0.85) are made if the user fails to progress any further than they did in their previous life.

After these initial changes are applied to all monsters, the AI counts how many times the player has died in a row in the same checkpoint; this allows the AI to compute the significance of the modifications that will be carried out next. Then, it determines which one of the monsters dealt the most damage to the player before they died. Next, the AI concludes if the same monster has killed the player in consecutive lives; if so, it determines how many lives in a row this has occurred. Additional modifications are then performed on this monster, as demonstrated in the figure below:

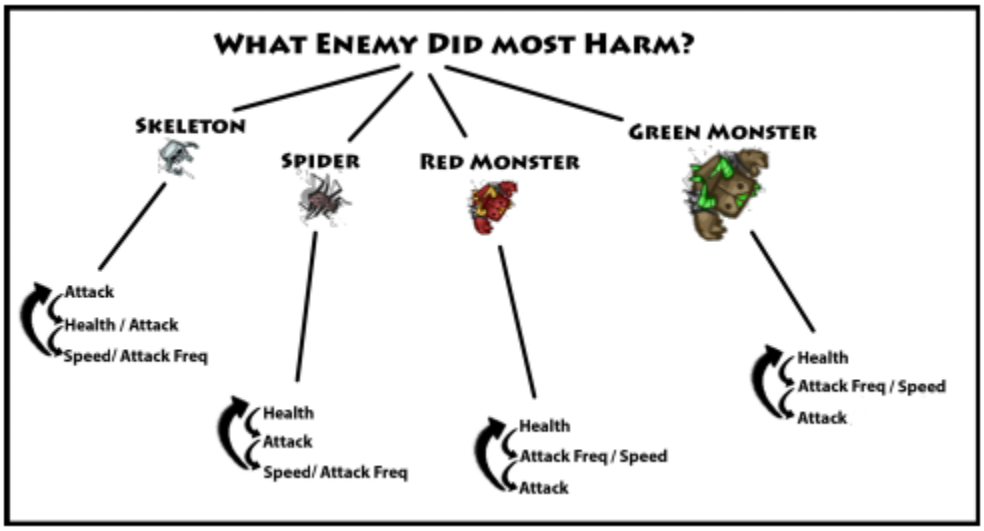


Figure 11: Decision tree for further modifications upon a death, retrieved from [66]

The figure above illustrates that the significance of these modifications is dependent on the number of consecutive deaths that have occurred on the same checkpoint. The number of consecutive deaths caused by the same monster determines the number of parameters will be affected by the additional modifications. If this was the first time, only the monster's health is further decreased, whereas a second consecutive death will also affect the monster's attack power, and so on. An example can be seen in the table below:

Monster	Health	Speed	Attack Power	A. Speed
Skeleton	0.893	0.921	0.867	0.921
R. Mons	0.921	0.921	0.921	0.921
G. Mons	0.921	0.921	0.921	0.921
Spider	0.921	0.921	0.921	0.921

Table 1: Sample output of modified monster parameters

This table was generated for the output file from one of the participants from this study. For context purposes, this table is the result of the modifications made after the participant died for the second consecutive life on the same checkpoint, where the skeleton was monster that dealt the most damage to the player. The values in the table represent the multipliers for each monster's stats; each monster has different base values for each one of these parameters, and the multipliers are used to increase/decrease those values.

For the vast majority of the stats, the multipliers have been set to 0.921, meaning that upon respawn, those parameters will be set to 92.1% of their original value. Since this is a second consecutive death, and because the skeleton was the enemy that killed the participant in both lives, the multipliers for the skeleton's health and attack power vary from the rest: 0.893

and 0.867, respectively. This is due to the fact that the DDA has been designed to make additional adjustments to the enemy that posed the greatest challenge the player. The first time the player died, only the skeleton's attack power was further adjusted. Then, once the player died a second time, additional adjustments were also made to the skeleton's health, which is why those two values are not the same. This table also demonstrates that the changes that are made between lives are small enough not to be noticed by the user, as two consecutive deaths from the beginning only result in an 8% adjustment for all but two of the parameters.

3.2 User Interface

In section 2.8, both throwaway and evolutionary prototyping methods were discussed. In order to develop the interface that was going to be used for this experiment, an evolutionary prototyping process was employed, as the initial requirements were well understood. However, the best method for executing those requirements remained unclear. Furthermore, as is discussed in sections 3.2.3 and 3.2.5, experimenters were able to uncover several requirements that were not established from the outset. As such, an evolutionary prototyping method was clearly the most appropriate for of prototyping for this situation.

Knowing how the DDA makes its adjustments was essential, as the basic requirements for the interface emerged primarily from this facet alone. As previously mentioned, the Wizard has no knowledge about how the adjustments are made, which monsters those adjustments are applied to, or about the significance of those adjustments. However, in order for the Wizard to

make adjustments in a similar fashion, it was important for them to have access to the same information as the DDA. As a result, prototypes were required to not only allow the Wizard to make changes to the parameters, but to also inform the Wizard as to the progress made by the participant and the damage being dealt to the player by each of the monsters. Several different prototypes were created with these characteristics in mind. Furthermore, it was important for the interface to follow the design principles mentioned in section 2.5. Sections 3.2.1 through 3.2.4 address the evolutionary prototyping procedure used, while section 3.2.5 discusses the final product that was generated as a result.

3.2.1 Designing the Interface: Phase One

The first prototype that was created can be seen below:

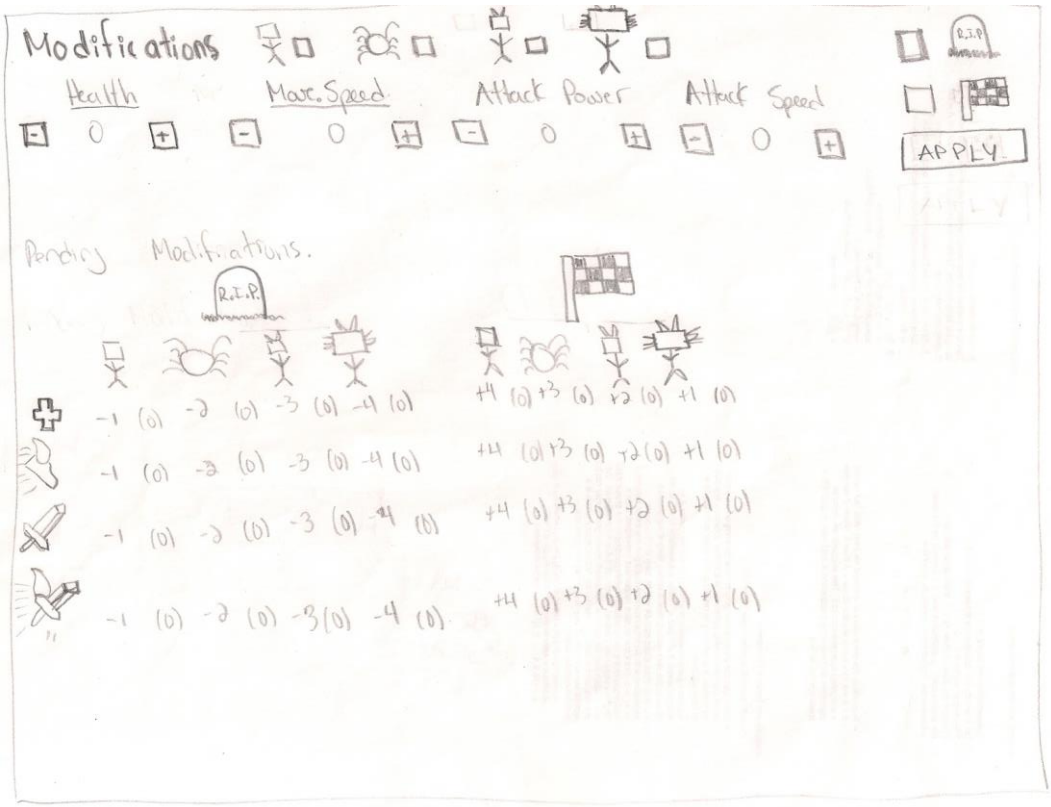


Figure 12: Prototype A

At first glance, this prototype was noticeably designed with the intent of using only checkboxes and buttons to manipulate the interface. The user would first select the monsters to which they wish to make modifications. They would then use the "+" and "-" buttons under each parameter to alter their values, check off whether they wanted those modifications to be applied the next time the player died or the next time they made it to a checkpoint, and then hit the apply button. The tables at the bottom of the screen are used to remind the user of the changes that are pending in the event of either a death or a checkpoint.

Figure 10 displays the second prototype that was created. This version of the interface replaces all of the checkboxes and buttons with knobs.

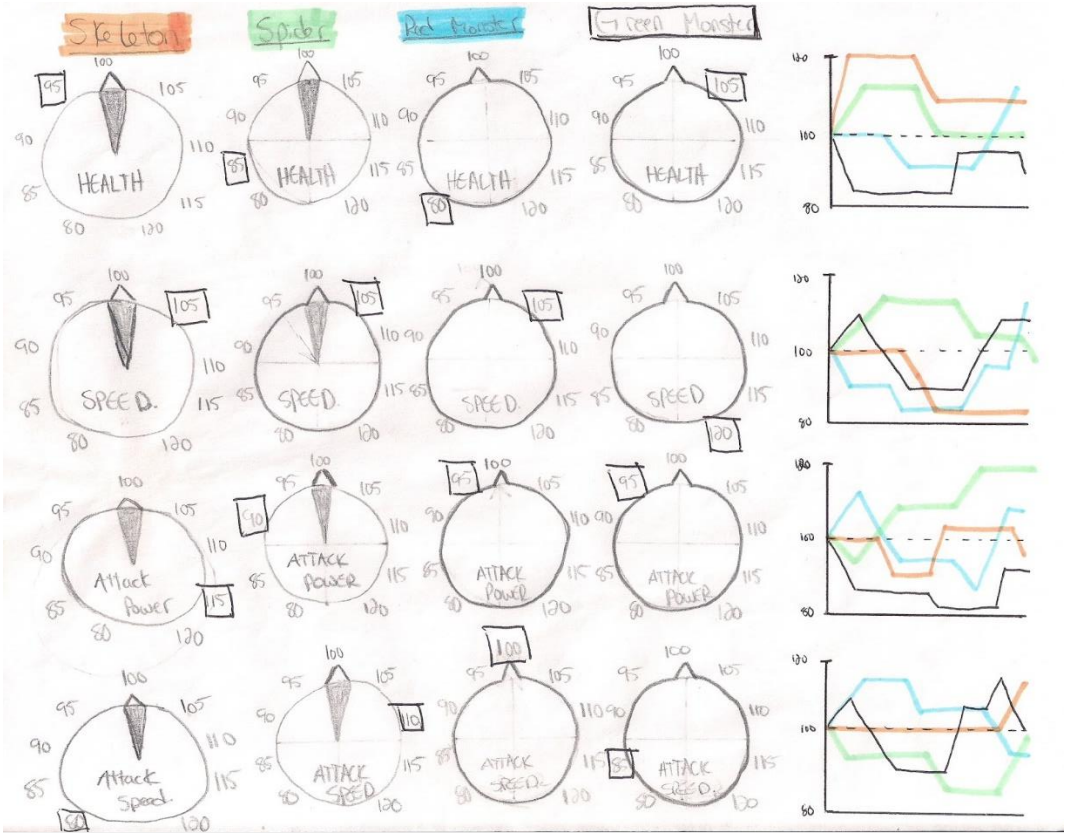


Figure 13: Prototype B

Given that each monster has four parameters that can be altered by the user, four sets consisting of four knobs per monster were placed on the screen for the user to manipulate. The potential values to which each parameter can be set encompass the knob, with the current setting being the one found at the top. The users would then be required to click on the tip of the knob of the parameter they wish to change and drag it until it aligns with the value with which they wish to see it change. When they let go of the knob, the value that is being pointed to by the tip will be circled/highlighted. The knob would then revert to its initial position, allowing users to set pending modifications for both a death and a checkpoint. Given that it is intended that the game be made easier upon a death and more difficult upon arriving at a checkpoint, it is implied that choosing a value to the left of the dial would represent the changes to be made upon a death, whereas the values to the right represent the changes that would take effect on a checkpoint. Finally, the graphs to the right of the screen represent a time-lapse of the changes that have been made to each monster. These graphs have been colour coded to correspond with the colours that were used to highlight each monsters name, and the row on which the graph is found corresponds to the parameter that the user is altering.

