Aalborg University Copenhagen

Semester: 4th Semester (MSc) – Spring 2020

Title: The Impact of Basic and Active Interaction Schemes on Player Involvement

Aalborg University Copenhagen

Frederikskaj 12,

DK-2450 Copenhagen SV

Semester Coordinator: Stefania Serafin

Secretary: Lisbeth Nykjær
Phone: 99402470
iny@create.aau.dk

Semester No.: MED10

Semester Theme: Master's Thesis

Supervisor(s): Henrik Schønau Fog

Project group no.: N/A

Members:

Christoffer Nicolai Kiaby Kjær

Ervin Hamulic

Abstract:
This study investigates how appliances of basic and active-action based interaction schemes can be used in mobile games and how they impact player involvement in 2D-platformers through their unique input gestures that are based around basic tap input and the more mobile-centric input methods such as active swiping or dragging. An evaluation was performed using methodology mostly centered around methods such as questionnaires and card sorting. The findings show that there is a significant difference in kinesthetic and overall involvement between the basic and active action-based interaction schemes, in favor of the basic one. However, because of a sudden need for changing the evaluation strategies due to ethical concerns, certain important compromises had to be made, which lessened the legitimacy of the findings. Therefore, further testing should be conducted in order to enhance the validity of the results.

Keywords: Player Involvement, Active Interaction, Basic Interaction, Mobile Interaction, 2D-platformer, Playability Heuristics, Player Engagement

The Impact of Basic and Active Interaction Schemes on Player Involvement

Master’s Thesis

Members
Christoffer Kjær
ckjar15@student.aau.dk

Ervin Hamulic
ehamul15@student.aau.dk

Supervisor
Henrik Schønau Fog
hsf@create.aau.dk

Table of Contents

Table of Contents **1**

1. Introduction **3**
1.1. Initial Problem Statement 4

2. Analysis **5**
2.1. Mobile Interactions 5
2.1.1. Mobile Interaction Techniques 6
2.2. Mobile Touch Interaction 9
2.2.1. Active Actions 9
2.2.2. Basic Actions 10
2.2.3. Virtual Controllers 11
2.3. Design Principles in 2D-platformers 13
2.3.1. Building blocks 13
2.3.2. Design patterns 14
2.4. Measuring the User Experience in Games 16
2.4.1. Playability Heuristics 16
2.4.2. Player Focus and Flow Theory 18
2.4.3. Player Engagement 19
2.4.4. Player Involvement 21
2.4.4.1. The Six Dimensions of Player Involvement 22
2.5. State of the Art 24
2.5.1. Virtual Controller Designs 24
2.5.2. Active Action Controls 25
2.6. Analysis Conclusion 26

3. Final Problem Statement **28**
3.1. Design Requirements 28

4. Methods and Evaluation Strategies **29**
4.1. Ethical Concerns 29
4.2. Usability Testing 30
4.3. Pilot Testing 30
4.4. Final Evaluation 31
4.4.1. T-test and Questionnaire 31
4.4.2. Card Sorting 32
4.4.3. Observations 33

5. Design **34**
5.1. The Concept and Visuals of Dark Squares 34
5.2. Components 34
5.3. Level Design 35

5.3.1. Rhythmic Jumping 38
5.4. Input Design 38
5.4.1. Basic Action Input 39
5.4.2. Active Action Input 40
5.5. Sound Design 41

6. Implementation 42
6.1. Detecting Interaction 42
6.1.1. Detecting the Gestures 42
6.1.1.1. Tapping 42
6.1.1.2. Swiping 43
6.1.1.3. Dragging 45
6.1.1.4. Buttons 45
6.2. Player Actions 46
6.3 Changing the Environment 47

7. Results 50
7.1. Usability Test 50
7.1.1. Setup and Procedure 50
7.1.2. Results 51
7.2. Pilot Test 52
7.3. Final Evaluation 53
7.3.1. Demographics 53
7.3.2. Procedure 54
7.3.3. Questionnaire and Statistical Test 55
7.3.4. Cards Sorting Results 61
7.3.4.1. Card Frequency Results 61
7.3.4.2. Card Sorting Score 65

8. Discussion 67
8.1 Sources of Error 68

9. Re-design 70

10. Conclusion 72

11. Future Works 74

References 76

1. Introduction

Video games have long been a stable form of digital entertainment. This is not only because it allows the player to actively take part in the experience but also because it has the capabilities to reach a wide audience through many different types of platforms such as computers, consoles, tablets and mobiles. Particularly the mobile platform, which include smartphones, smartwatches and other handheld devices, has over the last decade become the most widely used digital platform available (StatCounter, 2020).

Along with the rise in popularity of the smartphone, a growing number of video games tailored to the mobile platform has also seen the light of day. While some of these mobile games present input controls that are carefully tailored to fit the mechanics of the experience, other games instead rely on traditional input schemes that are heavily inspired by console controllers. By merely adopting these traditional input schemes, mobile games exclude some of the platform's many available features such as the gyroscope, camera, and a wide range of touch gestures, which is unfortunate since making use of these features could potentially enhance the player experience.

Certain developers have attempted to tailor the input scheme to fit the mechanics of their game, while also adhering to the traditional scheme. However, utilizing this approach can result in a range of issues, such as overuse of UI elements, which can become too intrusive and thus negatively affect the player's experience with the game. However, while certain games hold onto these traditional input schemes, other games have adapted to the mobile platform by utilizing the many features it possesses. This raises the question: is the traditional approach of simply adopting the input scheme from the console controllers really the most optimal way of designing input controls, or does embracing the alternative features of the smartphone enhance the player experience?

This study aims to investigate how the level of player involvement is influenced when comparing certain mobile interaction schemes. To achieve this, the study will explore how mobile interaction schemes are built and applied to a game. In order to measure player involvement, the study will investigate said subject along with other related topics. With this knowledge, a 2D-platformer and evaluation methodology will be designed to answer how different mobile interaction schemes influence player involvement.

1.1. Initial Problem Statement

This leads to an initial problem statement being formulated.

Which type of interaction schemes induce a higher sense of player involvement in terms of basic gameplay actions in a mobile 2D-platformer?

2. Analysis

The following chapter aims to research several aspects related to the initial problem statement in order to acquire useful knowledge on how to properly develop a 2D platformer that will be suitable to function with different types of mobile interaction schemes. The analysis will cover elements such as mobile game interaction techniques, touch interaction techniques, how to design a 2D-platformer, player involvement and other models for user experience, how user experience is measured, as well as current state-of-the-art as these are considered related to the evaluation performed in this study.

2.1. Mobile Interactions

In order to determine which type of mobile interaction techniques that should be evaluated it is important to understand which ones exist, how they are used as well as how widely used each of them are.

Game interaction, as a concept, refers to the relationship between the user's input and the response outputted by the smartphone. The general relationship is based around the smartphone continuously taking haptic input through either finger touches on the screen or the device being shaken; or by taking auditory input through the built-in sound recorder. The smartphone can then output a wide range of visual feedback, audio cues, or haptic vibrations, which the user can act upon and then provide the smartphone with new input (Chaichitwanidchakol and Feungchan, 2018).

Figure 1. The relationship between user and device (Chaichitwanidchakol and Feungchan, 2018).

Savari et al. (2016) state that game interactions can be divided into two types, with the first type being known as natural interactions. This concept refers to techniques that attempt to mimic real life interactions through body movements that are similar to those performed in the physical world. These movements are then captured by hardware which can potentially simulate said movement in a virtual environment or on a screen (ibid). Virtual reality and the Xbox Kinect are common pieces of hardware that utilize natural interactions. The difference between these two particular pieces of hardware is that virtual reality uses physical controllers and sensors to capture and simulate body movements, while the Xbox Kinect eliminates any need of a physical controller and instead relies on a camera with sensors to capture body movements. The second type is called non-natural interactions, and this form of interaction revolves around techniques that are performed through more conventional means such as the usage of a mouse, keyboard, joystick, or console controller (ibid.)

Both of the aforementioned types of interaction can be utilized for mobile games. Many of today's smartphones do not have any physical buttons since on-screen buttons are used for input commands (Chaichitwanidchakol and Feungchan, 2018). However, physical buttons or controllers such as a keyboard or a joystick can be utilized through a Bluetooth connection, and then used for non-natural interactions. Moreover, most of today's smartphones are equipped with many different sensors that can allow for implementation of more natural forms of interaction (ibid.)

2.1.1. Mobile Interaction Techniques

Chaichitwanidchakol and Feungchan (2018) view the concepts of natural and non-natural interaction as more general categorizations of game interaction. They argue that mobile interaction could be categorized into many interaction techniques that each utilize a single (or multiple) features of today's smartphones.

Touch

The touchscreen is one of the most standard methods for providing input on today's smartphones, and it allows the users to use their fingers to perform various actions. These actions include a wide range of touch gestures such as, tapping, double tapping, long tap (holding), dragging and flicking; all of which can be used for unique gameplay purposes (Kim and Lee, 2014). The concept of multi-touch can provide even more variation in touch gestures, by allowing users to perform actions by placing two or more fingers on the screen simultaneously (Chaichitwanidchakol and Feungchan, 2018). Though not seen on many of today's smartphones, innovative touch technology such as pre-touching (which can detect users' fingers before they touch the screen) (Hinckley et al, 2016), and rich touch (which can detect whether the player is touching the screen with their fingers, nails or knuckles) (Harrison, 2014) do exist and can allow for even more unique touch gestures .

Motion

Motion interaction revolves around physically moving the smartphone as a way of providing it with input, and this is done through built-in sensors such as the accelerometer or the gyroscope. One of the more popular examples of motion detection on smartphones is *tilting*, which works by the sensors detecting declination and inclination of the smartphone (Du et al, 2011; Gilbertson et al, 2008). Tilting is sometimes used in driving games to turn vehicles or in classic maze-like games where players have to guide a marble ball through a maze. While tilting-based motion controls might not be optimal for many games; it has been found that motion controls can induce a higher level of player satisfaction compared to conventional touch controls (for games that allow players to use either of the two) (Gilbertson et al, 2008). Examples of non-gaming applications that use motion detection include 3D viewing-modes and standard camera applications.

Video

Video interaction concerns itself with utilizing the camera as an important tool for interaction. It is often used by applications that aim to combine real-world and virtual imagery into a singular digital experience. The concept is more commonly known as augmented reality, and it allows the user to view the combined imagery through the smartphone’s camera (Das et al, 2017). It is however important to note that augmented reality applications usually do not rely on video interaction alone. This is because much of the actual interaction still occurs through conventional touch controls, while the video-aspect focuses more on how to present the imagery. Popular examples of mobile games that use augmented reality are Pokémon Go! (2016) and Harry Potter: Wizards Unite (2019) by *Niantic*.

Location

This type of interaction utilizes the player’s location as a key element in its gameplay through GPS tracking. The aforementioned mobile game Pokémon Go! (2016) is an example of a location-based game that takes into account where in the world the player is so that it can determine which Pokémon they should be able to find in their vicinity. Smartphones built-in magnetometer is also used in location-based interaction to determine which direction the player is looking (Chaichitwanidchakol and Feungchan, 2018). However, just like video interaction, location-based games often still rely on touch input from the player.

Sound

Sound interaction revolves around the player providing the smartphone with input through voice commands that are detected by its microphone. Sound interaction can be based around concepts such as speech recognition, where the player can e.g. make their tetrominoes in a game of tetris drop down by saying the word “down” (Sporka et al, 2006); or sound processing, which is often used in karaoke games where the input is more focused on the quality of the sound than the content of it (Gokul et al, 2016).

Other techniques

Other techniques highlighted by Chaichitwanidchakol and Feungchan (2018) include, social interaction (where multiple users are competing or cooperating in the same experience); date and time interaction (where the game can take into account the time of the day from the real world and then mimic it in the game); weather interaction (where e.g. in Pokémon Go (2018), water type Pokémon will be stronger if it is raining in real life); bioinformatics interaction (where biodata such as heart rate can be used as a gameplay interaction feature); and special purpose interaction (where an external device is connected to the smartphone as an alternative way of interaction), among others (ibid).

A general pattern that can be seen in a few of these mobile interaction techniques is that many of them rely on touch controls even though their name does not imply it. For example, while video and location interaction utilize the camera, magnetometer and GPS in order to function, these still rely on touch input, while the aforementioned features act almost like secondary elements to the interaction (see figure 2). Based on these findings, it is possibly to state that touch plays a much larger role in general mobile interaction than other techniques. It would therefore be more suitable for the techniques evaluated in this study to be influenced by some form of touch input, since it can be viewed as the most utilized interaction technique.

Investigated Interaction Techniques	**Utilized Smartphone Features**
Touch	Touchscreen
Motion	Gyroscope, accelerometer
Video	Touchscreen, camera
Location	Touchscreen, GPS
Sound	Microphone

Figure 2. An overview of the investigated interaction techniques and the smartphone features each of them utilize for proper interaction.

2.2. Mobile Touch Interaction

Before designing a game that utilizes touch techniques as input, it is important to determine which specific types of touch techniques should be investigated. It is furthermore also important to understand how input interfaces for the chosen techniques are designed, as well as what limitations that come with touch interaction.

As mentioned in section 2.1.1, the various forms of touch interaction techniques include tapping, double tapping, long tap (holding), dragging, flicking and multi-touch. Kim and Lee (2014) state that the aforementioned touch techniques can be divided into two categories; basic and active actions. Single tap, double tap, long tap (holding) and multi-touch tap are considered basic actions, while dragging, flicking, rotation and free motion are defined as active actions. The active actions are defined as such because performing them requires an open motion where the finger interacts with multiple areas of the touchscreen, while the basic actions revolve around a closed motion, since these require the player to only interact with a single point on the touchscreen (ibid). Apart from the motion involved in the actual gestures, basic and active touch techniques also differ in the fact that the former might require visually represented input buttons, while the latter can easily function without buttons (Insight, 2020).

2.2.1. Active Actions

As previously mentioned, active actions revolve around touch interaction techniques that require the player to touch multiple areas of the screen without breaking contact before the end of a gesture. These included swiping, dragging, rotation, and free motion. Contrary to basic action techniques, active action techniques allow for a cleaner user interface that is not cluttered with buttons, since a gesture such as swiping usually can be performed anywhere on the screen, and thus does not need any visual indicator (Insight, 2020).

This is the case in games such as Duke Dashington (2014) by *Adventure Islands*, Angry Birds (2019) by *Rovio Entertainment*, Vector 2 (2016) by *Nekki*, and Fruit Ninja (2010) by *Halfbrick Studios* (see figure 3); all of which include active actions as the core interaction technique (see figure 2). Furthermore, gestures such as dragging and swiping are directional, which can allow developers to couple specific directional gestures to a specific action, which will then provide developers with more options when having to implement more advanced controls (Insight, 2020).

One of the downsides of active actions is that some of these gestures are relatively similar. Gestures such as swiping, dragging and free motion all involve the player's finger being moved across the screen in a certain pattern or direction; and at a certain speed. Dragging and swiping can e.g. be very similar in their movement pattern and direction, whereas the time it takes to perform the gestures is what sets them apart (Kim and Lee, 2014). It is therefore very essential that the smartphone can properly identify a specific action and act accordingly, since not doing so will provide the player with an undesired output.

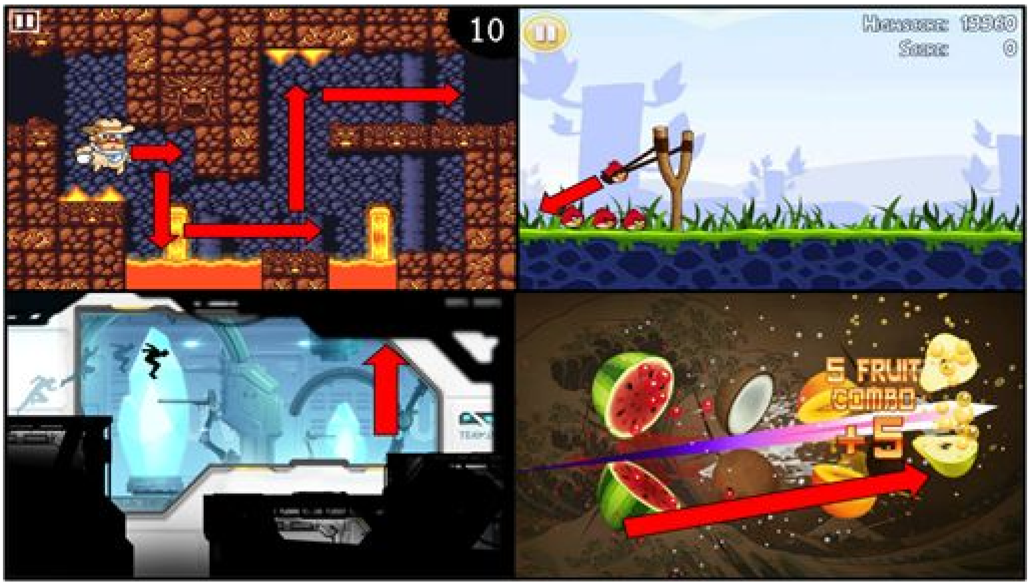

Figure 3. Duke Dashington (upper left) asks players to perform a directional swipe in order to dash through levels, while Angry Birds (upper right) asks players to drag and aim a slingshot toward destructible structures. Vector 2 (lower left) and Fruit Ninja (lower right) ask players to parkour past obstacles and cut fruit, respectively, through swiping motions.

2.2.2. Basic Actions

Touch interaction techniques that involve single tap, double tap, long tap (holding), and multi-tap gestures are collectively known as basic actions (Kim and Lee, 2014). One of the strengths of games that utilize these tap-based methods for input, is that they will (in most cases) provide players with a clear visual representation of where they should tap in order to provide the smartphone with the desired input (Insight, 2020). Individual buttons can also be tailored to respond differently to specific types of input; such as providing different output depending on whether the button was tapped a single or two times.

There is however, only a limited amount of ways to tap a button, which means that the amount of functionality that can be coupled with each individual button is limited. The downside of basic actions is that if a mobile game is supposed to have advanced controls, it needs a lot of buttons in order to function (ibid). This will present an issue due to the smartphone's screen size, which has to display both the actual game as well as the input buttons. Ideally, the number of buttons used for gameplay input therefore have to be limited to only what is absolutely necessary; otherwise the button layout will take up too much screen space and thus block the visuals of the game (Kim and Lee, 2014).

Many mobile games that use basic action techniques (or a combination of basic and active techniques) for interaction can utilize what is known as a virtual controller when designing the virtual button layout.

2.2.3. Virtual Controllers

The concept of a virtual controller revolves around eliminating the need for an external physical controller by making the smartphone itself act as the physical controller. This means that the button layout on the smartphone's screen is reminiscent of more conventional controllers. Mimicking the button layout from other well-known controllers, virtual controllers will place the movement controls on the left side of the screen, while action buttons are located on the right side (ibid). Examples of such a virtual controller can be seen in mobile fighting games such as Street Fighter IV (2010) by *Capcom* and Hero Versus (2019) by *Creative Bytes Studios*, which attempt to mimic the button layout of console controllers and old arcade machines, respectively (see figure 4). The familiarity gained from usage of console controllers or arcade machines can therefore allow players to more easily get accustomed to the controls of a virtual controller-based mobile game (ibid).

While it can definitely be viewed as beneficial that no external devices are needed, virtual controllers do possess a few limitations that can hurt the user experience if not taken into account during the design process. When a mobile game requires more than four input buttons for actions, the developer can no longer replicate the button layout of standard controllers and this can create three issues. Firstly, the virtual buttons will have to be smaller since the increased number of buttons will otherwise become too obtrusive. Park and Han (2010) found that virtual tap-buttons should be at least 8-9 mm in size, while Conradi, Busch and Alexander (2015) found that buttons of about 8*8 mm is the lowest size that does not result in too many unintentional inputs. Second, the space between the buttons will also have to be decreased due to the same reason. Hobart (2005) state that having at least 3 mm in between each virtual tap-button is recommended for better input performance, since lowering the size of the buttons or the distance between them could result in unintentional button taps and thus cause a decrease in player satisfaction (Kim and Lee, 2014). Finally, having more than four buttons means that the developer has to design a more unique button layout that might never have been seen before and thus could take longer for players to learn (ibid). Dolls Order (2018) by *Gumi* and KOF: Allstar (2018) by *Netmarble* are examples of mobile games where the number of action buttons surpass four. For these two specific titles, the action buttons are squeezed together and placed in a much more unconventional layout that does not follow any known pattern (figure 4).

Figure 4. Four different virtual controllers in mobile games. Street Fighter IV (upper left), Hero Versus (upper right), Dolls Order (lower left) and KOF: Allstar (lower right). The red dot represents where the virtual (movement/direction) analog stick is placed, while the yellow dots represent action buttons.

Kim and Lee (2014) argue that merely adopting the button layout from more conventional controllers should not necessarily be viewed as an advantage for virtual controllers but rather an issue, since it means that the game does not take full advantage of the smartphone's many features. In their study, they found that adding other interaction functionalities (such as swiping), to an otherwise purely tap-based interaction scheme, resulted in increased level of interest and fun that was not found when previously restricted to only tap-based interaction (ibid). This shows that combining basic and active actions can lead to an increase in player satisfaction, and that developers should not necessarily stick to a single type.

