Lasse Takkunen

# MOBILE GAME FOR KAAMOS GAMES

# MOBILE GAME FOR KAAMOS GAMES

Lasse Takkunen
Bachelor's Thesis
Autumn 2016
Degree Programme in Business
Information Technology
Oulu University of Applied Sciences

# ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Business Information Technology

Author(s): Lasse Takkunen
Title of Bachelor´s thesis: Mobile game for Kaamos Games
Supervisor(s): Liisa Auer
Term and year of completion: Autumn 2016 Number of pages: 33 + 2

This thesis was conducted for a startup game company called Kaamos Games Oy. The company specializes in the development of mobile games, and was looking to expand their expertise and list of developed games. The aim of this thesis was to develop a mobile game for Kaamos Games.

The thesis introduces concepts which are relevant to mobile game development and the game development process. The general stages of a game development process are explained and the activities that went into this development process are documented. Additionally, the thesis showcases the latest version of the game.

The result of this thesis is a free-to-play action mobile game for Android, developed using the Unity game engine. The development process involved working in a group environment, which entailed daily collaboration and weekly meetings. The developed game satisfied all the requirements set by the client. The product is well made and performs competently even on low-end devices.

Keywords: mobile games, game development, game optimization, Unity

## TIIVISTELMÄ

Oulun ammattikorkeakoulu
Degree Programme in Business Information Technology

Tekijä(t): Lasse Takkunen
Opinnäytetyön nimi: Mobile game for Kaamos Games
Työn ohjaaja: Liisa Auer
Työn valmistumislukukausi ja -vuosi: Syksy 2016 Sivumäärä: 33 + 2

Tämä opinnäytetyö tehtiin Kaamos Games Oy-nimiselle startup-peliyritykselle. Yritys erikoistuu mobiilipelien kehittämiseen, ja halusi laajentaa osaamistaan ja kehitettyjen pelien kokoelmaansa. Opinnäytetyön tavoitteena oli kehittää mobiilipeli Kaamos Games Oy:lle.

Opinnäytetyö esittelee mobiilipelin kehittämisen kannalta olennaisia käsitteitä, sekä pelikehityksen vaiheet. Yleiset pelikehityksen vaiheet selitetään ja tämän kehitysprosessin eri toimet dokumentoidaan. Näiden asioiden lisäksi opinnäytetyö esittelee pelin viimeisimmän version.

Opinnäytetyön tuloksena on ilmainen mobiilipeli Android-käyttöjärjestelmälle, joka on kehitetty Unity-pelimoottorilla. Kehitysprosessiin kuului ryhmätyöskentely, joka sisälsi päivittäistä yhteistyötä ja viikottaisia tapaamisia. Kehitystyön tuloksena syntynyt peli tyydytti yrityksen asettamat vaatimukset. Tuote on hyvin tehty ja suoriutuu pätevästi myös halvoilla laitteilla.

Asiasanat: mobiilipelit, pelikehitys, pelin optimointi, Unity

# CONTENTS

# 1 INTRODUCTION

Mobile games are video games which are specifically designed to be played on mobile devices, such as smartphones, smartwatches, handheld PC's or tablet computers. Today, mobile games are produced for a wide variety of platforms, with the most significant portion being developed for tablet computers or smartphones. A great deal of mobile games are free-to-play with in-app purchases, and they are easily available in Google Play Store or Apple App Store. Consequently, in 2013, 78% out of 1.2 billion gamers in the world were using a mobile device to play games, and the number of players keeps growing (Sommer 2013, cited 26.5.2015). The high demand for mobile games has been met by game development companies who are aspiring to create successful mobile games.

This thesis describes the development process of a mobile game for a startup game development company Kaamos Games Oy. The goal is to develop a fully functioning mobile game which Kaamos Games can sell, license or distribute. The requirements Kaamos Games had for this development were that the game is to be developed for Android smartphones and tablets, and that 3D techniques were to be used. It was also requested that the game had an orthographic camera setup. The final version of the game is expected to perform well on low-end mobile devices. Development tasks include prototyping, programming, design, and use of the game engine Unity. The tools used are MonoDevelop development environment, Git Extensions for version control, and the game engine Unity. All the programming will be done using C# (C Sharp) programming language.

The thesis will start by introducing basic concepts used in this mobile game development. It is followed by a chapter for the game development process, which is divided into two main stages: pre-production and production. The pre-production stage holds the initial planning and design phase for the game, as well as an early prototype which is used to test ideas and algorithms. The production phase is the primary stage of development, where all the assets are created, programming and game engine work is done, design changes are made, the game is tested and made ready for later stages of production. In addition to the normal steps in the production phase, an optimization section is added, due to it being a key requirement in this development.

The fourth chapter is about the final product. Its purpose is to exhibit the current version of the game at the time of completion for this thesis. The chapter emphasizes mobile playability, the user-interface, and certain aspects of the game, such as gameplay, characters and objects.

The third stage of game development, post-production, is not included as it extends beyond the schedule of the thesis process. The development of graphical assets has also been omitted, as it is handled by Kaamos Games. The audio side of the game has not been developed yet, and therefore not included.

# 2 MOBILE GAME

This chapter explains what a mobile game is. The sections in this chapter explain concepts which are relevant to game development, and specifically for a mobile game. The concepts in this chapter are ones which fit the development process for this thesis. In addition to describing these concepts in general, thoughts are given on how each concept can affect a development, and how they affected this one.

A mobile game is a video game which can be played on a mobile device, especially on smartphones and tablets. They are an easily accessible and fun way to spend time when there is nothing else to do. The popularity of mobile games can explained by the fact that they can be played anywhere, and with almost any modern mobile device. Nowadays the majority of mobile games are available for Android and iOS devices.

## 2.1 Monetization model

There are various ways a game company can earn income, and those are referred to as monetization models. When choosing the right monetization model, a few factors need to be taken into consideration: the platform, where the game is to be published, the strategies of competitors, the expectations of players, and the chosen game genre. The monetization model should already be considered during the planning stage of a game development so it can be integrated optimally into the content of the game. (Määttä & Nuottila 2016, 89.)

It is common nowadays for game companies to give a game away to be downloaded for free in hopes of attracting a large amount of players. The game is designed in a way that it can convince a player to use real money to buy something which can benefit him inside the game. (Määttä & Nuottila 2016, 96.)

In the free-to-play model, the game is free, or at least a significant portion of it. However, the game is designed in a way that without making any in-app purchases, progressing in the game can be slow or sometimes even impossible. At the start, the game hooks the consumer into playing, while gradually introducing “frustration”. The frustration has to be enough to motivate the consumer to buy in-game items to make his gaming experience easier, but not too much so that he quits the game because it is too hard. (Määttä & Nuottila 2016, 98.)

In the premium model, the consumer pays to get a copy of the game. If the consumer wants additional content, the game may offer it in the form of a DLC (Downloadable content), which the user can purchase. In the mobile game market, using the premium model seems to be an option only for well established brands, as free-to-play games have seized almost the entire market. (Määttä & Nuottila 2016, 93.)

The freemium model can be said to combine free and premium. The game can be downloaded for free, but the consumer has to pay for additional content, such as extra equipment, levels or characters. The consumers who decide not to pay for extra content will see advertisements in the game, which ensures some income for the developer from every consumer. The ads are usually removed when the consumer makes a purchase. (Määttä & Nuottila 2016, 97.)

Kaamos Games is not a well-known brand and trying to make income with a mobile game that uses the premium model would not work. Therefore, the choice for this development was the free-to-play model. The current monetization plan is that the consumer plays the game a few times a day, after each play session he is prompted to watch advertisements to free items, which generates ad income.

## 2.2 Game genre

Games can be divided into categories called game genres. Each of these categories have different challenges and style of gameplay. Games that offer different types of experiences also appeal to different people (Määttä & Nuottila 2016, 22).

An estimate of monthly active users per mobile game genre are presented on the next page in Figure 1. The game genres with the most monthly active users in the United States in July 2016 were Arcade, Adventure, Puzzle, and Action. It is important to note that a game can belong to many game genres.