In the following prototype, the knobs have disappeared entirely. In their place are a series of sliders that the user can drag to set the pending modifications:

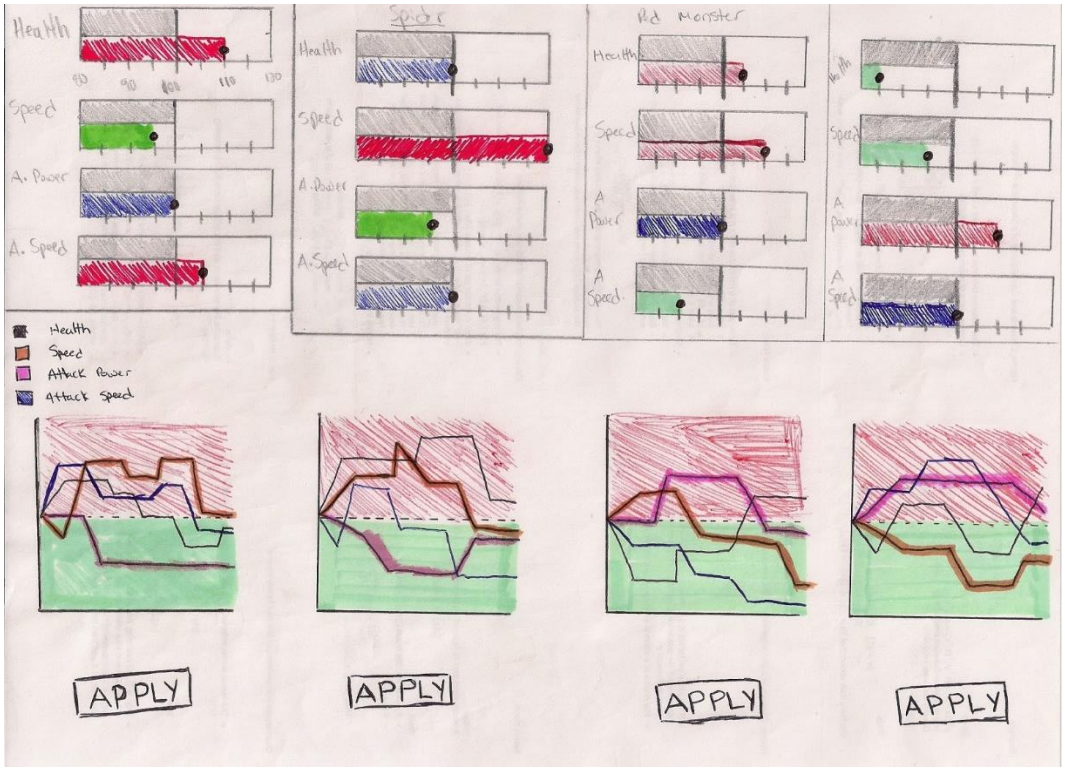


Figure 14: Prototype C

Once again, there are four sliders available per monster. The current value is designated by the grey bar above each slider. The user would click the handle at the end of the sliders below, and drag them to whichever value they desire. Should they choose a value higher than the current setting, the slider would turn red, signaling that they intend on making that particular setting more difficult. Conversely, the handle would become green if the user selects a value that would make the game easier. If, however, the user decided to leave the settings as they are, the slider would remain blue. The graphs located in each column represent a time-lapse of the modifications made to each monsters' parameters over the course of the experiment. This provides the user with some insight with regards to the impact their modifications are having on the players' performance. Also note the "Apply" button located below the graphs, which must be pressed when the users have finished making their desired modifications.

The next prototype developed during this stage can be seen below in Figure 12. Although it may appear confusing at first glance, the concept behind this design is similar to the prototype that was equipped with knobs. There are four dials per monster, one for each modifiable parameter. The current setting for each parameter is identified with the grey needle. The user would then click and drag the red and green needles in order to set pending modifications for a checkpoint or for a death, respectively. Like the prototype above, this prototype is equipped with graphs that reflect the changes made to each individual monster. Each of the parameters has been colour coded (health in yellow, speed in pink, attack power in purple and attack speed in brown) to correspond with the lines in the graphs below them. Furthermore, horizontal grid lines have been added to the graphs to represent events (deaths and checkpoints) that have occurred. These gridlines are also colour coded, with green used to signal that the game was made easier due to a death, while red means that the game was made harder due to the player arriving at a checkpoint. Unlike the prototype above, however, this prototype has also been equipped with two additional graphs on the right. The first informs the user which of the monsters has done the most damage to the player (the bar representing the damage dealt to the player by that monster becomes red while the others remain blue), and the other demonstrates the amount of progress that was made on the current checkpoint before the player died. These two graphs provide information that guide the user when making their adjustments, as the first will notify the user to the monster causing the player the most trouble, suggesting that adjustments should be made to make them easier, while the second shows how the changes over the course of the current checkpoint have affected the players' progress. If the player seems to be making less progress from life to life,

this would suggest that the modifications have made the game too difficult, whereas a dramatic increase from one life to the next could suggest that the game was made too easy.

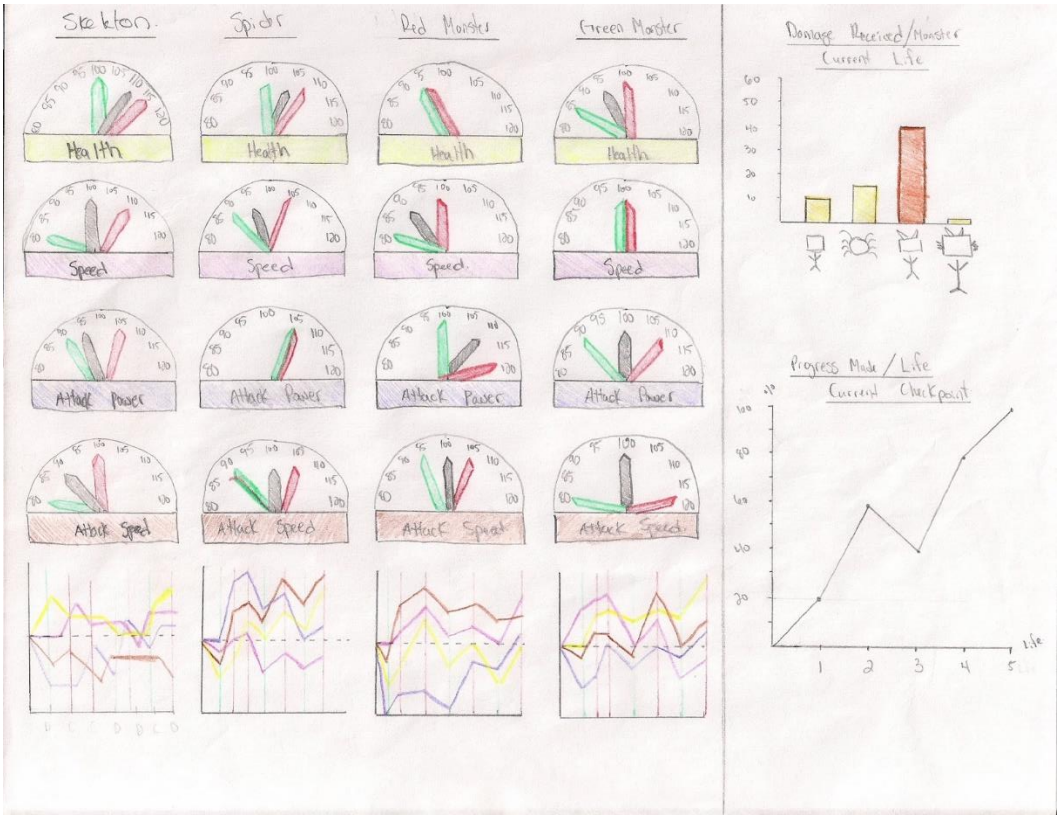


Figure 15: Prototype D

The prototype seen below was the final prototype created during the first phase of the design process. Although it may appear that there are only graphs, with no means to make adjustments to the monsters' parameters, that's not the case. Similar to the two previous prototypes, the four graphs to the left of the screen represent the changes made to each of the parameters over the course of the experiment. However, these graphs are unique: they are also dynamic, which is to say the parameters are adjusted within the graph itself:

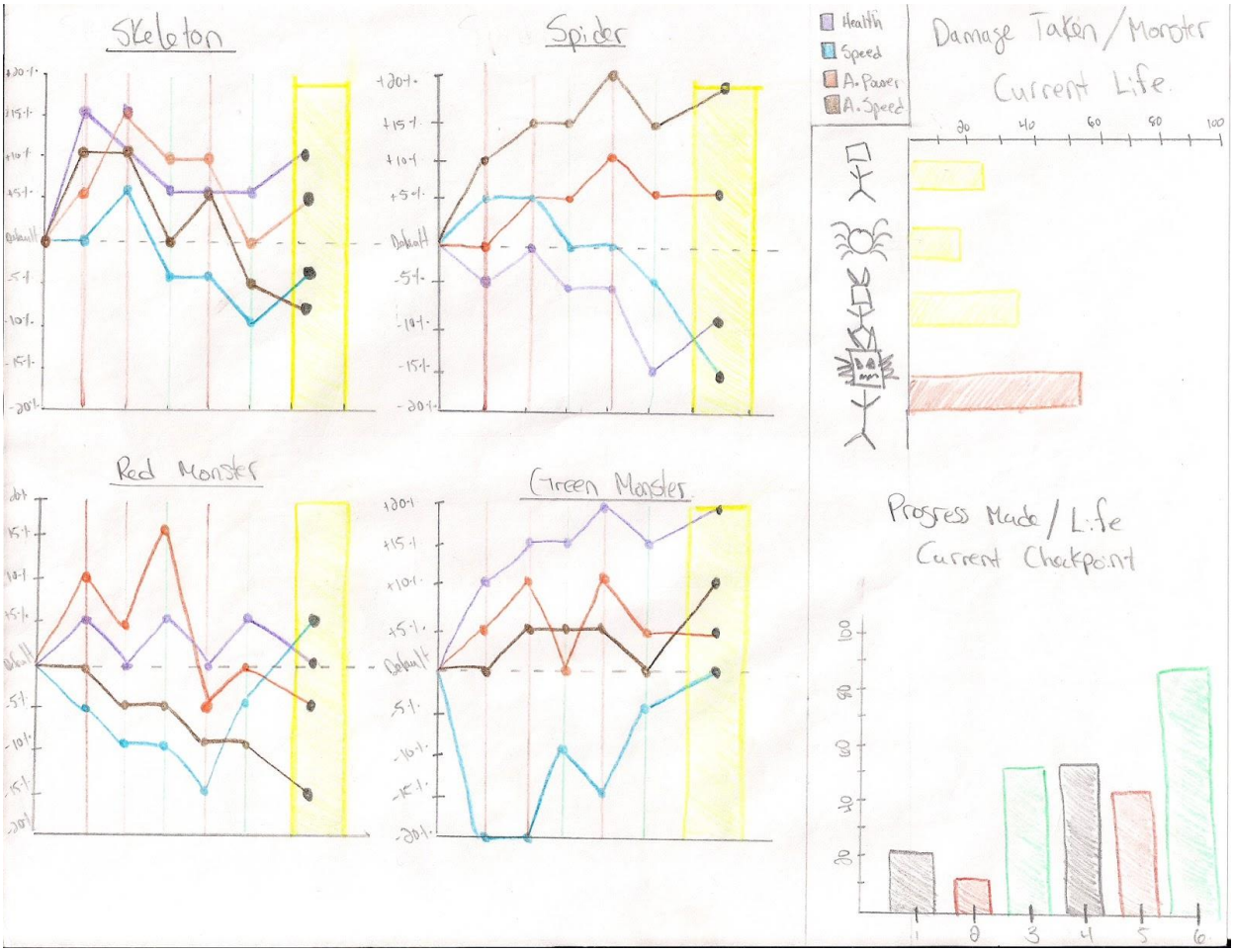


Figure 16: Prototype E

For example, in order to reduce the skeleton's attack power, the legend demonstrates that attack power is represented by the orange line. Therefore, in the yellow column of the graph (which represents pending modifications), the user would click the handle on the end of the orange line and drag it down to the desired value. The graphs to the right of the prototype remain the same as its' predecessor, where the first illustrates the damage dealt to the player by each monster and the second demonstrates the progress made by the player each life over the course of the current checkpoint.

