Dynamic Difficulty Adjustment In Games Using Physiology

Casper Gade Kranker Vollmers

Master Thesis

Medialogy

Aalborg University Copenhagen

June 2018

Indhold

1. Introduction

The video game industry has evolved exponentially during the last couple of decades into a multimillion industry, and as the technology to build video games improves more and more, so does the rules and requirements for making good games. Video game companies have to appeal to a large group of people to sell products, and they have to make games that are equally entertaining for each consumer, which brings very different challenges when it comes to game design. A game can be boring when it is too easy, but a game can also be frustrating when it is too hard. Implementing a difficulty system (Hunicke, R.) to automatically enhance the player's performance will in turn enhance the player's overall experience with the game.

However, these difficulty adjustment systems are often linear, which may leave players wanting a different kind of difficulty adjustment. The linear set of difficulty systems are often designed to fit the skills of "the average player", but players often deviate from the idea of an average player. This means, that the linear difficulty system will often respond very differently to the individual player. Either the game becomes too easy or too hard to the player. There is high interest to appeal to large demographics in the video game (*Um, Sang-Won et. al., 2007*), as this means that they will sell more units of video game products. Difficulty levels may lose potential comsumers' interest in playing a game if it is too hard for them to play.

This issue is elaborated by the concept of "Dynamic Difficulty Adjustment" (DDA) (*Um, Sang-Won et. al., 2007*). DDA is an in-game system that takes the player's performance as a data input and alter the game's difficulty in different ways by changing specifically selected parameters. Using this kind of system will allow for a more stable difficulty system, which will respond to the individual player's skills and performance within a game. Instead of building the system on the player's skills and performance, it should instead take the player's frustration as a data input and provide a better experience to the player.

2. Problem Investigation

Many video game researchers today has a great focus in implementing adaptive video games, which takes various factors into account, such as player motivation, experience and skill. This is because both game developers and publishers want to appeal to as broad a demographic as possible by offering the best experience in the video game industry that today's technology can bring. And one of the ways to bring a good experience for all people is to make games fair and balanced for everyone (*Gilleade, 2004; Um, Sang-Won et. al., 2007*).

The games that do not adapt to players' motivation, experience and skills, will face a challenge on the market, since they are tailored to a narrow and more specific audience. This means that some players may feel alienated by a game, whose concept was initially found appealing to play, though turned out differently. They may find the game too easy or too hard, to keep the interest of playing. Most games have adapted to these circumstances, by implementing a range of pre-set difficulty settings, such as 'easy', 'normal', and 'hard', yet these difficulty settings are still fixed and do not adapt during a game session. This solution neglects that players are dynamic entities. In this way many players will feel confused, unfairly challenged or underestimated during play, making the game lose its appeal for the individual player (*Gilleade, 2004*).

Looking into game difficulty, and how to design and manipulate it this way, became a particular area of interest. Researching this area may provide insight to what could be interesting and may help this project build upon current methods. Not only in regards to game design, but also in regards to commercial appeal of videogames and continuously enjoyable gameplay experience based on a more personal involvement. This begs the question, which will be referred to as the *initial problem statement* (IPS):

"How can a difficulty system be implemented to automatically adjust a game's difficulty and enhance the player's experience?"

3. Analysis

Now that the IPS has been set up, the research has to provide a clear understanding and in-depth look of the DDA system and what causes frustration within a game.
With the DDA system, the goal of this project is to design and implement a product that can eliminate any sources of frustration within a game and improve the overall experience. However, before this can be set into motion, a proper research has to be conducted in order to establish a proper test.
The find out how to develop a DDA system for a game that takes the player's physiological data as an input and uses it to automate the difficulty adjustment, the research has to look into how a player can do so. To do this, this chapter will look into how the difficulty of a game should be a part of the initial game design and implemented into the game itself. Therefore, it is important to look into what makes a game hard and challenging as the first stop, and how these parameters can be adjusted before any DDA system can be implemented.

As the goal with this project is to adapt the game's difficulty based on the player's frustration, this project will also look into theoretical knowledge of game – and level design in order to design a proper game for the initial testing.

3.1 Galvanic Skin Response

Galvanic Skin Response (GSR) is the measurement of variations in the electrical characteristics of the skin, like the conductance, caused by the human body sweating. The amount of sweat secretion from sweat glands varies from each area of the body. The homunculus model suggests that parts of the body more dedicated to motoric functions are more sensitive to sensation than other (Neil, 2013). Therefore, sweat secretion is higher in the palms than other areas of the body. GSR is an optimal indicator of a person's emotional state. It is also more reliable compared to other methods of measuring a person's emotional state, like heart rate, which to some degree can be controlled, be heavy breathing for instance.

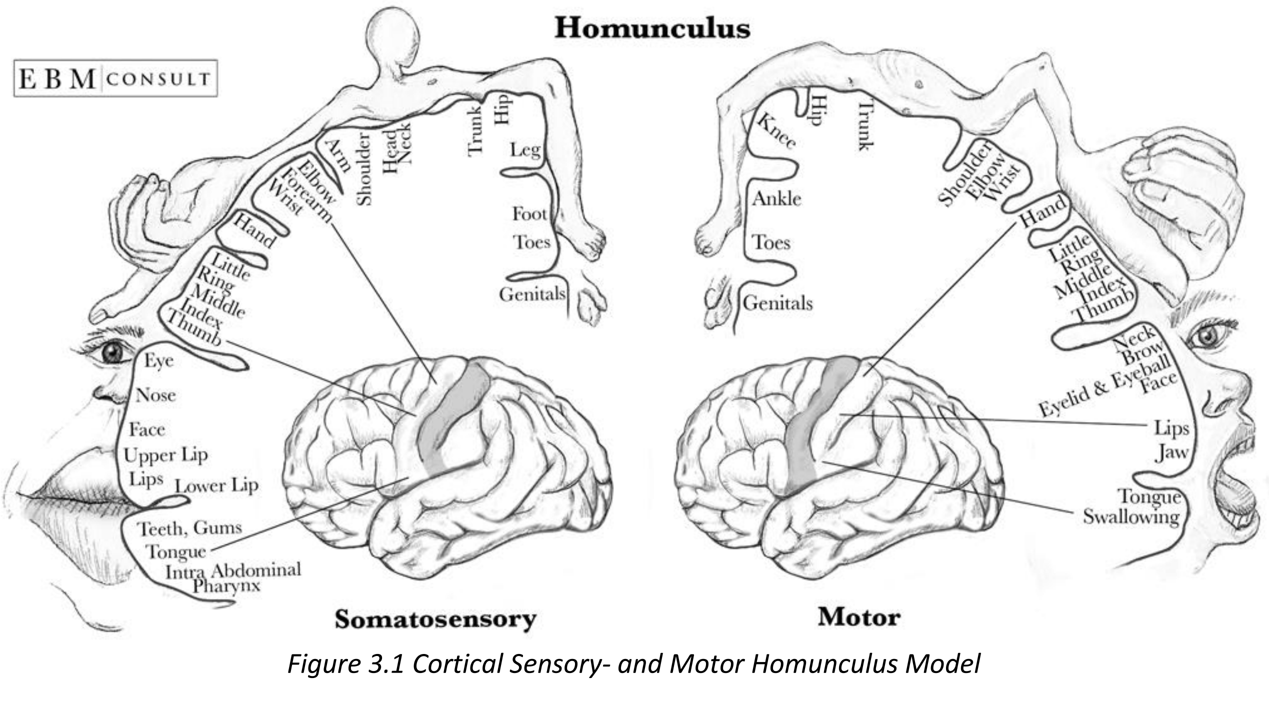

Figure 3.1 Cortical Sensory- and Motor Homunculus Model

3.2 Game difficulty

For a game to be interesting for players to play, and to preserve game flow, a game has to provide challenges that improves on the player's knowledge of the gameplay mechanics and use them to overcome these challenges (*Vorderer, Hartmann & Klimmt, 2003*).

A game's difficulty can be challenging in two different ways. The first option is to repeat the same challenge twice, but make the second iteration of the challenge harder than the first one. This can be done by making the enemies in the game have more health or making their attacks stronger. The second option is to have composed challenges (*Aponte, Levieux & Natkin, 2011*). Composed challenges are challenges where the player combines gameplay mechanics previously learned from other challenges to complete harder challenges later in the game. An example would be to jump over platforms while defeating enemies at the same time.

Isolating and understanding what makes a game more or less difficult is not a direct task and often depends on context (*Mourato, Birra , Prosperro dos Santos, 2014*). This means as Nicollet (*2004*) proposes, a series of points that grant insight into what difficulty means. These points are (*Nicollet, 2004*):

- *Surprise is not the same as difficulty.*
- *Difficulty implies possible and probable failure while performing a task that requires specific skills.*
- *Reducing the time window of an action or a set of actions increases difficulty.*
- *Performing a continuous sequence of actions has greater difficulty rather than tackling those same actions independently with structural separations in between.*
- *Reducing control over the character generally increases difficulty.*
- *Increasing the precision to overcome a challenge increases difficulty regarding that specific challenge.*

Difficulty means that there is a risk of failure (*Mourato, Birra , Prosperro dos Santos, 2014*). Juul has researched into what constitutes failures in game design (Juul, *2009*). These are visual feedback that can be displayed in the game, such as health depletion and setbacks. Setbacks means that player is taken back to a previous spot in the game after having lost a life, meaning that they are forced to replay sections of the level they have already gone though earlier. Life depletion means that the player has to out more concentration into avoiding enemy attacks and improve their skills in order to complete the challenges. Losing all lives means the player loses the game entirely and has to start all over from the beginning.

3.2.1 Dynamic difficulty adjustment

A game has to dynamically change its own design in order to adapt to individual players' effort put into the game. To do this, it is necessary to consider how to keep the player engaged with a game and have them keep playing a game (*Gilleade, 2004*). A game designer has to account for motivation, experience and skill.

Motivation is the reason why players want to continue playing a game. Players may have different reasons to play a game. It may be for the fun the game provides, the challenges, the goal or to beat the high score. If a player loses several times during a section of a game, the repetition of having to play the same level over and over may invoke frustration to the player and break the flow overall. The simple solution to this problem would be to change one or more parameters that makes the player keep losing, such as the enemy health, interval of enemy attacks or the strength of enemy attacks based on the player's level of frustration. However, such a solution may not work in all cases, as some players may take such a downgrade of challenges as patronizing, and completing the challenges won't feel as satisfying (*Gilleade, 2004*).

Experience and skill dictates how much time and effort a player has put into learning and mastering the gameplay mechanics in order to complete the game's challenges. How well they do in the game is a very important factor when looking into their rate of successes and failures and when the game needs to adapt based on the player's effort (*Gilleade, 2004)*.

A game can be developed to defect automatically when adapting its own difficulty is most appropriate. This could be done by detecting whether they have done very well or very badly in a certain time period of the game, and the game will be adjusted to become either easier or harder. Using this data, the game can automatically adjust the difficulty to adapt to the player's gameplay skills and preserve the game flow. However, how well the player does in a game and how a player feels are two completely different things. A game cannot take into account how a player feels when playing a game and cannot adjust its difficulty properly.