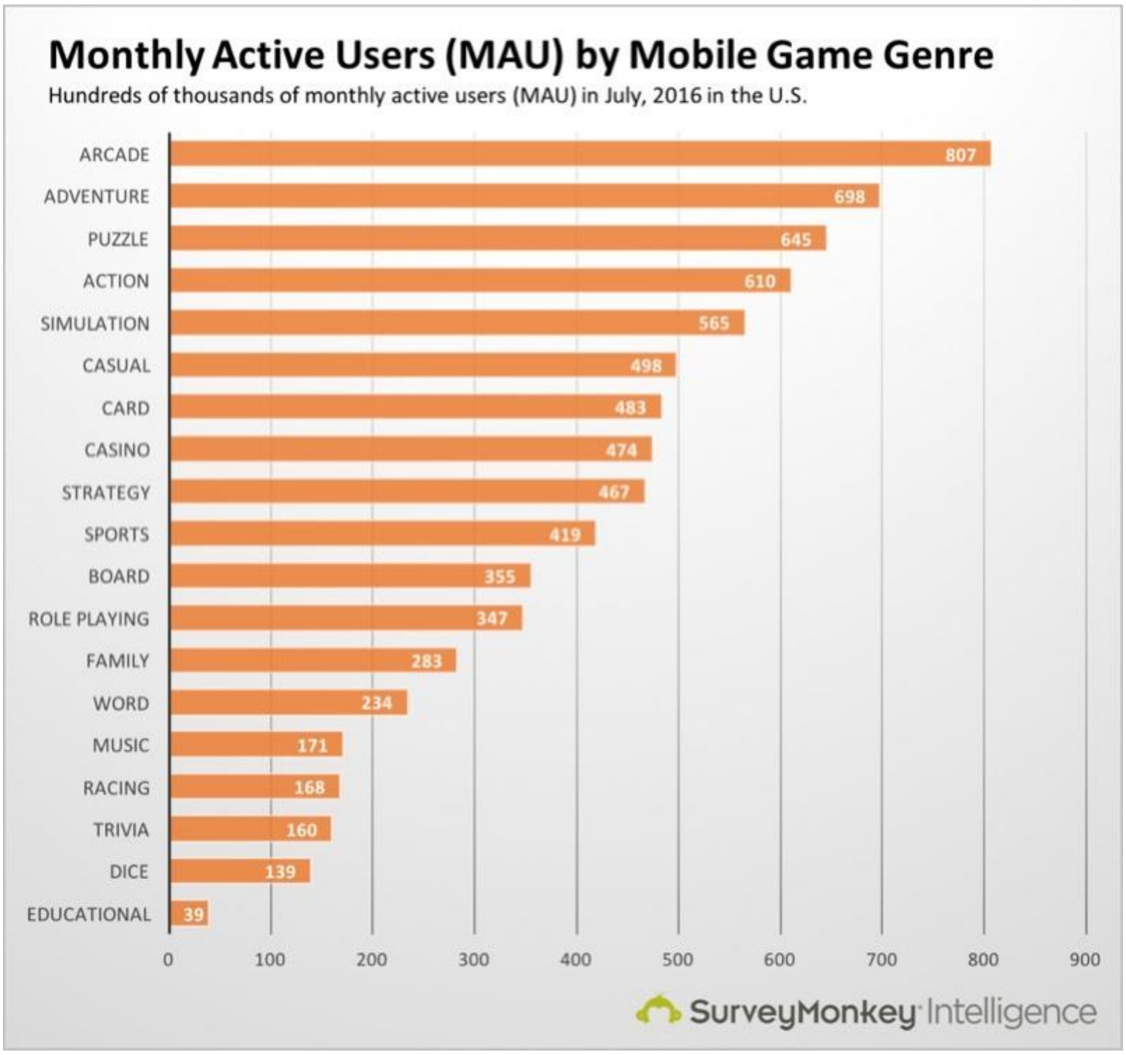


*FIGURE 1. Monthly Active Users by Mobile Game Genre in the United States in July 2016 (SurveyMonkey 2016, cited 7.11.2016)*

Action is the main game genre chosen for this development. An action game is a fast-paced game where the player has to use their hand-eye coordination to get through levels or to make progress in the game. The appeal of action games is that they are challenging. As games may belong to many genres, the game can also be considered an Arcade game due to its simple controls and short levels that become more difficult as the game progresses.

## 2.3 Fun

Games create emotion. People want to feel things while playing a game, and the emotions experienced by the player affects how much the game is played. The experience can be Hard Fun or Easy Fun, it can be a relaxing experience which helps you avoid boredom, and it can also be a feeling of socializing (Lazzaro 2004, cited 9.11.2016). A game can contain many types of fun, and the types of experience your game offers determines who will play it.

Hard Fun is a type of fun for players who enjoy challenges and overcoming obstacles. They like a game that requires you to focus on being good, and rewards you with a feeling of triumph when you succeed, and frustration when you fail. They want to beat the game or other players to see how good they really are. Easy Fun is for players who like to experience the activity of playing the game. To them, it is all about immersion, gaming is an adventure. They enjoy deep storylines, fine details, and are generally curious to find more things about the game. Some players may seek something different from their gaming experience other than just fun. A player such as this may play to clear his head, to feel better about himself, or simply avoid boredom. These players experience a change in their emotional state while playing. They can feel relaxed, excited and relieved. There are also people who only enjoy games which offer a way to socialize. The feeling of playing with or against other players is what makes the game fun for them. (Lazzaro 2004, cited 9.11.2016.)

The fun in this game is mainly Hard Fun, but it offers other types of experiences as well. The game is challenging, and the player enjoys honing and using his skills to reach new heights. While the game is challenging, it can still be played casually to avoid boredom. The social aspect, the feeling of playing with others comes from beating a friend’s high score and sharing your high score with them.

## 2.4 Game loops

In many mobile free-to-play games, the length of a play session is usually quite short, but repeated often. In such a case, the gaming experience is thought of as a loop, lasting only a few minutes, but something that the player wants to continue repeating. The player is driven to do actions in order to get rewarded. This is what can be considered a game loop. The objective of a core loop design is to give the player a quick and fun experience. However, the longer the player spends playing the game, the harder it becomes to complete the loop. The player can either grind, in other words play more, or use real money to advance faster in the game. (Both Guns Blazing 2013, cited 23.6.2016.)

Games may contain more than one loop. Common activity loops in a gamified system are the engagement loop and the progress loop. The primary goal of an engagement loop is to motivate a player to do an action, which gives the player feedback and rewards, which in turn motivates the player to do it again. A progression loop is a broader activity loop which is intended to make it easier for players to progress in a game. The progression loop divides the goal of the game into small steps which the player feels more comfortable doing. (Werbach 2016, cited 9.11.2016.)

The game being developed mainly uses a core loop design, offering a challenging and quick play session which gives a sense of excitement. The game contains an engagement loop to some extent. In return for playing, the game gives in-game currency and free gifts in exchange for watching ads. The consumer is motivated to play more to get a better score and to get rewarded each time he plays. Once the game contains more playable characters, a progression loop may be implemented.

## 2.5 Target audience

A target audience is a group of people who you want a game to appeal to. The target audience may affect the type of content the game can have, the difficulty level and other details. How well the game company knows its target audience’s expectations determines how well it can match them (Määttä & Nuottila 2016, 22).

The game is initially targeted for mobile gamers, especially the western market. To be more specific, the game is targeted for people who enjoy challenging free-to-play games, and can spare a few minutes of their time each day, whenever they like, to play a few game sessions.

# 3 STAGES OF DEVELOPMENT

This chapter describes the different stages of a video game development process. At first, each relevant concept is described in general. The description of each concept is followed by a report of what it embodied in this game development process.

Different game companies have different philosophies on how a development process works. Generally a development process consists of three main stages, pre-production, production and post-production. (Wikipedia 2016, cited 4.6.2016.) This particular development process does not include the post-production phase. Below in Figure 2 you see a concept map of this development process.