Many virtual controllers do however, utilize a combination of basic and active action techniques. The movement controls that are usually placed on the left side of the screen can e.g. either be individual directional buttons or a drag-based joystick, with the latter attempting to mimic the analog-sticks that are found on console controllers. A study has shown that individual direction buttons require more attention from the player compared to a drag-based joystick, but also that the joystick has a tendency to cause more unintentional operations (Baldauf et al, 2015).

Based on these findings, it is possibly to state that there are two broad categories of touch interaction; basic and active actions. It could therefore be argued that the two touch interaction schemes that should be evaluated in terms of player involvement should be based on basic and active actions, and that the basic action scheme should be designed with the virtual controller and its limitations in mind, since allowing player to quickly learn the controls is important when having to perform an evaluation.

2.3. Design Principles in 2D-platformers

In order to properly design a 2D-platformer that can help evaluate the chosen touch interaction techniques, it is important to investigate how games within this genre of video games are designed in terms of game mechanics, challenges, level layout and other important aspects of 2D game design.

2.3.1. Building blocks

Smith, Cha and Whitehead (2008) developed a framework that aimed to explore the different elements that make up many popular 2D-platformer games. Their framework presents a hierarchy that is made up of two sections: components, and structural representations.

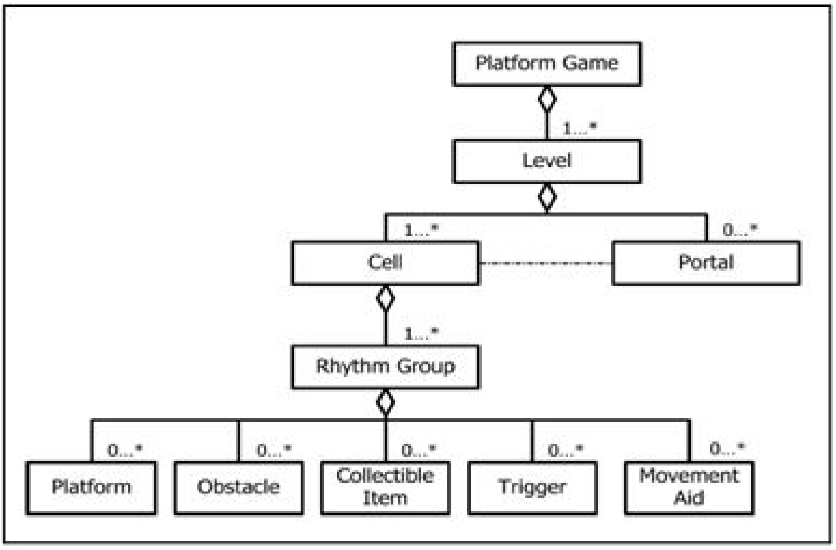

Figure 5. Framework for levels in 2D-platformers (Smith, Cha and Whitehead, 2008)

The components refer to individual objects and features that are found within a 2D-platformer level and these make up much of the gameplay experience. Smith, Cha and Whitehead (2008) categorize these as: platforms, obstacles, collectible items, triggers and movement aids. *Platforms* refer to objects that the player can safely traverse in order to progress the level, while *obstacles* refer to elements within the level that are trying to halt the player's progress (ibid). These can include enemies, gaps that the player has to jump across, or objects that are partially blocking the way forward. *Collectible items* deal with objects within the level that can be picked up and reward the player in some form. *Triggers* refer to objects or mechanics within the game that alters the state of the level and thus might change the gameplay principles, while *movement aids* are a reference to elements, such as ropes or springs, that aid the player in progressing the level (ibid). The concept of structural representations revolves around the structure of 2D-levels. The framework categorizes these structures as cells, portals and rhythm groups. *Cells* refer to linear gameplay sections where the player only has a single way forward, while *portals* are sections where these linear sections meet and then potentially provide the player with multiple new linear paths that they can choose from. *Rhythm groups* are short sections within each cell that contain a higher degree of challenge through components such as obstacles (ibid). These rhythm groups are usually separated by sections of less challenge where the player can rest.

2.3.2. Design patterns

This resting-section is also mentioned by Khalifa, Silva and Togelius (2019), as one of the six common level design patterns in 2D games, defined as *safe zones*. They define these as sections within a 2D-level where players are not exposed to any enemies or obstacles, but instead given a breathing space where they can analyze their surroundings and plan how to overcome the next obstacle. Safe zones can also be used as checkpoints in case the player fails a challenge (ibid). Another pattern is *foreshadowing*, which revolves around gameplay mechanics being introduced to the player in a controlled environment before challenges are met where the mechanics are required in order to proceed. The third pattern is called *pace breaking*, and deals with increasing or decreasing tension through e.g. subtly changing the music or visual of the game to fit the current or incoming change in tension. Calm music could e.g. imply that the player has reached a safe zone while more action-packed music could signify that danger is close by and that the player has to prepare for a tough challenge (ibid). *Guidance* is the fourth pattern, and deals with principles of guiding the player down certain paths through attention cues that will draw players toward secrets, collectibles or the final objective. These cues can be based on sounds (Peerdeman, 2010), lighting, (Dickley, 2015), motion (Abrams and Christ, 2003) or color; particularly colors such as red, yellow, green and pink are known to more easily catch human attention (Gelasca, Tomasic and Ebrahimi, 2005). Other patterns include *layering*, where players are introduced to harder challenges without new elements being added to the game, and *branching*, where players are provided with multiple paths to reach an end goal (Khalifa, Silva and Togelius, 2019).

Reyno and Cubel (2013) define the 2D-platformer genre as *"[…] a game genre characterized by a protagonist who moves and jumps into platforms, collecting prizes and destroying enemies in various ways […]".* Particularly the aspect of *jumping* is at the heart of the 2D-platforming experience, since this is the main method for the avatar's traversal in many platformers (Smith, Cha and Whitehead, 2008). Similar to how Smith, Cha and Whitehead (2008) define short sections of obstacles as rhythm groups, Compton and Mateas (2006) likewise state that platforming games rely on rhythm since rhythmic actions can aid the player in achieving a state of flow; a mental state where the player will be completely absorbed in the experience (Csikszentmihalyi, 1990).

Compton and Mateas (2006) present four different types of design patterns that can aid developers in e.g. designing levels and obstacles with rhythm in mind. The first is called the *basic pattern*, which presents the player with a sequence of obstacles that have no variation. *Complex patterns* likewise present a sequence of similar obstacles but with a slight change with each obstacle. This could e.g. be a series of gaps that the player has to jump across but each consecutive gap is larger than the former. *Compound patterns* take two different basic patterns and alternate between them. For example, the player could be faced with first having to jump three gaps, then dodge three incoming enemies, and then jump three more gaps. The final pattern presented by Compton and Mateas (2014), is the *composite pattern*. This pattern combines two challenges in a single one, and forces the player to take into consideration multiple actions in order to overcome this combined challenge. The player can e.g. be challenged to jump across a gap in the platforms, while simultaneously also having to dodge an incoming enemy projectile.

The works presented in this chapter should be viewed as helpful building blocks that can be utilized to develop and design a proper 2D-platforming experience that can be used to evaluate the chosen touch interaction techniques. The patterns presented by Khalifa, Silva and Togelius as well as those presented by Compton and Mateas (2006) can be useful in designing the structures of the 2D-levels, while the framework presented by Smith, Cha and Whitehead (2008) gives insight into the individual building blocks that make up a 2D-platformer.

2.4. Measuring the User Experience in Games

Now that two interaction patterns have been chosen, a way to measure a difference has to be explored. This has been attempted through certain methods such as the player involvement model, engagement model and usability-based models, and these methods all measure the player experience through different ways. This chapter will explore how the aforementioned models measure player experience while also investigating the theory that defines the different measurement categories within each model.

2.4.1. Playability Heuristics

General usability techniques have been used to improve products; be it games, physical products etc. Jakob Nielsen created usability heuristics to aid developers in identifying areas of a product that can be improved (Nielsen & Adelson, 1994). These heuristics describe avenues that can be explored and provide a foundation for what designers have to consider when developing a product. These heuristics have been further built upon to better model specific platforms such as mobile phones. This has e.g. been necessary because of the mobility factor of the mobile platform (Korhonen & Koivisto, 2004).

Korhonen and Koivisto (2004) created updated heuristics centered around the mobile game experience. These heuristics placed focus on three main categories; the game usability, mobility and gameplay heuristics (see figure 6). These categories furthermore specify multiple heuristics for each of the aforementioned categories (ibid).

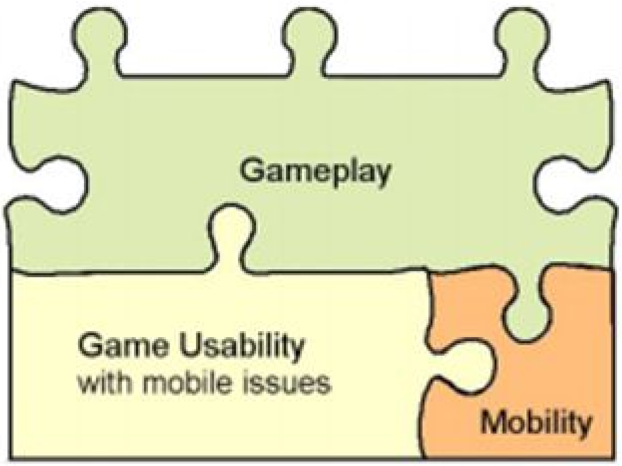

Figure 6. Model of the three playability heuristics (Korhonen and Koivisto, 2004).

The game usability heuristics focus on the controls and interface that the player is exposed to. As a general rule, it is recommended that the player is able to maneuver the avatar without issues and have access to all necessary information required to complete the task such as when the character can jump, what should the character avoid etc. (Korhonen & Koivisto, 2004). In the table (see figure 7), the GU1-GU5 heuristics are centered around visual design and how information is presented, while GU6-GU8 heuristics revolve around the control scheme and navigating the character. The remaining heuristics are about the help that players have access to and the feedback that the game provides.

The gameplay heuristics (see figure 7) are centered around the gameplay and aspects that contribute to the overall game experience. These heuristics are inspired by a concept such as flow theory (Csikszentmihalyi, 2014) (see section 2.4.2). These elements also appear in other models that are not directly tied to usability but more centered around descriptors such as engagement, as is the case with e.g. GP1 (see section 2.4.3).

The mobility heuristics are heavily centered around the platform. This is evident with MO2 and MO3 that encompass troubles a mobile platform might have such as outside lighting, prioritizing phone calls, unexpected pauses or looking out of place when using it in public (Korhonen & Koivisto, 2004). All of these issues are tied to the mobile platform and it is important that these issues are addressed early in the design process.

No.	Game Usability Heuristics
GU1	Audio-visual representation supports the game
GU2	Screen layout is efficient and visually pleasing
GU3	Device UI and game UI are used for their own purposes
GU4	Indicators are visible
GU5	The player understands the terminology
GU6	Navigation is consistent, logical, and minimalist
GU7	Control keys are consistent and follow standard conventions
GU8	Game controls are convenient and flexible
GU9	The game gives feedback on the player's actions
GU10	The player cannot make irreversible errors
GU11	The player does not have to memorize things unnecessarily
GU12	The game contains help

No.	Mobility Heuristics
MO1	The game and play sessions can be started quickly
MO2	The game accommodates with the surroundings
MO3	Interruptions are handled reasonably

No.	Gameplay Heuristics
GP1	The game provides clear goals or supports player-created goals
GP2	The player sees the progress in the game and can compare the results
GP3	The players are rewarded and rewards are meaningful
GP4	The player is in control
GP5	Challenge, strategy, and pace are in balance
GP6	The first-time experience is encouraging
GP7	The game story supports the gameplay and is meaningful
GP8	There are no repetitive or boring tasks
GP9	The players can express themselves
GP10	The game supports different playing styles
GP11	The game does not stagnate
GP12	The game is consistent
GP13	The game uses orthogonal unit differentiation[4]
GP 14	The player does not lose any hard-won possessions

Figure 7. Gameplay heuristics (right), game usability heuristics (upper left) and mobility heuristics (lower left).

2.4.2. Player Focus and Flow Theory

Since many theories attempt to define user experience based on aspects related to Csikszentmihalyi's flow theory, this concept will be covered by itself in this section. This will create an overview of the theory behind the other models mentioned in this chapter. Csikszentmihalyi (1990) mentions that a flow experience produces differentiation and integration which are two psychological processes. When the experience is over the ''self'' becomes differentiated, which means that the user managed to acquire new skills during the experience. This leads to integration as feelings, thoughts and senses are being focused towards the task the user is performing. According to Csikszentmihalyi (2014) there are certain conditions that have to be met in order for players to enter a state of flow.

These conditions involve a few important design requirements that have to be taken into account in order for the product to let the player reach a state of flow. These requirements are:

- A clear set of goals
- A balance between perceived challenges and perceived skills fig (8)
- Clear and immediate feedback

When flow is achieved there are certain signs that can be observed from the player that indicate that a state of flow has been achieved:

- Altered sense of time
- Forgetting their surroundings
- A sense of control of action

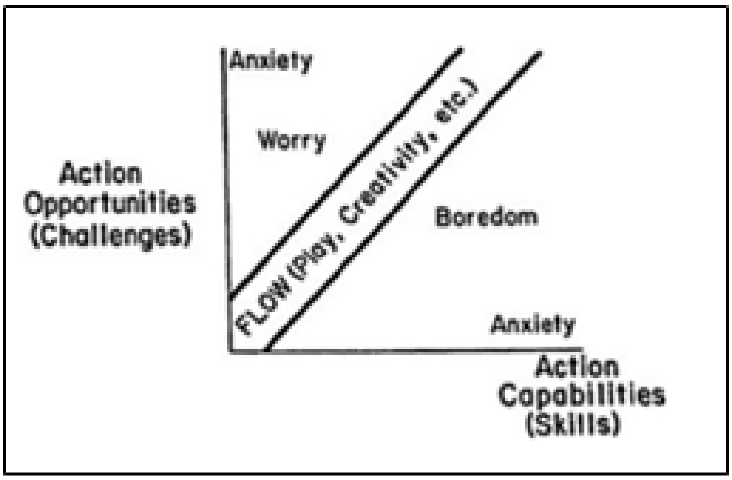

Figure 8. The relationship between player skill and level of challenge.

With these observable elements, there are clear behaviors that indicate a state of flow and these are often related to a positive user experience in other models such as player engagement (see section 2.4.3) and player involvement (see section 2.5.3).

2.4.3. Player Engagement

Within interactive entertainment, player engagement is closely related to concepts such as immersion, presence, enjoyment and flow (Schønau-Fog, 2011). Brown and Cairn (2004) define engagement as the first step of immersion; a stage that is centered around the player investing time into the interactive experience, as well as the effort and attention they exert (Brown and Cairn, 2004). In order to enter this first stage of engagement however, players have to overcome two barriers (ibid). The first barrier is about the player's preference. Here, it is important that the player e.g. enjoys the visual style, gameplay controls, overall genre etc, since not doing so will immediately disengage the player. The second barrier revolves around player investment in the form of losing track of time, putting effort into their actions and becoming focused. The elements of the second barrier are closely aligned with Csikszentmihalyi's (2014) flow theory which is used to determine and define the state of flow (see section 2.4.2).

Schønau-Fog (2012) presents the OA3 model as a framework that describes how player engagement works and which elements of games that can successfully induce engagement. The OA3 model (see figure 9) describes four core categories that contribute to player engagement; objectives, activities, accomplishments and affect.

Objectives

There are two types of objectives in video games, and these are called extrinsic and intrinsic objectives. Extrinsic objectives are objectives or goals that are created by the designers of the game. These are often delivered through story missions, side missions, collectibles etc. (Schønau-Fog, 2011). The intrinsic objectives refer to player-created goals and objectives. These can include seeing a landmark in the distance and wanting to reach it, or completing a game in a certain way such as playing on the highest difficulty or wanting to reach the ending in a specified amount of time; deliberately making the game harder for themselves even if they do not have to.

Activities

The OA3 model defines multiple activities that player's can perform. Since this study is centered around certain types of input on a mobile phone, the activity of interfacing will be the primary focus in this study. Interfacing is centered around how the user is in control of the game, as well as the physical actions of the protagonist. Whether or not the player becomes engaged or disengaged through this activity is dependent on their subjective view of the quality of the designed interfacing (Schønau-Fog, 2011). Other activities within the OA3 model, such as solving problems and sensing are relevant for this study and contribute to the player's overall level of engagement, but since these can be considered secondary to the purpose of this study, it is important to primarily focus on interfacing.

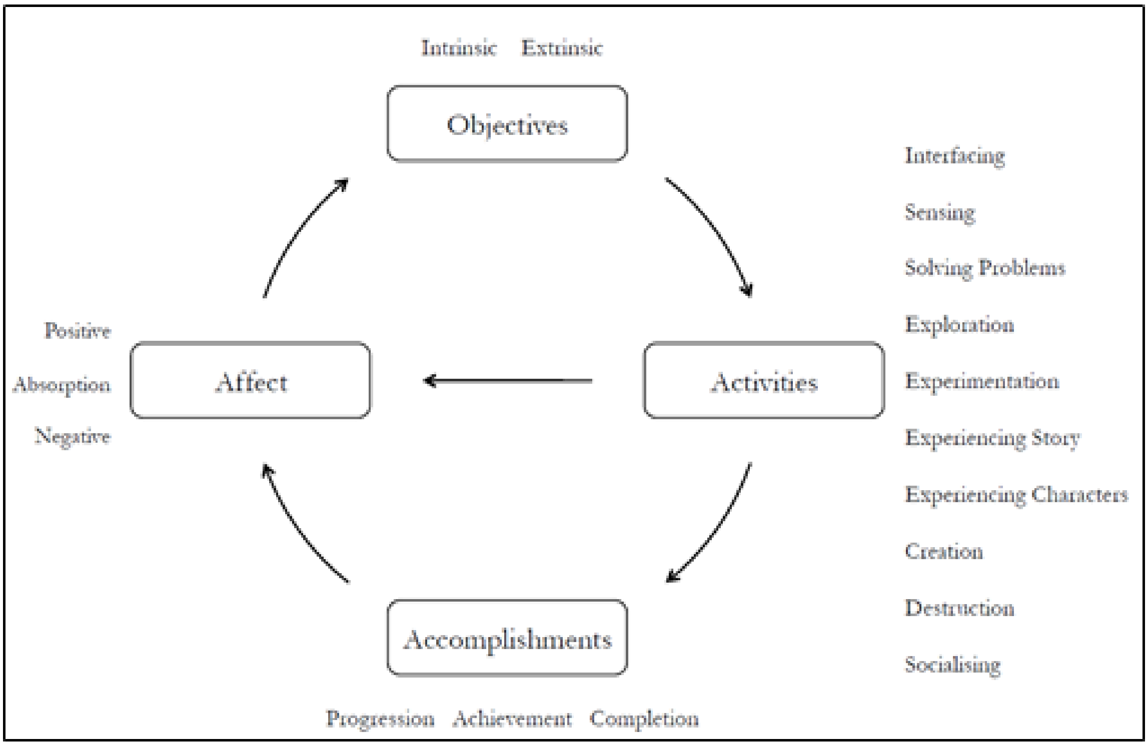

Figure 9. The OA3 model of engagement.

Accomplishments

Accomplishments in the OA3 model is centered around what happens after the player completes objectives. The OA3 model distinguishes between three types of accomplishments; progression, achievement and completion. Progression and achievement are closely related and revolve around the player's progress throughout the game in the form of leveling and receiving new and better equipment that can be used to complete increasingly difficult challenges (Schønau-Fog, 2011). Completion is centered around the player's desire to complete the game and witness the ending or to complete every single piece of content that the game has to offer.

Affect

Affect is about the emotions experienced by the player during activities. These felt emotions can both be positive and negative. Positive emotions can be enjoyment, excitement, fulfillment etc. and these emotions can be induced by completing extrinsic and intrinsic objectives. The negative emotions can be caused due to boredom, confusion or frustration and can result in the player becoming disengaged from the game. However, these negative emotions are not necessarily bad. Frustration from not being able to complete a very difficult challenge can lead to engagement through the determination of wanting to complete said challenge.

The OA3 model highlights many different elements of video games that can ensure a high level of player engagement. This study will focus especially on changes in the interface activity and how two types of interfacing in terms of basic and active control schemes can impact the overall experience.

2.4.4. Player Involvement

In order to measure a potential difference in user experience between two control schemes, a predefined method has to be used, and a potential method for doing so is Calleja's player involvement model (Calleja, 2007); a framework that explores how to measure user experience through clearly defined dimensions of involvement. The player involvement model can be used to describe what type of involvement the player is experiencing when interacting with a game.