3.2.2 Prototype Evaluation: Which Design is Preferred by Users?

Objective

Vastly different designs were created as prototypes for the interface to be used by the Wizard. An experiment was conducted in order to determine which design was the best candidate to be pursued for further development and potential implementation.

Hypothesis

H_0: Participants will not experience any differences with regards to usability between designs.

H_1: Participants will experience differences with regards to usability between designs.

Methodology

The participants were asked to sit at a desk, where they would find one of the five prototypes described above. After allowing them time to briefly examine the design, participants were given a task to complete using the interface. Because the prototypes have not yet been implemented, participants were asked to verbally explain how they would perform the task. For instance, if the experimenter asked them to set the spider's speed to a given value, the participants would describe how they would complete the task. The experimenter noted if the steps described by the participants were correct. A total of three tasks were given to each participant, and this process was repeated for all five prototypes. The order of presentation of the prototypes was counterbalanced across all participants. Once all five prototypes had been

presented, participants were asked to rank them based on their ease of use and how easy they were to understand.

Results

Based on the rankings provided by participants at the end of the experiment, a table containing the number of votes that each prototype earned as the most preferred design was created. It should be noted that the number of entries in this table (N = 16) does not reflect the number of participants (N = 12) that took part in this experiment. Some voted for two prototypes as their most preferred design, and therefore both of their selections were included.

Prototype	Chosen as preferred design
A	2
B	2
C	9
D	0
E	3

Table 2: Votes for each prototype as the preferred design in the first phase of the design process

Discussion

The results in the table above indicate that participants had a preference with regards to usability for one design. As such, the null hypothesis was rejected in favour of the alternate

hypothesis. Closer inspection of these results reveals that prototype D was the least preferred, having not been ranked first by a single participant. Participants commented that they were confused about the 3 different needles on each gauge. Furthermore, they often mixed the green and red needles for one another. For some, red meant that the player had died, and green meant that the player had made it to a checkpoint. However, the red needle was to be used to make the game more difficult (after a checkpoint) and the green needle was to be used to make the game easier (after a death). Due to the confusion that it caused, as well as to the lack of votes that it received as the preferred design, prototype D was excluded from further consideration in the design process.

Prototype A was another design that was viewed negatively. A majority of the participants commented that the tables used to demonstrate the pending changes took up too much space. Each one of the other designs were able to communicate the same information in much less space, and participants believed that some of the informative graphs found in some of the other designs would have been more useful. Furthermore, it was pointed out that the need to constantly click on the "+" or "-" arrows to arrive at the desired value was much more time consuming than the alternate designs. For these reasons, prototype A was excluded from further consideration.

Another design that failed to receive many votes as the most preferred was the prototype B. Despite having been one of the designs that participants found to be the most familiar, they said that it would work better if it was implemented in a physical manner. Although the idea to use the knobs covered several design principles discussed earlier, participants felt that a digital representations of the knobs simply wasn't the best method for making the modifications to the

parameters within the game. As such, this prototype was not included for further consideration in the design process.

The final design that was excluded from the next phase of the design process was prototype E. Most participants were confused as to how the pending modifications were to be made; they were unaware that the graph itself could be manipulated in order to make the changes. As a result, the experimenter was forced to give them a brief explanation, something that was typically not required for the other designs. While this prototype was viewed as the second most preferred design, it received three times fewer votes than the slider prototype. Furthermore, it should be noted that each of the three participants that voted for the dynamic graph design were participants that voted for two designs as their most preferred. Additionally, two of those three participants commented that if they had to choose between their two votes, the dynamic graph prototype was a "close second" to their other option, which was the slider prototype. For these reasons, the dynamic graph prototype was excluded from the next phase of the design process, and the slider prototype was chosen as the optimal candidate for further design and potential implementation.

Below is a table summarizing the pros and cons of each prototype created during this phase of the design process. While not every prototype received many votes as the preferred design, we can see that many of them contained elements that are ideal when designing an interface:

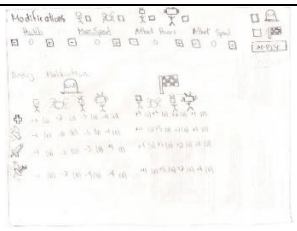
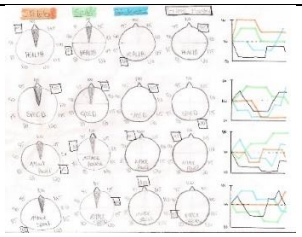
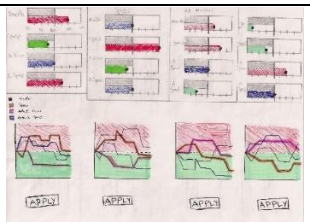
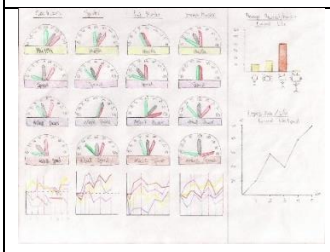
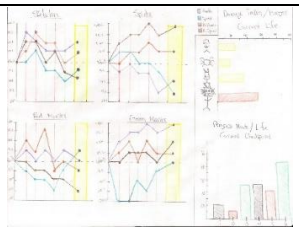
Prototype	Pros	Cons
	- Interface is easy to understand - Lacks any distracting visuals	- Process of making changes is time consuming - Tables showing pending changes take up too much space - Lacks informative graphs
	- Interface is familiar; takes advantage of mental models - Contains graphs reflecting previous changes made	- Lacks other informative graphs - Design ideal only if implementing physical version
	- Interface is easy to use/understand - Contains graphs reflecting previous changes made	- Impossible to set changes for both a death AND a checkpoint - Apply button an unnecessary step in process - Lacks other informative graphs
	- Contains informative graphs - Contains graphs reflecting previous changes made	- Interface was confusing to use - Design itself, while attractive, is cumbersome
	- Contains informative graphs - Contains graphs reflecting previous changes made - Changes made directly on graphs reflecting previous changes saves space/allows for bigger graphs	- Participants were unaware of how changes are carried out until told by experimenter - Impossible to set changes for both a death AND a checkpoint

Table 3: Pros and cons of each prototype from the first phase of the design process

This table demonstrates that while prototype C was chosen for the next phase of the design process, it lacked some of the ideal features found in other prototypes. Most notably, the final two prototypes in this table contained the “Progress Made/Life” and the “Damage Dealt/Monster” graphs, features that were not present in the slider prototype. During the next phase of the design process, prototypes were created that incorporated as many of the ideal features listed in the table above while also trying to limit the number of unfavourable features found in the cons column.

3.2.3 Designing the Interface: Phase Two

While prototype C was an overwhelming favourite during the first phase of the design process, it had several shortcomings. Users could only set one pending modification, rather than being able to set a modification in the event of either a death or a checkpoint. Also, participants commented that the “Apply” button felt like an unnecessary step in the process. With these suggestions in mind, and with the list of desirable features obtained after the previous phase of the design process, three more slider prototypes were created. The first such prototype can be seen below in Figure 14:

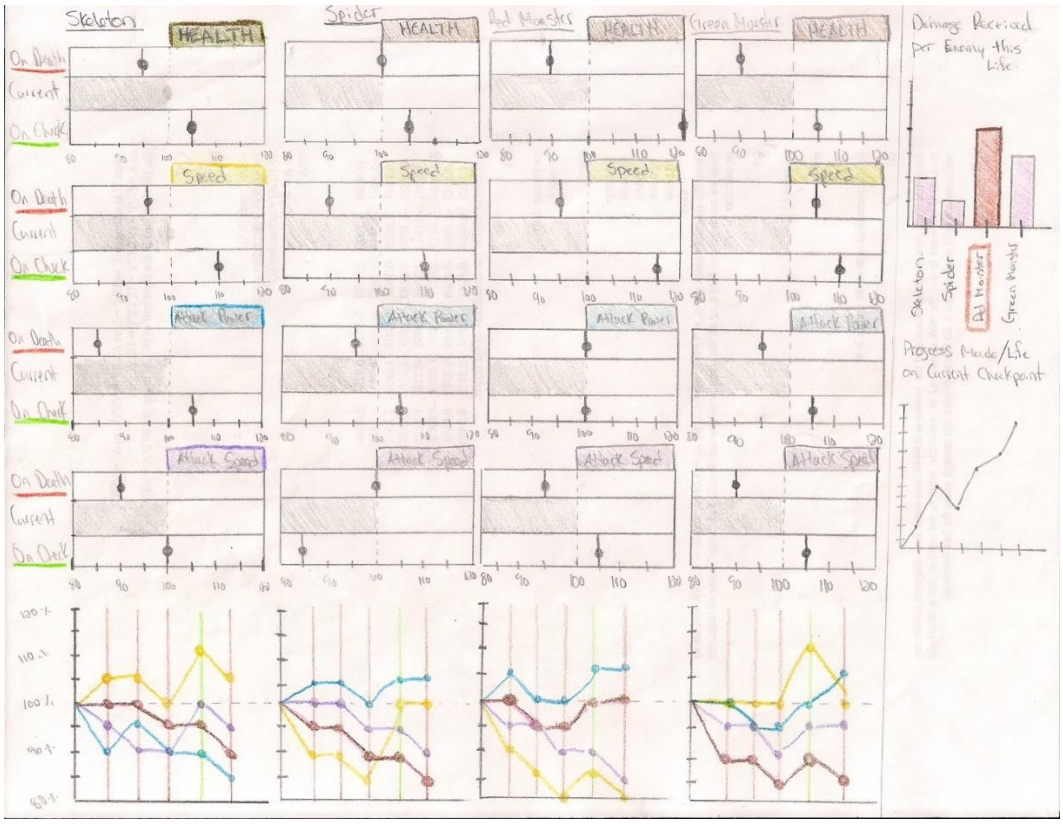


Figure 17: Prototype F

This design mimics the original slider prototype with regards to the grey bar used to demonstrate the current setting. However, unlike the original, an additional slider was added to this design in order to give users the opportunity to set pending modifications for both a death and a checkpoint. To the right of the prototype, the Progress Made/Life and Damage Received/Monster graphs were incorporated, while the graphs showing previous changes made to each parameter for each monster were included at the bottom.