Now the aspects of difficulty and the characteristics of challenges have been described. But before the design process can be approached, it is important to understand how difficult a challenge needs to be in order to adapt properly to the player's skills.

Mourato, Birra and Santos (2014) uses the example of jumping over gaps and jumping from wall to wall in a side scrolling platformer game, like Super Mario Bros. (Nintendo, 1985). The difficulty of a jump is defined by the maximum jump length and height of a player's character compared to the length they have to jump to reach over a gap. They explain that is becomes more challenging for a player to jump over to a platform that is positioned higher up than the platform they are standing on. Platforms that move are also elaborated on as being more challenging, as the player has to time the jump just right and how long to jump. They also explain that the difficulty is altered depending on how precisely players can perform a jump. If a player struggles making a jump just right to reach the next platform, if they have no option to move around freely in mid-air, the challenge becomes increasingly harder. Player's rarely ever performs a perfect jump, and therefore the challenges should be designed according to how well the player can jump (*Mourato, Birra and Santos, 2014*).

Games are designed to have different experiences with different challenges. Therefore, a DDA system can be used very differently based in a game's design. A DDA system is developed to provide improve each individual player's experience with a game based on their skills. Each game can be altered in different ways by changing specific parameters that won't necessarily change the game overall. An example would be a first-person shooter, where the game could be made easier by having the shoot less often to give the player a better chance at beating the enemies.

3.3 Flow

Flow is important to take into account for this project, as it provides knowledge that game designers use to make their games engaging and challenging. Flow is the state that a player falls into when performing any activity. This means that they are immersed in the activity that they are performing. Game flow is the state that a person falls into when playing a video game. The experience can be split into 8 parts (*Csikszentmihalyi, 1990*):

- Clear goals
- High degree of concentration
- Not feeling self-conscious while playing
- Missing sense of time
- Direct and immediate feedback
- Balance between ability level and challenge
- Sense of personal control
- Rewarding.

Flow is important to video games as the principles are used to maintain player engagement in a game and the player's determination to complete its objectives (*Csikszentmihalyi, 1990*). This can occur, even if the objective or the path to the objective doesn't seem to be straight forward to the player.

The reason why flow is such in important aspect of a video game is that it gives an experience that gives the player motivation to keep playing the game and complete its objectives. When the goal is reached, there often comes a reward of any variant. This can be something that represents the player's struggle with reaching the goal, like a high score that shows the players total amount of points collected. It can also be an item that spawns when defeating an enemy (*Sweetser & Wyeth, 2005*). Flow can also be correlated with the positive experience of frustration. Flow can occur differently from person to person, and they may all feel immersed within an activity in very different degrees.

One of the subtopics to flow, relating to video games, is called game flow *(Sweetser and Wyeth, 2005).* Game flow is another aspect within flow in which the level of flow is used to observe how engaged a person is when playing a game. Game flow is used to affect enjoyment within people playing games. This is divided into 8 (*Sweetser & Wyeth, 2005*):

- Games should require *concentration* and the player should be able to concentrate on the game.
- Games should be sufficiently *challenging* and match the player's skill level.
- Games must support player *skill* development and mastery.
- Players should feel a sense of *control* over their action in the game.
- Games should provide the players with *clear goals* at appropriate times.
- Players must receive appropriate *feedback* at appropriate times.
- Players should experience deep but effortless *involvement* in the game.
- Games should support and create opportunities for *social interaction*.

All elements each plays an important role in the overall enjoyment of a game. These components are not mandatory to use in a game, and some of them may not even be present in a game. This is a matter of choice, based on the game's concept and its mechanics (*Sweetser & Wyeth, 2005*). Game designers should choose carefully what components to use in a game when initiating the design stage of a game production. The peak of immersion when playing a game is that players lose sense of their surroundings and sense of time. Each component of game flow is made up of objectives and goals that can only be completed with specific gameplay skills in mind. Having this in mind can be used to design and implement a game design with a correlating level design that takes the players experience into account. With all the knowledge regarding the concepts of flow and how they are used to design games, it will be easier to design a game with the intent to maintain frustration based on a good DDA system.

3.4 Game design

Raph Koster has come up with important features, which constitutes what makes a game fun (*Koster, 2005*). First of all, there needs to be a flexible cooperation between input and output. A player should always expect a proper output that makes sense from the inputs that they are given. A game also needs to have elements that are unpredictable, but doesn't break the player's suspense of disbelief of a game world. They also needs to be things that the player can react on quickly and know how to approach the given situation. If a game's inputs give unpredictable outputs, the overall experience may decrease (*Koster, 2005*).

Koster proposed that the human brain is a pattern seeking and problem solving system. Games are just satisfying systems that presents the opportunity to be solved by the brain (*Koster, 2005*).

Consistency is something that should also be taken into account when designing a fun and satisfying game, with multiple patterns and problems to solve.

3.4.1 Inputs

A good game flow also depends on good controls. Any given set of input has to give feedback either visually or auditory in order to give the player a clear understanding of the gameplay mechanics and incorporate all the gameplay elements together to make progress within the game. As seen below is an illustration of an Xbox controller with specified input used to play the prototype. The controller was chosen instead of a keyboard is that there are a lot fewer buttons to push while playing the game, and it may also give the player a better experience with the game.

Figure 3.2 – Gamepad Controller Inputs

3.4.2 Level design

With the concepts and rules of game design being explored and elaborated, the level design of a game needs to be explored as the next step.

Feil and Scattergood (2005) did some research into gameplay mechanics, but they also explored the design principles of level designs within the games themselves. They state, that level designs can be many things and can overall be a challenging aspect of game design in many different ways, depending on the individual game.

A lot of old games, such as Tetris or Pong present challenges that are constant throughout their sessions. Challenges in a lot of modern games today are spaced out between sections where the player is allowed to get a break from completing the previous challenge before they proceed to the next one. Challenges can also be different as well. They can either be presented by defeating a squad of enemies, solving a pulse or reaching a destination and avoid obstacles and traps. Where a challenge will occur and how often they will occur depends on the game's flow and pacing. A challenge that level designers face when designing levels is what the player should be up against at the beginning of the game until the last part in order to maintain the flow. Resting can be a good thing for a player, but it can also be interpreted as a boring experience where the player can't do anything other than walking to the next challenge. Tension and engagement within a game can be split just as challenges can be as well by having the player be occupied with something else while resting in order to maintain the overall experience of the game during the whole session (*Feil & Scattergood, 2005*).

Feil and Scattergood have also elaborated on the aspects of level design by setting up a four-act structure. This structure focuses on the presentation of the game's fundamental rules and mechanics to the player in order to maintain a satisfying and challenging experience to the player. The 4 acts are each presented below (*Feil & Scattergood, 2005*).

The first act concerns about the first-hand impression of the game. The player should have a clear impression of the game within the first 10 minutes of the play session. It needs to provide a good enough experience for the player to be engaged enough to keep on playing. Player's should feel in control of the game and have a good understanding of how to use the gameplay mechanics, even if their knowledge of the game is only on a surface level during the beginning. There is often a introductory cutscene that shows the setting, story and characters of the game that gives the player enough information to understand the situation they're put into. There can also be small portions of texts that is presented to the play which explains to the player how to play the game step-by-step (*Feil & Scattergood, 2005*).

The second act is the middle part of the level design. The peak of the experience will be within the middle of the level section. The most important part of a good level design is consistency and growth. If players can destroy boxes during the beginning of the game, they will expect to break any kinds of boxes during the rest of the game. This is what constitutes consistency. Breaking the rules of interactivity could become problematic. If the game shows that the player can break one object but not another object it will risk breaking the immersion of the game. The challenge of maintaining consistency is balance. A game should have different varieties of enemies, but one enemy should not become stronger than the other should. Another important rule to maintain player engagement is to introduce game elements that builds upon the rules of the gameplay mechanics and the world the game takes place in in order to avoid contradictions. You cannot introduce a robot enemy in the beginning of the game and then introduce something else that would not exist in the world the game takes place in. Growth constitutes that the player's skills is built up from the bottom by presenting new and harder challenges one after another in order to get a feeling of empowerment of the game (*Feil & Scattergood, 2005*). But a challenge that level designers also face is to determine when a challenge is too easy or too hard, depending on the individual player.

The third act of the structure is the final part of the game. This is where all the player's collective knowledge of the gameplay mechanics and level design are put to a final test. This is where the player will have to combine all the skills that they have acquired through the whole game by completing a challenge harder than any of the other challenges they have faced during the game (*Feil & Scattergood, 2005*).

To make sure that the player have a consistently positive experience and engagement with the game it is needed to have different challenges, which can be split apart and with quiet times in between to build tension.

3.5 Emotions

When it comes to reading emotions and how they can be used to read and analyze player's experience with a game, it is important to take into consideration some of the problems with the definition. There are 3 phases of emotions; emotion, mood and core-effect (*Ekkekakis, 2012*).

Core-effect is described by Russell and Barrett as being "*a neurophysiological state consciously accessible as a simple primitive non-reflective feeling most evident in mood and emotion but always available to consciousness"* (*Russell & Feldman Barrett, 2009*). These emotions often include pleasure, tension, relaxation, activation and being tired.

The next is emotion. This is less direct in its definition, as it is described *as "a complex set of interrelated sub-events concerned with a specific object"* (Russell, Barrett, 1999). This means that emotions often are psychological emotions to some kind of stimuli. These emotions include anger, fear, pride, love and jealousy.

The last is mood. The difference with this kind of emotion is that it lasts longer. Compared to an emotion, which is a reaction to a certain kind of stimuli, mood is defined as "*the appropriate designation for affective states that are about nothing specific or about everything-about the world in general"* (*Frijda, 2009*). This means that mood can occur with great delay. A person could walk around all day in a bad mood, but the amount of time this mood is present is not identifiable, making it harder to determine what caused this specific mood to appear within this person.

There are also models to visualize human emotions. These models measure emotions in 2 dimensions. One of these models used to measure emotions is the circumflex model of affect, proposed by Russell (*1980*). This model is a 2-dimensional model used to measure emotions. The 2 dimensions are "valence" (a range between pleasure and displeasure), and "alertness" (a range between arousal and sleepiness). This model has proven to be a reliable method to describe the human affect-space (*Russell, 1980; Posner, Russell, Peterson, 2000*). The circumflex model will be combined with the self-assessment manikin model to analyze and visualize the players' overall experience with playing the game.