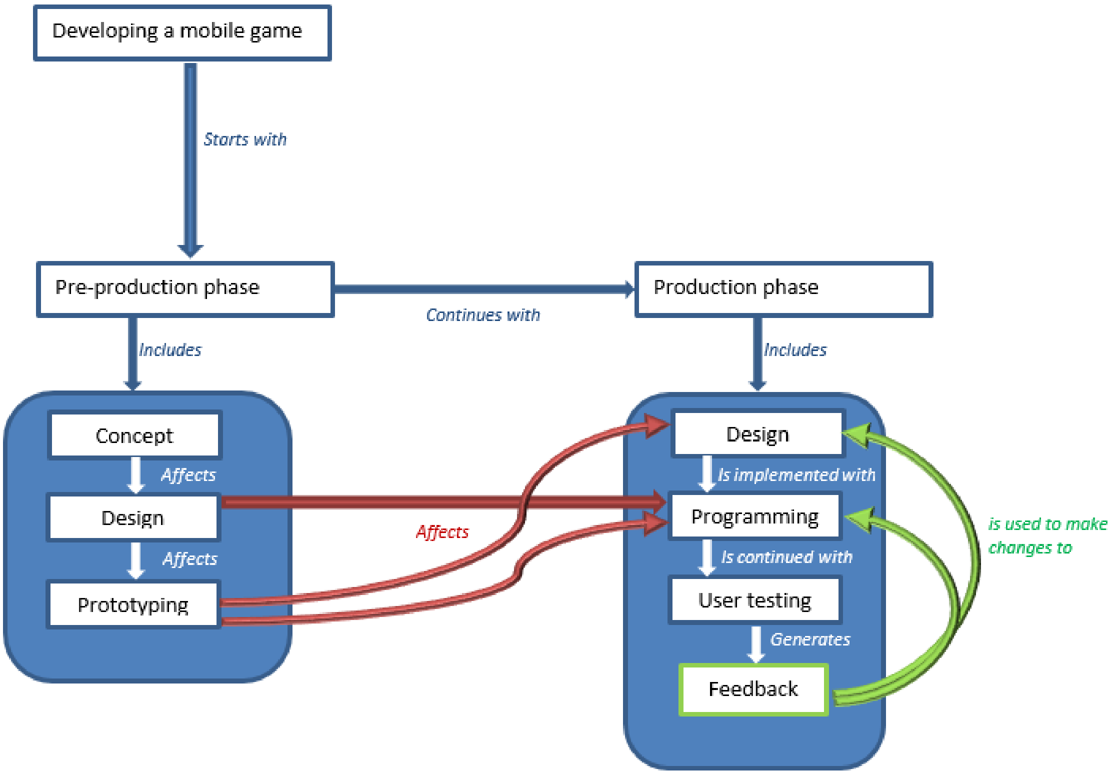


*FIGURE 2. Game development stages for this thesis*

## 3.1 Pre-production stage

In the pre-production stage the development team works on creating a plan for the game, and the game goals, level design and gameplay mechanics are put together in a design document (Edwards 2006, cited 27.9.2016). The pre-production stage can be said to be the stage where the initial idea is created and molded into something that resembles a game. This stage is mostly planning, but it may also include prototyping.

Our development team consisted of three members: the project leader, the graphical designer, and me, the programmer. Each of us had a specific role and task. The project leader was the one who came up with the idea for the game and had the vision for it. The graphical designer was there to create the visual outlook, while my role was producing the technical side. Since our development team was small, and the timeframe for this development was short, it was decided that all the members of the development team would participate in designing the game.

### 3.1.1 Concept

The concept is a plan for the development, and it includes the most important elements of the game, such as description of gameplay, features, the target audience and the game genre. (Wikipedia 2016, cited 4.6.2016.)

The first idea for our game was introduced to us by our project leader. The idea was based on several video games: a common video game concept Snake, the PlayStation game Crash Bash and a popular mobile game Crossy Road. The development team held a meeting where the first concept for the game was created.

For the game concept, the freemium model was chosen as the monetization model. The game was defined as a single-player action game. The playing field of the game would be an 7x7 grid platform, so the platform would be divided into 49 different areas. Some gameplay elements were influenced by other games, most notably the game character leaves a trail which can be used to create a “lasso”. When the lasso is completed, all the areas inside the encircled area is painted into a different color. Everyone agreed that for the visual side of the game, a style similar to Crossy Road would be great. It was decided that the game character’s speed and trail length would depend on the items the player picks up. These attributes would affect the potential size of the lassoed area. The bigger the area painted in one combo, the more points the player receives.

### 3.1.2 Game design document

In the design part of the pre-production phase, the first version of a game design document is created. The game design document describes the concept of the game and how the gameplay works. The document may include sketches and prototypes of different sections of the game. The document becomes a living document from here on out, as it will constantly change throughout the development process. (Wikipedia 2016, cited 4.6.2016.)

After the initial game concept was created, the development team held a few meetings to discuss the game design. The concepts of the chosen monetization model, target audience and game genre were explored and the findings, if applicable, were incorporated into the design. A few free-to-play elements such as unlockable skins and game currency were added promptly so they would not have to be incorporated on a later date. The logic of the gameplay was scrutinized to remove any errors. After all the important issues were discussed, an early game design document was produced.

It was hard to differentiate who did what in the design meetings, as we did all the decisions together. However, it could be said that my part in the design phase was to ensure the game logic was solid and simple enough that it could be carried out efficiently in a relatively small time frame. Additionally, I ensured that we would avoid any heavy computing by making all the technical side decisions that needed to be added to the game design.

A sketch produced during the design process can be seen below in Figure 3. It shows much of the initial plan for the game, with at least key elements of the game being mentioned. At the time of the drawing, some issues have not been decided yet, such as the question of whether the game is 2D or 3D. What we can see in the sketch is the gameplay idea of lassoing cubes, the progression of the game through the layers of cubes, the game controls, and a list of all the proposed items for the game.

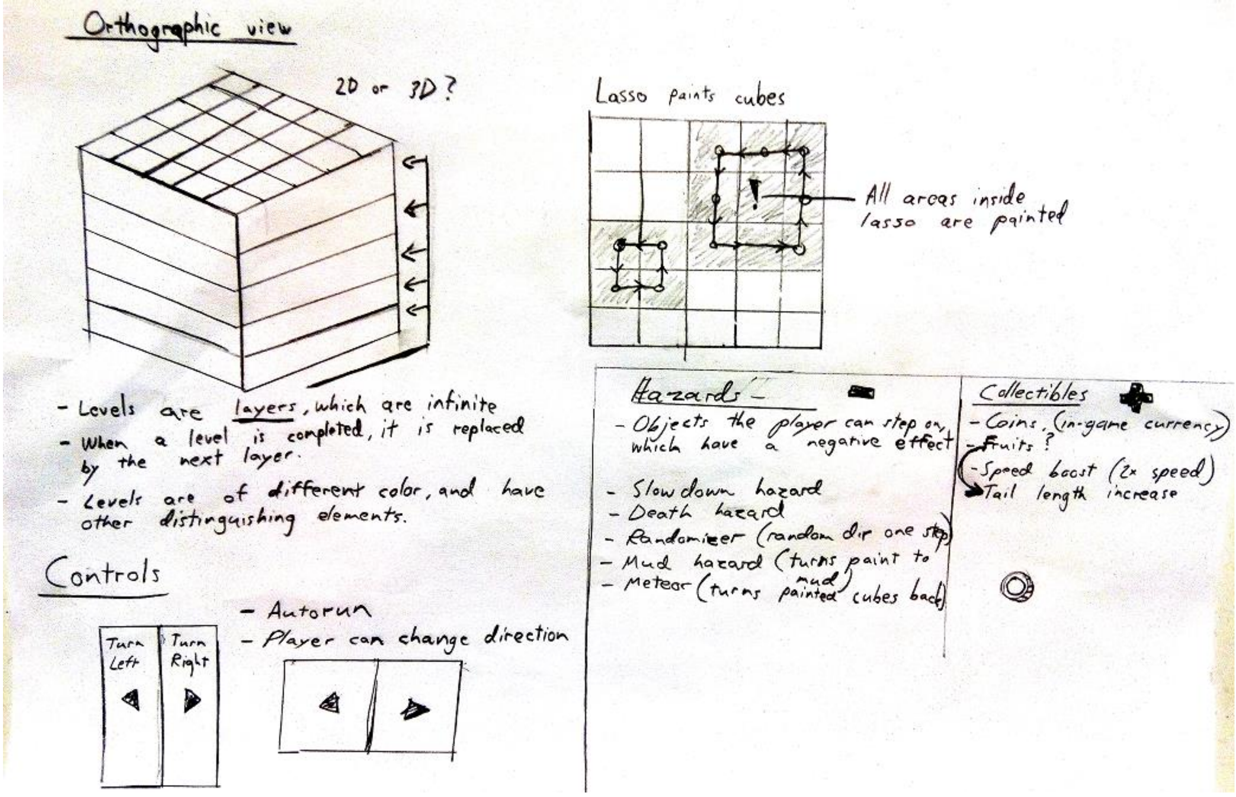

Orthographic view
2D or 3D?
Lasso paints cubes
All areas inside lasso are painted
- Levels are layers, which are infinite
- When a level is completed, it is replaced by the next layer.
- Levels are of different color, and have other distinguishing elements.
Controls
- Autorun
- Player can change direction
Turn Left
Turn Right
Hazards
- Objects the player can step on, which have a negative effect
- Slow down hazard
- Death hazard
- Randomizer (random dir one step)
- Mud hazard (turns paint to mud)
- Meteor (turns painted cubes back)
Collectibles
- Coins, (in-game currency)
- Fruits ?
- Speed boost (2x speed)
- Tail length increase