The model is split into six dimensions and two temporal phases (Calleja, 2007), and both of these categorizations contribute to the phenomenon known as incorporation, which is set in the middle of the involvement model (see figure 11). No single dimension is working independently when the player is interacting with a game. This is due to the fact that the game inherently contains pieces of the multiple dimensions. Incorporation can be induced when the player achieves a state in which the distance between the virtual avatar and the player is miniscule or non-existent. In this state, the player is able to control the avatar by instinct without having to think about how to perform actions. A second requirement for achieving a state of incorporation is to completely inhabit the virtual environment. The state of 'being incorporated' is often referred to as immersion which, according to (Calleja, 2011) can be divided into two different types. The first type is called *absorption*, which refers to players being absorbed into a task that can be as simple as a crossword puzzle or sudoku. The other type is *immersion as transportation*, which refers to the feeling of being in another place than the one that is physically inhabited. The two temporal phases, micro-involvement and macro-involvement, are centered around when the experience is taking place. The micro temporal phase is the player’s involvement in the moment-to-moment gameplay whereas the macro temporal phase is about the long-term experience (Calleja, 2007).

2.4.4.1. The Six Dimensions of Player Involvement

Performative/Kinesthetic involvement

The dimension of kinesthetic involvement describes the relationship between the player and the in-game avatar. It deals with the agency that the player is provided and the rewards received from controlling the avatar (PCG Lecture, 2015). It is therefore important that when designing a game that aims to achieve kinesthetic involvement, focus should be placed on controls that adheres to incorporation. This means that the game’s control scheme has to let the player learn and control the avatar to the point of not needing to think about the input.

Tactical/Ludic involvement

Ludic involvement refers to the player’s exploration of the game’s rules. This type of involvement is often induced when creating and following intrinsic objectives (PCG Lecture, 2015). In the macro phase, it encompases all forms of plan formulation such as planning a strategy in a tactical shooter before the round begins (Calleja, 2007).

Affective involvement

This type of involvement revolves around the game’s aesthetic and mood-altering properties (Calleja, 2011). Affective involvement can be achieved by designing and creating an experience that adheres to the player’s personal preferences in terms of graphics and sound design. In terms of micro and macro temporal phases, this involvement is e.g. induced when the player buys the game due to the art on the game’s cover, and after a play-session when the player talks to friends about the gameworld (PCG Lecture, 2015).

Narrative involvement

This type of involvement can be induced through either the scripted narrative and the tools utilized to portray it such as in-game cutscenes, cinematics and dialogue; or through narratives that players might generate themselves through stories that are not directly witnessed (PCG Lectures, 2015).

Spatial involvement

Spatial involvement is defined by the incorporation generated from navigating and exploring the game world. An example of the user being spatially involved with a game is when the player has created a mental map of their surroundings and are able to navigate the gameworld using vistas and recognizable locations without having to pull out a map. It is also shown when the player is blinded by a flash grenade but is still able to navigate their immediate surroundings and successfully find cover.

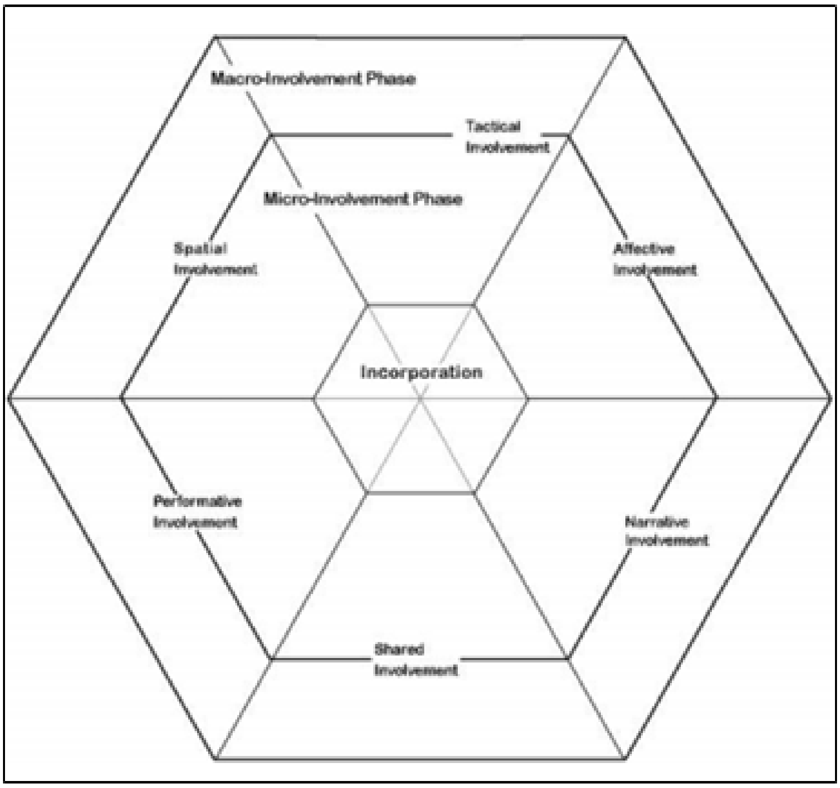

Figure 11. The player involvement model (including the temporal phases and six dimensions of involvement).

Shared involvement

This section of the involvement model contains the social aspects of involvement. This is often strongly shown in games that put heavy emphasis on competitive and cooperative gameplay modes (PCG Lecture, 2015). It also is very prevalent in the macro temporal phase (see section 2.4.4) due to players talking about the game after their play session, such as talking about guild policies in an MMO (Calleja, 2007).

Since the focus of this study is on controlling an in-game avatar and interacting with certain game objects, this study will attempt to measure the performative/kinesthetic involvement dimension of Calleja's player involvement model or the interfacing aspect of OA3 model of player engagement (see section 2.4.3). The knowledge gathered on playability heuristics can furthermore be combined with the related concepts from the engagement and involvement model to further improve the final evaluation by ensuring that the questionnaire will cover multiple theoretical frameworks. It is however important to utilize multiple testing methods in order to gather a broader perspective on the evaluated concepts; especially when said concepts are of a very subjective matter. For example, player engagement can be measured through a combination of interview-based and card sorting methods (Schønau, 2014), and since engagement and involvement share similarities, said evaluation methods could also be applied for an evaluation that is centered around involvement.

2.5. State of the Art

With the two evaluation conditions utilized in this study being based around basic (virtual controller) and active action techniques, it is important to investigate how developers have designed the input layout for these two types in other 2D-platformers. The following chapter will highlight two control schemes utilized by 2D-platformers that at least focus on traversal and jumping as some of the core gameplay mechanics.

2.5.1. Virtual Controller Designs

Contrary to the fighting games presented in section 2.2.3, many 2D-platformers have very simple input controls that mostly revolve around traversal, jumping and potentially an attack for dealing with enemies. Therefore, 2D-platformers with simple controls require less than the four buttons that are needed to replicate the button layout of a console controller, which thus allow the developers to opt for an even less obtrusive button layout. In terms of the movement controls located on the left side of the screen, many 2D-platformers seem to take the same approach by utilizing actual movement buttons rather than a virtual joystick.

Figure 12. Four 2D-platformers that present their movement buttons in a similar fashion. In terms of the action buttons on the right side of the screen, Swordigo (upper left), Sword of Xolan (lower left) and Star Knight (lower right) present these in a curvier pattern (represented by the red line) than League of Evil (upper right) does.

Mobile games such as Swordigo (2012) by *Touch Foo*, League of Evil (2016) by *Ravenous Games*, Sword of Xolan (2015) by *Alper Sarikaya*, and Star Knight (2016) by *LeftRight* all place the directional buttons next to each other and in the same height. These highlighted games do however present the action buttons of the right side of the screen differently. Swordigo, Star Knight, and Sword of Xolan shapes the button layout in such a way that it allows the player to use a curvy thumb movement when hovering over the action buttons, while League of Evil opts for placing the two action buttons at the same height; making the player perform a horizontal thumb movement when moving between the action buttons.

2.5.2. Active Action Controls

As mentioned in section 2.2.1, 2D-platformers that utilize an interaction scheme based on active actions ensure that the visuals of the game are not blocked by input buttons, which allows for a much cleaner gameplay experience that is not cluttered with UI elements (see figure 13). Like the aforementioned virtual controller-based games, 2D-platformers that utilize active actions also present their touch interaction in a somewhat similar fashion when compared to one another.

Figure 13. Four 2D-platformers that utilize a combination of basic and active actions in their respective input scheme. Limbo (upper left), Oddmar (upper right), Within (lower left) and Leo's Fortune (lower right).

Games such as Limbo (2010) by *Playdead*, Oddmar (2018) by *MobGe Limited*, Within (2020) by *Silver Lining Studio*, and Leo's Fortune (2014) by *1337 & Senri LLC* all make use of a drag-based touch input in order to control the movement of the playable character, while a swipe-based touch input is used for making the character jump over obstacles. Most of these games do however also make use of a basic action in different ways. Limbo and

Within utilize a hold (long tap) input for environmental interaction, while Oddmar uses a tap input for weapon attacks. Contrary to these, Leo's Fortune doesn't use any basic actions. This game only uses a dragging input for movement and a swipe for jumping. However, the swipe can be seamlessly turned into an upward drag input by not breaking contact with the screen. This allows the player to inflate the playable character; making the character descend towards the ground much slower. Thus, two gameplay mechanics are parented with a single touch input.

Another way that these highlighted games differ is through the fact that (for some) certain touch input can only be performed on specific areas of the screen. The three touch inputs (drag, swipe and hold) can be performed anywhere on the screen in both Limbo and Within, meaning that players can freely choose whether they want to use their right or left thumb for the dragging-based movement or the swiping-based jumping. Oddmar and Leo's Fortune instead separates the screen down the middle and limits which input can be performed with each thumb. In both Oddmar and Leo's Fortune, the dragging-based movement input can only be performed if it is initiated on the left side, while the swipe-based jumping can only be initiated on the right side of the screen. This method is somewhat similar to the aforementioned virtual controller-based games that also clearly separates movement and action buttons to certain areas of the screen.

2.6. Analysis Conclusion

In order to understand how to properly develop and evaluate a mobile game that caters to the purpose of this study, several relevant research topics related to the initial problem statement were investigated.

With this study aiming to explore the concept of mobile interaction, different mobile interaction techniques were explored in section 2.1. It was found that touch interaction is the most common type of interaction on the mobile platform. While other techniques such as video and location-based interaction do utilize other features of the smartphone (such as the camera, gps, and accelerometer), most of these techniques still often rely on touch interaction for input, while these aforementioned features are secondary in the overall interaction. Thus, touch interaction was chosen as the interaction technique that should be explored in this study.

Section 2.2. explored which types of touch interaction are commonly used for mobile games and general interaction on the mobile platform. It was found that mobile touch interaction can be categorized into two types; basic and active actions. These two categories of touch input differ a lot in the specific types of input that defines them, but both are still heavily utilized within mobile game development. It was furthermore found that basic actions are designed through virtual controllers that attempt to mimic the button layout of console controllers. It was thus concluded that the two touch techniques that should be evaluated should be based on active and basic actions.

In order to properly design a 2D-platformer, different fundamental elements of the genre were investigated in section 2.3. it was found that the building blocks of many 2D-platformers consist of components such as platforms, triggers, obstacles, and collectibles. It was also uncovered that 2D-platformers are usually designed with rhythm groups in mind, which are sections of a level that present challenges, and these sections are then separated by safe zones. Finally, it was found that concepts such as guidance, foreshadowing and layering are often used for guiding and presenting new challenges within a 2D-level.

Measuring the player experience is necessary to discern which type of interaction is best suited to a game, and thus different theory within measurement of player experience was explored in section 2.4. It was found that the player experience is dependent on multiple aspects and with each of these contributing to a whole. The aspects defined by models such as playability heuristics, player involvement and player engagement provide different methods for defining user experience. With this knowledge, a test can be performed that takes into account certain aspects that are shared across the presented theories but primarily through the involvement model, answer the final problem statement.

With these findings in mind, a final problem statement and relevant design requirements can be formulated.

3. Final Problem Statement

Based on the findings from the analysis, a final problem statement was formulated as follows:

How does changing a basic action-based control scheme with an active action-based control scheme impact the player's overall level of involvement in a 2d-platformer?

3.1. Design Requirements

The following design requirements are based on the findings from the analysis and involve aspects that are considered needed in order to develop a proper 2D-platformer that can successfully answer the final problem statement.

- Should utilize touch as the input method (see section 2.1.1)
- Should involve control schemes that are based on active (see section 2.2.1) and basic touch interaction techniques (see section 2.2.2)
- Should design the basic touch interaction technique through a virtual controller design (see section 2.2.3)
- Should include some of the basic components of 2D-platformer mechanics and level design (see section 2.3)
- Should include rhythm groups that contains rhythmic patterns such as basic, complex and composite (see section 2.3)
- Should utilize specific colors to highlight unique 2D-platforms (see section 2.3)
- Should design the levels with concepts such as guidance, foreshadowing and layering in mind (see section 2.3)
- Should adhere to some of the playability heuristics (see section 2.4.1)
- Should attempt to include a difficulty progression that works in accordance with flow theory (see section 2.4.2)
- Should tie game progress to activities involving interfacing (see section 2.4.3)
- Should enable the concept of player involvement but measured by performance in kinesthetic involvement (see section 2.4.4)

4. Methods and Evaluation Strategies

In order to answer the final problem statement (see section 3), fitting methodology has to be used. Since this study attempts to measure the difference in players' level of involvement between two test conditions, the methods and evaluation strategies should be tailored to provide proper data. Since player involvement is a very subjective concept that is best measured by allowing participants to elaborate their opinions, qualitative data gathering methods should be included as part of the final evaluation.

An iterative design process will be utilized in order to ensure that the 2D-platformer will be up to par for the final evaluation. This means that two smaller tests will be conducted, with each of them aiming to improve certain aspects of the application. The first one will be a usability test that will be centered around testing the general difficulty, uncover technical issues, as well as aspects related to the on-screen buttons and input methods for a certain gameplay mechanic. After the usability test, a different test centered around testing the chosen evaluation strategies will be conducted in order to ensure that the final evaluation will function as intended and provide the desired quality of data. This test will also function as a pilot test so that it can be ensured that the application has no technical issues that could negatively impact the gathered data. After ensuring that the evaluation strategies and the application function as intended, the final evaluation will be conducted.

4.1. Ethical Concerns

There are certain ethical concerns that need to be taken into consideration when setting up the different tests that are to be conducted. When utilizing qualitative methods that explore participants' subjective opinions and attitudes, it is essential that the participants are protected (Bjørner, 2015). This requirement leads to a necessity for a consent form that delivers transparency about how the test will be conducted, what data will be gathered and what said data will be used for. This gathered data would include recordings of voiced conversations and card sorting sessions, as well as answers to the questionnaires.

Due to the current state of the world, another ethical concern is the subject of COVID-19. The situation of COVID-19 will have to be closely monitored, since a resurgence of the virus potentially could force the evaluation strategies to be altered out of safety for both researchers and participants. If such a resurgence should happen around the same time that the usability test or the final evaluation is to be conducted, these tests will have to be changed to a different format. In such a case, the card sorting and questionnaire will be performed online, while the gameplay observations will be removed from the evaluation strategy. The reason for observations removal from the online strategy is because it will require participants to download an external screen-recording application as a requirement for participating in the evaluation. The researchers fear that requiring participants to download and set up such an application, will be viewed as an irritating inconvenience in the eyes of the participants and cause them to lose interest in participating in the evaluation. Coupled with the already very difficult process of gathering participants during a global pandemic, the researchers do not want to risk losing potential participants because of an inconvenience.

4.2. Usability Testing

To ensure that the application reaches a level of quality that can achieve a state of player involvement, a usability test will be performed in order to make adjustments such as determining a few design and gameplay elements as well as uncovering technical issues.

This usability test will focus on four different aspects. The first involves figuring out the most optimal way of designing the layout for the action buttons that will be placed on the right side of the screen in the control condition (basic action) (see section 2.2.3). The second aspect will focus on testing two different types of touch input for one of core action-based mechanics in the experimental condition (active action). Finally, the two remaining aspects will revolve around uncovering technical issues and perform adjustments in case the game is too easy or difficult.

For the usability test, 3-5 participants will be gathered through the simple random sampling method (Bjørner, 2015), and data will be gathered through a retrospective think-aloud protocol (van den Haak, De Jong & Schellens, 2003), which involves having the participants provide feedback immediately after the game session on the aforementioned topics such as difficulty and control designs. The test procedure and results can be viewed in section 7.1.

4.3. Pilot Testing

In order to ensure that the final evaluation can be carried out as intended and that the conductors follow the necessary test procedures to avoid differences in handling of participants and gathering data, a pilot test will be conducted. The general aim of the pilot test will be to do a test of the chosen evaluation strategies, which will ensure that the gathered data is of the desired quality. If any last minute technical issues appear, these will also be sorted before the final evaluation begins. The test procedure and results can be viewed in section 7.2.

4.4. Final Evaluation

As mentioned in section 3, the goal of the final test is to determine whether or not there is a difference in player involvement between the two test conditions. When dealing with concepts of player experience, it is not ideal to rely on quantitative data alone, and thus methods utilized for the final evaluation will be a mixture of quantitative and qualitative methods. The evaluation strategies will involve a questionnaire, a card sorting session and observations of the participants gameplay session, which in combination can uncover how involved the participants were in the experience. For the questionnaire and the card sorting, certain values will be applied to the possible answers in order to quantify their level of involvement for statistical tests.

Participants will be gathered using the simple random sampling method (Bjørner, 2015), and the aim is to reach 30 participants for each condition so that the result can be used in a t-test. The test will be done in a controlled environment in order to limit outside factors such as distractions, natural lighting and noise. The test procedure and results can be viewed in section 7.3.

4.4.1. T-test and Questionnaire

Upon having finished the gameplay session, participants will be presented with a questionnaire that aims to measure their involvement through the use of weighted likert items. The scores gathered from these weights will be used to describe the difference in player involvement across the two conditions. The questionnaire will not solely revolve around kinesthetic involvement but also other aspects of involvement so that a broader understanding of the participants' overall level of involvement can be gained.

Each likert item in the questionnaire will have a set range from 1 (strongly disagree) to 7 (strongly agree) so that these results can be compared to the scoring system made for the card sorting session. The scores gathered from the questionnaires will be put through a t-test so that it can be uncovered whether or not there is a significant difference between the two control schemes. The questionnaire can be seen in appendix A.4.

4.4.2. Card Sorting

After having completed the questionnaire, the participants will partake in a card sorting session. The type of card sorting session utilized for the final evaluation will be that of a closed card sorting session (Affairs, 2018), which means that participants will not be able to make their own cards. The downside of participants not being able to make their own cards is that if they do not understand or identify their experience with the predetermined cards, then their experience with the game might not be accurately represented in the data. However, the benefit is that each card can be given a weighted score which can be used to compare player involvement with the data gathered from the questionnaire, as long as the scoring system also fits within a range that goes from 1 to 7. The cards sorting session will present five categories that each contain six cards, and participants will be asked to pick the two cards from each category that best describe their experience with the game and elaborate on why those cards were chosen. Finally participants will be asked to choose three of the ten cards they have chosen across the five categories, and describe why those three cards more prominently describe their experience compared to the other other seven.

In this study, a scoring system will be used for the card sorting session. This system is not statistically proven and used only to describe differences between the data collected from the questionnaire and the card sorting session. The way this scoring system works is by giving cards with a negatively-loaded connotation a negative score, whereas positively-loaded cards will have a positive score. As participants choose their ten cards from across the five categories, these cards will add or subtract one point from the participant’s overall score based on the cards’ connotation. The three cards that are chosen from the ten initial cards will then add or subtract their score a second time. The reason for participants to choose the three cards that more prominently describe their experience is done to more accurately represent their overall feeling of involvement in case that their initial ten cards do not properly represent it. This score will be compared to the score thresholds of the level of involvement scale (seen figure 14) in order to determine where on the scale they lie. The collective mean of the participant group will then be compared to the questionnaire data mean.

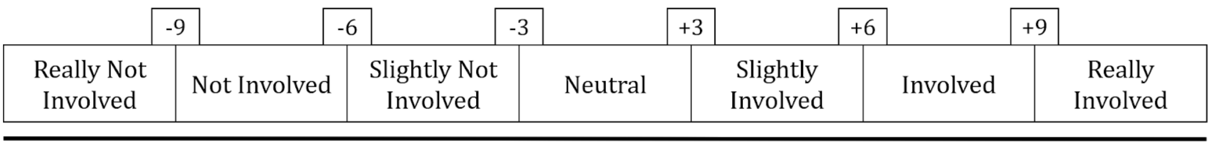

Figure 14. Level of involvement scale.

Along with the weighted score gained from the chosen cards, qualitative data will also be gathered through the dialogue between participants and researchers, since the participants will be asked to elaborate on their choice of cards. Finally, data will also be gathered on the frequency distribution of the chosen cards. This will indicate where differences between the conditions may be, which can then be further supported by the earlier dialogue during the card sorting session.

4.4.3. Observations

To ensure that the data gathered from the card sorting session and questionnaire are valid, observations will be performed during the participants' gameplay session. The method utilized will be the human observation method (Bjørner, 2015), in which the observations will be carried out by the researchers who will then note down errors, visual observations e.g. expressions from the participant, and unintended use of the controls that could impact the gathered data from the questionnaire or card sorting session. The observations will also be cross examined with the results of the other methods to see if they correlate, and thus further strengthen the validity of the results, as well as adding a level of clarification to why the results turned out the way they did.