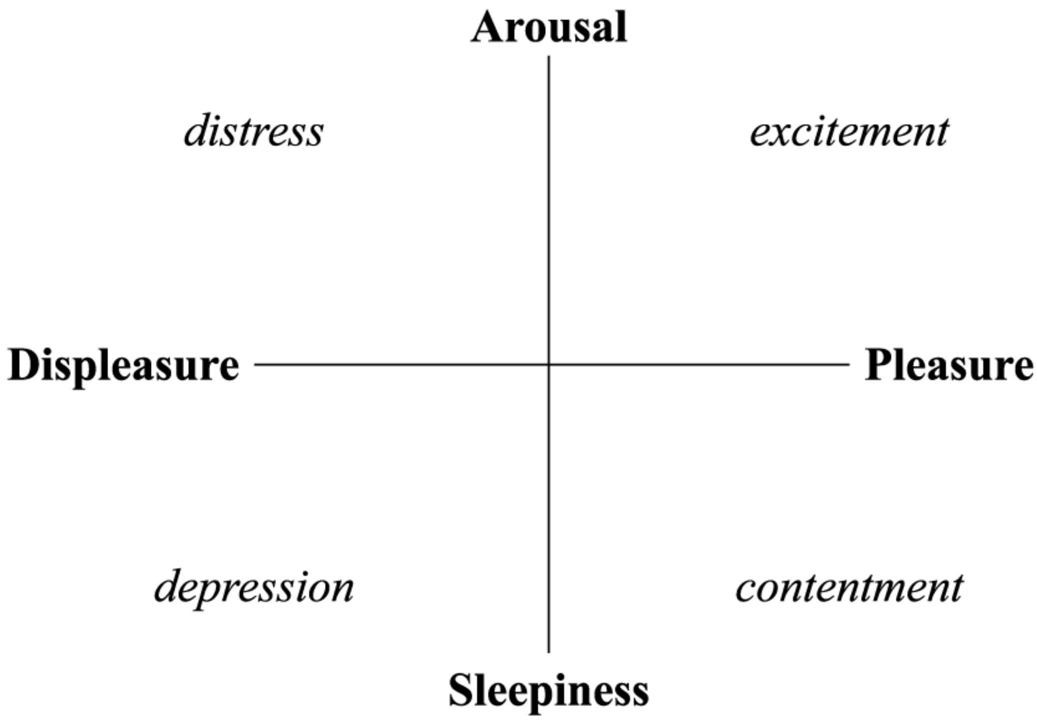

Figure 3.3 – Russell's model of affect

3.6 Final problem statement

With the knowledge gathered in the analysis chapter, it was concluded that a DDA system based on physiological data could improve a game's experience of flow for an individual player. But it was also apparent that such a method wouldn't work equally for all players. This could cause a problem to some players, who seeks to take up a challenge and complete them with their own skills and knowledge of the game's mechanics.

But if there is no frustration present within the player, there is no motivation or engagement to continue playing the game. But the same can happen if frustration becomes too high. This means that the player's frustration needs to be maintained at a constant level. Research has found a correlation between frustration and GSR, as mentioned in an earlier chapter (Chapter 4, page 14).

It was now possible to come up with a final problem statement:

Can a game take player frustration as a valid data input through GSR, and dynamically alter the game's difficulty of a game, in order to improve the maintenance of a player's level of frustration and improve the game's experience of flow?

4. Method

The focus of this study is to find a DDA system a proper way of adjusting difficulty in game. According to a research conducted by Darenn Lunn and Simon Harper (S. H., 2010), GSR has been shown to be closely related to the feelings of frustration under any given activity. The more frustrated a person becomes, the more GSR increases.

It was decided that one sample group would be tested on in order to find whether the desired DDA system was sufficient enough to maintain a stable level of frustration. This means that a null hypothesis and an alternative hypothesis would need to be set up to conduct the experiment, which are the following:

- Null hypothesis: A DDA system using physiological data does not do a good job at maintaining player frustration.
- Alternative hypothesis: A DDA system using physiological data is good at maintaining player frustration.

The test conditions are determined beforehand in order to make sure that all participants have an equal chance of playing and completing the game. The test participants will be playing in an isolated room to make sure that they will not be interfered while playing the game. While they are playing the game, a researcher will be present from a place where the participant will not notice them. While the test participant is playing the game, their level of frustration will be recorded and stored to be analyzed, using the Empatica wristband, which is a device used to measure a person's physiological data. The researcher will keep track of how the participant are managing with the game from a tablet device, while also keeping track of their level of frustration. When their level of frustration has increased a certain amount, the researcher will manually adjust the game's difficulty from the tablet in order to emulate the game to react to the participant's level of frustration and adjust the difficulty automatically.

Figure 4.1 – Empatica Wrist Band used to measure GSR

It was necessary that a pre-test would need to be conducted before the final test. This was to make sure that all at-game frustration was found and eliminated to remove any bias for the final testing. At-game frustration occurs when the game's mechanics does not respond accordingly to the player's commands. If this happens, it can drastically affect the overall experience of the game and create bias within the data. The response gathered from the pre-test would then be used to conduct a redesign of the gameplay mechanics and the level design to meet the requirements set for the experiment.

Test participants would be gathered from the campus of Aalborg University Copenhagen. This was decided because it was decided to have participants with at least some minimum knowledge and experience with video games in order to make sure that they understand how to play the game properly.

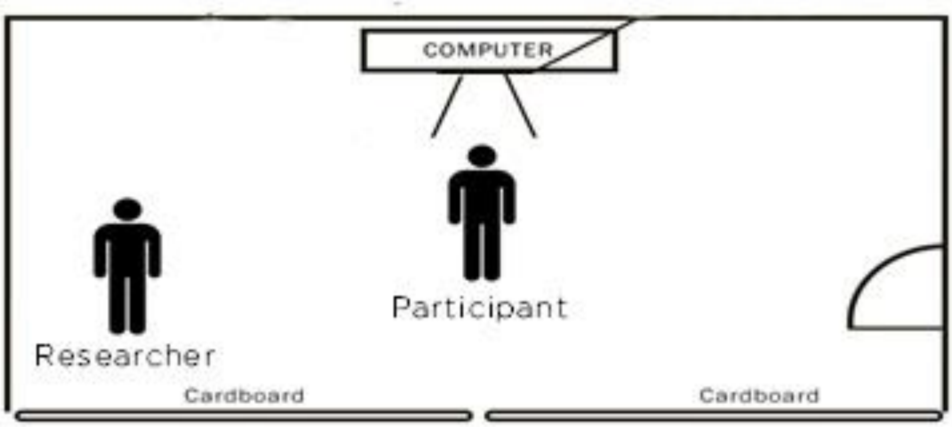

Figure 4.2 – Setup for testing

The measurement of GSR will be recorded during the play session and plotted into a graph which can be accessed for later analyzing. After all data has been gathered, a mean value will be calculated individually for each participant and plotted into a graph. The standard deviation of each participant will then be calculated from the mean values and used to accept or discard the null hypothesis.

5. Design requirements

5.1 DDA

The game should be automatically adjusted to the player's level of frustration. This should be done by adjusting one single parameter throughout the game and make the game more fair for the player to make progress. It should also be done in a way so the difficulty adjustment does not seem too obvious for the player to notice in order to not affect the game experience.

5.2 Game Design

Game design has a set of rules of principles that is being used by most game companies today. They are important to follow in order to convey the most important information to the player, most importantly the gameplay mechanics. This can be done by making each input from the player give a pleasing and consistent output that seems like something their player avatar could do. It is also important to make the challenges satisfying by having the player build up their knowledge and skills of the gameplay mechanics and combine them all in order to overcome obstacles and defeat enemies.

5.3 Level Design

When creating the game, the three acts of level design will be made. This builds upon the importance of players progressively building their skill, and using the knowledge of the mechanics to complete challenges, and use all the knowledge for the final act.

5.4 Frustration

Frustration occurs when the game's challenges exceeds the player's mastery of the gameplay mechanics, but also when the challenges and obstacles become too hard to overcome. Frustration can be used as an input to the Dynamic Difficulty Adjustment system. A proper system that can take this input and use it in a game was not found when researching the technical difficulties to implement the game. Another solution to this was to emulate the Dynamic Difficulty Adjustment system by adjusting it manually while the game was being played. When the design document has been established, the implementation of the game can be conducted.

5.5 Flow

When playing the game, it is important that the game maintains the player's flow during the whole session. As explained in the Flow Chapter (Chapter 3, Page 8), flow incorporates different factors from the player that makes the player experience flow. These are their level of concentration, challenging gameplay, level of skills and a feeling the gameplay outputs respond accordingly to the player's input. Also to make it clear to the player what the main goal is, and to give them a reason to continue playing.

6. Game Design

6.1 DDA

The DDA system should change the game's difficulty based on the player's change in level of frustration. It is necessary to know when to decrease the game's difficulty based on how high the player's level of frustration. Therefore, the game will not take the player's current level of frustration into account, but rather the change in their level of frustration. For this, the DDA system is set up with certain requirements in mind.

The test should focus on increasing the game's difficulty from the player's level of frustration. Therefore, the game will get easier if they get too frustrated, and harder if they stop being frustrated. The DDA system will change one parameter in the game to decrease the difficulty. The parameter is set to be the enemies' maximum health. The enemies are set to have a certain amount of health that will take some trial and error for the players to defeat in order to provoke the player and increase their level of frustration. When the player gets frustrated, the enemies' maximum health will decrease, making it easier for the players to overcome the game's challenges.

6.2 Game flow

When the game becomes too hard to play, it will by result break the game flow. The purpose with The Dynamic Difficulty Adjustment system is to improve and maintain the overall experience of the game, which requires the player to concentrate and decide every move quickly. To maintain flow and create good challenges for the player, the challenges should appear consistently throughout the game, and force the player to keep concentrating while playing. The challenges should not appear everywhere across the game, as it would become too much of an exhaustion to the player. Completing each challenge should give the player a momentarily break and allow the player to move around freely to collect points and prepare themselves for the next challenge. Though these challenges should appear consistently throughout the game, they will still be adjusted accordingly to the DDA system when they become too much of a hindrance to the player.

6.3 Inputs

The game has different inputs which each of them executes different outputs that the player can control freely. However, it is also important to place these player inputs in a way so the player can memorize and have a clear understanding of each input. For this, the Xbox controller was chosen to be used as the controller input for the game. The reason behind this is that with only a limited amount of gameplay mechanics to choose from, it is easier to memorize each button with a controller compared to a keyboard, where there are more buttons to take into account and can become more confusing to the player.