*FIGURE 3. Early design sketch*

### 3.1.3 Prototyping

Prototyping is the act of making a model or a sample of your game and exploring the thought process of it. A prototype may even be created from physical materials such as paper and can prove useful when trying to explain the game to someone else (Clark 2014, Chapter 3). Game companies tend to make prototypes using a game engine. Prototyping is useful for visualizing the game and correcting design flaws. It can also be used to produce initial code for the game.

In our development, the game prototype was created using Unity. The prototype served as a quick way to test new ideas and playability. The prototype was shown the potential product to investors in various events, such as Game Spring Oulu. A screen capture from the prototype version of the game can be seen below in Figure 4.

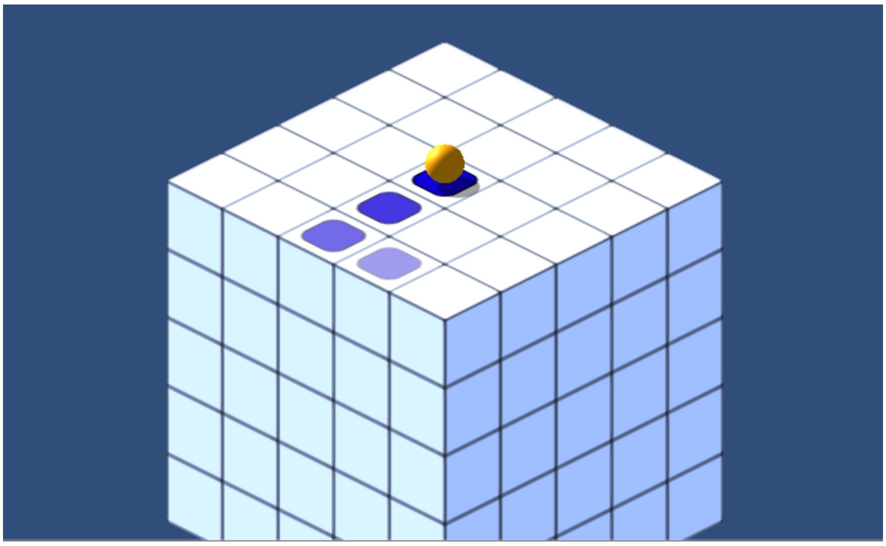


*FIGURE 4. A screen capture from an early prototype version of the game*

My task in the prototyping phase was building the prototype in Unity. This task mostly involved setting up the project in the Unity game engine. The main focus was to insert the most fundamental features of the game, such as the orthographic camera view, character movement, and character interaction with the environment. Few things, such as the camera view were achieved using Unity's own Camera component, but others, such as the character movement and objects had to be scripted. From a programming perspective, the prototyping phase provided a great way to build a foundation for the game code for the production stage. In addition to just simply implementing and coding assets, many hours were spent testing the game logic, how playable the game was, and whether it offered the experiences we wanted a player to have while playing. At this stage, it was deemed that the game could become more challenging as the game progresses. That way, players who seek for hard fun can enjoy the game more.

## 3.2 Production stage

The production stage is where the main product is produced, and it commences after the pre-production phase is completed and a plan for the game is ready. The development team works together to build the game according to the design while limited by a set schedule. The designer's job is to make sure the details of the game are properly implemented by programmers and artists. If a hole in the design is found, whether it is a technical limitation or an omission by the designer, the whole team works to find a solution for the issue. (Edwards 2006, cited 27.9.2016.)

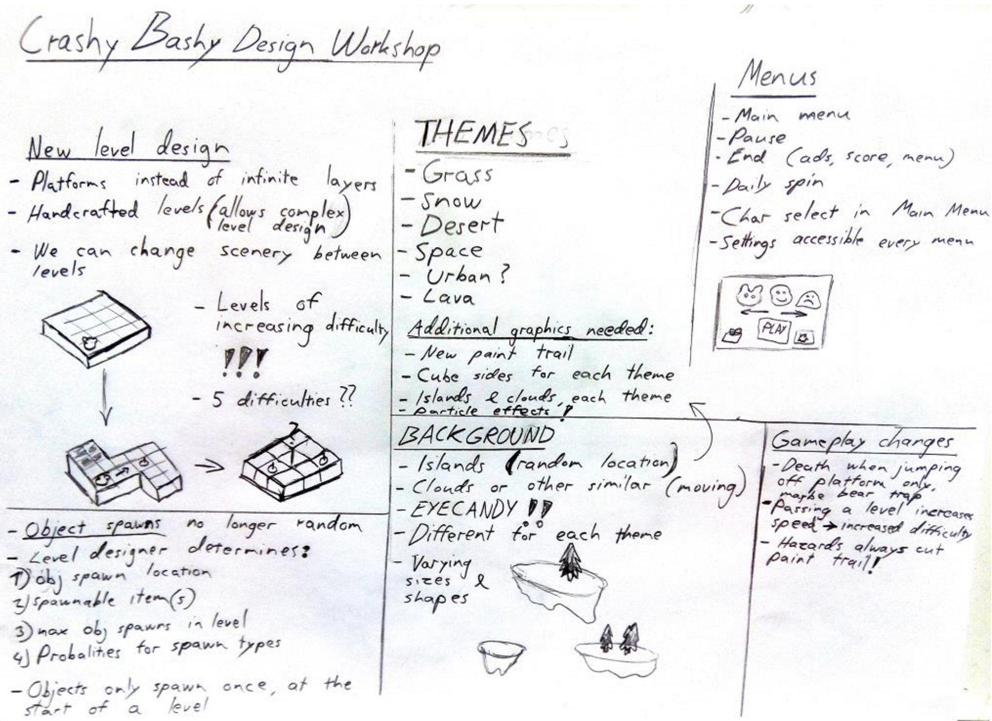
Our production stage commenced after we realized that our game prototype was functioning well and that it could made into a real game. We organized a few meetings to discuss the design and the changes we needed to make. After the majority of changes to the game were written, we started making the product from what we had with the Unity prototype. The prototype had a lot of reusable assets, and it was possible to build on it to create the final product. Aside from a few placeholder graphics, the main graphical assets were produced by our graphical artist.

Overall my part in this phase was the same as in the pre-production phase. I was included in the design meetings along with the project leader and graphical artist, where we planned and worked out what was needed to make this game better and more attractive to players for a longer period of time. Throughout the production phase I kept check of the integrity of the game, and made sure that the game worked flawlessly, logic-wise and free of programming errors. I filled my main role of programming, handled all Unity related work, implemented graphics provided by our graphical artist, and scripted all the objects in the game. In the later stages of production, I patched bugs and added any last changes brought on by feedback or design changes.

### 3.2.1 Design

Game design embodies the process of designing the content and rules of a game. When a development is ongoing, the design is implemented and also modified to ensure it correctly demonstrates the current vision for the game (Wikipedia 2016, cited 27.9.2016). The need to do alterations to the design may come from information discovered in testing sessions, design meetings or they can even come from requirements set by potential investors.

Everyone in the development team felt that the game was too simple as it was and needed something to keep players interested for a longer period of time. Therefore, a major design overhaul was conducted. We held weekly meetings to discuss any issues we found. The known design flaws were corrected, new features were implemented and the design was refined to be as flawless as it can be. A summary from a design meeting in Figure 5.