The second new slider prototype can be seen below in Figure 15:

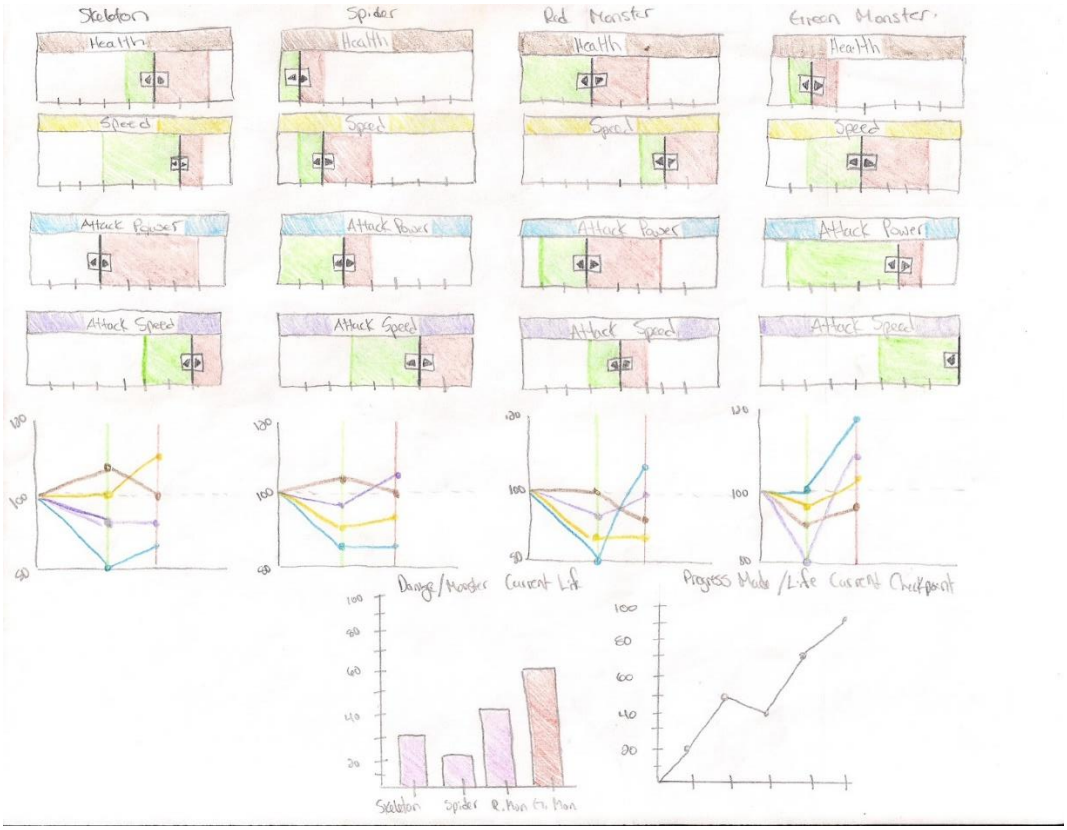


Figure 18: Prototype G

Similar to the previous design, this prototype incorporated the graphs providing information regarding previous changes, the Damage Dealt/Monster, and the Progress Made/Life. The main difference between this prototype and the former involves the sliders and how they operate. In prototype F, there were two sliders that could be manipulated. This design, however, makes use of what will be referred to as the "snapback" method. The current value is designated by the black line that runs through the slider. Two handles are attached to either side of that line. In order to set pending modifications, the user must click one of the handles and drag it to the desired setting. Upon letting go of the handle, it "snaps back" to its original position, while the space between the current value and the desired setting becomes red or green (red if it's making the game harder, green if it's making the game easier) to illustrate that a monsters'

parameter has a modification pending. The fact that the handles snap back to their original position allows the user to change the pending modification if they wish to do so.

The third and final slider prototype created during this phase of the design process can be seen in Figure 16:

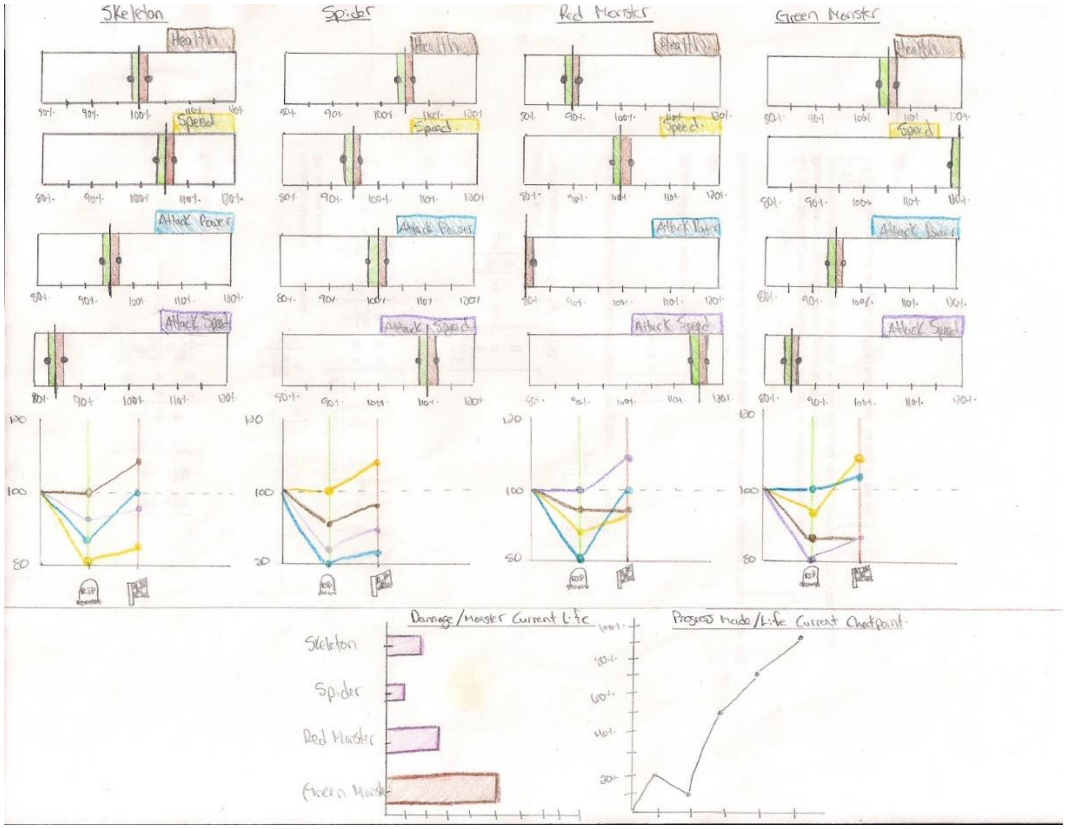


Figure 19: Prototype H

Once again, the Damage Dealt/Monster and the Progress Made/Life graphs were included in this design, along with the graphs showing the changes that were made to each of the monsters' parameters. Unlike prototype F, which has three sliders per parameter, this prototype only has one. While this is similar to the previous design, this one does not make use of the snapback feature. Instead, once the user clicks and drags the handles to the desired value, rather

than return to their original positions, the handles remain on the pending value. In order to change any of the pending modifications, the user would simply drag the handle to a new position.

These three prototypes were created to reflect the preferences of the test group from the first phase of the design process. Since the votes showed an overwhelming preference for prototype C, sliders were incorporated in each new design. Additionally, while the other designs were significantly less preferred, they contained aspects that participants found to be very useful. As a result, as many of those aspects as possible were incorporated in the new prototypes. In order to determine which new variant should be implemented as the interface for this experiment, a second experiment was conducted.

3.2.4 Evaluation of the Slider Prototypes: Selecting a Design for Implementation

Objective

The objective of this experiment is to determine which of the three slider prototypes created during this phase of the design process is preferred by users with regards to usability and how easy they are to understand. The preferred design will be the one implemented for use by our Wizard.

Hypothesis

H_0: Participants will not prefer any one design over the others with regards to usability.

H_1: Participants will prefer one design over the others with regards to usability.

Methodology

The same participants that were used during the experiment in section 3.2.2 were used for this experiment. They were given one of the three prototypes, as well as a series of tasks to be carried out. Once again, participants were asked to verbally explain how they would perform these tasks. The experimenter made note of their answers to make sure they were using the prototype correctly. This process was repeated for all three prototypes, with the order of presentation counterbalanced across participants. Similar to the previous experiment, the participants were asked to rank the prototypes based on their ease of use and how easy they were to understand.

Results

The number of votes each design received as the most preferred, according to the rankings provided at the end of the experiment, were tabulated and inserted into the table below:

Prototype	Chosen as preferred design
F	5
G	1
H	6

Table 4: Votes as the preferred design for the second phase of the design process

Discussion

The results from this experiment reveal that prototype G is the least preferred option of the three. The experimenter was required to give the participants a brief explanation of the snap back feature before they were able to complete the tasks. Although this explanation was quickly understood, a majority of participants felt that this feature was unnecessary. While this may have been the only negative feedback received regarding this design, the lack of votes this prototype received as the preferred design indicate that participants did not like for the snap back feature. However, because the vote did not reveal a clear preference for any one design, the null hypothesis was accepted; participants did not prefer any one design over the others with regards to usability.

Next, we see that there is nearly a tie in votes between the last two prototypes. In both cases, participants commented that it was evident how to make changes to a monster's parameters from the beginning, and that neither design required any explanation whatsoever. It was interesting, however, that there was such a disparity between prototype G and prototype H.

Given that the only difference between them was the snapback functionality, a similar number of votes for these two prototypes was expected. Provided that this was not the case, it suggests that—snapback feature aside—users prefer the single slider approach. Furthermore, it was mentioned that the sliders in prototype H took up less space on the interface than the ones in prototype F, which allowed for larger graphs that were easier to read. The table below summarizes the pros and cons of each design:

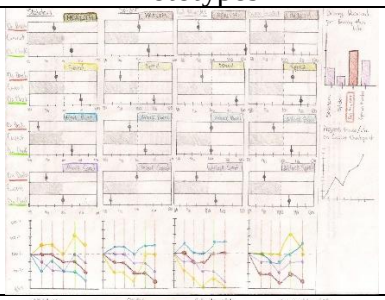
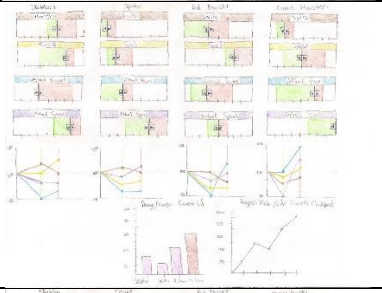
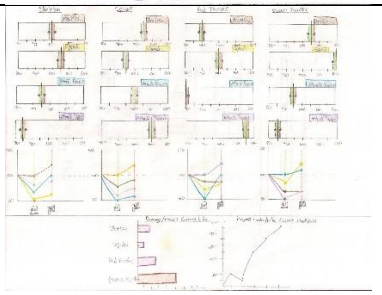
Prototypes	Pros	Cons
	- Easy to use/understand - Participants can make adjustments for deaths and for checkpoints	- Three sliders/bars per parameter take up significantly more space than the other designs - Feels cluttered
	- Two handles on single slider save space; allow for larger graphs - Participants can make adjustments for deaths and for checkpoints	- Snap back feature required some explanation - Participants did not like the snap back feature; felt it was unnecessary
	- Easy to use/understand - Participants can make adjustments for deaths and for checkpoints - Two handles on single slider save space; allow for larger graphs	

Table 5: Pros and cons of each design for the second phase of the design process

Examination of this table reveals that all three prototypes share several desirable traits, while also having very few flaws. As a result, the participants' rankings were used to determine which prototype would be implemented as the interface for the main experiment. Given that prototype G received the fewest number of votes, this design was not selected for implementation. The two other prototypes received nearly the same number of votes as the preferred design. While this is not a significant difference, participants commented that prototype F was too cluttered due to the amount of space taken up by the three sliders per parameter. Furthermore, prototype H failed to receive any negative feedback; as such, it was chosen as the interface for this experiment.

3.2.5 Implementation of the Interface

As mentioned in the section above, prototype H was chosen for implementation. The final product can be seen in the image below:

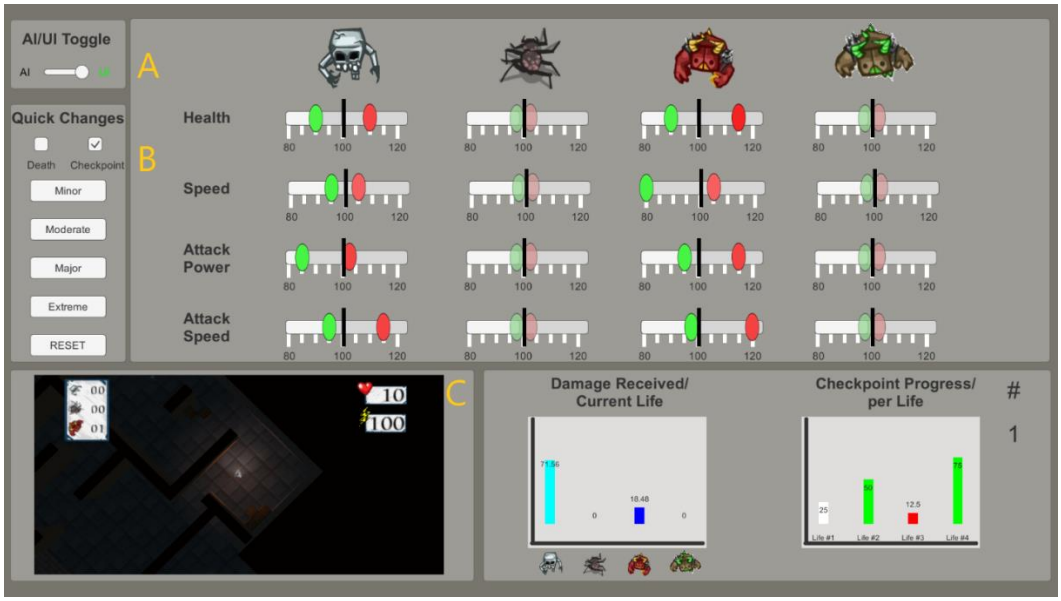


Figure 20: Implementation of prototype H

Although this interface was inspired by one of the designs created during the prototyping process, there are a few differences between the prototype itself and the final implementation. In the top left corner, there is a AI/UI switch (A), a component that was overlooked when creating the prototypes. Because the experiment calls for repeated measures, there needed to be a way to toggle between the AI making changes to the games' parameters and the Wizard fulfilling that same role. This switch allows the experimenter to make that change when necessary.