In conclusion; through the use of a questionnaire, a card sorting session and observational data, this study will perform an evaluation that will attempt to measure the participants' level of involvement. The quantitative data from the questionnaire and card sorting will be compared to further strengthen validity of the given results, while the observational and qualitative data from the card sorting session, will further document the findings.

5. Design

The following chapter will be centered around the design choices made in order to develop a simple 2D-platformer that can help answer the final problem statement, while also including the design requirements specified in section 3.1. The chapter will focus on choices made in regards to the level design, input design and sound design of the platformer which has been given the title *Dark Squares*.

5.1. The Concept and Visuals of *Dark Squares*

The general gameplay loop of *Dark Squares* is relatively simplistic. The player will control a yellow square across five 2D-platforming levels that will task players with jumping gaps, dodging enemy squares, gathering collectibles and utilize a gravity-reverse mechanic to overcome puzzles and other challenges. A tutorial level will precede the five main levels, allowing players to become familiar with the controls in a controlled environment that does not pose any danger to the playable character.

The 2D-platformer was designed with simplicity in mind. This design choice was partially based on the fact that even 2D games consisting solely of quadrilateral and other simple geometric shapes have been well received by critics, as is the case with games such as Flat Heroes (2016) by *Parallel Circles* and Thomas Was Alone (2012) by *Mike Bithell*, indicating that more advanced visuals are not a necessity in order to develop an engaging game. Putting less emphasis on visuals also allows more development time to be allocated towards gameplay mechanics and controls; two aspects that are much more important for answering the final problem statement. The game was made in the 2019 version of the Unity game engine and three asset packs from the Unity asset store were used to develop the visuals; the *Dropping Box*-asset pack (2018) by *Nak-Studio*, the *Dynamic Space Background Lite*-asset pack (2017) by *DinV Studio*, and the *Clean Vector Icons*-asset pack by *Poneti* (2018). Finally, an image with the text *"You Win!"* (Shutterstock, n.d.), was used to let players know that they had completed the game.

5.2. Components

In section 2.3.1. it was found that 2D-platformers consist of certain building blocks, which are called components. These components served as the foundation of the game in terms of level design and game mechanics. Platforms and obstacles were designed based on simple quadrilateral shapes and given a distinct color that would allow the player to distinguish them from each other. Shapes that served more distinct purposes were given the strongest eye-catching colors while standard platforms that players will traverse were given less eye-catching colors (see section 2.3.2). Dangerous platforms that will transport players back to the nearest checkpoint if touched, were given the strongest eye-catching color (red), since the player actively needs to avoid these quadrilateral shapes. The second and third most eye catching colors, yellow and green, were given to the player square and the checkpoint squares, respectively. Yellow and green squares also serve a secondary purpose; namely that yellow squares signify the end of a level, while green squares enable the gravity effect on the next gravity wall that players will come across. The gravity walls are tall collections of pink (fourth most eye-catching color) squares that are influenced by the gravity mechanic in the same way that the playable square is, meaning that getting past them demands a bit of thinking from the player. The gravity walls can also be moved to another location by connecting them to safe moving platforms, and they also destroy the dangerous platform upon impact, thus adding more layers to the game's puzzle solving elements. Finally, the game will present two types of platforms that can be traversed safely. One will be static while the other will be in constant movement between two points. These will be given the colors blue and cyan respectively; two colors that are less eye-catching but since these will be relatively common in all levels, attention does not need to be drawn towards these.

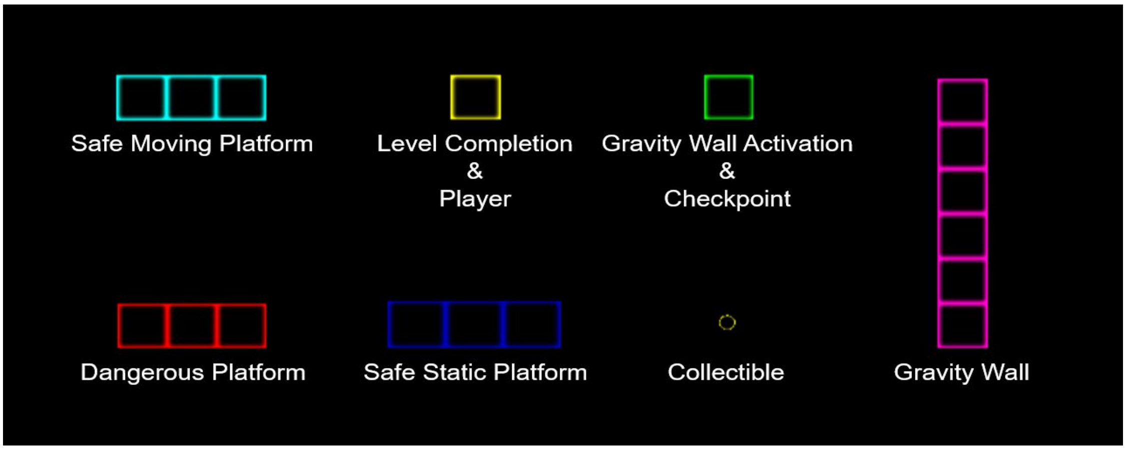

Figure 15. All visual components of Dark Squares.

Other components mentioned in section 2.3.1. that serves as building blocks in *Dark Squares* are collectibles and triggers. The collectibles add an optional task for players to complete, while the gravity-reverse mechanic will function as the game's trigger, since said mechanic performs an altering of the game's state.

5.3. Level Design

As mentioned in section 2.3.1. 2D-platformers often alternate between presenting sequences of a higher level of challenge (rhythm groups) and sequences of rest (safe zones). Much of the level design for *Dark Squares* was designed with these two types of sequences in mind. The safe zones have been placed towards the end and in the very beginning of each level; allowing players to collect themselves before they continue onwards towards the next set of rhythm groups. What can be considered the middle section of each level function as the rhythm groups where players will be faced with dodging dangerous platforms, solving gravity-based puzzles, jumping gaps, and getting onto safe moving platforms. Throughout the rhythm groups, there will be very small platforms where the player can stand still without being in danger, but the aforementioned challenges will in these cases still surround the player; something that is not the case in the safe zones (see figure 16).

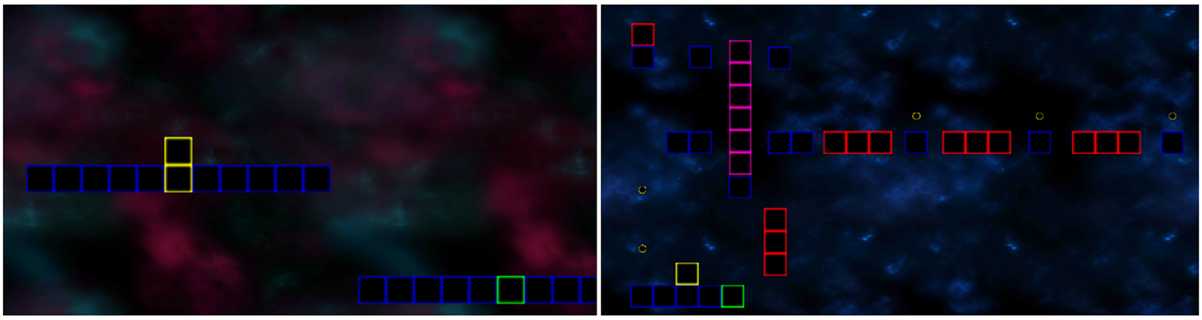

Figure 16. One of the safe zones (left) and one of the rhythm groups (right) in Dark Squares.

In section 5.2, it was mentioned that the collectibles served as an added optional task for players to complete, but they are also part of *Dark Squares'* usage of the *guidance* pattern (see section 2.3.2). None of the collectibles have been placed in locations that are not part of the main path throughout the level, meaning that as long as the player follows the collectibles then they are heading in the right direction. Occasionally, the player will come across locations where the next safe platform will be located off-screen. Here, the collectibles will highlight the path towards said off-screen platform and indicate to players where they have to go and that they have to take a leap of faith (see figure 17).

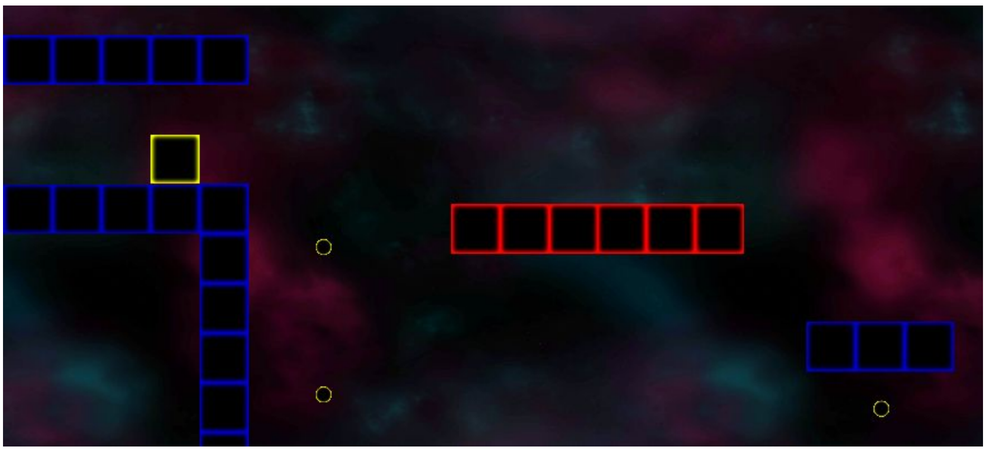

Figure 17. The collectibles highlighting a safe path towards the next platform which is off-screen.

Another utilized pattern is *foreshadowing*, which (as mentioned in section 2.3.2) is centered around subtly introducing new mechanics in a controlled environment because said mechanic is about to feature in upcoming challenges or puzzles. In *Dark Squares*, this foreshadowing pattern is used to introduce players to the gravity wall's ability to destroy the dangerous platforms. At the start of the third level, players will come across a gravity wall located below a dangerous platform that is blocking the way forward. By performing a gravity-reverse up to the collectible (see figure 18), players (unbeknownst to themselves) will cause the gravity wall to collide with the dangerous platform and destroy it, and thus players have subtly been introduced to this new mechanic.

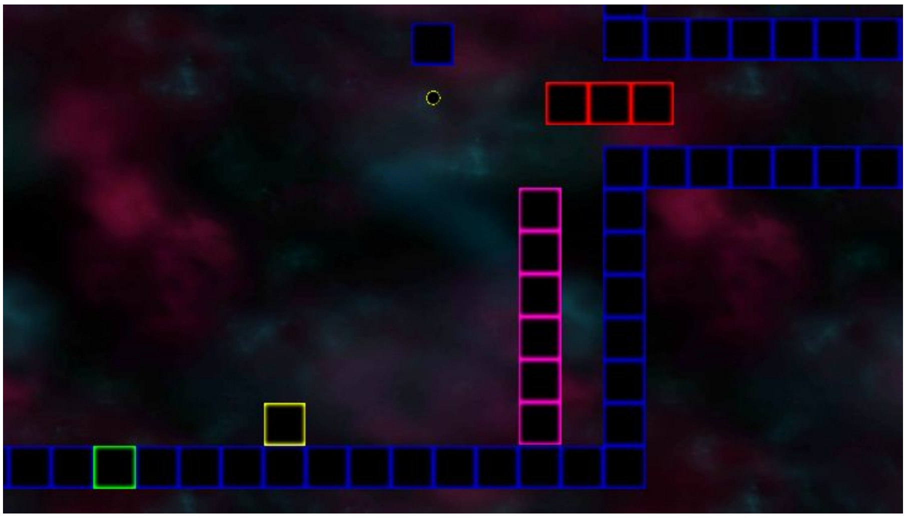

Figure 18. Usage of the foreshadowing pattern in the beginning of the third level. In a controlled environment, the player will be introduced to gravity wall's ability to destroy the dangerous platforms upon impact.

Layering is the final pattern mentioned in section 2.3.2. that is used in *Dark Squares*. This pattern is utilized by introducing new gameplay mechanics without the need of introducing new components. For example, the gravity wall, dangerous platform and safe moving platform components are all introduced in the first level, but the fact that the gravity wall can destroy the dangerous platforms is not introduced until the third level. Furthermore, the gravity wall's ability to be moved around the level by being connected to the safe moving platforms is not introduced until the fifth level. Thus, new mechanics related to already existing components are slowly introduced throughout the game. The inclusion of these puzzles also cater to the *solving problems* aspect of the activities category within the OA3 model of player engagement (see section 2.4.3); thus adding a second property (apart from interfacing) that can induce player engagement. The increasing level of difficulty in the puzzles also cater to Csikszentmihalyi's flow theory which highlights the importance of increasing difficulty as the player's skills increases (see section 2.4.2)

5.3.1. Rhythmic Jumping

The jumping mechanic is central to the gameplay loop of *Dark Squares*, and thus research was made to uncover beneficial methods for designing jumping patterns. In section 2.3.2, it was found that designing rhythmic jumping actions can aid players in achieving a state of flow (see section 2.4.2), and thus increase the chance of inducing player engagement. Therefore, a few jumping sections were designed based on the patterns presented in that section. One of these patterns utilized in *Dark Squares* are *basic patterns* (see section 2.3.2). This type of pattern is used a few times through the five main levels and serves as the most basic type of rhythmic actions performed in the game. The *Complex* and *composite* patterns have also been included in the levels, though not as prominently as the basic one.

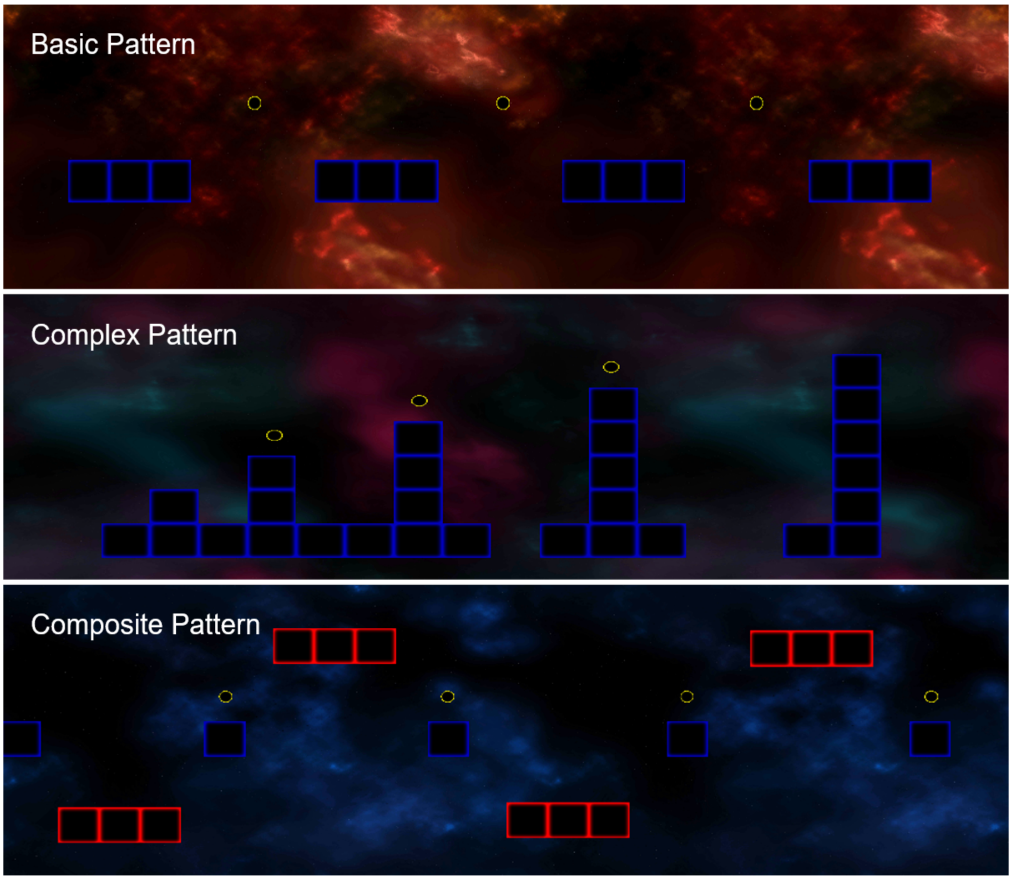

Figure 19. Basic patterns (upper) have no variation in the challenge, while complex patterns (middle) utilize similar obstacles but with a slight change in each consecutive obstacle, which in this case is a longer jump. Finally, composite patterns (lower) combine two challenges into a single challenge, which in this case requires the player to both jump over a gap and dodge a vertically moving enemy platform at the same time.

5.4. Input Design

As mentioned in the final problem statement (see section 3), two different input schemes will be tested in this study; one based on basic actions (control condition) and another based on active actions (experimental condition). Korhonen and Koivisto wrote in their playability heuristics (see section 2.4.1) that it is important that *"control keys are consistent and follow standard convention"* and that *"game controls are convenient and flexible"* (see figure 7). Based on playability heuristics such as these, as well as the general findings from the analysis, the input design for the two conditions were developed.

5.4.1. Basic Action Input

The basic action-based control scheme for *Dark Squares* is centered around the design philosophy of virtual controllers (see section 2.2.3). This means that movement related buttons have been placed on the left side of the screen, while action buttons (jumping and gravity-reverse) are located on the right side. The specific input type used in the game is the single tap gesture (the most conventional basic action), and this was determined as the most optimal input method since the amount of buttons are relatively low. Had the game required more individual buttons, other basic action gestures such as double tap and long tap (hold) could have been used to map multiple actions to a single button, but since this was not the case those basic action gestures were disregarded.

The size of the buttons and the gap between them measure 1cm and 5mm respectively on a Samsung Galaxy S9 (the smartphone used in development). This is slightly higher than the necessary requirements found in section 2.2.3, which stated that buttons should be at least 8mm in size and have a gap of at least 3mm between them in order to avoid unintentional button inputs from the player. The decision of slightly increasing these two measurements was made in case the final evaluation would have to be performed online, since participants would then have to use their own smartphone, which potentially could have a smaller screen than the Galaxy S9, and thus decrease these measurements automatically.

Section 2.5.1. explored virtual controller designs specifically for 2D-platformers and two different button layouts were found to be popular among mobile developers. The first layout places all input buttons at the same horizontal height, while the second layout would place the action buttons on the right side of the screen in a more curvy or diagonal pattern, while the movement button remains similar to the first layout (see figure 20). As mentioned in section in section 4.2, the purpose of the usability test was (partially) to uncover which of these two button layouts should be utilized in *Dark Squares*. The results from said usability test (see section 7.1.2) indicated that participants found the second button layout more convenient and practical compared to the other layout.

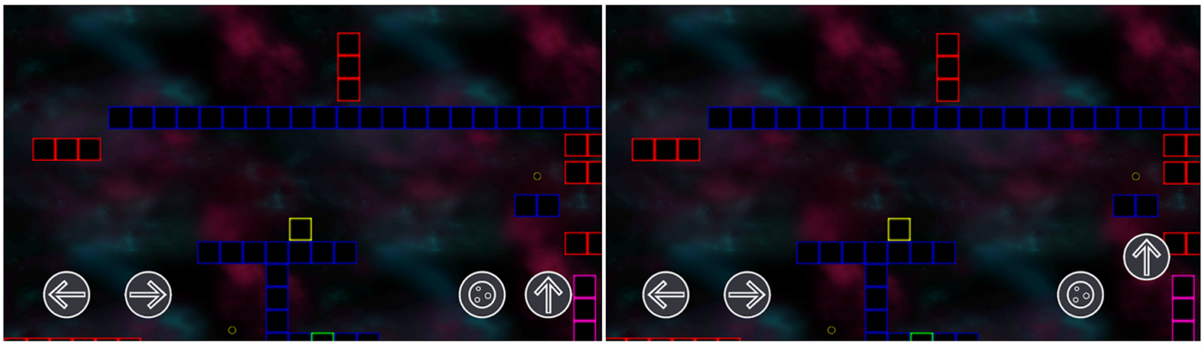

Figure 20. The two button layouts that were tested in the usability test (see section 7.1.2). The results from the usability test indicated that participants prefered the layout on the right.

Based on the gathered data from the usability test, the button layout for *Dark Squares* was then designed with a layout that presents the action buttons on the right side of the screen in a more diagonal pattern.

5.4.2. Active Action Input

The active action-based control scheme was designed with the dragging and swiping gestures in mind as these two gestures are commonly used in mobile games that do not include on-screen buttons (see section 2.2.1). In section 2.5.2, it was found that 2D-platformers that utilize active action gestures often differ in one particular way, which is whether or not they combine basic and active gestures. In order to figure out which approach *Dark Squares* should utilize, the usability test (see section 4.2) involved testing these two design approaches. Two different types of gesture input schemes were tested; one which combined basic and active action by mapping movement to a dragging gesture, jumping to an upward swipe, and gravity-reverse to a single tap, and another which purely relied on active actions by changing the gravity-reverse input from a single tap to a downward swipe. The results from the usability test (see section 7.1.2) indicated that participants prefered the input scheme that combines basic and active actions, and that mapping gravity-reverse and jumping to different directional swipes made the controls confusing and frustrating. With this gathered data in mind, the experimental condition would thus utilize a combination of active and basic actions.