Crashy Bashy Design Workshop

New level design

- Platforms instead of infinite layers
- Handcrafted levels (allows complex level design)
- We can change scenery between levels
- Levels of increasing difficulty !!!
- 5 difficulties ??

- Object spawns no longer random
- Level designer determines:
1) obj spawn location
2) spawnable item(s)
3) max obj spawns in level
4) Probalities for spawn types
- Objects only spawn once, at the start of a level

THEMES

- Grass
- Snow
- Desert
- Space
- Urban?
- Lava

Additional graphics needed:

- New paint trail
- Cube sides for each theme
- Islands & clouds, each theme
- Particle effects?

BACKGROUND

- Islands (random location)
- Clouds or other similar (moving)
- EYECANDY !!
- Different for each theme
- Varying sizes & shapes

Menus

- Main menu
- Pause
- End (ads, score, menu)
- Daily spin
- Char select in Main Menu
- Settings accessible every menu

Gameplay changes

- Death when jumping off platform only, maybe bear trap
- Passing a level increases speed → increased difficulty
- Hazards always cut paint trail!

*FIGURE 5. A summary from a design meeting*

### Gameplay changes

Before design changes, the game character could only die if he/she stepped off the level platform. After the changes, death could occur in two ways; stepping outside the level or touching a hazard which causes death. Bridges were added so there could be something other than cubes linking different areas of the game.

### Level design changes

Instead of having only square-shaped levels, a level could be of any shape as long as it fit inside the game view and each of its areas could be lassoed. As mentioned above in gameplay changes, bridges were added which brought more depth to level design. The object spawn mechanics were also changed so that the objects only spawned once and the probability of a certain object spawning could be decided by the level designer. These changes opened up more possibilities for handcrafted levels, as a level designer could create a level of any shape, and also decide where and which objects would spawn when the level starts. A few basic graphical themes were decided added to the design. The thought behind this was that the game needed to look different in order to feel different.

### User interface design

We wanted our players to be able to play on any resolution and using the screen orientation they prefer. Therefore, UI was designed in a way which supports both portrait mode and landscape mode. When the UI was tested by volunteers, the most common opinion was that the game controls were challenging but fun, which is exactly what we wanted, and therefore the controls remained unchanged. However, the speed of the game was slowed down to help new players adjust to the challenge. Some testers also noted that the menu elements seemed stiff, so menu animations were also added to the design and later implemented to the game.

### 3.2.2 Programming

The programming part of the production phase is where the final source code for the game is created. Programmers work using scripting tools and a game engine to build a game. In the production stage, a programmer's job is to fix bugs and implement any changes to the game according to design requirements (Wikipedia 2016, cited 4.6.2016).

All the programming for this game was done by me. C# programming language was used for scripting. A few programming mistakes were made while the prototype was being developed, and therefore a large portion of the code had to be rewritten. Each class only had their own purpose and methods, and the code was carefully commented in order to make changes and bug fixes a bit easier. Figure 6 is an illustration of the game's main algorithm. It shows a step-by-step operation of events in the game.

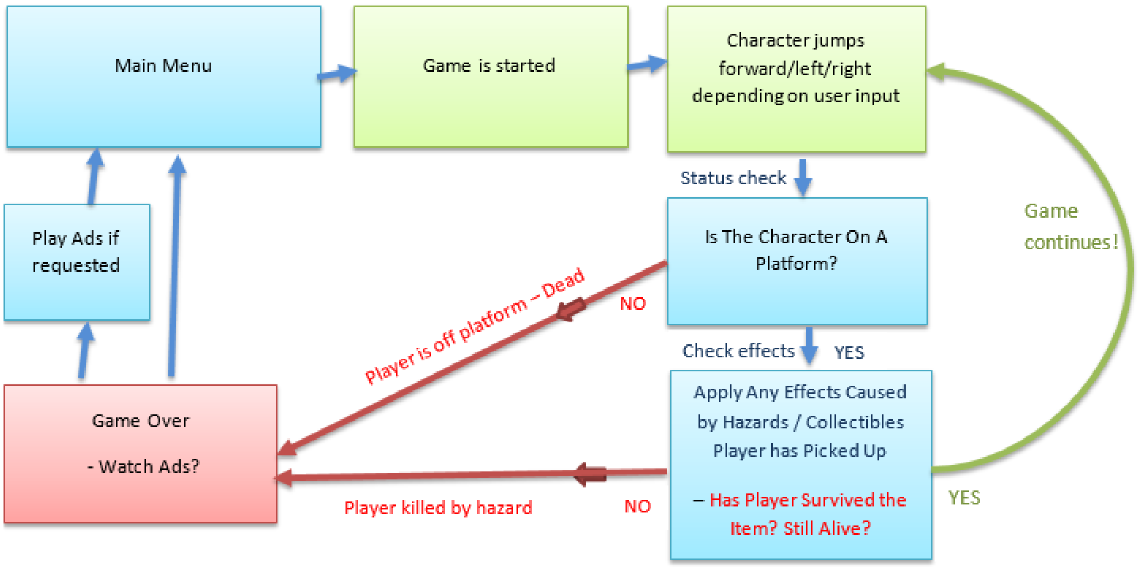

Main Menu
Game is started
Character jumps forward/left/right depending on user input
Status check
Is The Character On A Platform?
NO
Player is off platform – Dead
Check effects
YES
Apply Any Effects Caused by Hazards / Collectibles Player has Picked Up
– Has Player Survived the Item? Still Alive?
YES
Game continues!
NO
Player killed by hazard
Game Over
- Watch Ads?
Play Ads if requested

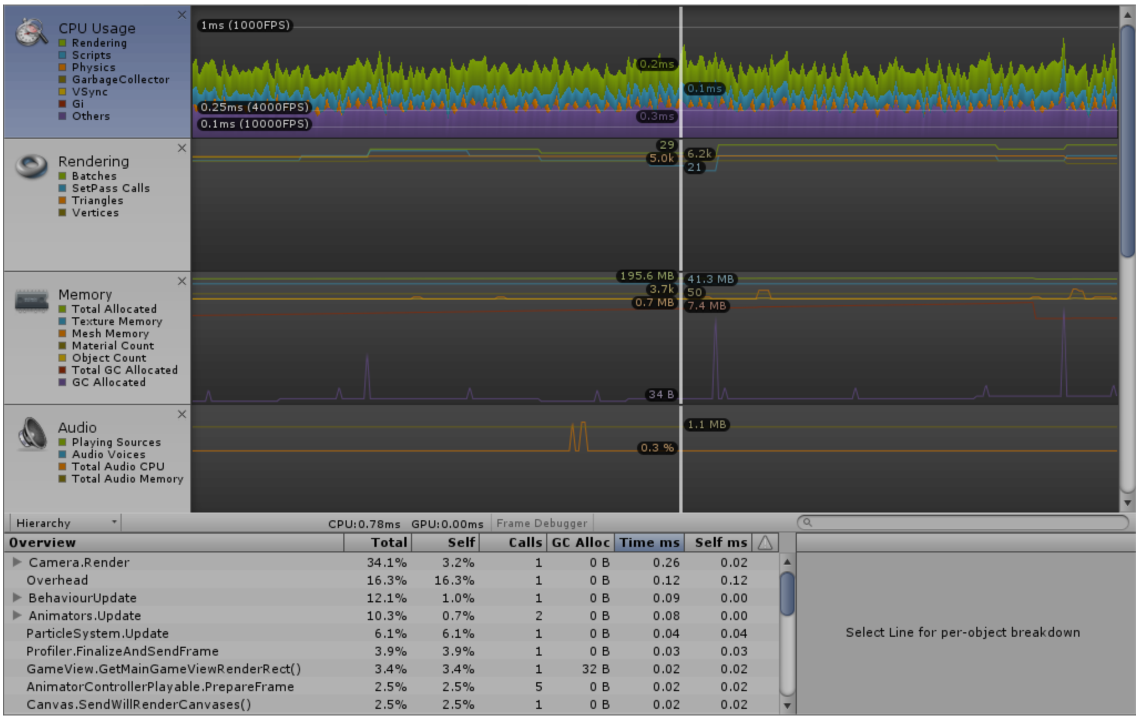
*FIGURE 6. Simplified illustration of the game's main algorithm*

The design changes mentioned in the design section also brought a great deal of changes programming-wise. A lot of the game logic was rewritten in order to incorporate the design changes, and new objects such as bridges had to be implemented. There were some optimization issues as well. The design changes brought a lot of new handcrafted levels, and with them came a significant issue of how to load them without taking up too much memory or CPU. The aforementioned issue is described in more detail in the optimization section.