Below the switch, there is a "Quick Changes" panel (B). This was incorporated into the interface for the sake of speed. Because the pending modification are applied after a death or after a checkpoint, the Wizard must have the desired modifications in place before either of those events occur. Given that each monster has four parameters that can be altered, it was possible that the Wizard would have to adjust sixteen parameters in a limited amount of time. Furthermore, two different values could be set for the same parameter—to reflect changes to be made either on a death or on a checkpoint—meaning that the Wizard would in fact need to move thirty-two handles. Therefore, in order to compensate, the quick changes panel was added; allowing the Wizard to alter all sixteen parameters in as few as two mouse clicks.

First, they would choose either the death, the checkpoint, or both checkboxes. Then, they would click the button representing the degree of changes they wish to apply. The least impactful change that could be made, thanks to the "Minor" button, was a variance of five percent. Each successive button represented five percent increases in variance, with the "Extreme" button allowing a maximum of a twenty percent variance from the current setting. Although it was deemed unlikely that the Wizard would want to apply the same changes to each monster for every parameter, this feature was designed to reduce the total number of click and drags

necessary to make the desired modifications, as it is still possible to alter the pending modifications once they've been set.

Another difference between the final product and the prototype can be seen in the bottom left corner of the interface. Since the Wizard was located in a different room than our participants, it was impossible for them to know where the participants were in the game, and whether they were dying or making it to checkpoints. One of the options to resolve this issue would have been to modify the configuration of the lab so that the players' backs were facing the one-way mirror, allowing the Wizard to see their screen over their shoulders. However, this would inhibit the Wizard from seeing the participants' facial expressions. Therefore, a small portion of the interface was dedicated to creating a mirror image of the players' screen (C). One drawback from this solution is that four of the six graphs that were present in the prototype—more specifically those illustrating the changes made to each monsters' parameters over the course of the experiment—needed to be removed. With the inclusion of the "Progress Made/Current Checkpoint" and the "Damage Dealt/Monster" graphs, however, those four graphs were considered less essential; the Wizard was provided with enough feedback about how their changes were affecting the participant's performance.

Another aspect worth discussing about the implemented interface is the use of colours. As discussed in section 2.7, it's important to use colours carefully and sparingly. Consequently, the AI/UI switch uses green to highlight the option that is currently active, while the other remains black. This allows the user to discern at a glance which of the two has been enabled. Furthermore, most electronic devices use a green light to signal to their users that they are powered on. This was taken advantage of, as it gives the user a sense of familiarity, another one

of the design principles outlined in the literature. The next component wherein colours were incorporated are the handles on the sliders; they are red and green. These colours were chosen to give the user a hint as to which handle makes the game easier (green) and which one makes the game harder (red). Therefore, the association can be made that the game should be made easier when the player dies, or harder when the player makes it to a checkpoint, thus eliminating confusion as to which handle needs to be moved for either scenario. Finally, the bar graphs in the bottom right corner of the interface use colours to highlight important information contained within. For the "Damage Dealt/Monster" graph, three of the four bars are navy blue, while the fourth is a light blue. Because the light blue bar stands out from the rest, it designates the monster that has dealt the most damage to the user over the course of the current life, which suggests that this monster should be the one to have its parameters altered. In the second graph, the bars can be one of three colours: green, yellow or red. The colours are used to inform the Wizard about the impact of their most recent changes. If the right-most bar is green, it means the user has made more progress during the previous life compared to the life prior. A yellow bar means the same amount of progress was made, while a red bar means less progress was made. Although green bars could indicate that the game was made too easy, it could also mean that the player has learned the layout of the checkpoint, and has altered their strategy accordingly. The same can be said for a red bar; it could indicate that the game was made too hard, or it could mean that the user has become frustrated and is simply trying to make their way through the level using brute force, without any regard for the damage they are taking. Although not conclusive, the information provided from the colour of the bars in the graph—along with the

observations made by the user—can be used to determine what kind of changes should be made next.

The colour of the dashboard and panels themselves was also carefully considered. Perceptual Edge, a data visualization consultancy of which Stephen Few is the founder, has held competitions in the past wherein companies had to design the best possible dashboard to accommodate the data provided, which are then critiqued and used to highlighting various desirable aspects pertaining to dashboard design. One such aspect was the use of colours in a particular entry. Although he commended the entrants for their use of panels, he noted that the colours they chose for those panels would be distracting to the user; they were too bright, which subsequently nullified the effect of the significant data having been highlighted within. Therefore, using gray tones for this interface's background and panel colours allowed significant data to stand out and attract the Wizard's attention.

The next component of the interface worth discussing is the use of panels, as they were carefully used to achieve a specific effect. Few touches on this aspect by referencing Gestalt principles of visual perception. One such principle is that of enclosure, which dictates that any objects that are secluded or delimited from others are seen as a group. The AI/UI toggle can be found in its own panel because it does not serve the same functionality as any of the other parameters of the UI. The sixteen different sliders, however, are all grouped together because they serve the same purpose; to alter a given monster's parameters. One could argue that additional panels should be used to group the sliders belonging to each individual monster together, but another Gestalt principle was used to aid in this regard instead; the principle of proximity. The Skeleton's health slider, for instance, is closer in proximity to the slider below it

than to the slider beside it. Therefore, according to the aforementioned principle, a user's brain tends to see the sliders below the Skeleton's health slider as a group; it encourages their eyes to move from the top to the bottom as opposed to left and right. As such, the additional panels mentioned previously are not required, as the effect of keeping the sliders grouped by monster has already been achieved. Given that this panel is reserved for aspects of the UI that allow users to make modifications to the game's parameters, this raises the question of why the "Quick Changes" panel was separated from the sliders. However, it was decided to keep this panel secluded from the sliders to avoid confusing the user. As mentioned above, applying quick changes involves selecting one of two checkboxes and then pressing one of four buttons. These changes are applied to all monsters. If this panel was merged with the slider panel, it was possible that users might be inclined to manipulate the sliders in order to apply quick changes, which is not necessary.

An additional panel was used to group the graphs used to communicate information to the user. Clearly they do not belong with the sliders, as those are used to make adjustments; the graphs are simply used to monitor the impact of those changes on the player's performance. They could have been included in the same panel as the duplicate player's screen, but it was decided that the dashboard was more aesthetically pleasing when those components were kept separate from each other.

The type of graph that was used is another noticeable difference between the prototype and the implemented interface. Another key component of dashboard design highlighted by Stephen Few, the type of graph used can have an impact on how the information contained within is perceived by the user. He noted that only interval data should be displayed using a line

graph, while nominal and ordinal data is best viewed in the form of a bar graph. As such, it stands to reason that the "Damage Dealt per Monster/Current Life" graph be displayed using bars, as the information it displays is considered nominal in nature. The information contained in the "Progress Made per Life/Current Checkpoint" graph, on the other hand, is interval data, which should be viewed in the form of a line graph. Although this is the type of graph that was used in the prototype, Few points out that interval data is best viewed in the form of a bar graph when individual values are more pertinent to the user than the trends found within. This allows for comparison between adjacent values, which is more difficult to accomplish with line graphs. Since the goal of this graph is to compare two values—the progress made two lives ago to the progress made in the last life—trends are not the focus; hence the switch from a line graph to a bar graph during implementation.

Finally, another element worth discussing is the dual handled sliders. Although they remained consistent with the prototype, their implementation proved to a difficult task. Because the interface was implemented in Unity—an engine known mostly as a game making tool—the user interface tools are still under development. Simple items—such as buttons, checkboxes, and sliders—are readily available and easy to implement. However, when more complex tools were required, such as the graphs seen in the bottom right corner, for instance, users are left to their own devices with regards to their creation. In order to create the dual handled sliders, it was necessary to make use of two different sliders. The first is seen on the interface itself. The second, however, can only be seen in the editor, as it has been placed outside of the camera view. Its handle was then placed along the first slider. In order for the newly added handle to slide along its new slider without going past either end, the slider from which it was stolen had to have

the same X coordinates as the former. Then, through the use of scripts, boundaries were set to avoid a scenario where the handles would cross one another. Once the first dual-handled slider was created, a prefab was created, allowing for the implementation of the fifteen remaining sliders.

4 Experiment

The purpose of this experiment was to provide empirical support for the inclusion of biofeedback in the implementation of DDAs through the use of the WoZ method.

4.1 Methodology

4.1.1 Participants

A total of 20 participants (N = 11 males, N = 9 females) were recruited via direct solicitation. Participant's ages ranged between 19 and 30+ (participants were asked to circle their age range as opposed to listing their age).

4.1.2 Measures

Self-report data: The revised User Engagement Scale (UESz) was used to collect data from the participants following each of their two play sessions. This questionnaire is composed of 30 statements; participants must rate the degree to which they agree with each statement on a 5-point Likert scale, with 1 being "Strongly Disagree" and 5 being "Strongly Agree". These statements are then divided into four groups, representing the four subscales found within: focused attention, perceived usability, aesthetics and reward. The average of each subscale

computed, and an overall engagement score—ranging between four and twenty—is obtained by adding the sum of those four averages.

4.1.3 Procedure

A repeated-measure design was used, with 2 levels for the independent variable (AI vs Wizard) as well as one dependent self-report measure (UESz). Upon arrival at the lab, participants were placed in the testing room and asked to complete the consent form and demographics questionnaire. The layout of the lab can be seen in the Figure below:

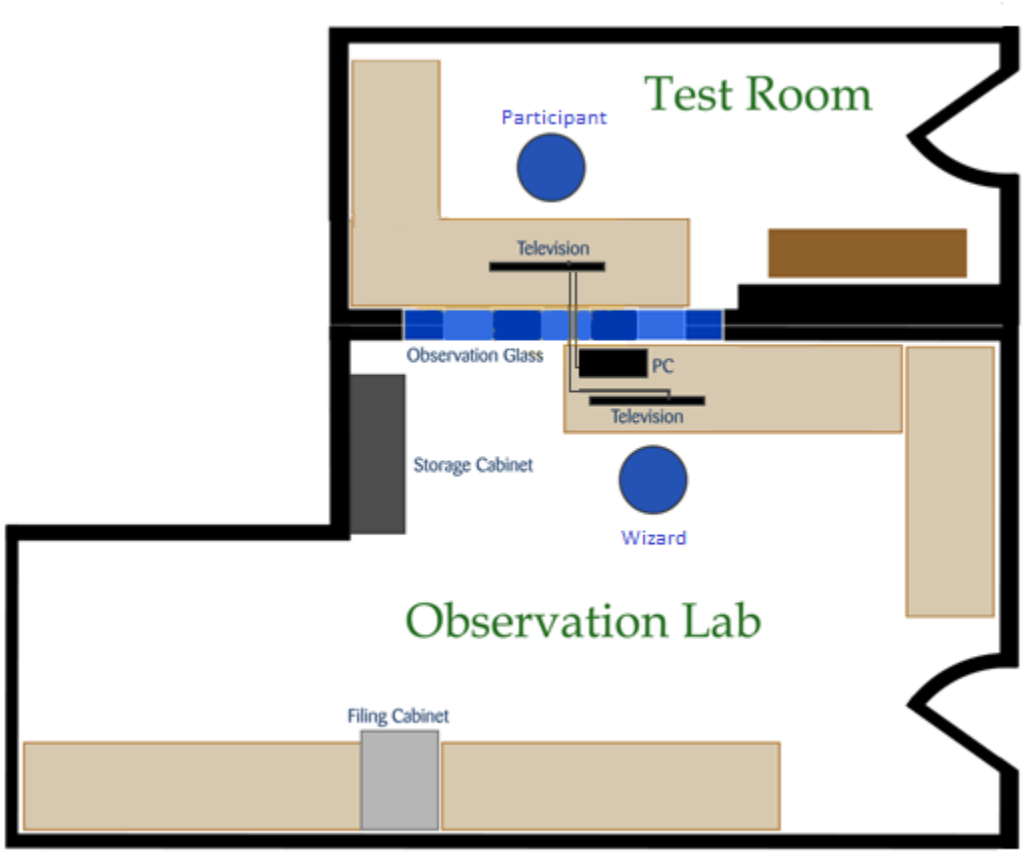


Figure 21: Lab setup

The setup above allowed the Wizard to observe the participant through one-way (observation) glass. This was essential, as the adjustments made by the Wizard were influenced by the participant's facial expressions and body language, as outlined in Appendix 1. During the AI trials, the Wizard was asked to sit at the table located to the bottom-left of the above Figure with their back facing the monitor. The Wizard would use this time to work on their homework or browse the internet; therefore, incapable of watching the participant or how they were performing.