In section 2.5.2. it was also found that certain games based around active action controls will split the screen into two sections; ensuring that certain gestures can only be performed on specific areas of the touch screen. For *Dark Squares*, it was decided that no screen-separation should be included. This decision was made due to one of the playability heuristics in section 2.4.1, which states that it is important that *"the game supports different* *playing styles”*. By allowing players to choose which thumb they wish to use for a specific gesture, the game will thus cater to more than a single playing style.

5.5. Sound Design

Like the visuals, the sound design for *Dark Squares* is very simplistic. It involves a piece of music that continuously runs on a loop, and a few sound effects that indicate actions taken by the player. The music (entitled *Triumph*) is composed by Paul Taylor (2011) and is borrowed from the soundtrack of the video game Frozen Synapse (2011) by *Moby 7 Games*. Sound effects have been added to various interactions within the game such as picking up collectibles (brandwesson, 2011), stepping on green blocks as an indicator for checkpoint activation (ZvinbergsA, 2015), as well as the destruction of red dangerous platforms (MATRIXXX_, 2018). The latter has been given a sound effect because certain red blocks need to be destroyed while they are off-screen, meaning that a non-visual indicator was needed to inform the player that they have been successfully destroyed.

With the inclusion of many design principles such as rhythm groups, components, rhythmic patterns and other general design patterns, Dark Squares should be ready to serve its purpose and aid in answering the final problem statement.

6. Implementation

The following chapter will document the implementation and thought processes behind the development of the application. The documentation will contain a description of the necessary elements needed in order to create the core features of the application utilized for the evaluation such as gesture detection, player movement and abilities as well as how obstacles in the environment function.

The application will be made in the 2019.3.15f1-version of the Unity game engine. The engine was chosen because of its availability and good online documentation. It furthermore includes an API (Unity, 2020) that is used for mobile interaction, which makes the engine suitable for developing mobile games. The game will be developed for android-based smartphones since the final version of the game (used for the final evaluation) will run on a Samsung Galaxy S9.

6.1. Detecting Interaction

The application itself has to be able to test multiple ways of interaction on a mobile platform. The two main types of interaction are an active and a basic-based interaction model. For the active interaction version, three main gestures have to be detected which are tap, swipe and drag. It also has to be able to detect multiple fingers interacting at the same time. For the basic action-based version, Unity's UI system will be utilized to implement the on-screen buttons.

6.1.1. Detecting the Gestures

In order to detect the aforementioned gestures, it is important to understand how they work. In essence, the code will be based around three phases of interaction: the beginning, the movement and the end of the gesture.

6.1.1.1. Tapping

The tap gesture is an interaction type that begins and ends in the same place of the touchscreen. Distinguishing between different types of tap gestures is therefore more centered around the amount of time it takes between the beginning and the end of the gesture. When scripting in Unity, these two phases of interaction can be called through the use of the Touch API (Unity, 2020), which is able to detect separate touches and call their position. In order to create tap gesture detection, the phase's beginning and end will be used to store the touch positions (see figure 21), which is then compared.

However, simply comparing the values between the beginning and the end of the gesture can make the tap detection inconsistent because of the precision required to detect the tap. Therefore, a distance threshold is added so that small inaccuracies such as miniscule movements will not be an issue.

```
if (Input.touches[i].phase == TouchPhase.Began) // touch begin
{
    startTouch[i] = Input.touches[i].position; //store the start touch location in an variable
}
else if (Input.touches[i].phase == TouchPhase.Ended || Input.touches[i].phase == TouchPhase.Canceled)
{
    mouseUp = true;
    tapE[i] = Input.touches[i].position; // the ending position
    tapDelta[i] = tapE[i] - tapS[i]; // the distance between start and end
    startTouch[i] = swipeDelta[i] = Vector2.zero; // reset variables
    //swipeTrue = false;
    runOnce = true;

    //Debug.Log(timer - tempTime);

    if (timer < tempTime) // if a certain time threshold has passed
    {
        timeGate = true;
    }
    else
    {
        timeGate = false;
    }

        if (tapDelta[i].magnitude < minSwipeDistance && timeGate) // interaction is has less distance than swipe and is quicker than 0.x seconds
        {
            runTap = true; // allows tap to run 1 time
        }


}
```

Figure 21. Code used for tapping which is based around using the Touch.phase API (Unity, 2020).

6.1.1.2. Swiping

The difference between a swipe and a tap gesture is that for swipe, the distance between the beginning and the end of an interaction is larger. A swipe can however happen without the user ending the interaction, meaning that a swipe gesture can be performed during the execution of another gesture such as a dragging-based one. This is done by focusing on the distance between points in a time frame (see figure 22). This type of swipe behavior is used in a limited capacity in the application but can be expanded by adding a dynamic distance detection for swipe.

```
public void TouchDeltaAndSwipe(int i)
{
    if (timeGate)
    {

        //Debug.Log("calculating");
        swipeDelta[i] = Vector2.zero; // reset variable incase of old interaction
        if (startTouch[i] != Vector2.zero)
        {
            // check delta for interaction
            if (Input.touches.Length != 0) // if an gesture is detected
            {
                swipeDelta[i] = Input.touches[i].position - startTouch[i]; // compare current pos with start pos (start pos can be an temp but is not used)
            }
            else if (Input.GetMouseButton(0)) // mouse update incase of PC debugging
            {
                swipeDelta[i] = (Vector2)Input.mousePosition - startTouch[i];
                // Debug.Log("Swipetrue = " + swipeTrue);
                //  Debug.Log("mouseUp////  " + swipeDelta);
            }
        }

        ExecuteSwipe(i); // if statement at the begining in this if (swipeDelta[i].magnitude > minSwipeDistance && timeGate)
                         // greate than distance of minimum swipe and withing an certain time limit
    }
}
```

Figure 22. Code for the initial swipe check.

In regards to the swiping gesture however, the application is mostly centered around detecting vertical motions. The way to detect the direction of swiping motion is by locating the direction with the furthest distance. In the case of this application, rightward, leftward, upward, and downward directions can be detected, meaning that diagonal motions will be converted to the most similar of the aforementioned directions. The distance of the swipe on the x and y-axis are compared in order to detect if the gesture is performed either vertically or horizontally, and then comparing the values to know if the gesture movement is upward, downard, left or right (see figure 23).

```
if (swipeDelta[i].magnitude > minSwipeDistance && timeGate)
{

    // Swipe has happened
    float x = swipeDelta[i].x; // the x value of the swipe
    float y = swipeDelta[i].y; // the y value of the swipe

    if (Mathf.Abs(x) > Mathf.Abs(y)) // take the absolute value (-10 -> 10) or (10 -> 10)
    {
        // user swiped left or right
        if (x < 0) // detact which side on x there was swiped
        {
            swipeL = true;
        }
        else
        {
            swipeR = true;
        }

    }
    else
    {
        //user swiped up or down
        if (y < 0) // lower than initial position
        {
            swipeD = true;
            if (!player.GetComponent<movement>().isJumping)
            {
                runSwipe = true;
               // player.GetComponent<movement>().jump();
               // swipeTrue = true;
            }
        }
        else
        {
            swipeU = true;
            if (!player.GetComponent<movement>().isJumping)
            {
                runSwipe = true;
                //player.GetComponent<movement>().jump();
                //swipeTrue = true;
            }
        }
    }
```

Figure 23. Code that detects direction and executes a swipe.

6.1.1.3. Dragging

In the application, the dragging-based gesture works by comparing the position of the initial touch with the position of the current touch but only after a certain time threshold has expired. This is done in order to distinguish the drag gesture from the swipe gesture, since these two types are relatively similar. To record the current position of the touch, the movement phase has been used (see figure 24).

```
if (Input.touches[i].phase == TouchPhase.Moved) //When touch moves
{
    if (timer > tempTime) // time exeeded swipe time
    {
        tempPos = Input.touches[0].position; // store temp location
            dragDelta = tapS[i] - tempPos; // the delta of the drag

            runDragL = false; //reset variables
            runDragR = false;

        if (tempPos.x > tapS[i].x) // if the delta i more than the inital location
        {
                // go right
                runDragL = false;
                runDragR = true;

            }
        else if (tempPos.x < tapS[i].x) // if the delta i less than the inital location
            {
                // go left
                runDragR = false;
                runDragL = true;

            }

    }
}
```

Figure 24. Code used for the dragging detection.

With the application being able to detect the different gestures, these can be tied to specific mechanics such as movement, jumping and other special abilities. In order to ensure that the detected difference is seamless, fine tuning the variables such as minimum distance or how long the duration of the time variable is for swipe is necessary.

6.1.1.4. Buttons

For the basic action-based condition, buttons from Unity UI have been used. By using the event trigger (see figure 25), it becomes easy to detect when the button is being pressed and when it is released.

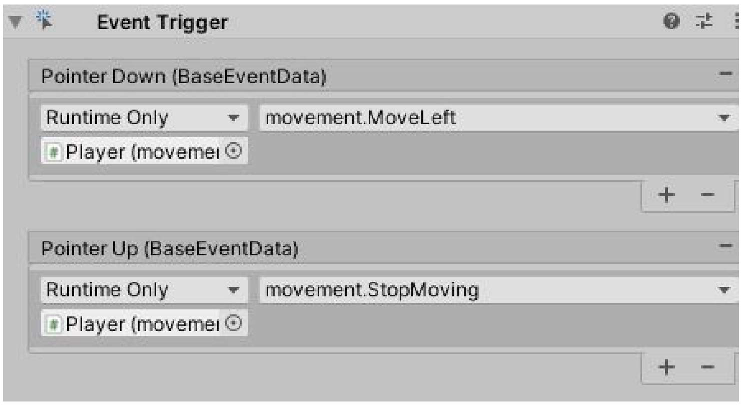

Figure 25. Event triggers used for a button input.

With the ‘pressed’ and ‘released’ system in mind, a few simple functions were added for player actions (see figure 26).

```
public void MoveLeft()
{
    // go left if button is pressed
    moveRight = false;
    moveLeft = true;
}

0 references
public void MoveRight()
{
    // go right if button is pressed
    moveLeft = false;
    moveRight = true;
}

0 references
public void StopMoving()
{
    // stop moving when a button is released
    moveLeft = false;
    moveRight = false;
}
```

Figure 26. Boolean system that controls whether the player can move.

Because the active action-based interaction scheme has to mechanically work similarly to the basic action-based interaction scheme, boolean-values are used to activate the movement in the player movement script, meaning that the speed of the playable character is the same across both versions (see section 6.2).

6.2. Player Actions

When evaluating different input controls, it is important that the two versions do not behave differently in terms of player movement. Therefore (as mentioned in section 6.1.1.4), the player movement is shared between the two conditions, as to isolate the difference of the input detection schemes.

To achieve this (as seen in section 6.1.1.4), the input detection will be responsible for activating booleans (see figure 27) which then allows for player movement to happen. The base movement itself is performed by simply using the gameobject's transform and moving it accordingly.

```
if (moveLeft)
{
    // go left
    //Debug.Log("running is true");
    Vector3 movement = new Vector3(-1f, 0f, 0f); // movement direction
    this.transform.position += movement * Time.deltaTime * horizontalSpeed; // movement * moded speed over time
}

if (moveRight)
{
    //go right
    Vector3 movement = new Vector3(1f, 0f, 0f); // movement direction
    this.transform.position += movement * Time.deltaTime * horizontalSpeed; // movement * moded speed over time
}


if (usingJoystick) // joystick for virtual controller with analog
{
     Vector3 movement = new Vector3(joystick.Horizontal, 0f,0f);
     transform.position += movement * Time.deltaTime * horizontalSpeed; // movement * moded speed over time

}
```

Figure 27. Executing player movement using its local transform.

Beyond basic movement, the player has two actions which they can perform. The first action is centered around reversing the gravity of the playable character and gravity walls (see section 6.3), while the other is a basic jumping mechanic (see figure 28).

```
public void jump()
{
    if (!isJumping)
    {
        isJumping = true; // limits jump to 1 until reset when hit the ground
        playersRigid.velocity = new Vector2(0f, 0f);

        if (!gravIsReversed) // checks the current state of gravity
        {
            gameObject.GetComponent<Rigidbody2D>().AddForce(new Vector2(0f, jumpStrength), ForceMode2D.Impulse); // adds force to the characters rigidbody
            Debug.Log("Run jump+"); // debugging

        }
```

Figure 28. Code for knowing when to jump as well as executing a jump.

6.3 Changing the Environment

As mentioned in section 6.2, one of the actions that players can perform is changing the gravity-state of the playable character and gravity walls. This is done by manipulating the gameobjects gravity scale settings in the rigidbody of both the character and the walls. This means that the game operates in two distinct states and edits vertical movement to accommodate this. This section mostly focuses on how the gravity-state of the gravity walls is changed, but the same logic can be applied to the playable character.

In order for the gravity-reverse to function properly, three steps are taken; detection of which objects that are affected, calling the change in gravity, and executing it. To determine which objects should be affected by the gravity-reverse mechanic, Unity's tag system was used to store the objects in an array (see figure 29).

```
void Start()
{
    gravityState = startingGravState; // starts with grav reversed or normal
    gravWalls = GameObject.FindGameObjectsWithTag("gravWall"); // store all objects with the custom tag gravWall
}

public void ReverseGravity() // changing gravity function
{
    GameObject.FindGameObjectWithTag("Player").GetComponent<movement>().ChangeGrav(); // changing the players gravity

    foreach (GameObject item in gravWalls) // for every object with the tag gravWalls
    {
        Debug.Log(item.name); // debugging
        item.GetComponent<reverseGrav>().reverseGravity(); // reverse their gravity
    }
}
```

Figure 29.Code that stores all tagged objects in an array and calls the execution for changing gravity.

To call the gravity-reverse, the "reverseGravity" function is called from scripts (see figure 30 and 29). The function's primary purpose is to change the values that are updated when the gravity-reverse is executed.

```
public void reverseGravity()
{
    //this.gameObject.GetComponent<Rigidbody2D>().constraints = RigidbodyConstraints2D.None;
    //this.gameObject.GetComponent<Rigidbody2D>().constraints = RigidbodyConstraints2D.FreezeRotation;

    this.gameObject.GetComponent<Rigidbody2D>().constraints = RigidbodyConstraints2D.FreezePositionX | RigidbodyConstraints2D.FreezeRotation;
    // freezez X and rotation

    startingGrav *= -1; // swapping the direction of gravity
    gravModState = true; // is true until it hits collision (changes speed of gravity)
}
```

Figure 30. The changing of gravity that happens through reversing the startingGrav variable.

The execution of the reverse gravity function (see figure 30) changes the gravity scale value of the object. This in turn, reverses the gravity of the object in-game.

```
void Update()
{
    //Debug.Log(colliding);
    if (Input.GetKeyUp(KeyCode.S)) // debugging key on PC
    {
        reverseGravity();
    }

    if (gravModState) // check state
    { // while in air gravity is much stronger
        this.gameObject.GetComponent<Rigidbody2D>().gravityScale = startingGrav * gravMod;
    }
    else
    { // resets the gravity to normal value after collision
        this.gameObject.GetComponent<Rigidbody2D>().gravityScale = startingGrav;
    }
}
```

Figure 31. With startingGrav reversed the Rigidbody2D is updated and will change the gravity of the object.

With these components of the implementation presented, core elements of the game and gesture detection could be replicated by others. It also highlights how different gesture detection can be used in terms of design (see section 6.2), or how an analog stick could be utilized (see section 6.2). This should allow for expansion of specialized interactions on smartphones that fit a different type of game or a different control scheme than the ones used in this study.

7. Results

To answer the final problem statement (see section 3), multiple tests were conducted. The following chapter will present the purpose, procedure and results from each of these tests which include: a usability test that was performed in relation to a few design decisions centered around the chosen control schemes, a pilot test that was conducted to test the evaluation strategies and the final evaluation which aimed to investigate the final problem statement.

7.1. Usability Test

The first purpose of the usability test was to figure out how the on-screen buttons for the control condition should be displayed, and therefore two possible layouts (based on findings from the section 2.5.1) were compared. The first layout presented the two action buttons on the right side of the screen in a diagonal pattern, while the other layout placed both buttons at the same horizontal height (see figure 20). The second purpose was to test two different types of gestures for the gravity-reverse mechanic in the experimental condition; one based on a basic gesture (single tap) and another based on an active gesture (downward swipe). Furthermore, the usability test also aimed to uncover any technical issues in both conditions and to investigate if the level of difficulty should be adjusted before the final evaluation.

Due to a slight resurgence in daily COVID-19 cases around the time in which the usability test was conducted, the entire test ended up being conducted online rather than in-person as it had originally been planned. This meant that participants were instead gathered through the criterion sampling method (Cohan and Crabtree, 2006), since it was necessary for participants to own an android themselves.

7.1.1. Setup and Procedure

The only piece of equipment needed for the test was an android smartphone and thus only people with such a smartphone were contacted as potential participants for the test. Participants were gathered through the criterion sampling method and a total of four participants took part in the test. Four different versions of the game were sent to participants as apk-files, (two control conditions with different button layouts and two experimental conditions with different input gestures for the gravity-reverse mechanic) before they were contacted via a Discord call. Participants were then instructed to simply play through all versions of the game back-to-back. Data was gathered using a retrospective think-aloud protocol in which they were asked to provide feedback, on the button layouts in the control conditions, input gestures in the experimental condition, difficulty, and potential technical issues, immediately after finishing the gameplay session. In order to save time, only the second and third levels were included in the usability test.

7.1.2. Results

In terms of difficulty, all participants thought that the control condition was fairly well balanced after they had gotten used to the game's controls. One participant felt that the last set of challenges in the second level had somewhat of a surprising difficulty spike compared to the other challenges presented across the two levels. For the experimental condition however, all participants noted that the game was a bit more difficult, mostly because participants felt that the active action-based controls took slightly longer to get used to. The same participant who had a bit of difficulty getting past a specific section in the second level, had even more trouble getting past said section in the experimental condition, but eventually did succeed in progressing to the third level. While not necessarily related to difficulty, two participants noted that the safe moving platforms were way too slow and should have their speed increased since it ruined their momentum.

In regards to technical issues, a few of these were experienced across both variations of the level. One participant got stuck between two quadrilateral shapes in the level design and had to restart the level since it was impossible to free the playable character. One technical issue was however experienced by all four participants for the experimental condition. For some reason, the playable character would suddenly not respond to the participants' dragging gesture which should move the character forwards or backwards. The jumping and gravity-reverse mechanics could still be performed without issue, but the movement gesture stopped responding. This issue could however quickly be resolved by performing a dragging gesture followed by a swiping gesture, without breaking contact between the dragging thumb and the touchscreen; thus allowing participants to progress.

Looking at the gathered data for the two button layouts in the control condition, it was clear that the difference between these two was fairly small. All four participants found both of the layouts suitable for the game but when asked which one they prefered, three of them chose the layout that presented the action buttons on the right side of the touchscreen in a more diagonal pattern, while one participant prefered the layout where both buttons were located at the same horizontal height. In terms of the two different gravity-reverse gestures however, the gathered data clearly showed that the single tap gesture for the gravity-reverse mechanic was prefered. This was an opinion shared by all four participants with two of them noting that having one distinct gesture for each mechanic worked better, than having to constantly keep in mind which way to swipe in order to either jump or activate the gravity-mechanic.

Based on the gathered data from the usability test, it was thus decided that the control condition should utilize a diagonal layout for the action buttons (jumping and gravity-reverse), and that the experimental condition's gravity-reverse should be activated by a single tap rather than a downward swipe. The difficulty in the second level was also lowered a bit in order to ensure that future participants who do not have much mobile gaming experience would not become stuck due to a sudden difficulty spike. Time was furthermore allocated towards figuring out why the dragging gesture would sometimes become deactivated in the experimental condition.

7.2. Pilot Test

One day before the final evaluation began, a pilot test was conducted. Due to the fact that the daily amount of COVID-19 cases had continued to rise since the usability test, it was decided that the pilot test and final evaluation would be performed online. This decision was made out of safety for both the researchers and participants, and meant that a few changes were made in regards to the evaluation setup and strategies presented in section 4.4. First of all, the evaluation now took place in a natural environment rather than a controlled environment since participants would perform the evaluation in their own home without being observed by researchers. The human observation elements of the evaluation strategies were also removed due to the reasons explained in section 4.1. Performing the card sorting session online meant that participant's elaborated reasoning for choosing specific cards was instead done through a written format rather than orally. Finally, the sampling method was changed from the simple random sampling method to the criterion sampling method, since it was now a requirement that the participants owned an android smartphone.