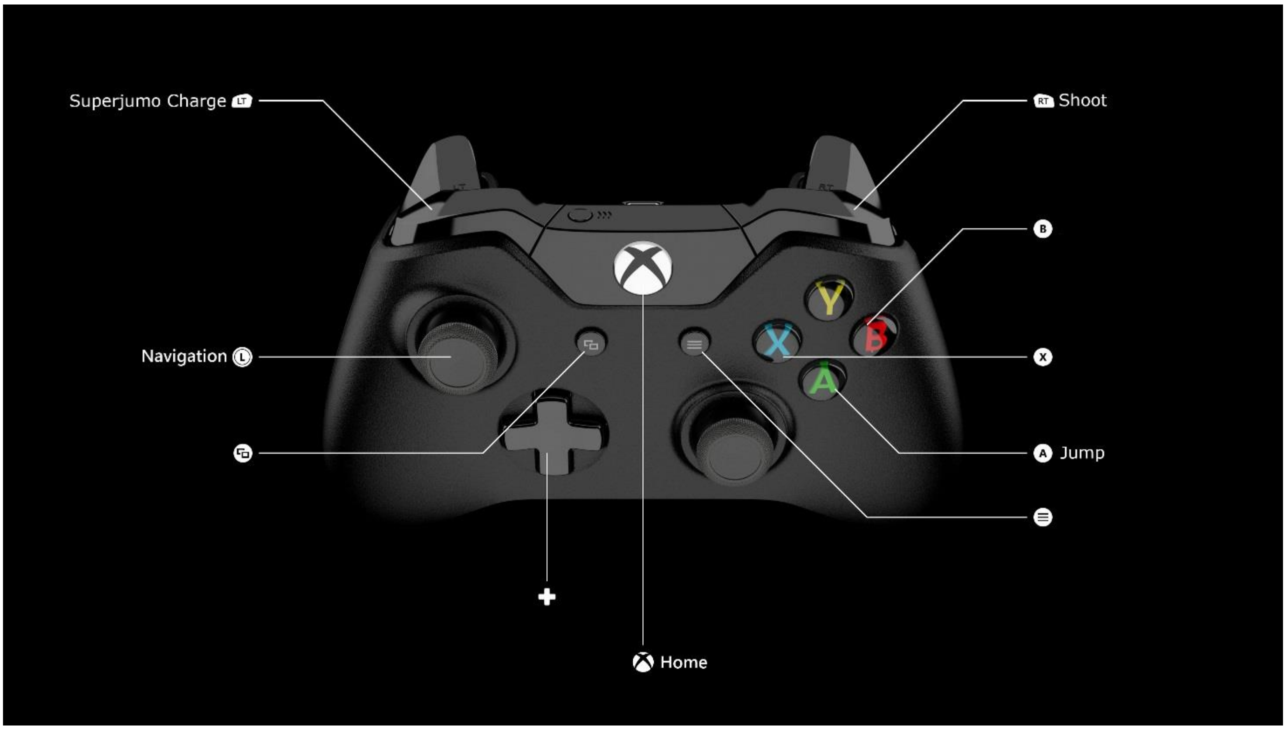

Figure 6.1 - Gamepad controller input for game demo

6.4 Level design

Level design was made consistent with having in mind that each gameplay mechanic should be elaborated and explored during the first segment of the game, as elaborated on by Feil and Scattergood (*2005*). As the game will only be played for 10-15 minutes, the introduction to the gameplay mechanics will be short and clear. Therefore the game should be explained simply and clearly. As there are only a few mechanics to learn, they can each be introduced in small simple steps.

Figure 6.2 – Introduction to input mechanics

The first challenge that the player approaches is a tall platform that the player has to get across. This is where they are introduced to the super jump function, which allows them to jump higher into the air than a normal jump would. There are also virtual points scattered across the level to help guiding the player in which way they need to go, but also to maintain the game's engagement by collecting points to beat the high score.

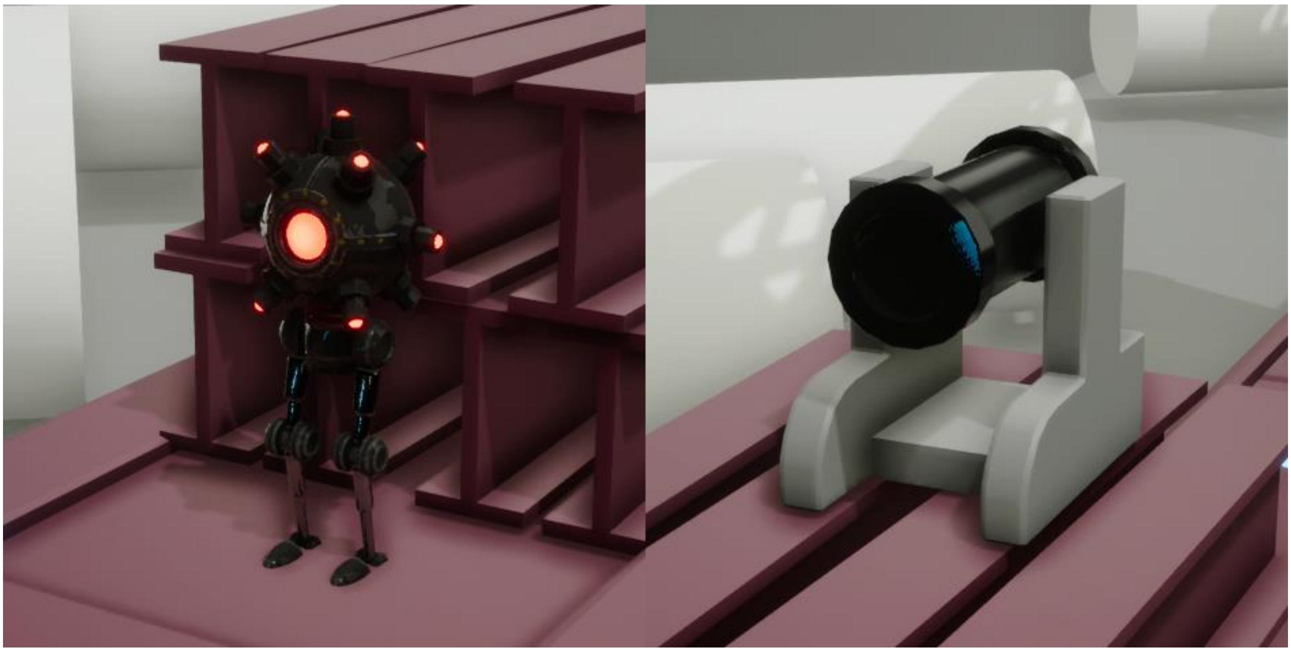

Figure 6.3 – Bomb enemy and cannon enemy

The next challenge they will face is a bomb enemy, which will approach the player as soon as they are within their range. The bomb is designed with intention to make it look like a mine to tell the player indirectly that touching it is dangerous and should be avoided. They also emit an audio queue, which the player will be able to detect when they are close to the bomb. Afterwards they will then face against the first cannon enemy, which will occur most often across the level. Letting the player defeat one cannon enemy at first will give them the impression how hard it is to defeat one cannon before fighting against multiple cannons at once.

All of this was sketched digitally before implementation could begin. This was a good way to get an idea of how to structure the level and the challenges that the player will have to overcome.

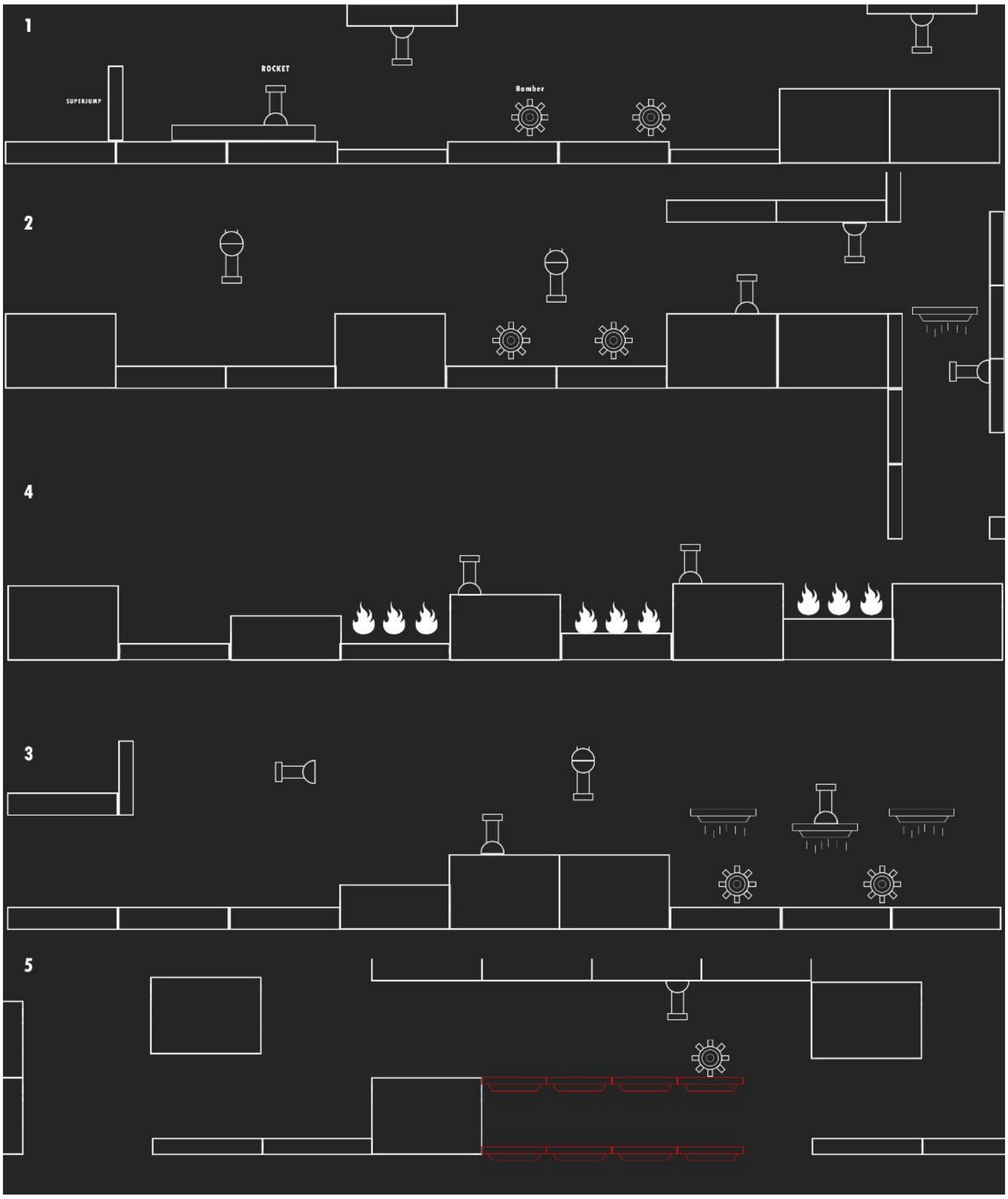

Figure 6.4 – Level design

7. Implementation

This chapter goes into details on how the design was implemented, as well as some examples on how the implementations were carried out. This chapter also goes into depth about some of the code that was created, how the 3D modelling was done, and how the sounds were used.

7.1 Programming

The game was implemented using the video game engine, Unreal Engine 4 (Epic Games, 2013). The engine takes use of the programming language, C++, but also offers another approach at programming, called the Blueprint Visual Scripting system. The Blueprint system in Unreal Engine 4 is a node-based scripting system that allows users to implement and test pieces of codes in a dynamic workflow. It follows the same logic and principles as any other object-oriented programming language. Using blueprints for the game allowed the implementation process to take less time to complete and brought just a good result.

7.1.1 Player

The Player blueprint is where all the player inputs are implemented. All the pieces of codes run on the "Event Tick" node. The Event Tick node updates all the nodes connected to it by every frame in the game. One of the player blueprint's core functions is the ability to shoot. The way this works, is that when a button us pressed, the code will spawn a Bullet object from a socket point, located in the player avatar's cannon. The bullet has a function assigned to it, that makes it move at a preset constant speed at the direction of a vector the moment it spawns. Whenever a bullet hits an actor marked with a tag named "Enemy", the bullet will destroy itself and inflict damage to the Enemy actor.