### 3.2.3 Optimization

When producing any game for a wide variety of devices, optimization is something that should be looked into. Optimization means increasing the performance of a video game by cutting down the resources it uses. The difference between an optimized and an unoptimized game is that if the game is not optimized, customers that own old or low-end devices may not be able to play it.

One of the key requirements for the game was that it is suitable for mobile devices and that it would perform well on low-end devices. Throughout the development process, Unity's profiler tool was used to monitor the performance of the game application, both on PC and in an Android device. The profiler revealed any intense memory, CPU or GPU usage, and consequently these issues were fixed as well as possible. One example of a critical issue we encountered and the solution for that issue is explained later in this section. Figure 7 below shows the profiler tool.



CPU:0.78ms GPU:0.00ms

| Overview | Total | Self | Calls | GC Alloc | Time ms | Self ms |
|---|---|---|---|---|---|---|
| ▶ Camera.Render | 34.1% | 3.2% | 1 | 0 B | 0.26 | 0.02 |
| Overhead | 16.3% | 16.3% | 1 | 0 B | 0.12 | 0.12 |
| ▶ BehaviourUpdate | 12.1% | 1.0% | 1 | 0 B | 0.09 | 0.00 |
| ▶ Animators.Update | 10.3% | 0.7% | 2 | 0 B | 0.08 | 0.00 |
| ParticleSystem.Update | 6.1% | 6.1% | 1 | 0 B | 0.04 | 0.04 |
| Profiler.FinalizeAndSendFrame | 3.9% | 3.9% | 1 | 0 B | 0.03 | 0.03 |
| GameView.GetMainGameViewRenderRect() | 3.4% | 3.4% | 1 | 32 B | 0.02 | 0.02 |
| AnimatorControllerPlayable.PrepareFrame | 2.5% | 2.5% | 5 | 0 B | 0.02 | 0.02 |
| Canvas.SendWillRenderCanvases() | 2.5% | 2.5% | 1 | 0 B | 0.02 | 0.02 |

Select Line for per-object breakdown

*FIGURE 7. Unity Profiler*

### Example issue

The game is planned to have hundreds levels, but only one level is visible and used at a time. The game only features one scene to achieve continuity, and the initial plan was that one Unity GameObject (see Unity 2016, cited 19.6.2016) would hold all the levels and characters inside as variables for us to instantiate from. This method however, significantly increased the amount of memory the game uses.

### Solution to example issue

The solution was to store the levels in a special Resources folder (Unity 2016, cited 19.6.2016) within the Unity project. The contents of the Resources folder are also packed in the game build. This way the necessary asset, in this case a level, can be accessed, and furthermore only loaded onto memory when needed. When the asset is no longer needed, it is unloaded to free some memory. This solution needed some work too however, as it created spikes in CPU usage while it loaded an entire difficulty category of levels to instantiate randomly from.

We needed a way to count all the levels in order to only load one level from the resources folder. There was no built-in way for Unity to count how many assets a resource folder contained, and therefore the levels were counted manually. The count of levels of from difficulty level were stored in a text file, which could be loaded at runtime. The levels were then named appropriately, so that the level loading script could load one exact level using a difficulty number and a random number that is below the count of levels in that difficulty category. This solution significantly reduced the CPU usage when loading a new level. The same solution is also applied to player models and skins.

### 3.2.4 Testing

A significant part of a game development process is testing. Tests are conducted to find errors, performance issues and any problems with user experience. Although tests are carried out throughout the development process, the production phase is where most play-tests are organized.

Throughout our development process, the game was constantly tested by the development team. After the game was fundamentally ready, we handed our game to be tested to anyone who walked into the office. These tests produced valuable feedback and information about bugs, game-breaking or otherwise. This feedback and error reports were used to make improvements to the game and patch all the bugs that were reported.

The frame rate was observed with the use of a modified version of an FPS (Frames Per Second) script (see Unify Community wiki 2015, cited 25.6.2016). The script is an adequate way to measure performance, as it shows how many frames are displayed each second inside the game application. The  modification added the possibility of showing the average, highest and lowest frame rate during application runtime. The result of 60 frames per second being drawn would be a desired result, and anything lower than 30 would not be acceptable.

The first test device, Samsung Galaxy J1, had minor trouble achieving a stable frame rate in the first minute of playing, but after that it achieved a very steady 60 frames per second, which kept going throughout the play-test. Another test device we used to get a comparison is a very popular mobile device from 2012, the Google Nexus 7. The Nexus 7 had no trouble maintaining a steady 60 frame rate.

Numerous Unity community members have reported that the profiler tool in the Unity editor may be unreliable. We also wanted to observe performance using a way than the frame rate script. Therefore, the game app was monitored using a tool provided by the Unity game engine, the Unity Built-in profiler. The built-in profiler provides more in-depth info of how the game runs. The frame time is a measurement given by the built-in profiler. The built-in profiler reported a frame time of 16.7ms, which is considered a good result for an application that runs at 60 frames per second.

# 4 THE GAME

This chapter exhibits the game as it is at the time of completion for this thesis. The gameplay is first explained briefly, and it is followed descriptions of each section of the user interface. The levels, the characters, and the objects of the game are also given a brief introduction.

## 4.1 Gameplay

The game is a fast-paced action game, in which the challenge is completing levels, getting a good score, and keeping the game character alive. A single mistake can result in the game being over. During a play session, a player affects the direction the character jumps towards. If the game receives no input from the player, the character just jumps forward, and off the level eventually.

The objective of the game is to jump on cubes to leave a trail, which can be used to create a lasso, which gives you points, and paints all the cubes inside the lassoed area (see Figure 8). When the entire platform is painted, a teleporter appears which takes the player to next level. Each time the player completes a level, the score multiplier is increased, but the game becomes faster and the difficulty of the levels increases. When the game ends, the player sees the score he has achieved and has the option to watch advertisements in exchange for a small benefit, such as coins or new characters.

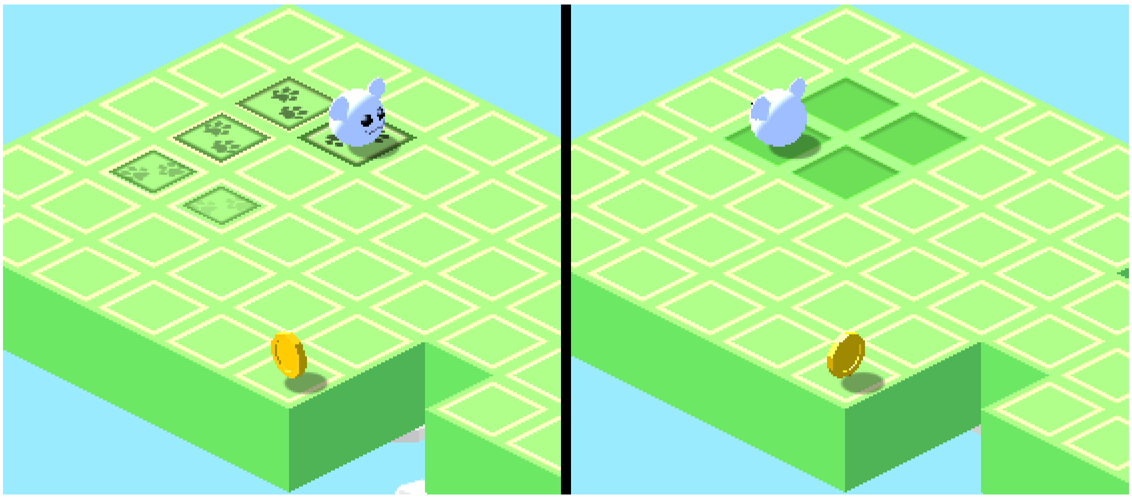


*FIGURE 8. The image on the left shows the trail, the image on the right shows the painted area*

## 4.2 User interface

The user interface is first and foremost designed to support mobile devices. As with any video game that is developed for different types of devices, screen resolution and screen orientation have to be taken into consideration. There are many examples of mobile games which only support either portrait mode or landscape mode. We have made sure that this game supports all screen resolutions and screen orientations, as we want to ensure our players can play anywhere, and using a screen orientation they prefer. This way their gaming experience can be as comfortable as possible. In Figure 10 at the bottom of the page you can see a side-by-side comparisons of landscape and portrait mode views of our game.