The demographics questionnaire was used to gather information about their gaming habits, such as time spent playing per week, time spent playing per session and the level of difficulty they typically play on. After a brief description about the game they were about to play, participants were given the opportunity to play through the first checkpoint of the game in order to familiarize themselves with the button mapping. For this experiment, an Xbox 360 controller was used, and there were only three controls the participants needed to learn: how to move, how to attack and how to use the speed burst. Movement was controlled with the left joystick. Attacks, for their part, were controlled with the right joystick. The direction in which the user pointed the right stick dictated the direction in which the player's avatar swung its sword. Therefore, if an enemy was to the player's right, pushing the right joystick to the right resulted in the player's avatar swinging its sword at the enemy. Finally, the dash function was controlled using the right bumper. While moving, if the participants pressed this button, their avatar would jolt forward. This function was used primarily in order to evade attacks.

The game was played on a 23" Sony PlaySation™ 3D TV, with the 3D mode turned off. Figure 19 illustrates an example of what the participants saw on their display:

Figure 22: Sample screenshot of participant's display

Participants were told that in order for the door to the next room to open (A), they were required to kill the number of enemies listed in the top-left corner of the screen (B), and that there were more monsters present in each room than those listed in the requirements. They were also told that there were six rooms in total. In the last room—once they'd killed all the monsters listed in the top-left—a door would open from which the final boss would emerge, which they needed to kill in order to beat the game. In the top right of the screen, we can see the player's health (C), as well as their stamina (D). At the beginning of each new life, health is set to 100, which cannot be exceeded. The participants character (E) died if it's health reached 0. Stamina also begins at 100, and is slowly depleted any time the player attacks an enemy or uses the dash function described above. However, unlike the player's health, their stamina would replenish itself between attacks or between dashes. Participants were told to be careful not to abuse their stamina, as the lower it was when they attacked an enemy, the less damage they would do. A copy of the script that was recited to participants can be found in Appendix 2. Once participants

felt comfortable with the controls, the experimenter left the room and allowed the participants to play through the game in its entirety. As previously mentioned, the independent variable was counterbalanced; for half of the participants, the Wizard was asked to make adjustments to the game during the first play session, while for the other half, it was the AI making those changes. Upon completion, the experimenter returned to the room asked the participants to complete the UESz, provided in Appendix 3. After having completed the questionnaire, participants were asked to play through the game again. If the Wizard made changes to the game in the first play session, then the AI was tasked with this assignment during the second play session, and vice-versa. When they finished the game, participants were once again asked to fill out the UESz questionnaire.

4.2 Hypothesis

H_0: There will be no significant differences in overall UES scores between the Wizard trial and the AI trials of the experiment.

H_1: The UES scores for the Wizard trial will be significantly higher than the UES scores for the AI trial of the experiment.

4.3 Results

A paired-samples t-test was conducted to compare player engagement (UES scores) when the Wizard made modifications to the game and when the DDA made modifications to the game. We also conducted a paired-samples t-test on each of the four subscales that compose the User Engagement Scale to compare each facet of player engagement when the Wizard made modifications to the game and when the DDA made modifications to the game. The results can be seen in the table below:

Trial	Overall UES Scores	Aesthetics Subscale	Reward Subscale	Focused Attention Subscale	Perceived Usability Subscale
Wizard	M=14.3, SD=2.2	M=3.81, SD=0.89	M=3.66, SD=0.81	M=3.11, SD=0.88	M=3.61, SD=0.70
AI	M=14.6, SD=2.1	M=3.88, SD=0.82	M=3.79, SD=0.67	M=3.08, SD=0.84	M=4.0, SD=0.63
Summary	t(19)=1.08, p=0.293	t(19)=-0.47, p=0.647	t(19)=-1.31, p=0.206	t(19)=0.20, p=0.843	t(19)=-2.71, p=0.014

Table 6: UES Scores and subscale scores

Based on the results in the table above, there is no significant difference in overall engagement between the Wizard and AI trials. There is also no significant difference between the Wizard and AI trials for three of the four subscales that compose the UES: aesthetics, reward, and focused attention. There is, however, a significant difference between the Wizard and AI trials with regards to the perceived usability subscale. Participants experienced a significantly higher degree of perceived usability during the AI trial of the experiment.

Paired-samples t-tests were then conducted on each of the four modifiers used to alter the parameters of each of the four monsters found within the game to compare the level of difficulty when the Wizard made modifications to the game and when the DDA made modifications to the game. The results of those paired t-tests can be seen in the table below:

Monster/Trial	Health	Movement Speed	Attack Power	Attack Frequency
Skeleton Wizard	M=0.74, SD=0.25	M=0.72, SD=0.24	M=0.71, SD=0.26	M=0.72, SD=0.27
Skeleton AI	M=0.58, SD=0.24	M=0.59, SD=0.25	M=0.52, SD=1.92	M=0.58, SD=0.26
Summary	t(19)=3.42, p=0.003	t(19)=2.45, p=0.024	t(19)=3.85, p=0.001	t(19)=2.55, p=0.020
Spider Wizard	M=0.84, SD=0.21	M=0.83, SD=0.20	M=0.85, SD=0.19	M=0.85, SD=0.20
Spider AI	M=0.62, SD=0.26	M=0.59, SD=0.25	M=0.62, SD=0.26	M=0.50, SD=0.29
Summary	t(19)=5.36, p=0.001	t(19)=5.64, p=0.001	t(19)=5.51, p=0.001	t(19)=6.16, p=0.001
Red Monster Wizard	M=0.69, SD=0.27	M=0.71, SD=0.29	M=0.61, SD=0.27	M=0.63, SD=0.29
Red Monster AI	M=0.55, SD=0.24	M=0.55, SD=0.24	M=0.57, SD=0.26	M=0.62, SD=0.26
Summary	t(19)=3.13, p=0.006	t(19)=3.22, p=0.005	t(19)=0.735, p=0.472	t(19)=0.101, p=0.921
Green Monster Wizard	M=0.81, SD=0.28	M=0.81, SD=0.28	M=0.80, SD=0.29	M=0.79, SD=0.29
Green Monster AI	M=0.60, SD=0.27	M=0.59, SD=0.27	M=0.61, SD=0.27	M=0.62, SD=0.26
Summary	t(19)=3.53, p=0.002	t(19)=3.77, p=0.001	t(19)=3.53, p=0.002	t(19)=3.15, p=0.005

Table 7: Final settings for monster parameters

The results presented in the table above reveal that in the case of the skeleton, there was a significant difference between the Wizard and AI trials for each of the four modifiers. The

skeleton's health, movement speed, attack power and attack frequency were all significantly higher during the Wizard trial. With regards to the spider, the results found in the table above show that there was again a significant difference between the Wizard and AI trials for each of the four modifiers. During the Wizard trial, the spider's health, movement speed, attack power, and attack frequency were all significantly higher than during the AI phase. The results in the table above—with regards to the red monster—unveil that two of the modifiers were significantly higher during the Wizard trial: the health and movement speed. There were no significant differences between the Wizard and AI trials with regards to the attack power and attack frequency modifiers for the red monster, however. Finally, in the case of the green monster, the results from the table above reveal that there was a significant difference between the Wizard and AI trials for each of the four modifiers. In each instance, the Wizard's settings were significantly higher than the AI's settings.

A paired-samples t-test was also conducted to compare the length of time it took the participants to play through the game when the Wizard made modifications to the game and when the DDA made modifications to the game, followed by a paired-samples T-test to compare the number of times the participants died while the DDA made changes to the game and when the Wizard made changes to the game. The results of those t-tests can be found in the table below:

Trial	Play through Time	Number of Deaths
Wizard	M=1475.95, SD=536.63	M=14.35 SD=6.01
AI	M=1181.53, SD=419.09	M=10.95, SD=0.969
Summary	t(19)=2.25, p=0.037	t(19)=3.164, p=0.005

Table 8: Playthrough statistics

Observations of these results reveal that there was a significant difference between the Wizard and AI trials with regards to both the play through time and the number of deaths. During the Wizard trial, it took participants significantly more time to play through the game than during the AI trial. The same can be said for the number of deaths; participants died significantly more during the Wizard trial than during the AI trial.

5 Discussion

The User Engagement Scale was used to assess player engagement following each of the play sessions. As previously mentioned, this scale is comprised of four subscales; focused attention, perceived usability, aesthetics, and reward. Again, the average of each subscale is added to give an overall score between four and twenty. Results from the analysis show that there was no significant difference in overall participant engagement across both conditions of the experiment, meaning that participants were equally engaged regardless of whether the DDA or the Wizard was responsible for altering the level of difficulty within the game. However, upon analyzing each of the four subscales separately, it was revealed that there was a significant difference in Perceived Usability scores.

As defined in O'Brien, Cairns & Hall (2018), the Perceived Usability subscale is a representation of the negative affect resulting from the interaction with the game, the degree of control the users felt they had, and the effort required throughout the interaction. Results from the comparison between trials revealed that participants experienced a higher degree of perceived

usability when the DDA was making modifications to the game. In order to determine the cause, play through times between the DDA and Wizard trials were compared. This comparison revealed that participants completed the game significantly quicker (4 minutes and 54 seconds, on average) when the DDA was making modifications to the game. This lead to the evaluation of the difficulty that the game was being set to by both the DDA and the Wizard. As such, the settings for the four modifiers for each of the four monsters once the game had been completed were examined. Since there are four modifiers per monster, and four monsters in total, there were sixteen modifiers that could be altered by both the DDA and the Wizard. Of those sixteen modifiers, fourteen of the modifiers were found to be significantly more difficult during the Wizard trial, with the other two showing no significant difference. Subsequent evaluation of the data revealed that there was also a significant increase in total number of deaths for the Wizard trial, which further demonstrates that this phase of the experiment was significantly more difficult than the DDA trial. Therefore, the significantly lower scores in Perceived Usability during the Wizard trials were attributed in large part to the settings being more difficult during the Wizard trial.

While analyzing the results, several statements found within the UESz were deemed to address some of the key elements that are essential in order to achieve flow. This makes sense, as O'Brien & Toms (2010) created the UES based on their prior work regarding the process of engagement (2008), wherein they investigated flow theory and its impact on engagement. While the UES has since been revised (UESz), the elements found within remain the same; it was simply the subscales that had seen any revisions. With this in mind, a majority of the statements

in the PU subscale were thought to be used in order to assess many different elements of flow. One of the elements that is addressed in this subscale is that of balance between skill and challenge. The statements found within that are indicative of this include PU.1, “I felt frustrated while playing the game”, as well as PU.3, “I felt annoyed while playing the game”. Recall that, according to flow theory, feelings of frustration only occur if the level of challenge exceeds the player’s skill level. Therefore, given that the game was more difficult during the Wizard trial, it’s possible that players were not situated within the flow channel, thus leading to feelings of frustration and annoyance, which would have been reflected in these two statements.

There are more statements within the PU subscale that possibly address other elements of flow theory. The paradox of control, for one, is assessed thanks to statement PU.7, “I felt in control while playing the game”. Csikszentmihalyi (1990) states that sensations of control arise from the feeling that one can perfect their skills to the point of minimizing the margin of error to as close as zero as possible. However, during the Wizard trial, it seems that the level of difficulty exceeded the player’s skill level to such an extent that they could not achieve this feeling of being in control. As mentioned above, fourteen of the sixteen modifiers were significantly more difficult during the Wizard trial. Therefore, it stands to reason that participants would not have experienced the same sense of control during this portion of the experiment. This is further illustrated by the significant increase in completion time for this trial, as well as by the significant increase in total number of deaths during this trial.