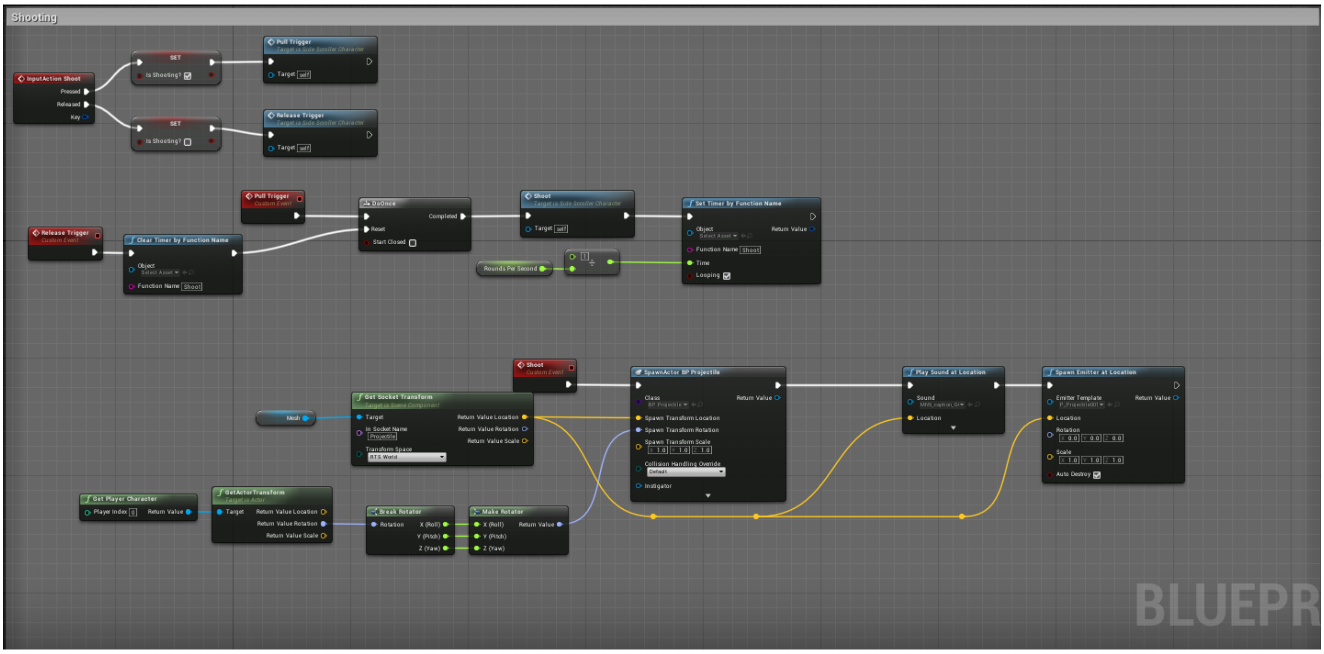

Figure 7.1 – Player blueprint

7.2.1 Enemy

One of the enemies that occurs during the game the most is the enemy turret. If the player character is within a specific range of the enemy turret, a Boolean, named “TargetLocked”, will be set to true and follow the player character’s movement. The second function is the way the enemy turret rotates while following the player character’s position. Firstly, a blueprint note called “Find Look at Rotation” is used to make a vector that determines the angle between every enemy turret and the player character throughout the level. Then another node is used to animate the cannon part of the turret between it’s current rotation to the vector that stores the angle between the turret and the player character. The speed of the transitioning rotation is pre-set to allow the cannon to rotate more smoothly. The third function is the enemy turret’s ability to shoot the player character. The way this works is that when the player character is within the enemy turret’s range, it will spawn a bullet object from the position and rotation of a socket object, which is attached to the cannon part of the turret, so that it will always move accordingly to the cannon’s current position and rotation.

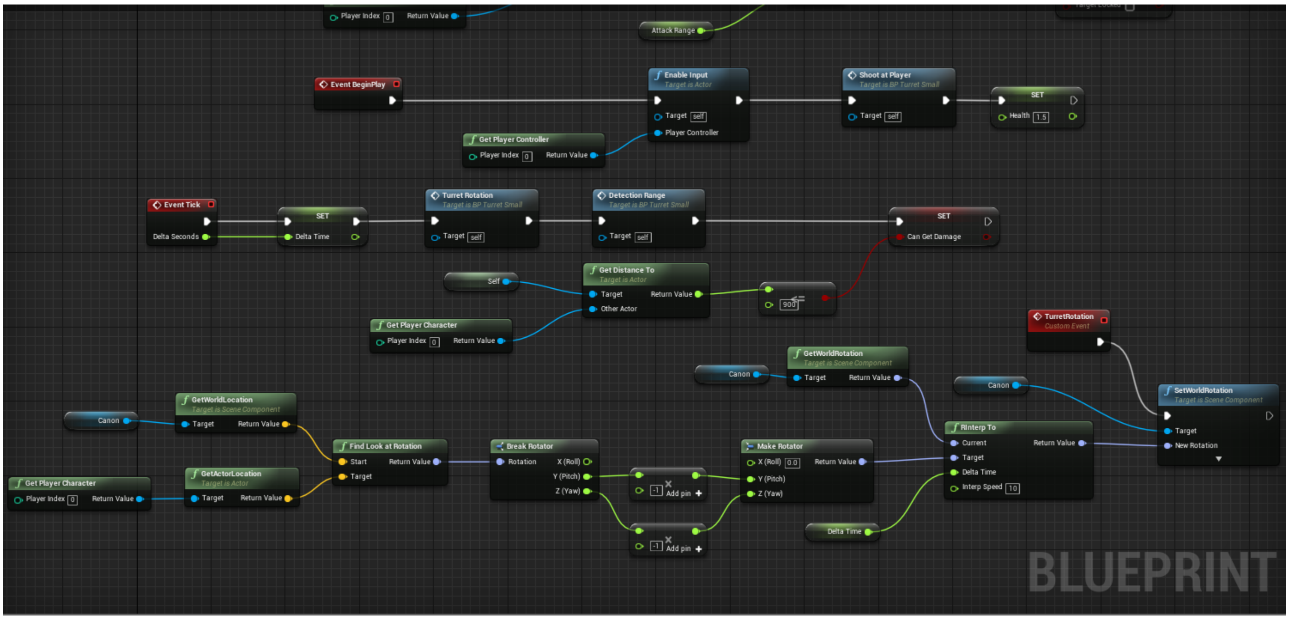

Figure 7.2 – Cannon enemy blueprint

7.3.1 Platform

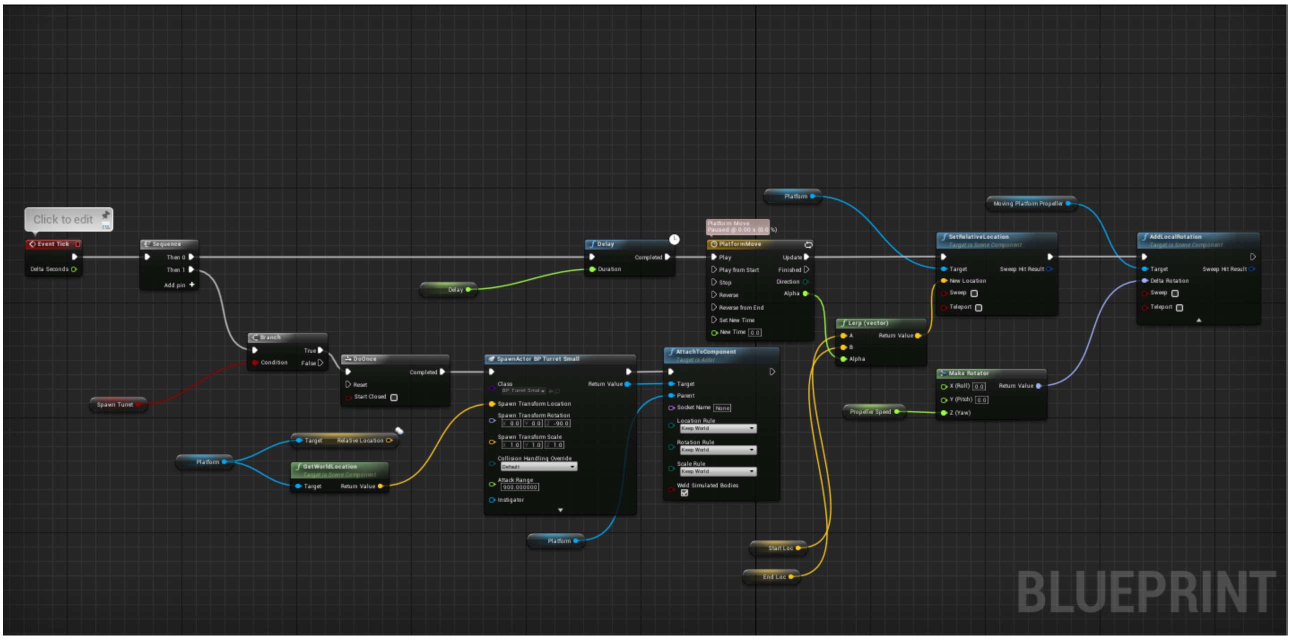

Figure 7.3 – Platform blueprint

7.4.1 Checkpoint

The Player Character blueprint has a variable, named “Player Spawn Transform”, which can be seen on illustration 7.4. The variable is being called from the Checkpoint blueprint whenever the player character touches the blueprint’s hitbox within the game. This will set the Player Spawn variable to store the position and rotation of the checkpoint that has most recently been hit.

Whenever the player character’s health reaches zero, a function will be called, which spawns a new player character from the position and rotation of the last called checkpoint, before the old player character object destroys itself.