### Main menu

The main menu is kept very simple. At the center of the screen, you can choose a character from a list. Directly below the characters is a play button, which you can tap to start the game. The game settings are also featured in the menu. The settings button, which is located on the bottom right part of the screen, opens a bar that holds audio toggle buttons and a help screen. To keep the player interested and visit the game app at least daily, we introduced a concept called daily spin. By using the daily spin, the player can receive rewards, such as skins or game currency. The button for it, which looks like a gift box, is located on the bottom left part of the screen. The entire menu view is visible in Figure 9 below.

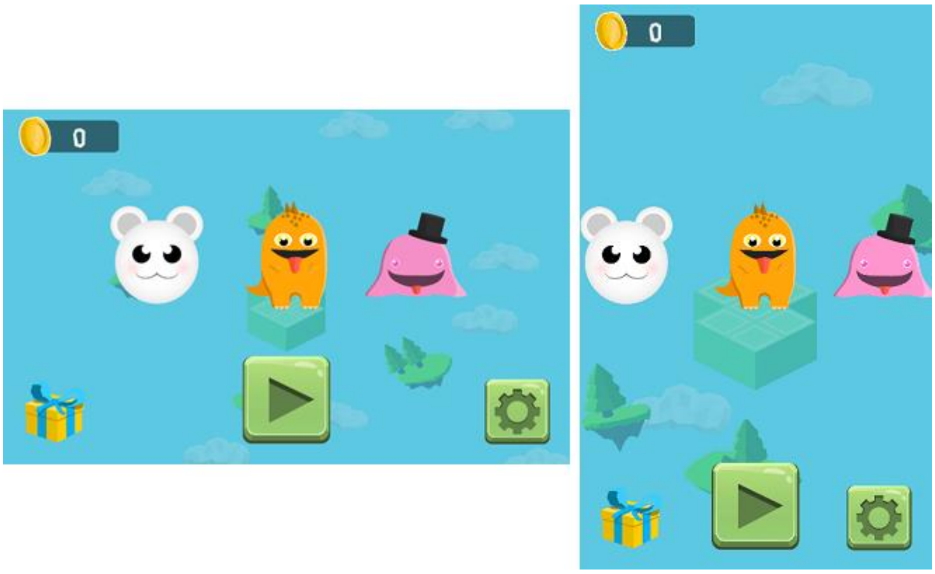


*FIGURE 9. Main menu in Landscape mode (left) and Portrait mode (right)*

### Game view and controls

The game view can be seen below in Figure 10. The controls in the game view are simple, touching the left side of the screen turns the player left and the right side turns the player right. Additionally, on the top right of the screen there is a pause button if you wish to go back to the menu or just simply pause the game. Aside from controls, the game view features some indicators of the player's progression. The top left and middle part of the screen shows how much game currency you have, what your current score is and the score multiplier.

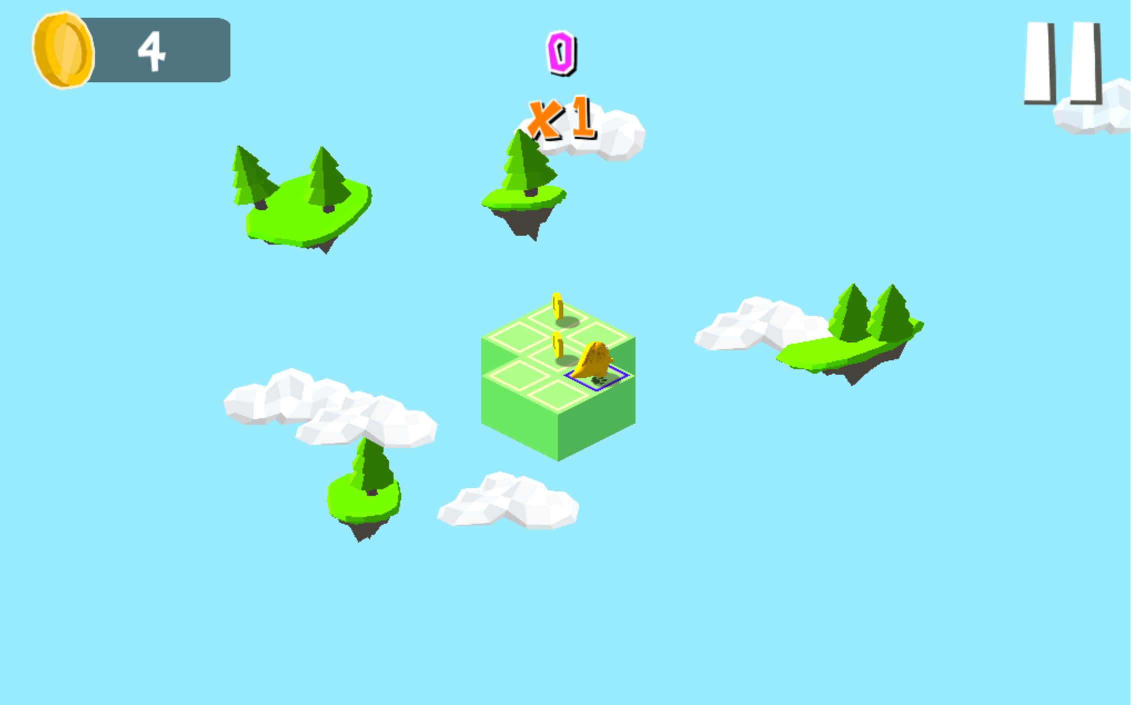


*FIGURE 10. Game view in Landscape mode*

### Pause menu

The pause menu pops up when the player wants to pause their game using the pause button (see Figure 11). When pause menu is active, a text indicating the paused status appears in the middle of the screen. The play and settings buttons can be found in the same locations as in the main menu. While in the pause menu, the player can change audio settings, resume the game by tapping the play button located in the center of the screen or return to the main menu by tapping the home button located on the bottom left of the screen. The pause menu is shown below in Figure 11.

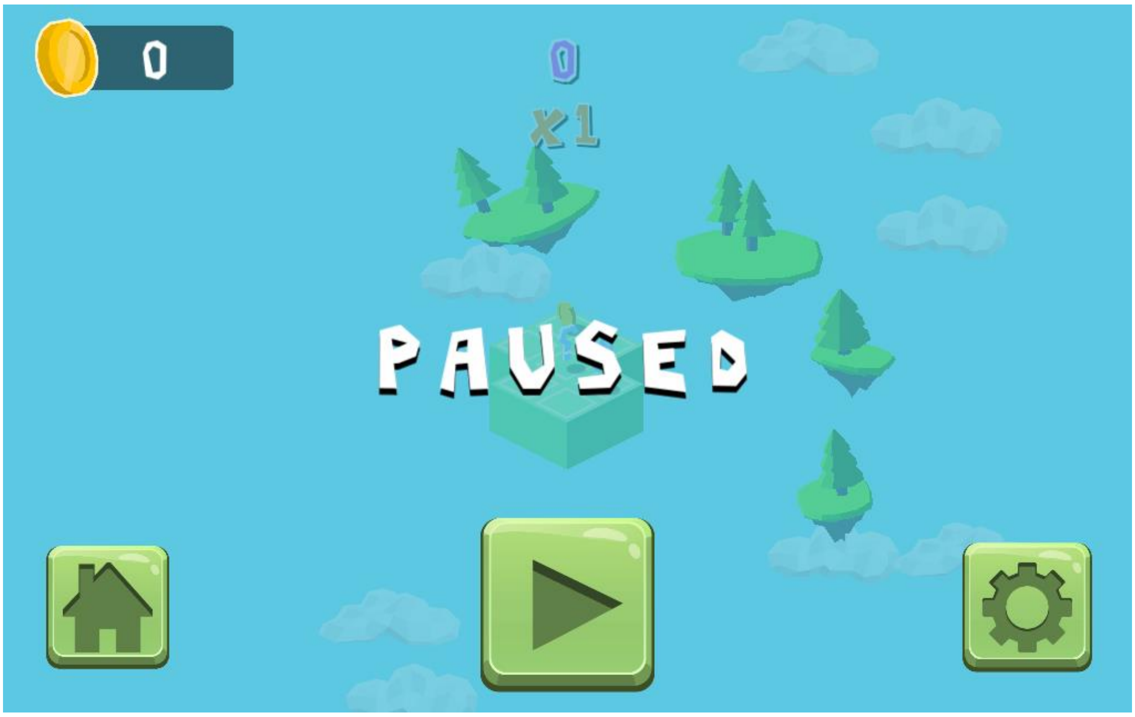


*FIGURE 11. Pause menu view in Landscape mode*

### Game over screen

The game over screen appears when the game character jumps off the platform or steps on a hazard which causes death (see Figure 12). The game over screen shows the score the player has acquired during the play session, and offers game currency and rewards in exchange for watching advertisements.