The purpose of the pilot test was to ensure that the evaluation strategies for the final evaluation functioned in an online format and that the data gathered was of the desired quality. The pilot test was furthermore also used to uncover any game-breaking technical issue that could impact the gathered data. Two participants were gathered using the criterion sampling method for the pilot test. The participant who played the experimental condition was made aware of said versions technical issue where the dragging-gesture would suddenly not function as intended - as this issue could not be resolved before the pilot test and final evaluation had to begin. No other technical issue appeared however, and both participants seemed to properly understand the questionnaire and the card sorting session, and thus everything was ready for the final evaluation.

7.3. Final Evaluation

With the usability test having shaped some key elements of the game's design, and the pilot test having ensured that the evaluation strategies worked in an online format and provided the desired quality of the gathered data, the final evaluation was conducted to investigate whether or not there is a difference in player involvement between basic (control condition) and action action-based (experimental condition) control schemes. The changes made for the evaluation strategies of the final evaluation were identical to those made in the pilot test (see section 7.2)

7.3.1. Demographics

The first section of the questionnaire asked participants to provide information about their age, gender and level of experience with games. This was done in order to get a clearer understanding of the general demographics of the participants, and also get an overview of how skillful the participants were with games prior to testing, since this could potentially have a clear relation to the gathered data. The age demographics in the two conditions were relatively similar with the majority being in an age range of 23-27. The control condition did have slightly more participants outside the aforementioned age range (see figure 32).

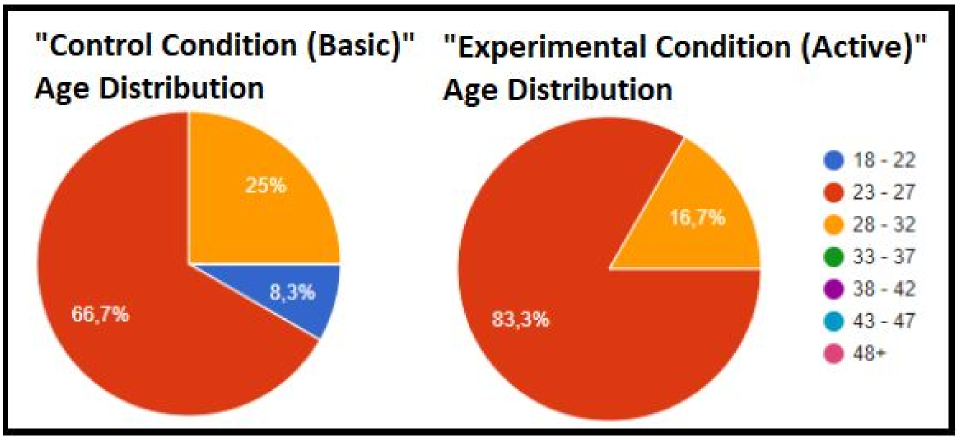

Figure 32. Age distribution between the two test conditions.

Participants were also asked to rate their level of experience in both games as a whole and in mobile games specifically on a scale between 1 (the equivalent of being very inexperienced) and 7 (the equivalent of being very experienced).

The participants' level of experience in both games and mobile games were higher for the experimental condition than in the control condition. However, both means for mobile games were between 4-5, which suggests that the participants have an average level experience playing mobile games (see figure 33).

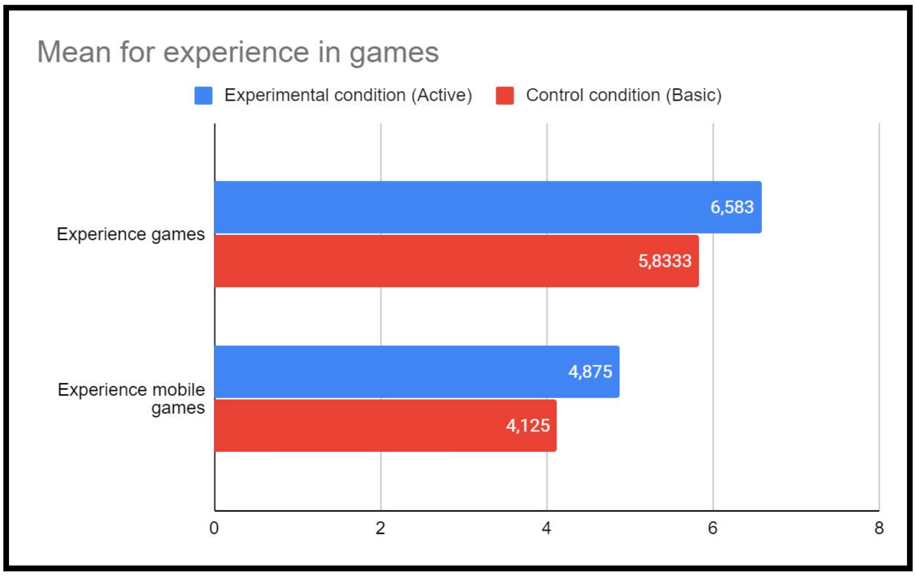

Figure 33. Mean scores for experience with games prior to testing.

In regards to the gender distribution between the two conditions, it is apparent that the control condition had a slightly more even representation between the genders than the experimental condition had (see figure 34).

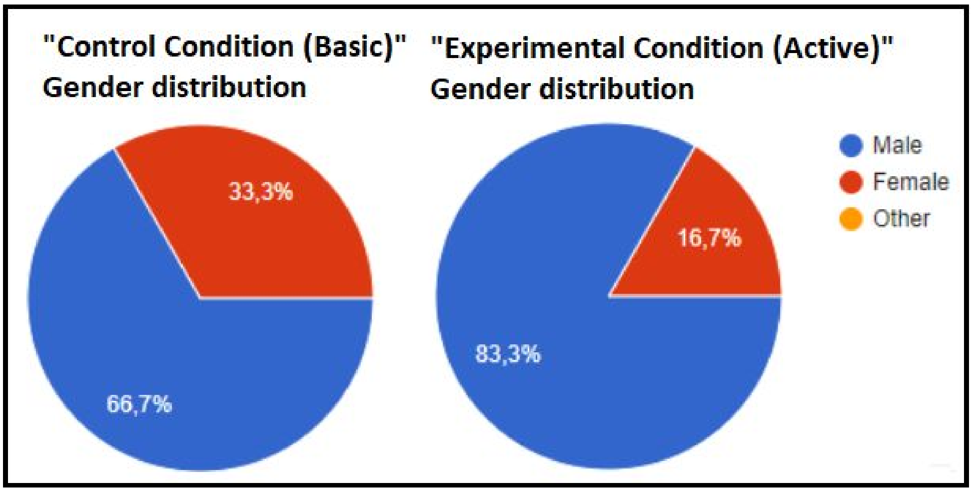

Figure 34. Gender distribution between the two test conditions.

7.3.2. Procedure

Given that the final evaluation was conducted online, interaction between researchers and participants was limited. In order to ensure that participants received the same introduction, a mail (see appendix A.7) which provided a detailed description of the evaluation procedure was sent to each participant. This email furthermore also included a link to downloading the game and another link to the questionnaire and card sorting. After having read the introductory mail, participants were asked to download and play the game. Upon completing the game, participants were asked to open up the questionnaire and read the consent form. The consent form explained that the evaluation was anonymous, while also briefly elaborating on what the gathered data was, what it was going to be used for and that by submitting the questionnaire, the participants were allowing the researchers to use the data for research purposes. Then participants would then fill out the questionnaire which presented several likert items – focusing on certain elements of the game such as controls, difficulty, visuals, sounds, and more (see appendix A.4). The questionnaire was then followed by the remote card sorting session where participants were introduced to five different categories centered around topics such as controls, overall experience, difficulty and more. Each category contained six cards and participants were asked to choose two cards from each category that best described their experience with the game. The participants were then required to briefly elaborate on the reasons for their choice of cards in a written format. Finally, they would be asked to choose three of their ten chosen cards, which they felt represented their experience more than the other seven. Participants were gathered using the criterion sampling method and a total of 48 participants took part in the final evaluation (24 for each condition).

7.3.3. Questionnaire and Statistical Test

To answer the final problem statement, a likert-scale questionnaire was used. The questionnaire included 25 separate likert items; each of them with scores ranging from 1 to 7. The 25 likert items were separated into five categories related to the concept of player involvement. These categories were *kinesthetic involvement (control of actions), spatial involvement, tactical involvement, affective involvement,* and *flow* (see appendix A.3). The data gathered from the two questionnaires turned out to be parametric which meant that each category could be analyzed using an independent t-test (see appendix A.2). If the gathered data had turned out to be non-parametric, a Mann-Whitney U test would have been utilized instead (see appendix A.2 or A.1)

Questionnaire Results for Kinesthetic Involvement (Control of Actions)

The category for *kinesthetic involvement* contains nine questions which were analyzed between the two conditions. The hypotheses for the kinesthetic involvement category were:

- H0: There is no significant difference in the kinesthetic involvement scores between the control and experimental conditions.
- H1: There is a significant difference in the kinesthetic involvement scores between the control and experimental conditions

Through the independent two tailed t-test for the kinesthetic involvement category, the H0 is rejected given its p-value of 1.0662e-05, which is significantly less than the alpha of 0.05. With the H0 having been rejected, uncovering how the data is different is important. Therefore a right tailed t-test with new hypotheses will be conducted.

- H0: The control and experimental conditions have equally good kinesthetic involvement scores
- H1: The control condition performs better than the experimental condition

The right tailed t-test also rejects the H0 with a p-value of 5.3311e-06, which is significantly lower than the alpha value of 0.05. This means that it can be assumed that the control condition performs better when kinesthetic involvement is measured. To further investigate how the control condition performs compared to the experimental condition, the mean scores for both conditions were plotted in a boxplot (see figure 35). The statistical descriptive data of the control condition (M = 4.5556, SD = 0.9139, SE = 0.1865) and the experimental condition (M = 3.2361, SD = 0.9356, SE = 0.1910) also differ. Here, it can be seen that there is a difference of 1.3195 between mean scores in favor of the control condition.

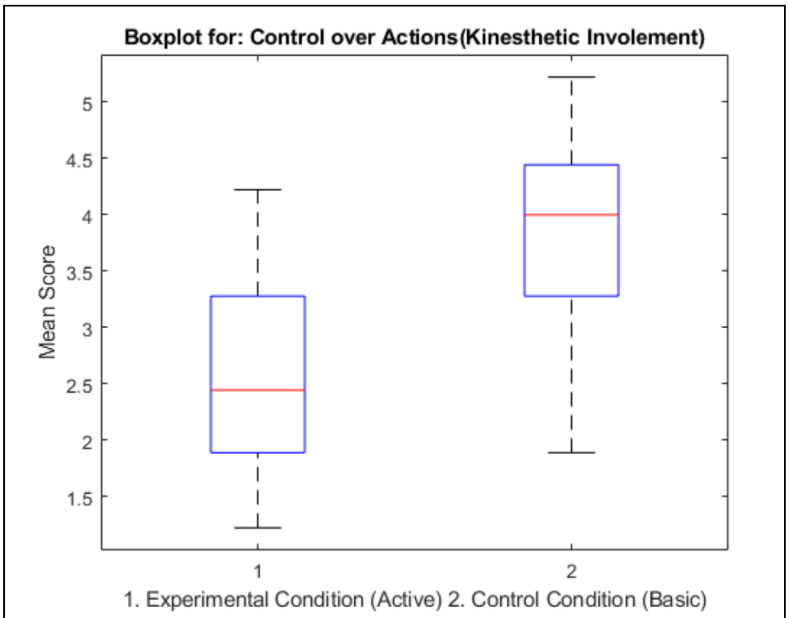

Figure 35. Boxplot for the 'control of actions' category

Questionnaire Result for Spatial Involvement

The category for *spatial involvement* contains five questions which were analyzed between the two conditions. The hypotheses for the spatial involvement category were:

- H0: There is no significant difference in the spatial involvement scores between the control and experimental conditions
- H1: There is a significant difference in the spatial involvement scores between the control and experimental conditions

Through the independent two tailed t-test for the spatial involvement category, the H0 is accepted with a p-value of 0.1652, which is more than the alpha of 0.05. This means that no further investigation is required since the difference is not significant enough to be used. To investigate how the control condition performs compared to the experimental condition (however significant), the mean scores for both conditions were plotted in a boxplot (see figure 36). The statistical descriptive data of control condition (M = 4.4750, SD = 0.9322, SE = 0.1903) and the experimental condition (M = 4.8333, SD = 0.8250, SE = 0.1684) also slightly differ. Here, it can be seen that there is a difference of 0,3583 between mean scores in the favor of the experimental condition.

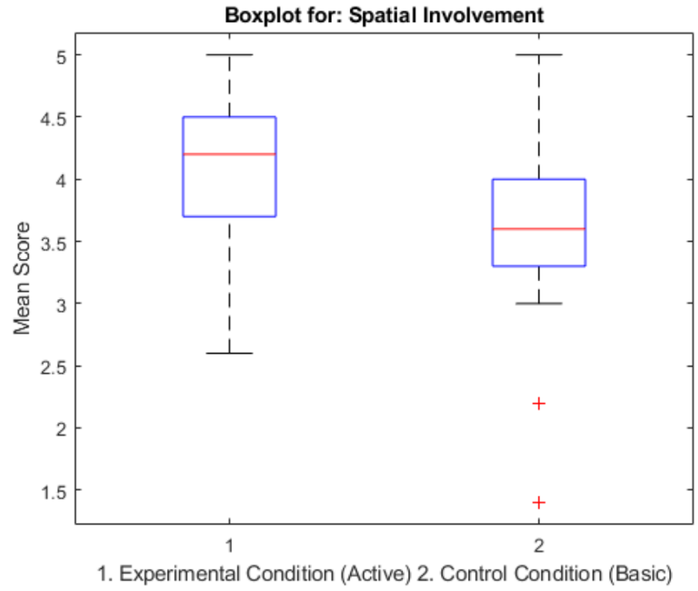

Figure 36. Boxplot for the 'spatial involvement' category

Questionnaire Results for Tactical Involvement

The *tactical involvement* category contains five questions which were also analyzed between the two conditions. The hypotheses for the spatial involvement category were:

- H0: There is no significant difference in the tactical involvement scores between the control and experimental conditions
- H1: There is a significant difference in the tactical involvement scores between the control and experimental conditions

Through the independent two tailed t-test for the tactical involvement category, the H0 is accepted given its p-value of 0.0939, which is more than the alpha of 0.05. This means that no further investigation is needed since the difference is not significant enough. To further investigate how the control condition performs compared to the experimental conditions in regards to tactical involvement (however significant), the mean scores were plotted in a boxplot (see figure 37). The statistical descriptive data of the control condition (M = 5.1389, SD = 1.0164, SE = 0.2075) and the experimental condition (M = 4.5417, SD = 1.3755, SE = 0.2808) is slightly different. Here, it can be seen that there is a difference of 0,5972 between mean scores in the favor of the control condition.

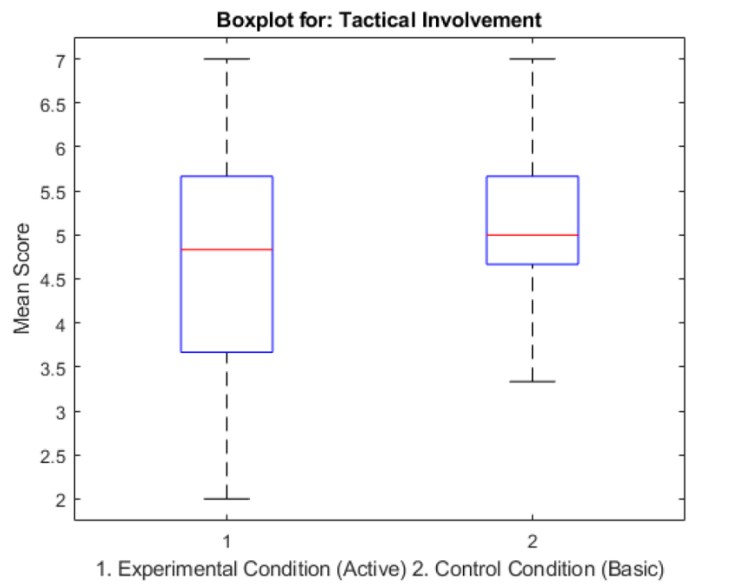

Figure 37. Boxplot for the 'tactical involvement' category

Questionnaire Results for Affective Involvement

The category of *affective involvement* contains three questions which were also analyzed between the two conditions. The hypotheses for the affective involvement category were:

- H0: There is no significant difference in the affective involvement scores between the control and experimental conditions
- H1: There is a significant difference in the affective involvement scores between the control and experimental conditions

For the affective involvement category, the independent two tailed t-test outputs a p-value of 0.7631, which means that the H0 is accepted since this value is above the alpha of 0.05. This means that no further investigation is required since the difference is not significant enough. To gain a broader understanding of how the control condition performs

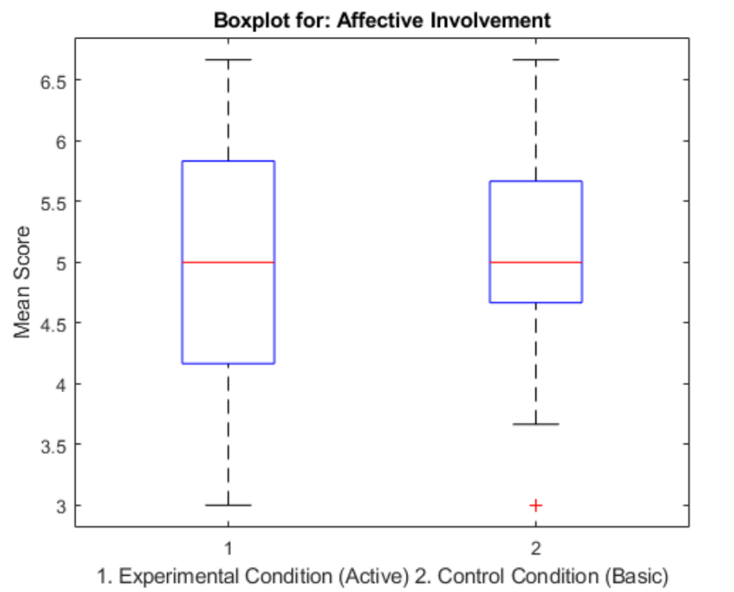

Figure 38. Boxplot for the 'affective involvement' category

in comparison to the experimental condition in regards to affective involvement (however significant), the mean scores were plotted in a boxplot (see figure 38). The statistical descriptive data of the control condition (M = 4.9722, SD = 0.8101, SE = 0.1654) and the experimental condition (M = 4.8889, SD = 1.0753, SE = 0.2195) is very similar. Here, a very small difference of 0,0833 between the mean scores can be seen in the favor of the control condition.

Questionnaire Results for flow

The *flow* category contains five questions which were analyzed between the two conditions. The hypotheses for the flow category were:

- H0: There is no significant difference in the flow scores between the control and experimental conditions
- H1: There is a significant difference in the flow scores between the control and experimental conditions

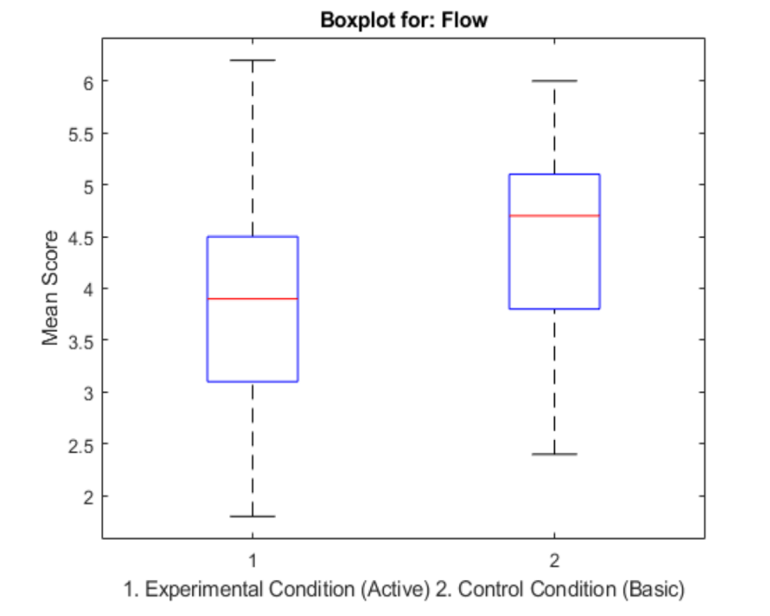

Figure 39. Boxplot for the 'flow' category

Through the independent two tailed t-test for the flow category, the H0 is accepted with a p-value of 0.0537, which is more than the alpha of 0.05. This means that no further investigation was performed. To investigate how the control condition performs compared to the experimental condition (however significant), the mean scores for both conditions were plotted in a boxplot (see figure 39). The difference in the statistical descriptive data of the control condition (M = 4.5083, SD = 0.9399, SE = 0.1919) and the experimental condition (M = 3.9250, SD = 1.0951, SE = 0.2235) is noticeable. Here, a difference of 0,5833 between the mean scores is apparent in the favor of the control condition.