Meanwhile, the merging of action and awareness is another element of flow that is addressed in the PU subscale. It is defined as “[the lack of] excess psychic energy to process any information but what the activity offer” (Csikszentmihalyi, 1990). In other words, if a person is

experiencing flow, they do not have the resources to question their actions, as all of their energy is being devoted to the task at hand. Statements such as PU.2, “I found this game confusing to play”, and PU.8, “I could not do some of the things I needed to do while playing the game”, were able to assess this element. Because certain participants commented that they wondered if the purpose of the experiment was to see how long they would tolerate the level of difficulty before they gave up, clearly this element of flow was not achieved during the Wizard trial.

Despite the fact that there was no significant difference in overall engagement scores between the two trials, a significant difference for the perceived usability subscale was uncovered. Several factors from this experiment that can account for this difference were identified, such as the increase in difficulty, the increase in the total number of deaths, as well as the increase in overall play through time. Upon further investigation of the subscales that compose the UES, it was established that several of the elements mentioned in Csikszentmihalyi’s description of flow were measured in this subscale, such as the balance between challenge and skill, the merging of awareness and action, as well as the sense of being in control. As mentioned above, several factors that can account for the significant difference in perceived usability between trials were found. However, not only do these results account for the said difference, but in combination with comments made by several participants, they suggest that participants did not experience flow during the Wizard trial. While not all elements of flow are assessed in the PU subscale, when the principle was discussed in Section 2.2, it was noted that each element of flow was dependent on one another. Therefore, since it was demonstrated that certain elements of flow were not achieved, and since the elements of flow are dependent on one another, the conclusion that flow was not achieved is valid.

6 Conclusion

In this work, we employed the Wizard of Oz method in order to determine if DDAs would benefit from having biofeedback from video-game players. In other words, we used the User Engagement Scale (revised, 2018) to determine if a Wizard could elicit higher levels of engagement than a DDA. The Wizard was given access to the same information made available to the DDA, which included the number of times the player had died on the current checkpoint, the amount of damage dealt to the player's character by each of the four enemies, as well as the amount of progress made by the player for each life over the course of the current checkpoint. Furthermore, through the use of colour-coding, the monster that dealt the most damage to the participants character, as well as the progress over the course of a given checkpoint relative to the previous life were made apparent to the Wizard. However, she was also able to watch the participants through one-way glass, meaning that she could see the participants' facial expressions and their body language. Given that the Wizard had access to this additional information, we hypothesized that she would be able to make more accurate estimations of the players emotional state, thus allowing her to make more appropriate modifications to the game, which in turn would have elicited higher levels of engagement than the DDA. Our results show, however, that this was not the case; there was in fact no significant difference in engagement between the two trials. As such, the null hypothesis was accepted. Nevertheless, the results demonstrated that participants experienced a higher degree of perceived usability when the DDA was making modifications to the game. We concluded that this difference was due to a variety of factors, including the significantly increased playthrough time for the Wizard trial, the significant increase in average death totals for the Wizard trial, and the fourteen out of sixteen modifiers that

were significantly more elevated during the Wizard trial, all of which indicate that the Wizard was unable to accurately scale the difficulty of the game to the participants skill level.

Future work on this matter would include repeating the experiment with several different Wizards. Individual differences exist for virtually every facet of human characteristics, which includes a person's ability to read another person's body language and facial expressions. As our experiment relied heavily on the Wizard's ability to do just that, it's conceivable that the results do not reflect the DDA's superiority over the Wizard at tailoring the game's level of difficulty to the player's skill level. Rather, it could indicate that our Wizard lacked the required skill with regards to reading body language and facial cues, and it's entirely possible that an expert in this field could have elicited higher levels of engagement than the DDA. Furthermore, if the experiment were to be repeated, we suggest that heart-rate or EEG data be used rather than facial expressions and body language. While research with regards to the use of psychophysiological data suggests that it's difficult to pinpoint the psychological cause for changes in physiological readings, we would be curious to see how a Wizard would respond to such changes.

References

1. Alexander, J., Sear, J. & Oikonomou, A. (2013). An investigation of the effects of game difficulty on player enjoyment. *Entertainment Computing, 4*, 53-62.

2. Agosti, M., Orio, N. & Ponchia, C. (2018). Promoting user engagement with digital cultural heritage collections. *International Journal on Digital Libraries*, 19(4), 353-366.

3. Andrade, G., Ramalho, G., Gomes, A. & Corruble. (2006). Dynamic game balancing: an evaluation of user satisfaction. In *Proceedings of the Second AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment.* pp. 3-8. Pittsburgh, PA: AAAI Press

4. Ang, D. (2017). Difficulty in video games: Understanding the effects of dynamic difficulty adjustment in video games on player experience. In *Proceedings of the 2017 ACM SIGCHI Conference on Creativity and Cognition*, pp.544-550. New York, NY: ACM

5. Ang, D. & Mitchell, A. (2017). Comparing effects of dynamic difficulty adjustment systems on video game experience. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, pp.317-327. New York, NY: ACM

6. Baddeley, A. (2003). Working memory: Looking back and looking forward. *Nature Reviews Neuroscience*, 4, 829-839.

7. Bakkes, S., Whiteson, S., Li, G., Viniuc, G. V., Charitos, E., Heijne, N., & Swellengrebel, A. (2014). Challenge Balancing for Personalised Game Spaces. In *2014 IEEE Games Media Entertainment (GEM)*: 22-24, pp. 10-17. Piscataway, NJ: IEEE. DOI: 10.1109/GEM.2014.7047971

8. Bernsen, N., Dybkjaer, H. & Dybkjaer, L. (1994). Wizard of Oz prototyping: When and how? *Proc. CCI Working Papers Cognit. Sci./HCI*, Roskilde, Denmark.

9. Boyle, E., Connolly, T., Hainey, T. & Boyle, J. (2012). Engagement in digital entertainment games: A systematic review. *Computer in Human Behavior, 28*, 771-780. doi: https://doi.org/10.1016/j.chb.2011.11.020

10. Brener, L. (1940). An experimental investigation of memory span. *Journal of Experimental Psychology.* 21, 271-282.

11. *Canada: How many hours per week do you spend playing video/computer games?* (2018, August). Retrieved from https://statista.com

12. Cass, S. (2002). Mind Game. *IEEE Spectrum, 39* (12), 40-44.

13. Caroux, L., Isbister, K., Le Bigot, L. & Vibert, N. (2015). Player-video game interaction: A systematic review of current concepts. *Computer in Human Behavior, 48,* 366-381. https://doi.org/10.1016/j.chb.2015.01.066

14. Christou, G. (2014). The interplay between immersion and appeal in video games (2014). *Computers in Human Behavior, 32,* 92-100. https://doi.org/10.1016/j.chb.2013.11.018

15. Chung, S., Kramer, T. & Wong, E. (2018). Do touch interface users feel more engaged? The impact of input device type on online shoppers' engagement, affect, and purchase decisions. *Psychology & Marketing*, 35(11), 795-806.

16. Clarke, D., & Duimering, P. R. (2006). How computer gamers experience the game situation: a behavioral study. *Computers in Entertainment*, 4(3), 6. http://doi.org/http://doi.acm.org/10.1145/1146816.1146827

17. Csikszentmihalyi, M. (1990). *Flow: The psychology of optimal experience.* New York, NY: Harper & Row.

18. Csikszentmihalyi, M. (2014). *Flow and the foundations of positive psychology: The collected works of Mihaly Csikszentmihalyi.* Dordrecht, NL: Springer

19. Davis, A. (1992). Operational prototyping: A new development approach. *IEEE Software, 9*(5), 70-78. DOI: 10.1109/52.156899

20. Dekker, A. & Champion, E. (2007). Please biofeed the zombies: Enhancing gameplay and display of a horror game using biofeedback. *DiGRA Conference*.

21. Demediuk, S., Raffe, W. & Li, X. (2016). An adaptive training framework for increasing player proficiency in games and simulations. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*, pp 125-131. New York, NY: ACM

22. Demediuk, S., Tamassia, M., Raffe, W., Zambetta, F., Mueller, F. & Li, X. (2018). Measuring player skill using dynamic difficulty adjustment. In *Proceedings of the Australasian Computer Science Week Multiconference*, New York, NY: ACM

23. *Do you ever play games at home or elsewhere in any of these ways?* (2018, April). Retrieved from https://statista.com

24. Elkin, A. (2012). Adaptive Game AI and Video Game Enjoyability. Retrieved from semanticscholar.org

25. *Essential facts about the computer and video game industry.* [White paper]. (2018). Retrieved from theesa.com

26. Few, S. (2006). *Information dashboard design: The effective visual communication of data*. Sebastopol, CA: O'Reilly.

27. Few, S. (2006, November). Dashboard design for rich and rapid monitoring. *Visual Business Intelligence Newsletter*. Retrieved from https://perceptualedge.com

28. Few, S. (2007, January). Pervasive hurdles to effective dashboard design. *Visual Business Intelligence Newsletter*. Retrieved from https://perceptual edge.com

29. Few, S. (2007). *Dashboard design for real-time situation awareness.* [White Paper]. Retrieved from https://www.perceptualedge.com/articles/Whitepapers/Dashboard_Design.pdf

30. Few, S (2012). *Show me the numbers: Designing tables and graphs to enlighten*. El Dorado Hills, CA: Analytics Press

31. Fraser, N. & Gilbert, N. (1991). Simulating speech systems. *Computer Speech and Language, 32,* 81-99.

32. Giannakos, M., Jaccheri, L. & Krogstie (2015). How video usage styles affect student engagement? Implications for video-based learning environments. *State-of-the-art and Future Directions of Smart Learning*, 157-163.

33. Green, P. & Wei-Haas, L. (1985). The Wizard of Oz: A tool for rapid development of user interfaces. *University of Michigan, Tech. Rep.* UMTRI-85-27.

34. Gunapati, S. (2011, 10). Key features for designing a dashboard. *Government Finance Review (0883-7856)*, 27, 47-50. Retrieved from www.gfoa.org

35. Hebb, D. (1949). *The organization of behaviour: A neuropsychological theory*. New York, NY: John Wiley & Sons

36. Hoefer, R. & Twis, M. (2018). Engagement techniques by human services nonprofits : A research note examining website best practices. *Nonprofit Management and Leadership*, 1-11

37. Hunicke, R. (2005). The case for dynamic difficulty adjustment in games. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, pp. 429-433. New York, NY: ACM

38. Hunicke, R. & Chapman, V. (2004). AI for dynamic difficulty adjustment in games. *Challenges in game artificial intelligence* (AAAI Workshop). Pittsburgh: AAAI Press.

39. Janes, A., Sillitti, A & Succi, G. (2013). Effective dashboard design. *Cutter IT*, 26(1), 17-24.

40. Johnson, D. & Wiles, J. (2001). Computer games with intelligence. In *2001 IEEE International Fuzzy Systems Conference Proceedings.* (pp. 1355-1358). New York, NY: IEEE

41. Kelley, J. (1984). An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Office Information Systems, 2*(1), 26-41.

42. Kelley, J-P., Botea, A. & Koenig, S. (2008). Offline planning with hierarchical task networks in video games. In *Proceedings of the Fourth Artificial Intelligence and Interactive Digital Entertainement Conference*. (pp.60-65). Cambridge, MA: MIT Press

43. Lidwell, W., Holden, K. & Butler, J. (2003). *Universal principles of design: 100 ways to enhance usability, influence perception, increase appeal, make better design decisions, and teach through design*. Gloucester, MA: Rockport

44. Lim C., Baumgarten R., Colton S. (2010) Evolving Behaviour Trees for the Commercial Game DEFCON. In: Di Chio C. et al. (Eds.), *Applications of Evolutionary Computation. EvoApplications*. (pp. 100-110). Berlin, Germany: Springer

45. Millington, I. & Funge, J. (2009). *Artificial intelligence for games.* Burlington, MA: Elsevier

46. Missura, O. & Gärtner, T. (2009). Player modeling for intelligent difficulty adjustment. In Gama, J., Costa, V., Jorge, A. & Brazdil, P (Eds.), *12th International Conference of Discovery Science* (pp. 197-211). Berlin, Germany: Springer

47. Nakamura, J. & Csikszentmihalyi, M. (2014). The Concept of Flow. In: *Flow and the Foundations of Positive Psychology*. Dordrecht, Netherlands: Springer.