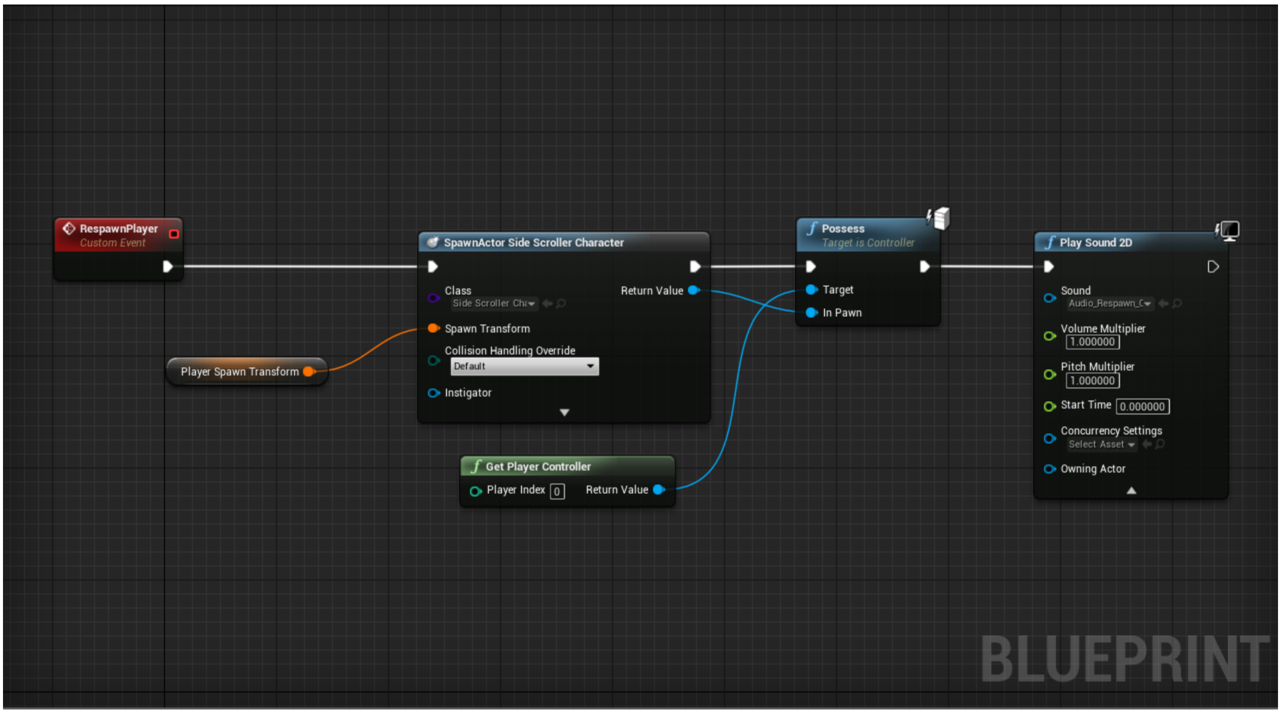

Figure 7.4 – Checkpoint blueprint

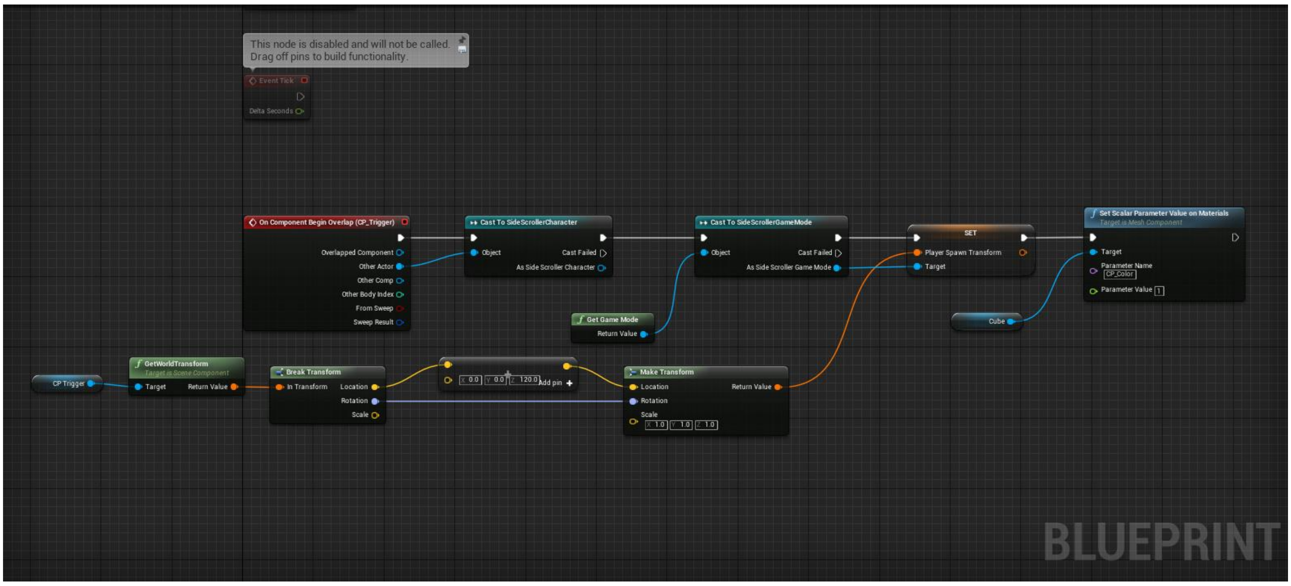

Figure 7.5 - Checkpoint blueprint

7.5.1 Difficulty Adjustment

For adjusting the difficulty while playing the game, a blueprint was made so that the difficulty could be adjusted manually by pressing a button. When the O button was pressed down, the blueprint would decrease all the enemies' maximum health, making it easier for the players to defeat the enemies and make progress. When you press the P button, the game would increase the enemies' overall health.

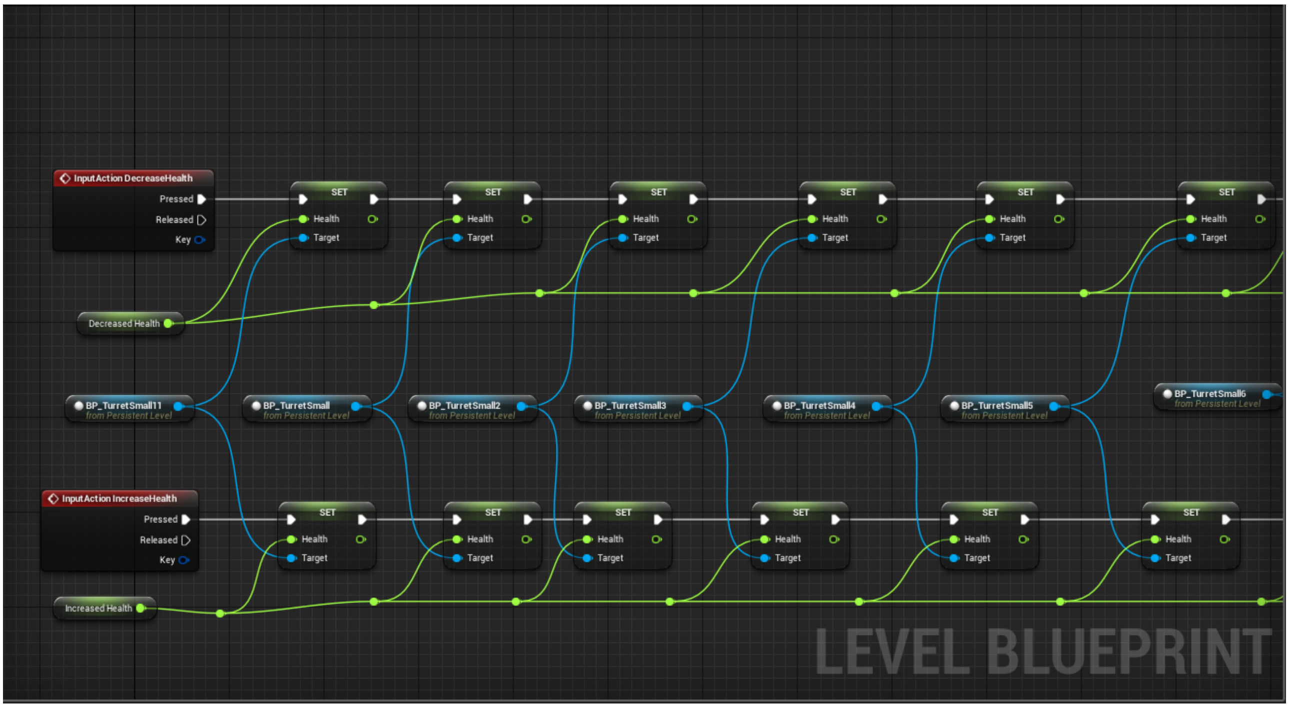

Figure 7.6 – Level Adjustment script

7.2 3D

This chapter will elaborate the implementation of all the assets and animations used for the game. It will also include some of the principles of art design used in the game to help guiding the player through the level.

There is no definitive way to implement 3D graphics in a game. There are several pipelines that can be used to approach the 3D process from a sketch to the finished asset. One of the requirements for a game is to keep each asset at a certain amount of polygons in order to maintain the performance of the game. Some engines can take a larger of amount of polygons than others, but the thing to keep in mind in here is to keep the assets simple, as they are not the main focus of the game.

7.2.1 3D Software

The modelling software used to create the assets and animations is 3D Studio Max (Autodesk, 1996), developed by Autodesk. It is a 3D modelling and animation software used in both the video game and movie industry. For this game, it is used to produce both the assets and animations for the main character.

The software used to create the textures for the assets was the 3D painting software, Substance Painter (Allegorithmic, 2018). Substance Painter offers tools that can be used to create procedurally generated textures on the 3D model itself. The textures can be modified by adjusting various parameters in order to make it fit the 3D asset most appropriately.

Figure 7.7 – Player character in animation software

7.2.2 Character design

When a video game is being developed, it also needs to have a character that the player can use and control within the game. One of the tasks of the art department is to go in-depth with character design and have it fit with the given art direction and art style. The requirements are to determine the character's appearance, color scheme, and their personality traits, such as strengths, weaknesses and behavior. This is all to make the character appealing to the player. How the character plays and how they behave in the world can tell everything you need to know about them. The design mainly has to follow the art style and the setting of the world the game takes place in. The character design can also tell how the character plays and how they behave within the game.

7.2.3 Avatar

An Avatar is a term primarily used by game creators. It is used to refer to the character that serves as character that the player controls in the game. Avatars can have a pre-determined design, but they can also be customized in open-world games or RPG’s, where the player can determine the Avatar’s gender, race, clothing, weapons etc.

For this game, the avatar’s design was based on the setting and gameplay mechanics. It had to make sense for the player how the avatar could perform certain actions and movements. It also had to make sense with the gameplay mechanics and level design. In the beginning, the character design was approached drawing several sketches of different character designs in order to get the right shape and color. The final design was used as reference to model the avatar in 3D Studio Max. The avatar was designed to be a robot in order to appeal to both men and women. Because the avatar can perform a boosted jump, the lower part of it’s body was designed to behave like a spring, where the avatar would descend it’s lower body in order to charge up a boosted jump.