Questionnaire Results for Overall Involvement

The following section covers the entire questionnaire with the aim of getting an idea of the *overall involvement* measured from the questionnaire. This section thus contains all 25 questions which were analyzed between the two conditions. The hypotheses for the overall involvement were:

- H0: There is no significant difference in overall involvement scores between the control and experimental condition
- H1: There is a significant difference in overall involvement scores between the control and experimental condition

Through the independent two tailed t-test for the overall involvement level, the H0 is rejected given its p-value of 0.0058, which is lower than the acceptable alpha value of 0.05. With the H0 having been rejected, investigating how the data is different is of interest. Therefore a right tailed t-test with new hypotheses will be conducted.

- H0: The control and experimental conditions have equally good overall involvement scores
- H1: The control condition performs better than the experimental condition in overall involvement

The right tailed t-test also rejects the H0 with a p value of 0.0029, which is also much lower than the alpha value of 0.05. This means that it can be assumed that the control condition performs better in regards to the overall involvement.

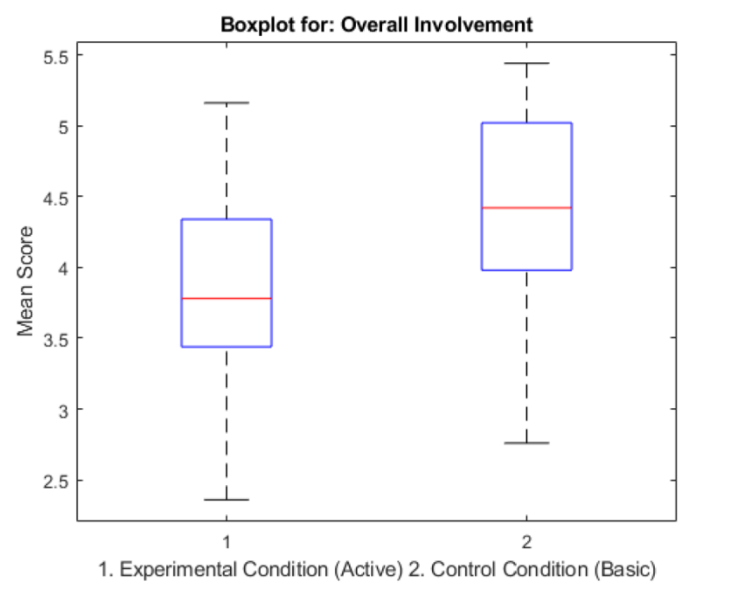

Figure 40. Boxplot for the 'overall involvement' category

To gain a broader understanding of how the control condition performs better in comparison to the experimental condition in regards to overall involvement, the mean scores were plotted in a boxplot (see figure 40). The difference in the statistical descriptive data of the control condition (M = 4.6500, SD = 0.6767, SE = 0.1381) and the experimental condition (M =

4.0483, SD = 0.7605, SE = 0.1552) also differ. Here, it can be seen that there is a difference of 0,6017 between mean scores in the favor of the control condition.

7.3.4. Cards Sorting Results

With the data collected from the card sorting session, the frequency of chosen cards from the five categories were analyzed and used to compare the overall feeling of involvement between the two conditions.

7.3.4.1. Card Frequency Results

Analyzing the frequency of cards chosen in the card sorting sessions can provide knowledge about which elements of the game causes a difference in the participants' overall level of involvement between the two conditions. This segment will document the main differences in frequency of cards between the control and experimental conditions.

In regards to the *controls of action* category, it is clear the participants in the control condition had a more positive experience than those in the experimental condition. Particularly, the *'Intuitive'* card was chosen by a total of 15 participants in the control condition with participant 14 noting; *"I felt that [the] controls were intuitive and the output of controls were precise enough"*, while participant 42 stated that; *"It was very easy to [use] the controls and they were very intuitive and precise […]"*. Participants in the control condition also chose cards such as *'Precise'* and *'Enjoyable'* more often than participants in the experimental condition. A clear pattern can also be seen in regards to the negatively-loaded cards in this category. The *'Frustrating'* and *'Unresponsive'* cards were chosen by participants in the experimental condition a total of 16 and 13 times respectively. The general consensus on the controls in this condition is summed up quite well by participant 9 who noted; *"The way the controls work is very intuitive, however the horizontal movement has too long [of a] startup, which makes it feel somewhat unresponsive. Sometimes when I wanted [to] move slightly to the side I ended up reversing gravity instead, which was very frustrating in the early levels with a long distance between checkpoints"*. Even though there is a clear pattern in which cards were chosen by participants in the two conditions, it should be noted that not all participants in the control condition were positive about this category, but in some of these cases the issues were also related to the game itself rather than only the controls. Participant 28 noted that; *"The controls do what they are supposed to. However, combining walking and jumping/gravity shifting did sometimes feel a bit awkward and controls lack tactility. Gravity seems a little inconsistent, sometimes I fall gently and* *other times I fall like a rock".* Based on the responses and the cards chosen in this category, it is clear that much of the frustration felt by participants revolve around the gesture detection aspect of the game (see figure 41 and appendix A.0).

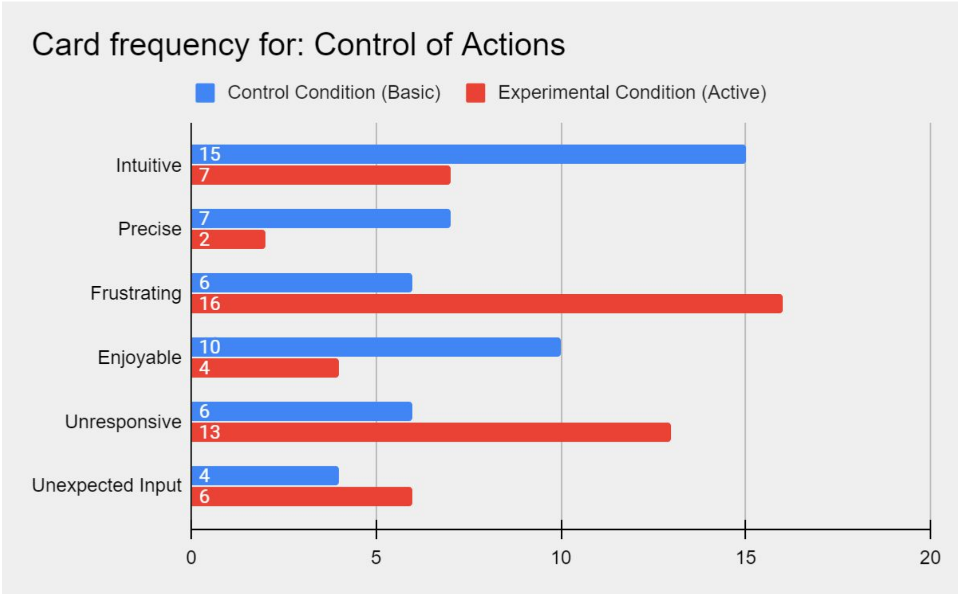

Figure 41. Card frequency results for the 'control of actions' category.

For the *visuals and sounds* category, the card frequency distribution between the two conditions are relatively similar and somewhat equally spread out over all possible cards. Theoretically, this outcome was to be expected since the visuals and sounds are identical in both conditions. Based on the responses, it is clear that the simplistic nature of the game's visuals were very hit and miss with the participants (see figure 42).

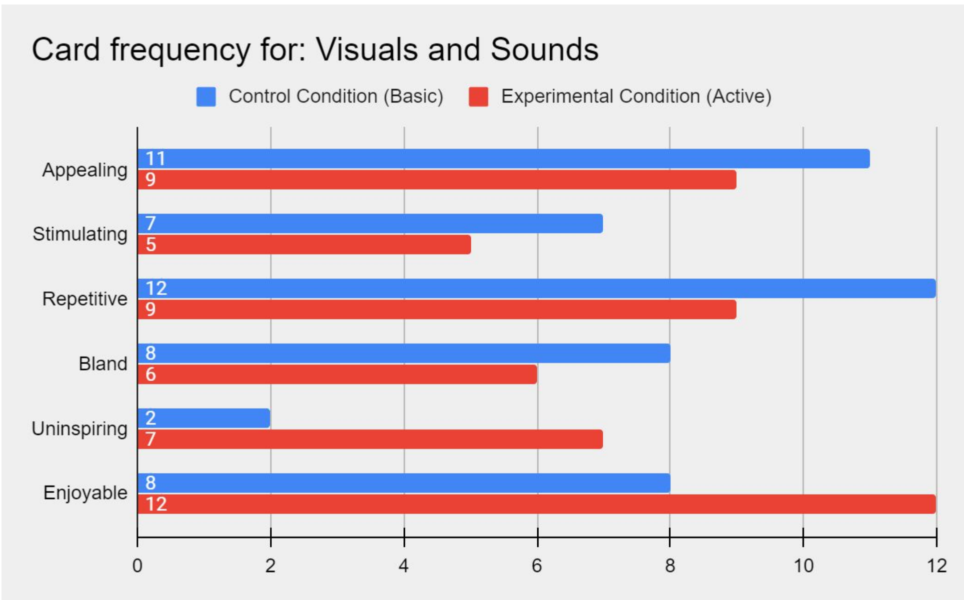

Figure 42. Card frequency results for the 'visuals and sounds' category.

In terms of the *navigation, rules and goals* category, the most substantial difference in chosen cards between the two conditions involve the cards; *'Confusing'* and *'Clear'*. In the experimental conditions there seemed to be a general consensus that the navigation, rules and goals were pretty clear with a total of 20 participants in said condition having chosen the *'Clear'* card. Participant 13 noted it was; *"very clear how the character is able to navigate the world […]"*, while participant 5 stated; *"My goal was clear and how to achieve it was clear as well"*. For some participants, what was not clear to them would eventually become so after some time, which can be seen in the response from participant 17; *"This stuff [was] clear and simple. Was a bit confused on the meaning of some of the colors but I figured it out by trial-and-error".* While a total of 10 participants in the control condition would also choose the *'Clear'* card, just as many would also choose the *'Confusing'* card, with participant 6 noting that; *"It was a little confusing that I could solve the puzzle in multiple ways […]"*. Even though 11 participants in the control condition chose the *'Confusing'* card, it is interesting to note that 15 of them also chose the *'Interesting'* card (see figure 43)

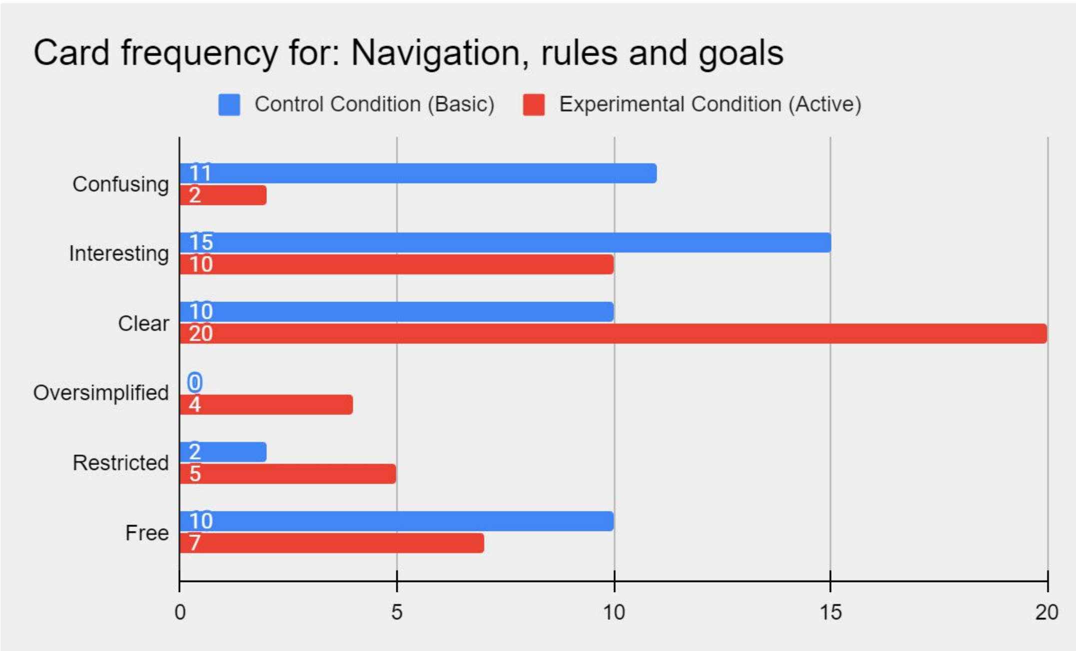

Figure 43. Card frequency results for the 'navigation, rules and goals' category.

The frequency distribution in the *difficulty* category is mainly split between the cards *'Fair'*, *'Balanced'* and *'Too difficult'*. There seems to be a trend between the two conditions in the sense that participants in the experimental condition found the game to be harder than the other participants. This is evident since the *'Too difficult'* card was chosen 14 times in the experimental condition whereas only four participants from the control condition chose said card. Participants in the control condition also chose the *'Fair'* (17 times) and *'Balanced'* (14 times) cards much more than the other participants. Participant 42 from the control condition noted that the game became too easy by stating; *"What i played seemed fine but if i had played any more i would have liked it to get a little more difficult [...]"*. All of this indicates that the level of difficulty is quite noticeable between the two conditions (see figure 44).

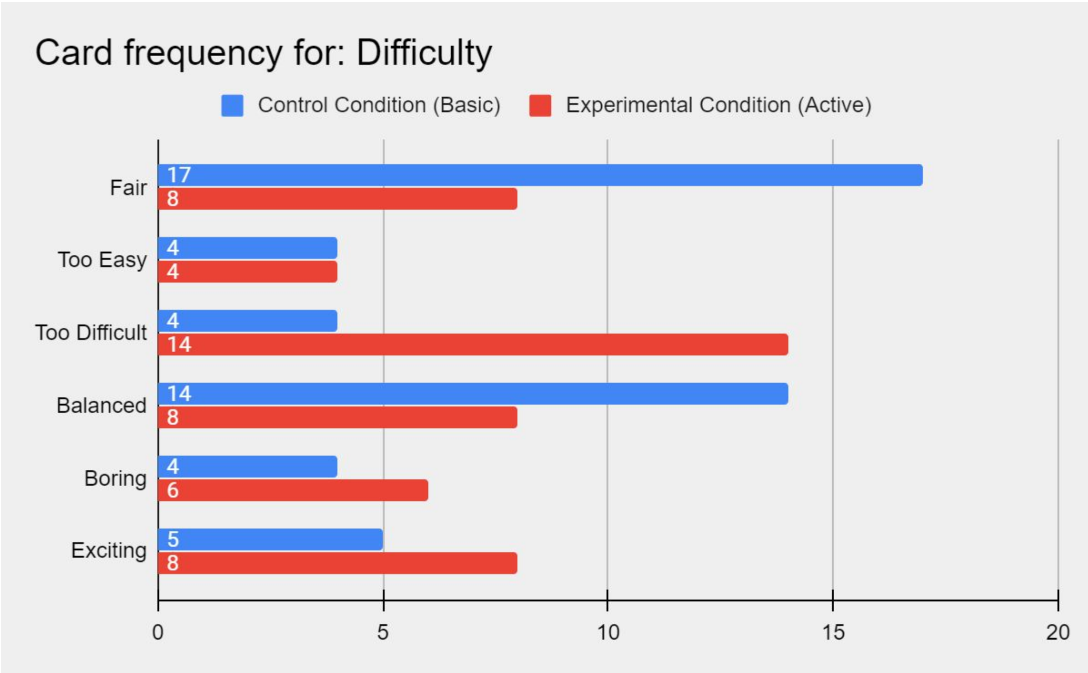

Figure 44. Card frequency results for the 'difficulty' category.

The *overall experience* category of the card sorting session indicates that participants who played the control condition had a more positive experience than those who played the experimental condition. A total of 21 out of 24 participants in the control condition chose the *'Entertaining'* card whereas only 10 would do so in the experimental condition. A card that was chosen more frequently in the experimental condition was the *'Frustrating'* card which was chosen by 19 participants. Contrary to this, the *'Frustrating'* card was only chosen by six participants in the control condition. The reasons for this difference in the *'Frustrating'* and *'Entertaining'* card lies in the participants opinion on the controls. Participant 5 who played the experimental condition noted that *"[...] my only real problem is the controls and the lack of precision"*, while participant 15 stated that *"The overall experience was frustrating due to the controls [...]"*. Participant 44 (who played the control condition) noted that the game was *"Entertaining because of the controls"*. Other reasons for *'Entertainment'* to be picked by participants include an interest in the gameplay loop and the puzzle elements. It should be noted that ten participants in the experimental condition also chose *'Entertaining'* while almost just as many chose the *'Forgettable'* card. For the control condition the second most chosen card was the *'Satisfying'* one (see figure 45).

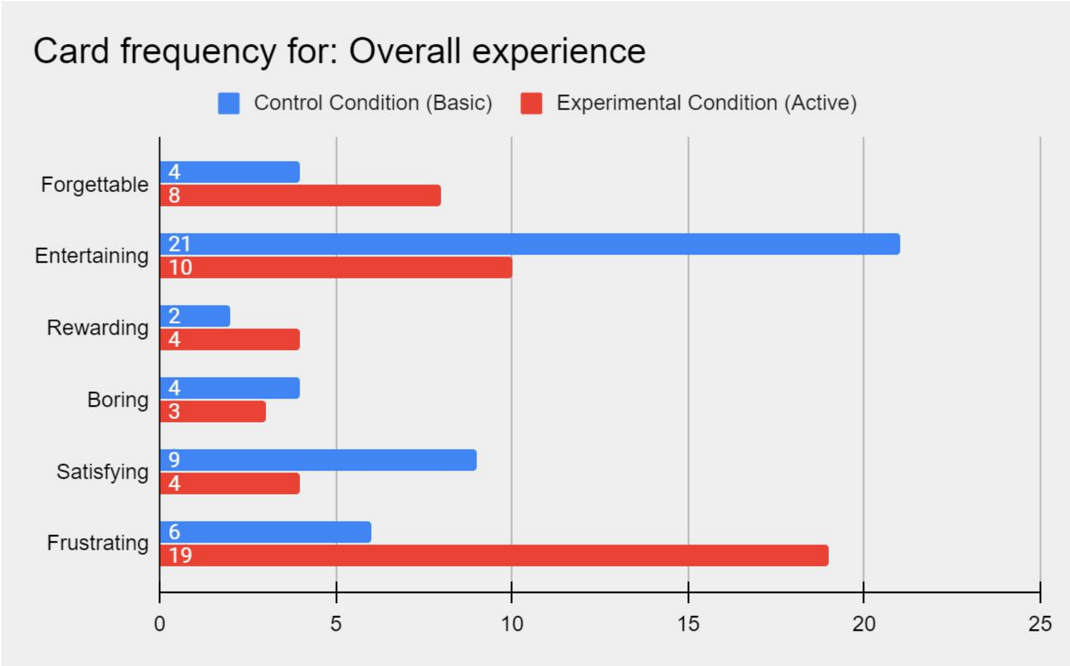

Figure 45. Card frequency results for the 'overall experience' category.

7.3.4.2. Card Sorting Score

The purpose of the card sorting score is to attempt to compare the results from this session with the questionnaire data. This technique is however not a proven method and is used only to describe differences between the card sorting and questionnaire data. Each card has a score that contributes to participant's overall score in the card sorting session; a score that is set between 1 (really not involved) and 7 (really involved), thus mirroring the questionnaire's setup of likert scale items in terms of ranging from answers between 1 and 7. The mean score for the processed data in the control condition is 5,25, while the mean in the experimental condition is 3,6666, meaning that there is a difference of 1.584 between the two conditions. Since a mean score of 4 is considered the neutral score, this means that (overall) participants in the experimental condition were leaning towards being slightly not involved while participants in the control condition were leaning towards being slightly involved. The card sorting scores seem to have a larger difference (1.584) between the two conditions compared to the scores from the questionnaire (0.6017). However, both scores favor the control condition. Generally, it seems that the means for the card sorting are more sensitive to smaller changes than in the questionnaire, due to fewer data points. Ideally, each participant would have been asked if they agreed or disagreed with their placement on the "level of involvement scale" in order to validate the card sorting scores, since they potentially still could have been involved even if their chosen cards indicated otherwise. This was however not done because participants most likely would not know the exact definition of involvement and therefore would not be able to answer the question faithfully, which would cause the data to become less valid.

8. Discussion

As concluded in the final evaluation (see section 7.3), the basic action-based control scheme performed better than the active action-based control scheme. This could be seen in the statistical test performed on the *kinesthetic involvement (control of actions)* category from the questionnaire, as well as the test performed on *overall involvement* (which included the entire questionnaire), whereas the other categories did not indicate a difference between the two conditions. The difference in overall involvement was further supported by the results gathered from the card sorting session. The following chapter will investigate why this was the case, what could have been done differently, and explore which bias and sources of error might have influenced the final results.

Changing the test to an online format did have some unwanted effects on the results in terms of bias and potential sources of error. Conducting the questionnaire and card sorting session online meant that participants did not have a chance to ask for clarification on likert items of card definitions in case of potential misunderstandings. Some participants might also not have read the detailed introductory mail, since a few participants ended up picking one or three cards in a category instead of two, which forced the researchers to add or eliminate a card based on the tone of the participant's general answers. Performing the evaluation online also meant that participants had to use their own phone rather than the intended Samsung Galaxy S9 that was used during development. The size of the buttons and the gap between them had been designed to ensure a reasonable size and gap for smaller smartphone's, but if participants used older phones it could results in the game running slower than intended which leads to less responsiveness since the gesture input is tied to the frame-dependant update function in Unity, which in turn could lead to frustration.