48. Norman, D. (1988). *The design of everyday things*. New York, NY: Doubleday.

49. O'Brien, H., Cairns, P. & Hall, M. (2018). A practical approach to measuring user engagement with the refined user engagement scale (UES) and new UES short form. *International Journal of Human-Computer Studies, 112*, 28-39. https://doi.org/10.1016/j.ijhcs.2018.01.004

50. O'Brien, H. & Toms, E. (2008). What is user engagement? A conceptual framework for defining user engagement with technology. *Journal of the American Society for Information Science and Technology, 59*(6), 938-955. doi: 10.1002/asi

51. O'Brien, H. & Toms, E. (2010). The development and evaluation of a survey to measure user engagement. *Journal of the American Society for Information Science and Technology, 61*(1), 50-69. doi: 10.1002/asi.21229

52. Perez, D., Nicolau, M., O'Neill, M. & Brabazon, A. (2011). Evolving behavior trees for the Mario AI competition using grammatical evolution. In: Di Chio C. et al. (Eds.), *Applications of Evolutionary Computation. EvoApplications*. (pp. 123-132). Berlin, Germany: Springer

53. Procci, K., Bowers, C., Jentsch, F., Sims, V. & McDaniel, R. (2018). The Revised Game Engagement Model: Capturing the subjective gameplay experience. *Entertainment Computing, 27,* 157-169. https://doi.org/10.1016/j.entcom.2018.06.001

54. Rani, P., Sarkar, N. & Liu, C. (2005). Maintaining optimal challenge in computer games through real-time physiological feedback. In: *Operational and Virtual Environments, Foundations of Augmented Cognition*. Mahwah, NJ: Lawrence Erlbaum Associates Publishers

55. Riek, L. (2012). Wizard of Oz studies in HRI: A systematic Review and new reporting guidelines. *Journal of Human-Robot Interaction, 1*(1), 119-136. doi: 10.5898/JHRI.1.1.Riek

56. Russell, S. & Norvig, P. (2010). *Artificial intelligence: A modern approach*. Upper Saddle River, NJ: Prentice Hall

57. Shaker, N., Togelius, J. & Nelson, M. (2016). *Procedural Content Generation in Games*. New York, NY: Springer

58. Sharp, H., Rogers, Y., & Preece, J. (2007). *Interaction design: Beyond human-computer interaction*. Chichester: Wiley.

59. Silva, M., Silva, V. & Chaimowicz, L. (2015). Dynamic difficulty adjustment through an adaptive AI. In *Proceedings of SBGames 2015*. pp.52-59. Piscataway, NJ: IEEE

60. Thalmann, M., Souza, A. S., & Oberauer, K. (2018). How Does Chunking Help Working Memory? *Journal of Experimental Psychology: Learning, Memory, and Cognition*. Advance online publication. http://dx.doi.org/10.1037/xlm0000578

61. *The truth is out there: Knowing the real addressable universe is crucial to TV buys*. (2018, February 1st). Retrieved from https://nielsen.com

62. *The state of online gaming – 2018* [White paper]. (2018). Retrieved from https://limelight.com

63. Tizkar, A. & Tabatabaei, N. (2009). Rapid Prototyping for Software Projects with User Interface. *Scientific Bulletin of University of PITESTI, Electronics and Computer Science Series*. 2. 85.

64. Togelius, J., Kastbjerg, E., Schedl, D. & Yannakakis, G. (2011). What is procedural content generation? Mario on the borderline. In *Proceedings of the 2nd International Workshop on Procedural Content Generation in Games*. New York, NY: ACM

65. Tracy, A. (2016, March 13th). *Two thirds of video gamers prefer to play alone*. Retrieved from https://forbes.com

66. Vallieres, D. (2017). *Achieving flow in gameplay through dynamic difficulty adjustment system*. (undergraduate thesis). Laurentian University, Sudbury, ON, Canada

67. Vijayan, J. & Raju, G. (2011). A new approach to requirements elicitation using paper prototype. *International Journal of Advanced Science and Technology, 28*, 9-16

68. Wiebe, E., Lamb, A., Hardy, M. & Sharek, D. (2014). Measuring engagement in video game-based environments: Investigation of the User Engagement Scale. *Computers in Human Behavior, 32*, 123-132. https://doi.org/10.1016/j.chb.2013.12.001

69. Xue, S., Wu, M., Kolen, J., Aghdaie, N. & Zaman, K. (2017). Dynamic difficulty adjustment for maximized engagement in digital games. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pp. 465-471. New York, NY: ACM

70. Yang, M. & Epstein, D (2005). A study of prototypes, design activity, and design outcome. *Design Studies, 26*, 649-669

71. Yannakakis, G. & Togelius, J. (2018). *Artificial intelligence and games*. New York, NY: Springer

72. Yun, C., Shastri, D., Pavlidis, I. & Deng, Z. (2009). O' game, can you feel my frustration?: Improving user's gaming experience via StressCam. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2195-2204

73. Yun, C., Trevino, P., Holtkamp, W. & Deng, Z. (2010). PADS: Enhancing gaming experience using profile-based adaptive difficulty system. In *Proceedings of the 5th ACM SIGGRAPH Symposium on Video Games*, pp 31-36. New York, NY: ACM

Appendices

Appendix 1: Wizard statement regarding how she determined what changes were necessary

Checkpoint changes

If a player got through a level, despite their health level, in one life, I would make the game a "minor" step more difficult. If a player got through multiple levels in one life, with more than fifty percent of their health remaining, I would make the game a "moderate" step more difficult to all monsters. If a player made it through the level with little (2-3) lives lost but it seemed like they breezed through it or took little damage (more than 70-80 percent of health remaining), I would make whichever monsters who caused them low damage 5-10 points higher.

At death changes

On the first death of a level, normally, I did not make any changes since most of the time the player was going in with less than full health. Most of the time a player died, I would make a change. The first few times they died I would make "minor" changes. I would than judge what monsters are causing them more trouble and make 10-20-point adjustments to those. If they continued to die, I would then make "minor" or "moderate" adjustments based on how close the points were to 0 (so I still had room to make changes in further levels without the points going negative and therefore, giving points to the player) to either all monsters if they were all giving them a hard time or just to the monsters that were being difficult. At the final level, with the final boss, if major changes (> 20 points) have already been done to all other monsters, equivalent changes would be done as soon as possible to the final boss. Lastly, if the player showed signs of frustration, I would make a "minor change, if I was not already planning on doing so. If I was already planning on making a minor change, I would then make a moderate change.

Appendix 2: Script recited to participants throughout experiment

Good morning/afternoon.

Thank you for taking the time to participate in my study.

We're going to start off by filling out the consent form. I'm going to ask you to read it thoroughly, and then sign and date it once you are done. If you have any questions, do not hesitate to ask. It's Laurentian University policy that I provide you with a copy of the consent form, so I have provided you with a second copy to keep.

Now that you have completed the consent form, I'm going to ask you to fill out a brief questionnaire. This is a simple demographics questionnaire, asking for some personal information such as age and handedness, as well as information regarding your video-game playing habits. Thanks.

This game was developed last year by an undergraduate student here at Laurentian. In a nutshell, you will be controlling a character that needs to make its way through a series of rooms. Each room will be filled with enemies, and in order to gain access to the next room, you must defeat a certain number of them (they are listed in the top left corner of your screen). For this, you will be using an Xbox 360 controller. In order to move the character, you use the left joystick, and in order to attack the enemies, you point the right joystick in their direction. Keep in mind that you are controlling a knight that uses a sword. This means that you must be close enough to the enemies when you attack in order to inflict damage. Furthermore, the knight must have enough stamina to swing the sword around. Each time you attack, you'll lose some stamina. However, it does replenish itself over time. Your health and stamina are located in the top right corner of the screen. Lastly, you have the ability to dash using the right bumper. This essentially shoots your character a short distance in the direction you're travelling. However, this takes a considerable amount of your stamina.

I'm going to launch the game from the adjacent room and let you have a quick trial run to get used to the controls. I'll come back to answer any questions that you might have, and when you feel ready, we'll start playing the game.

Go to adjacent room, watch participant through entirety of first play session

Alright, how was that?

We're going to take a little break. What I want you to do now is fill out this brief questionnaire. The questions are simply looking to assess your experience while playing the game. Circle the option that best applies. Thanks.

We're now going to play through the game one more time.

Go to adjacent room, watch participant through entirety of second play session

So, how was that?

I'm just going to as you to fill out the same questionnaire as before.

Debrieffing
Now that we have concluded the experiment, I must admit to you that you were "duped". During one of the test sessions, an Artificial Intelligence was adjusting the level of difficulty of the game based on your performance. However, during the other test session, my lab assistant was watching you play through the one-way glass. Based on your performance and your body language, she was able to make adjustments to the level of difficulty of the game. The reason behind this is that I'm actually trying to see if Artificial Intelligence would benefit from having information such as your body language or facial expressions fed into it in order to help it make adjustments to the game. Thank you so much for your time today, and have a great day.

Appendix 3: User Engagement Scale (revised)

1. When I was playing the game, I lost track of the world around me

Not at all		Somewhat		Completely
1	2	3	4	5

2. I blocked out things around me when I was playing the game

Not at all		Somewhat		Completely
1	2	3	4	5

3. The time I spent playing the game just slipped away

Not at all		Somewhat		Completely
1	2	3	4	5

4. I was absorbed in my gaming task

Not at all		Somewhat		Completely
1	2	3	4	5

5. I was so involved in my gaming task that I lost track of time

Not at all		Somewhat		Completely
1	2	3	4	5

6. During this gaming experience I let myself go

Not at all		Somewhat		Completely
1	2	3	4	5

7. I lost myself in this gaming experience

Not at all		Somewhat		Completely
1	2	3	4	5

8. I was really drawn into my gaming task

Not at all		Somewhat		Completely
1	2	3	4	5

9. I felt discouraged while playing the game

Not at all		Somewhat		Completely
1	2	3	4	5

10. I felt annoyed while playing the game

Not at all		Somewhat		Completely
1	2	3	4	5

11. I found the game confusing to play

Not at all		Somewhat		Completely
1	2	3	4	5

12. I felt frustrated while playing the game

Not at all		Somewhat		Completely
1	2	3	4	5

13. I could not do some of the things I needed to do while playing the game

Not at all		Somewhat		Completely
1	2	3	4	5

14. The gaming experience was demanding

Not at all		Somewhat		Completely
1	2	3	4	5

15. This gaming experience did not work out the way I had planned

Not at all		Somewhat		Completely
1	2	3	4	5

16. I liked the graphics used for the game

Not at all		Somewhat		Completely
1	2	3	4	5

17. The game appealed to my visual senses

Not at all		Somewhat		Completely
1	2	3	4	5

18. The game was aesthetically pleasing

Not at all		Somewhat		Completely
1	2	3	4	5

19. The screen layout of the game was visually pleasing

Not at all		Somewhat		Completely
1	2	3	4	5

20. The game was attractive

Not at all		Somewhat		Completely
1	2	3	4	5

21. The content of the game incited my curiosity

Not at all		Somewhat		Completely
1	2	3	4	5

22. I would continue to play this game out of curiosity

Not at all		Somewhat		Completely
1	2	3	4	5

23. I would recommend this game to my family and friends

Not at all		Somewhat		Completely
1	2	3	4	5

24. Playing this game was worthwhile

Not at all		Somewhat		Completely
1	2	3	4	5

25. I felt interested in my gaming task

Not at all		Somewhat		Completely
1	2	3	4	5

26. My gaming experience was rewarding

Not at all		Somewhat		Completely
1	2	3	4	5

27. The gaming experience was fun

Not at all		Somewhat		Completely
1	2	3	4	5

28. Rate the level of challenge this game presented you with

Very easy	Easy	Moderate	Difficult	Extreme

29. Rate your level of enjoyment while playing this game on a scale of 1-5, 1 being "Completely bored" and 5 being "Very entertaining"

1	2	3	4	5