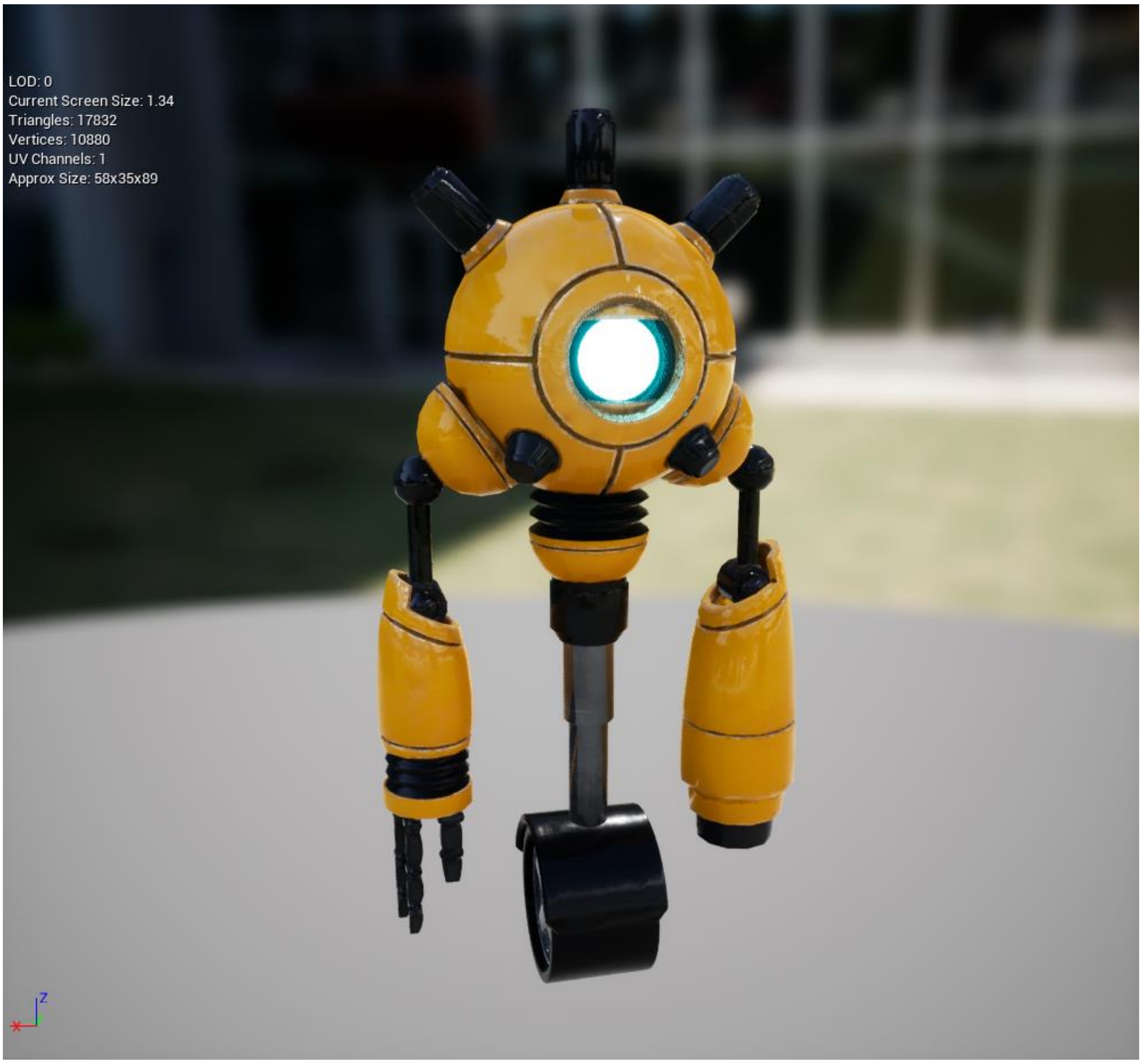

Figure 7.8 – Player avatar in game engine

7.3 Sound

7.3.1 Audio tracks

Audio tracks are being played continuously throughout the game. They are used to set the mood of the game and increase the overall experience of playing the game, but also to eliminate the deafening silence that would risk making the player feeling bored while playing the game. The audio tracks consists of 5 different tracks, each playing after one another, in a preset order. The software used to compose and mix all the audio tracks is the music software, FL Studio. It provides a set of electronic instruments that can be used to compose each instrument separately and more dynamically create music that fits the mood of the game.

7.3.2 Audio queues

Audio queues are short audio tracks that plays when a player input is being executed in the game. Here audio queues are being used whenever the player jumps and shoots. Audio queues are also used to warn the player of incoming enemies, most specifically the walking bombs. Each bomb plays an idle sound, which the player will be able to detect from a certain distance, making them aware that something will be assaulting them.

8. Test

8.1 Pilot Test

A pilot test was performed on a couple of test participants before the final test could be conducted. The purpose with the pilot test was to test whether the game was set to be too easy or too hard, but also to find and eliminate any at-game frustration. At-game frustration can occur from issues with controls, interface or errors in the game. The pilot test was conducted by having a couple of test participants play the initial prototype of the game to the end while they were being observed to find any at-game frustration or other kinds of issues. Any issues found would be noted for implementing the final version of the game for the final testing.

8.1.1 Re-design

5 participants played the game in order to test the usability of the game and to see whether the gameplay design worked efficiently enough for the players to understand how to play the game and how to make progress in the level. During the observation of all participants, a few noticeable issues occurred during the play session that would likely disrupt the test results in the final testing. These issues were noted down to be taken into account while implementing the final version of the game. The pilot test had a pre-determined setup. The test took place in an isolated location where the participants would play the game without being distracted by any outside elements. While they played the game, one researcher would place themselves in the same location a fair distance from the player, while observing their play session on a separate tablet, using the software, Team Viewer.

One noticeable habit that most participants seemed to follow was shooting nonstop without hitting anything as they were walking across the level. They did this hoping they would, by chance, hit any nearby enemies and overcome obstacles easier that way. This seemed to work for them to some degree, which was something that would have to be corrected for the final implementation. It was corrected so that players can only damage enemies when they are in the camera's field of view.
Another notable issue was when players felt frustrated by the enemies, they would try to get through the level and avoid being hit by the enemies. This was a big issue as the players should only be able to make progress by defeating enemies. This would be corrected by placing enemies in the level in a way to that the player couldn't avoid the enemies no matter what and would get killed trying to. This would force them to play the game as intended.

Eliminating all the issues coming from the game would ensure that the frustration coming from the test participants will be the result from their lack of skills and not the game itself.

8.2 Final test

The test setup consists of 3 parts. During the first part, each participant is briefly introduced to the game, the main objective, and how they are meant to play the game. They are given instructions on the gameplay mechanics, and what each button on the controller does. When they have had a clear understanding of the test, they will have the Empatica wristband attached to their wrists. The wristband will be connected to the empatica application on a smartphone through Bluetooth, and their physiological data can be recorded

The second part consists of the participant playing the game until they have completed the level or until the time limit runs out. They play the game on a stationary computer with headphones on, so that they can get a better focus on the game without being interrupted by any other sounds. Their level of frustration would be observed on a smartphone through the Empatica application, and their game session will be observed on a tablet by a researcher with the help of the software, TeamViewer (TeamViewer, 2005). When their level of frustration has increased to a certain level, the researcher will manually adjust the game's difficulty by pressing a button, which will decrease the enemies' health while the participant is playing the game.

The third and final part consists of the participants answering a questionnaire after playing the game. The questionnaire consisted of qualitative and quantitative questions, with the Likert scale being incorporated. The scale was a 7-point ranking scale from 1 to 7, where the values represent "not at all", "slightly", “moderately”, “fairly” and “extremely”. The questionnaire specifically asks about how the participants felt with the game, how fair they felt the game was and how strongly they felt the game got easier as the game’s difficulty decreased while playing the game. The questionnaire also consisted of a self-assessment manikin scaling to measure each participant’s valence and arousal after playing the game. This would then be plotted into the circumplex model plot of self-assessment manikin.

9 Results

Now that the test session has been finished, it was possible to take all the data gathered from the testing and questionnaire and evaluate the results. The testing was conducted on the campus of Aalborg University Copenhagen. 12 participants were gathered to play the game and fill in the post-questionnaire. They were all students at Medialogy.

9.1 Physiological Data

In order to score the data for each participants, the measured values of GSR recorded with the Empatica wristband were calculated using Excel (Microsoft, 1994). This software is useful in processing large quantities of data and accurately measure the level of which frustration was maintained throughout each participant’s play session. The standard deviation scores were then plotted into the following two histograms (Illustration 9.1 and 9.2). The mean level was calculated to be 3.904.

The box plot histogram illustrated below shows varied values of distributions within each participants. While some participants maintained a stable level of frustration, other participants had very different degrees of frustration during each play session. The graphed recordings of each participants GSR during the test session can be found in the appendix.