Two issues regarding the cards from the card sorting session might have had an unwanted effect on the results. First of all, the card *'Unexpected Input'* should have been named *'Unexpected Output'*. The reason for this is that the card deals with the playable character not acting in accordance with the input provided by the participants, and not with the participant thinking that they accidentally performed an incorrect input. This means that the card might not have served its intended purpose by inaccurately describing what it entails. The second issue is centered around certain words appearing more than once; specifically *'Frustrating'*, *'Boring'* and *'Enjoyable'* all appeared in two categories each. This could potentially have been an issue since those card's impact on the results could have been more prominent than the other cards.

The t-test performed for the *flow* and *difficulty* categories did not show a significant difference between the two conditions. Contrary to this however, the card frequency results suggested a trend centered around the fact that participants in the experimental condition more often found the game to be too difficult. It seems that the innate difficulty of the active action-based control scheme was much bigger than initially thought. This issue could be fixed by making the game activities easier but in doing so, another issue could be introduced; namely that the other condition would become too easy, since certain participants in that condition expressed a desire for more difficulty (see section 7.3.4.1). Certain design choices could however be made to somewhat alleviate the higher difficulty in the experimental condition, such as lowering the amount of precision needed to land on platforms, and increasing the number of checkpoints so that longer sections would not have had to be replayed and thus lessen the participants frustration.

Another potential issue with the evaluation strategy was the lack of any observational data. By not recording or witnessing the participants' gameplay session, there is no way of confirming what happened beyond the individual participant's own account and memory. This means that if participants did not describe the technical issues they experienced, there is no way of knowing that these issues exist. However, recording a gameplay session would have required the participant's themselves to set up an external screen-recording application and this was viewed as being too much of an inconvenience for the participants (see section 4.1). The most optimal way of circumventing this issue would have been to perform the evaluation in-person but this was not an option due to the COVID-19 situation.

8.1 Sources of Error

Three technical issues ended up having had an impact on the gathered data, and all three of them are contained within the experimental condition. The first issue was that sometimes, participants were not able to activate the gravity-reverse mechanic mid-air unless they performed two single tap gestures on the touchscreen. This meant that the playable character would (in some cases) either soar up or drop down onto the level boundaries and be sent back to the nearest checkpoint, even if they had performed the gravity-reverse gesture properly.

The second technical issue was centered around the dragging-based gesture (which purpose was to move the playable character) which would occasionally stop functioning. This was an issue that was discovered during the usability test but a fix for it was unfortunately not found in time before the final evaluation was conducted. Participants in the experimental condition were made aware of the issue in the introductory mail as well as the method resolving it in-game by consecutively performing two different input gestures (see section 7.1.2). However, even if participants were made aware of the issue, it was clear from the card sorting reponses that it (understandably) still caused some frustration when it happened.

Finally, the last technical issue had to do with the responsiveness of the controls. The issue seemed to be that participants would accidentally initiate a gravity-reverse when they meant to move the character forward; causing a lot of frustrating game over states when (in some cases) their playable character would collide with one of the level's boundary boxes. Contrary to the two other issues, this one was experienced by quite a few participants, which means that the data gathered for the *kinesthetic involvement* and *overall involvement* might have skewed the results in favor of the control condition. For this issue to be so prominent for multiple participants, it seems that the accepted distance between the initial press and the release for a single tap have been too large. Further polishing of these threshold values would be a necessity to fix this issue, especially since these small inaccuracies have led to such frustration among the participants in the experimental condition.

9. Re-design

Both technical and design-related issues were found during both the usability test and the final evaluation. In order to improve the quality of the game and the study as a whole, these issues should be fixed.

Based on the results from the card sorting session (see section 7.3.4.1), it was clear that some frustration was experienced among the participants in the experimental condition in regards to the gesture detection for the active action-based control scheme. Improvements that can be made to fix these issues would be to polish the variable threshold for touch input so that the gesture detection will be more accurate.

Certain changes should also be made to the game's difficulty, since many participants felt that the game was too hard. A few changes could be made to lessen this problem. One of these changes could be to design a better difficulty curve, since some participants expressed that the second level was the hardest to complete, while another change could be to add more checkpoints so that participants would have to replay longer sections in case they fail. Finally, given that the game aims to provide players with both challenging traversal and interesting puzzles, it could be beneficial for the game if some of the difficulty was placed onto the puzzles, rather than mostly being centered around the traversal elements.

Since quite a lot of participants in the control condition chose the *'Confusing'* card in the *navigation, rules and goals* category of the card sorting, it would be a good idea to design a more extensive tutorial level that properly introduce the player to the controls, the gameplay mechanics, and what the significance of the different colored platforms are. It should e.g. be much clearer to the player that the green platforms activate the next purple gravity wall. This could be improved by initially having the gravity wall be gray and then turn purple once it has been activated.

In order to enhance the relationship between the players' input and the gesture detection, visual indicators of the gestures performed by players in the active action-based control scheme would ensure that it is clear to players which input gesture they had performed.

In order to improve the evaluation strategies, any future evaluation should be performed in a controlled environment where participants and researchers are in the same room. This would provide more in-depth data from the card sorting session, since a natural dialogue can be held between the participant and the researcher. It would furthermore also allow participants to ask for clarification in case they do not understand the meaning of certain cards from the card sorting session or likert items from the questionnaire. Finally, observational data could also be gathered (without being an inconvenience for the participant) which means that technical issues can be clearly identified by researchers during gameplay sessions.

In order to gather more consistent data, any future evaluation should furthermore ensure that all participants play the game on the same type of smartphone; something that would be much simpler during an evaluation performed in a controlled environment, since the researchers could easily provide the smartphone. This would eliminate potential issues such as participants experiencing the game through different screen qualities and frame rates.

10. Conclusion

In the beginning of this study, an initial problem statement was formulated with the aim of investigating the difference in induced player involvement across certain mobile interaction schemes in a 2D-platformer. The initial problem statement was formulated as follows:

Which type of interaction schemes induce a higher sense of player involvement in terms of basic gameplay actions in a mobile 2D-platformer?

In order to research and evaluate the initial problem statement, a thorough analysis was made centered around related topics such as mobile interaction techniques, mobile touch techniques, design principles for 2D-platformers, and measuring of user experience. In regards to mobile interaction techniques, the analysis explored which types of interaction are commonly used for general mobile interaction and it was found that touch interaction was the most common type. By further investigating touch interaction, it was found that a meaningful way of categorizing touch gestures was through basic and active action-based gestures, which then formed the basis of the two interaction schemes evaluated in this study. In order to develop a mobile 2D-platformer, different aspects related to design principles of this genre were investigated, and the findings indicated that 2D-platformers are made from certain building blocks such as platforms, obstacles, triggers and rhythm groups. Research was also conducted to explore different methods for measuring user experience, and here it was found that different ways doing so can be done through playability heuristics, player engagement and player involvement. The player involvement model was chosen for this study as the method for describing and measuring the player experience. Finally, the touch-input controls of existing 2D-platformers were explored to uncover how other developers go about designing basic and active-based input controls. With these findings in mind, a final problem statement was formulated

How does changing a basic action-based control scheme with an active action-based control scheme impact the player's overall level of involvement in a 2d-platformer?

With the design requirements in mind, a simple 2D-platformer was designed to function with two different control schemes. The game was sent out to participants for evaluation, where their player involvement was measured through the use of a questionnaire and a card sorting session.

The results of the independent t-test performed on the gathered questionnaire data indicated that there was a significant difference between the two test conditions in the *overall involvement* and *kinesthetic involvement (controls of action)* categories. In the *overall involvement* category, the t-test showed a significant difference (p-value of 0.0058) in favor of the control condition (p-value of 0.0029). The *kinesthetic involvement (controls of action)* category, also showed a significant difference (p-value of 1.0662e-05) in the favor of the control condition (p-value of 5.3311e-06). This difference was further supported by the results from the card sorting session, where participants in the experimental condition often chose negatively-loaded cards when describing the control scheme of said condition. Also, the mean scores for the card sorting session between the two conditions showed a difference of 1.584 in the favor of the control condition. This shows that the basic action-based control scheme (control condition) performed better than the active action-based control scheme (experimental condition) in terms of inducing player involvement.

11. Future Works

This study focused on measuring the difference in player involvement between two types of touch input but as seen in section 2.1.1, other mobile interaction techniques were explored but not utilized in this study. In order to gain an even broader understanding on the subject of player involvement across interaction schemes, these alternative interaction techniques should also be investigated and compared to the results from the touch-based input. Particularly a motion-based control scheme could provide interesting data, since this type of involvement requires more physicality in providing the smartphone with input. Coupled with the results found in this study, data from a motion-based control scheme could lead to further studies that would attempt to combine these three type interaction techniques into the most optimal hybrid-based control scheme.

Since this study solely focused on 2D-platformers, the results are thus centered around said genre of games. In order to further investigate if a basic action-based control scheme is superior in inducing player involvement in comparison to an active action-based control scheme, other popular genres of games within the mobile gaming industry should be investigated.

An aspect that could improve the validity of the results, would be to investigate how player involvement could be measured through technical approaches such as facial recognition, GSR or heart rate. This investigation would aim to develop new and innovative methods that could function as an addition to the questionnaire and card sorting utilized for measuring player involvement.

In relation to the gesture detection thresholds (from the experimental condition) that needed polishing, more investigation should be made into finding optimal thresholds that are scientifically proven. This investigation would be more tailored towards a performance aspect rather than user experience.

It could also be beneficial to investigate if the results from the final evaluation would be similar if a different evaluation strategy was utilized. An alternative approach would be to perform a dependent test where participants would be allowed to try out both of the control schemes and thus provide valuable qualitative data about which control scheme they preferred. However, when performing a dependent test, one has to remember that the second play-session is biased so to counter this issue, the participants should alternate between which condition they would try first, meaning that some participants should start with the experimental condition, whereas others should start with the control condition.

References

- 1337 & Senri (2014), *Leo's Fortune*, video game, Tilting Point, Sweden.
- Abrams, R., & Christ, S. (2003). Motion Onset Captures Attention. Phychological Science 14(5), pp. 427-432.
- Adventure Islands (2014), *Duke Dashington*, video game, Adventure Islands.
- Affairs, A. (2018). Card Sorting | Usability.gov . [online] Usability.gov. Available at: https://www.usability.gov/how-to-and-tools/methods/card-sorting.html [Accessed 30 April 2020].
- Alper Sarikaya (2015), *Sword of Xolan*, video game, Alper Sarikaya.
- Baldauf, M., Fröhlich, P., Adegeye, F., & Suette, S. (2015). Investigating On-Screen Gamepad Designs for Smartphone-Controlled Video Games. *ACM Transactions On Multimedia Computing, Communications, And Applications*, *12*(1s), 1-21. doi: 10.1145/2808202
- Brandwesson (2011). *CollectCoin.wav*. Available at: https://freesound.org/people/bradwesson/sounds/135936/
- Brown, E., & Cairns, P. (2004). A grounded investigation of game immersion, 1297-1300. Retrieved from https://dl.acm.org/citation.cfm?id=986048.
- Bjørner, T. (2015) Qualitative Methods for Consumer Research. Hans Reitzels Forlag.
- Calleja, G. (2007). Digital Game Involvement. Games and culture, 2(3), 236-260. doi: 10.1177/1555412007306206
- Calleja, G. (2011). In-game. Cambridge, Mass.: MIT Press.
- Capcom (2010), *Street Fighter IV*, video game, Capcom, Japan.
- Chaichitwanidchakol, P., & Feungchan, W. (2018). Exploring Mobile Game Interactions. *International Journal Of Electrical And Computer Engineering (IJECE)*, *8*(5), 3954. doi: 10.11591/ijece.v8i5, pp. 3954-3965.
- Cohen, D., & Crabtree, B. (2006). Qualitative Research Guidelines Project. Available at: http://www.qualres.org/HomeCrit-3814.html
- Compton, K., & Mateas, M. (2006). Procedural Level Design for Platform Games. American Association for Artificial Intelligence, pp. 109-111.
- Conradi, J., Busch, O., & Alexander, T. (2015). Optimal Touch Button Size for the use of Mobile Devices while Walking. *Procedia Manufacturing*, *3*, 387-394. doi: 10.1016/j.promfg.2015.07.182
- Csikszentmihalyi, M. (2014). Flow and the foundations of positive psychology (pp. 227-236). Springer.

- Csikszentmihalyi, M. (1990). Flow: The Psychology of Optimal Experience. HarperCollins.
- Creative Bytes Studios (2019), *Hero Versus*, video game, Creative Bytes Studios.
- Das, P., Zhu, M., McLaughlin, L., Bilgrami, Z., & Milanaik, R. (2017). Augmented Reality Video Games: New Possibilities and Implications for Children and Adolescents. *Multimodal Technologies And Interaction*, *1*(2), 8. doi: 10.3390/mti1020008
- Dickley, M. (2015). Aesthetics and Design for Game-Based Learning. Routledge.
- DinV Studio (2017). *Dynamic Space Background Lite*. Available at: https://assetstore.unity.com/packages/2d/textures-materials/dynamic-space-background-lite-104606
- Du, Y., Ren, H., Pan, G., & Li, S. (2011). Tilt & Touch: Mobile Phone for 3D Interaction. UbiComp'11, Beijing, China, pp 485-486.
- Gelasca, E., Tomasic, D., & Ebrahimi, T. (2005). Which Colors Best Catch Your Eyes: A Subjective Study of Color Saliency. Ecole Polytechnique Federale de Laussane.
- Halfbrick Studios (2010), *Fruit Ninja*, video game, Halfbrick Studios, Australia.
- Gilbertson, P., Coulton, P., Chehimi, F., & Vajk, T. (2008). Using "tilt" as an interface to control "no-button" 3-D mobile games. *Computers In Entertainment*, *6*(3), 1-13. doi: 10.1145/1394021.1394031
- Gokul, G., Yan, Y., Dantu, K., Ko, S., & Ziarek, L. (2016). Real Time Sound Processing on Android. Proceedings of the 14th International Workshop on Java Technologies for Real-Time and Embedded Systems, 3, 1-10.
- Gumi (2018), *Dolls Order*, video game, Gumi, Japan.
- Harrison, C. (2020). Improving Mobile Interfaces with Rich-Touch Interactions | UX Magazine. Retrieved 14 August 2020, from https://uxmag.com/articles/improving-mobile-interfaces-with-rich-touch-interactions
- Hinckley, K., Heo, S., Pahud, M., Holz, C., Benko, H., Sellen, A., Banks, R., O'hara, K., Smyth, G., & Buxton, B. (2016). Pre-Touch Sensing for Mobile Interaction. Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16), San Jose, California, USA, pp. 2869-2881.
- Hobart, J. (2005). Designing Mobile Applications. Classic Systems Solution Inc.
- Insight, U. (2020). Tap Vs Swipe: The Good, The Bad, and The Ugly – User Insight. Retrieved 14 August 2020, from https://www.userinsight.com/tap-vs-swipe-good-bad-ugly/
- Khalifa, A., Silva, F., & Togelius, J. (2019). Level Design Patterns in 2D Games. 2019 IEEE Conference on Games.
- LeftRight (2016), *Star Knight*, video game, LeftRight.
- Kim, Y., & Lee, J. (2014). Game interface enhancement under smartphone platform focused on touchscreen interaction. *Computers & Industrial Engineering*, *80*, 45-61. doi: 10.1016/j.cie.2014.11.017

- Kim, Y., & Lee, J. (2013). Game interface enhancement under smartphone platform focused on touchscreen interaction.
- Korhonen, H., & Koivisto, E. M. (2006). Playability heuristics for mobile games. Proceedings of the 8th conference on Human-computer interaction with mobile devices and services - MobileHCI '06, 9-. doi: 10.1145/1152215.1152218
- MATRIXXX_ (2018). *Retro_Explosion:05.wav*. Available at: https://freesound.org/people/MATRIXXX_/sounds/441497/
- Mike Bithell (2012), *Thomas Was Alon*e, video game, Mike Bithell.
- MobGe Limited (2018), *Oddmar*, video game, MobGe Limited, Turkey.
- Mode 7 Games (2011), *Frozen Synapse*, video game, Mode 7 Games, United Kingdom
- Nak-Studio (2018). *Dropping Box*. Available at: https://assetstore.unity.com/packages/templates/packs/dropping-box-133040
- Nekki (2016), *Vector 2*, video game, Nekki, Cyprus.
- Niantic (2016), *Pokemon Go!*, video game, Niantic, United States.
- Niantic (2019), *Harry Potter: Wizards Unite*, video game, Niantic, United States.
- Nielsen, J., & Adelson, B. (1994). Enhancing the explanatory power of usability heuristics. Proceedings of the SIGCHI conference on Human factors in computing systems celebrating interdependence - CHI '94, 152-158. doi: 10.1145/191666.191729
- Netmarble (2018), *KOF: Allstar,* video game, Netmarble, South Korea.
- Parallel Circles (2016), *Flat Heroes*, video game, Parallel Circles, United Kingdom.
- Park, Y., & Han, S. (2010). Touch key design for one-handed thumb interaction with a mobile phone: Effects of touch key size and touch key location. *International Journal Of Industrial Ergonomics*, *40*(1), 68-76. doi: 10.1016/j.ergon.2009.08.002
- Peerdeman, P. (2019). Sound a Music in Games, VU Amsterdam, pp 1-18.
- PCG Lectures (2015) PCG2012 - Gordon Calleja - In-Game: From Immersion to Incorporation Available at: https://youtu.be/gs95e-L26SI?t=1s [Accessed: 15th April 2020].
- Playdead (2010), *Limbo*, video game, Playdead, Denmark.
- Poneti (2018). *Clean Vector Icons*. Available at: https://assetstore.unity.com/packages/2d/gui/icons/clean-vector-icons-132084
- Ravenous Games (2016), *League of Evil*, video game, Ratalaika Games, Canada.
- Reyno, E., & Cubel, J. (2013). Model-Driven Game Development: 2D Platform Game Prototyping. Valencia, Spain.
- Rovio Entertainment (2009), *Angry Birds*, video game, Rovio Entertainment, Finland.

- Savari, M., Bin Ayub, M., Bin Abdul Wahab, A., & Mohd. Noor, N. (2016). Natural Interaction of Game-based Learning for Elasticity. *Malaysian Journal Of Computer Science*, *29*(4), 314-327. doi: 10.22452/mjcs.vol29no4.5

- Schønau-Fog, H. (2011). The Player Engagement Process - An Exploration of Continuation Desire in Digital Games. Aalborg University.

- Schønau Fog, H., & Bjørner, T. (2012). “Sure, I Would Like to Continue”. Bulletin of Science, Technology & Society, 32(5), 405-412. doi: 10.1177/0270467612469068

- Schoenau-Fog, H. (2014). At the Core of Player Experience: Continuation Desire in Digital Games. Handbook of Digital Games, 388-410. doi: 10.1002/9781118796443.ch14

- Shutterstock. n.d. *Cool Neon Glitch You Win Stock-Video (100 % Royaltyfri) 1015498576 | Shutterstock*. [online] Available at: <https://www.shutterstock.com/da/video/clip-1015498576-cool-neon-glitch-you-win-text-animation?fbclid=IwAR26-JcCXefKiXsI1D5GvhJYqXOUna-kd2up0n0rFSWmP2kuAxaGSRfqxCI> [Accessed 9 July 2020].

- Silver Lining Studio (2020, *Within*, video game, Silver Lining Studio.

- Smith, G., Cha, M., & Whitehead, J. (2008). A Framework of Analysis of 2D Platformer Levels. Proceedings of the 2008 ACM SIGGRAPH symposium on video games.

- Sporka, A., Kurniawan, S., Mahmud, M., & Slavík, P. (2006). Non-Speech Input and Speech Recognition for Real-time Control of Computer Games. Portland, Oregon, USA, pp. 213-220.

- StatCounter Global Stats (2020). Desktop vs Mobile vs Tablet vs Console Market Share Worldwide. Retrieved 14 August 2020, from https://gs.statcounter.com/platform-market-share#monthly-200901-202007

- Taylor, P. (2011), *Frozen Synapse Soundtrack #08 - Triumph*. Available at: https://www.youtube.com/watch?v=HJKvoIvv6SE

- Touch Foo (2012), *Swordigo*, video game, Touch Foo, Finland.

- Unity (2020). Unity Dokumentation: Touch Phase. Available at:https://docs.unity3d.com/ScriptReference/TouchPhase.html [Accessed: 20th, March, 2020]

- van den Haak, M., De Jong, M., & Jan Schellens, P. (2003). Retrospective vs. concurrent think-aloud protocols: Testing the usability of an online library catalogue. Behaviour & Information Technology , 22(5), 339-351.

- ZvinbergsA (2015). *Radio.wav*. Available at: https://freesound.org/people/ZvinbergsA/sounds/273691/