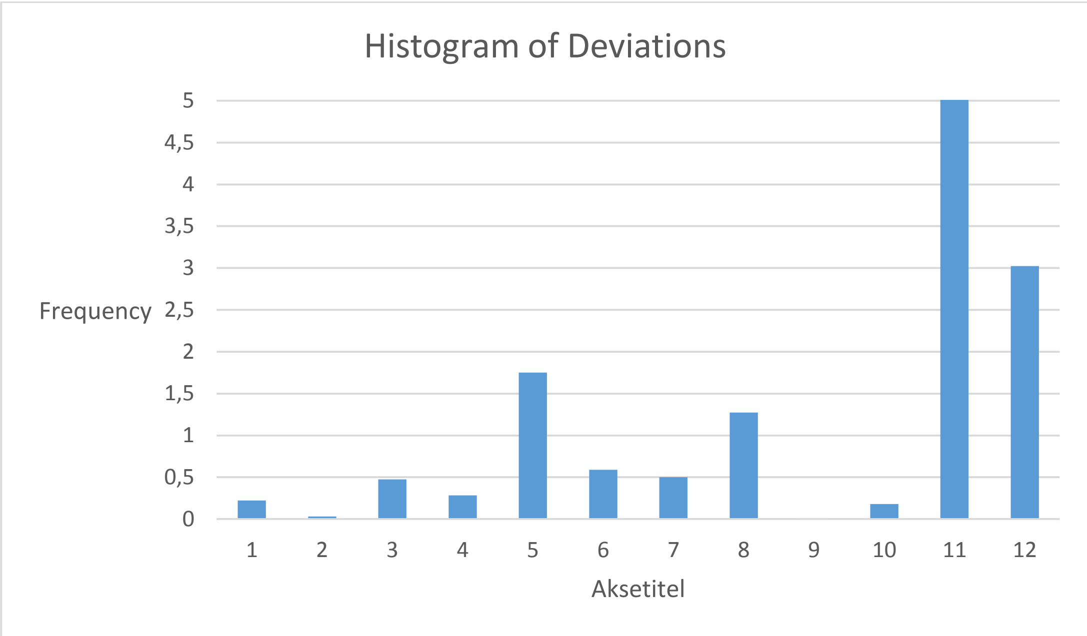

Figure 9.1 – Total score of standard deviation

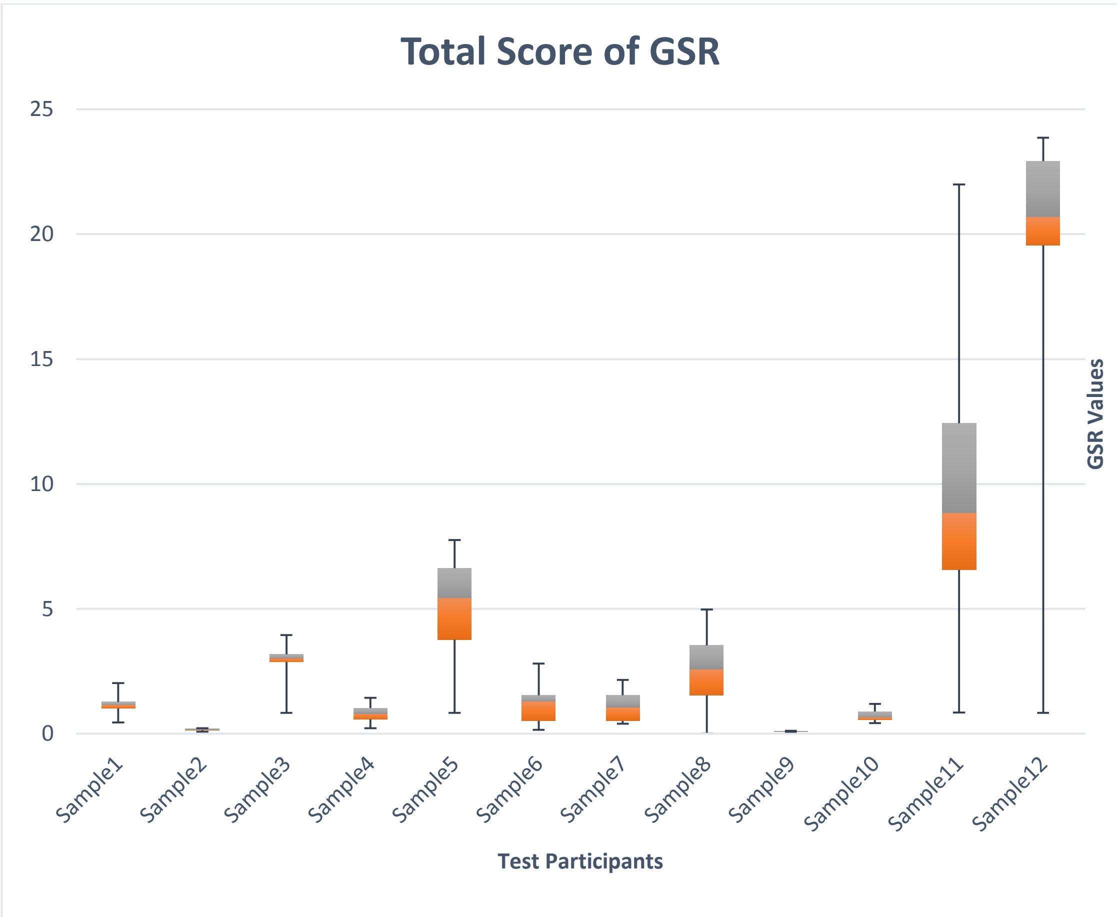

Figure 9.2 – Box plot of each participant's score

To see if the sample group is normally distributed, the Chi-Square Goodness of Fit Test was used.

P = 0.0144

9.2 Questionnaire

Of the 12 participants, most of them felt either indifferent or disagreed that the game felt easier to play when the game’s difficulty was decreased. Therefore, it was important to ask them what they think would make the game easier. One of them pointed out, that the player’s health could be increased. Another participant also pointed out that the order in which the fireballs fell could have been re-adjusted to be less random. Some player’s fell that the fireballs were unpredictable and hard to avoid. Giving them a rhythm to fall on would make it easier for the players to read their movements. All the filled in questionnaires can be found in the appendix.

9.3 Self Assessment Manikin

The data gathered from the Self Assessment Manikin shows that the majority of test participants are in the activated and the pleased area. The distributed data shows, that there are a few participants, who felt that they were in the unpleasant area after playing the game.

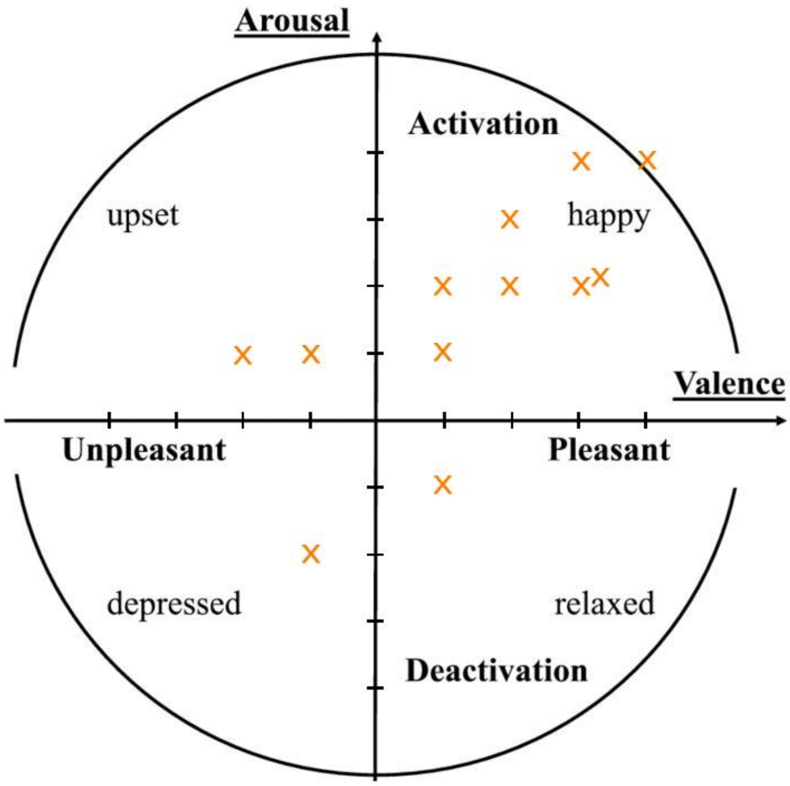

Illustration 9.3 - Circumplex model plot of Self-Assessment Manikin based on the data collected with color symbols.

10. Discussion

When looking at the gathered results, the setup and the predefined conditions for the testing sessions, there are several things to take into consideration, which could have made bias. To help future test design and conducting of testing conditions, the test will have to undergo some re-work. It would benefit the test not only to eliminate bias that would disrupt the data during each play session, but also have a larger sample group. Larger sample groups would also help increasing the validity of the results. At this point, it was hard to determine the validity of the results with a significance level of $p = 0.0144$

10.1 Room temperature

One reason for the high difference in the values of the GSR measuring could have occurred for different reasons. One reason could be that the testing sessions took place during spring season, and the hot weather outside could have affected the temperature of the room, meaning that it could have affected some test participants' level of GSR to some degree, and therefore affect the results.

10.2 Level of difficulty

One problem that might have affected the test overall was the level of difficulty. When a pre-test had been conducted before the final test, it was observed that all test participants were eager to avoid defeating the enemies and reach the end of the level as quick as possible. While the level had been purposely designed to eliminate this issue, some participants still had a tendency to take the risk and try to run past enemies, even if they failed numerous times. Perhaps the participants felt trying to defeat the enemies was too frustrating an experience and overall tried to avoid defeating the enemies. Or they were not aware of the fact that defeating the enemies would reward the participants with extra points. They they were informed of this function, it may not have been made clear enough to the participants. Perhaps the output generated from defeating the enemies was not satisfying enough to the player.

The game's usability also seemed to influence each participant's experience with the game positively. While the But it was concluded after the final test session that the game's level design should have been re-fined to focus around enemies solely instead of increasing the level of difficulty with obstacles. Though the intension was to incorporate obstacles with the enemies to increase the difficulty more and more as the player makes progress, some participants took longer to overcome certain obstacles than anticipated. This was not taken into account when determining the parameters that would be changed to decrease the level of difficulty. It would also benefit the DDA system to adjust more than one parameter within the game to decrease the difficulty and improve the experience with playing the game. Such parameters could have been the cannon enemies' rate of bullets.

11. Conclusion

This project sought to find out whether a DDA system based on a player's level of frustration was a proper method to adjust difficulty and over improve on the experience of a game.

Through some iterative design sessions, a game was designed and implemented to work with one form of a DDA system. This game would read a player's physiological, more accurately their GSR, and use the input to alter one parameter within the game which would decrease the game's difficulty to improve the experience and maintain the player's level of frustration. With this design in mind, it would be possible to either accept or reject the final problem statement: Can a game take player frustration as a valid data input through GSR, and dynamically alter the game's difficulty of a game, in order to improve the maintenance of a player's level of frustration and improve the game's experience of flow? Measuring of physiological may not be the most proper way to adjust level of difficulty within a game because of the difference in circumstances from each individual. This was observed during some of the testing sessions in the spring season. Because the weather was hot, and the testing room had no air-condition, it would cause the test participants to secrete sweat from their hands and disrupt the measuring of each player's frustration.

The conclusion that can be made from the test and the results is that a DDA system, which operates on a player's level of frustration, is not an appropriate method to adjust a game's difficulty. However, this is not a definitive conclusion. It would require a larger sample group to get more data in order to validate the p-value in order to accept or reject the null hypothesis. Based on the responses gained from the post-questionnaire, the game would also require a re-design to create a more dynamic DDA system that improves the overall experience with the game.

With that said, people who play games have different preferences when it comes to challenges, and they all have very different ways of playing games. Some people just like to have fun while playing, and some like to fail a challenge time and time again until the find the right solution to beat the challenge. Even if their frustration level increases exponentially, they crave for a pleasing emotion that only can be made from beating a hard challenge. Implementing a DDA system that reads on a player's physiological data shouldn't necessarily be a mandatory system built into every game, but it should be optional to use it for people who finds a game too hard and frustrating to play.

12. Bibliography

Hunicke, R., Designing Dynamic Difficulty Adjustment for Games. Ph.D. Thesis, Northwestern University, Evanston, IL, (forthcoming 2005).

Um Sang-Won, Kim Tea-Young, Choi Jong-Soo, (2007). Dynamic Difficulty Controlling Game System Transactions on Consumer Electronics.

Gilleade, Kiel M., (2004). Using Frustration in the Design of Adaptive Videogames. Lancaster University

Gilleade, K. M., & Dix, A. (2004). Using frustration in the design of adaptive videogames. In Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology

Carlson, Neil (2013). *Physiology of Behavior*. New Jersey: Pearson Education, Inc. ISBN 978-0-205-23939-9

Vorderer, P., & Hartmann T., and Klimmt, C., (2003). Explaining the Enjoyment of Playing Video Games: The Role of Competition. Proceedings of the Second International Conference on Entertainment Computing Carnegie Mellon University

Aponte, M.-V., Levieux, G., and Natkin, S. (2011). Difficulty in videogames: an experimental validation of a formal definition. Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology

Mourato F, Birra F, Prospero dos Santos, M, (2014). Difficulty in action based challenges: success prediction, players' strategies and profiling. Departamento de Sistemas e Informática, Escola Superior de Tecnologia

Juul, J. (2009). Fear of failing? the many meanings of difficulty in video games. The video game theory reader

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

Sweetser, P., & Wyeth, P. (2005). GameFlow: a model for evaluating player enjoyment in games. Computers in Entertainment, 3(3), 1-24. DOI=10.1145/1077246.1077253

Koster, R. (2005). A theory of fun for game design. Scottsdale, AZ: Paraglyph Press.

Feil, J., & Scattergood, M. (2005). Beginning game level design. Boston, MA: Thomson Course Technology.

Ekkekakis, P., (2012). Affect, Mood, and Emotion. Human Kinetics (http://www.public.iastate.edu/~ekkekaki/pdfs/ekkekakis_2012.pdf)

Russell, J.A., & Feldman Barrett, L., and Frijda, N.H., (2009). The Oxford companion to emotion and the affective sciences. New York: Oxford University Press.

Russell, J.A., & Feldman Barrett, L. (1999). Core affect, prototypical emotional episodes, and other things called emotion: Dissecting the elephant. Journal of Personality and Social Psychology.

D. L., & S. H. (n.d.). Using galvanic skin response measures to identify areas of frustration for older web 2.0 users. Retrieved April 26, 2010, from https://dl.acm.org/citation.cfm?id=1806032&dl=ACM&coll=DL

13. Figures