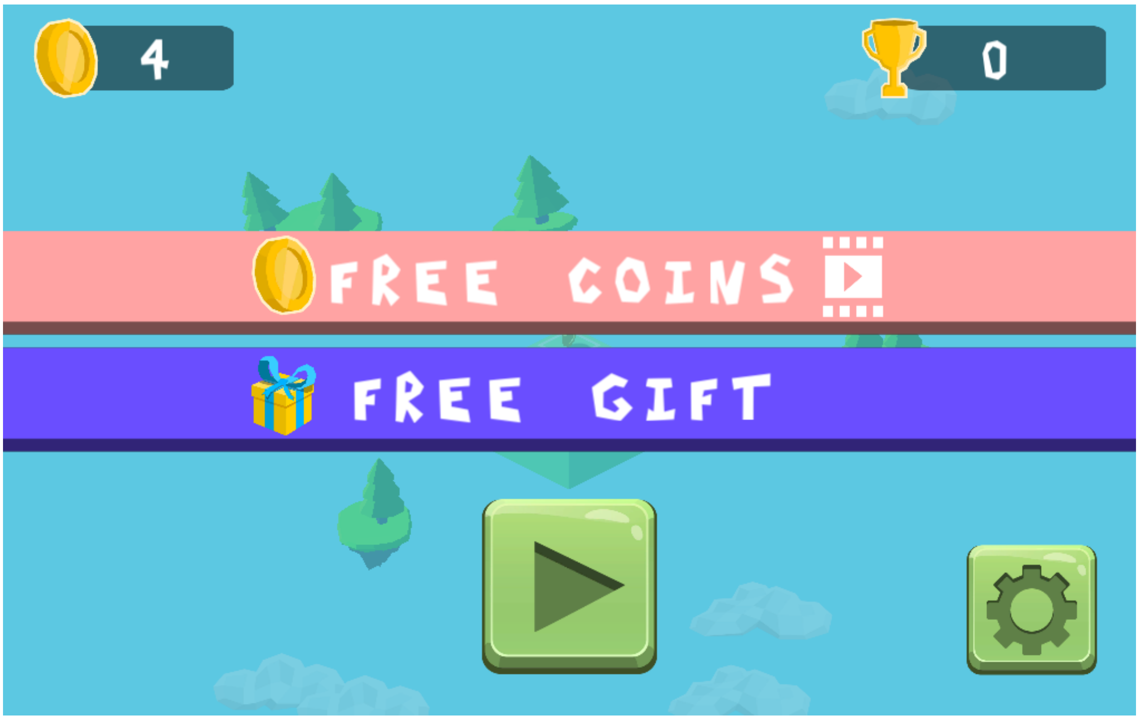


*FIGURE 12. Game over screen in Landscape mode*

### 4.3 Levels and themes

A set of handcrafted levels were created for the game. The levels are divided by a difficulty variable. The player's progress determines the difficulty of a level. By completing two levels in a row, the difficulty mode is increased. Each difficulty mode has numerous handcrafted levels, and one is chosen randomly for the player to play. Examples of each difficulty category can be seen below in Figure 13.

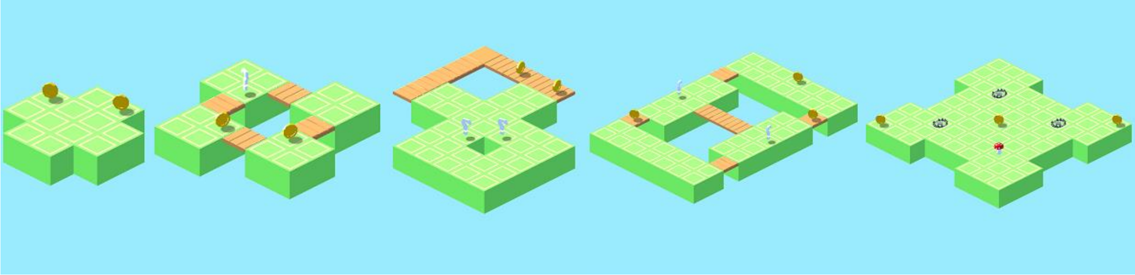


*FIGURE 13. Levels of different difficulty.*

The game also features a set of themes which changes the outlook of the game: the background, the color of the "cubes" and other props, such as the background islands, are swapped between levels to provide a visually pleasing experience for the user. Figure 14 below shows all the themes.

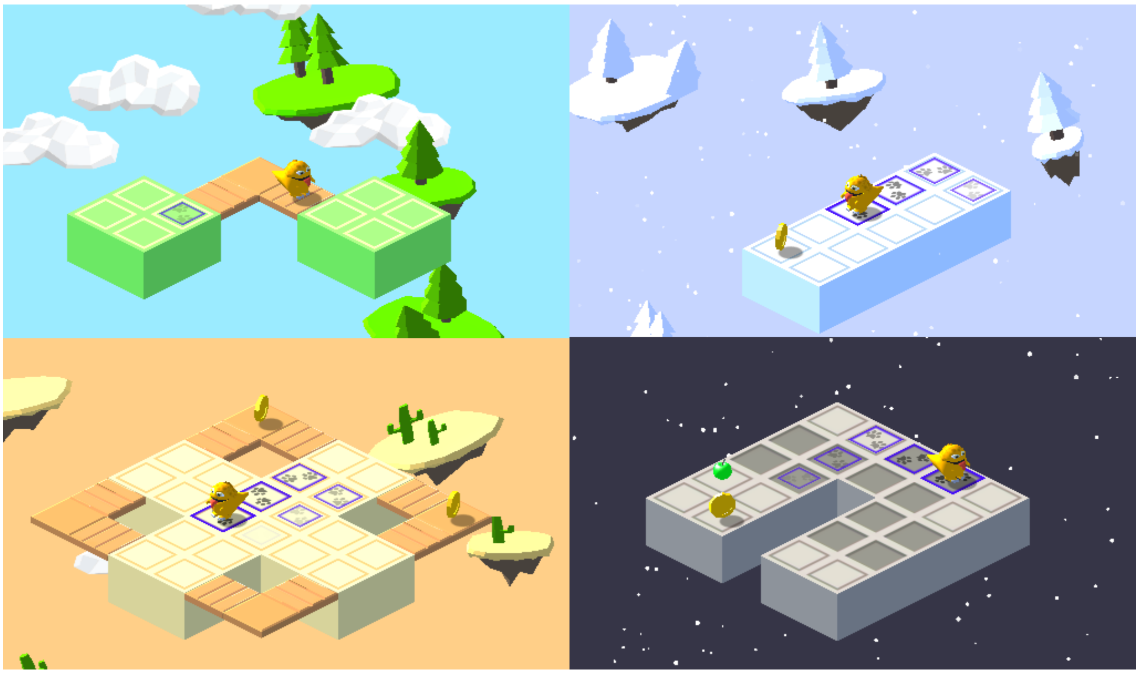


*FIGURE 14. Images representing the different themes of the game*

## 4.4 Characters

This game has a playable character which can jump around a level. As many free-to-play games do, this game also has a number of character skins which the user can unlock with game currency and select to use at any time. The skins are designed to look cheerful and positive, but also we are trying to please players with different tastes. Below in Figure 15 you can see all the characters currently in the game.

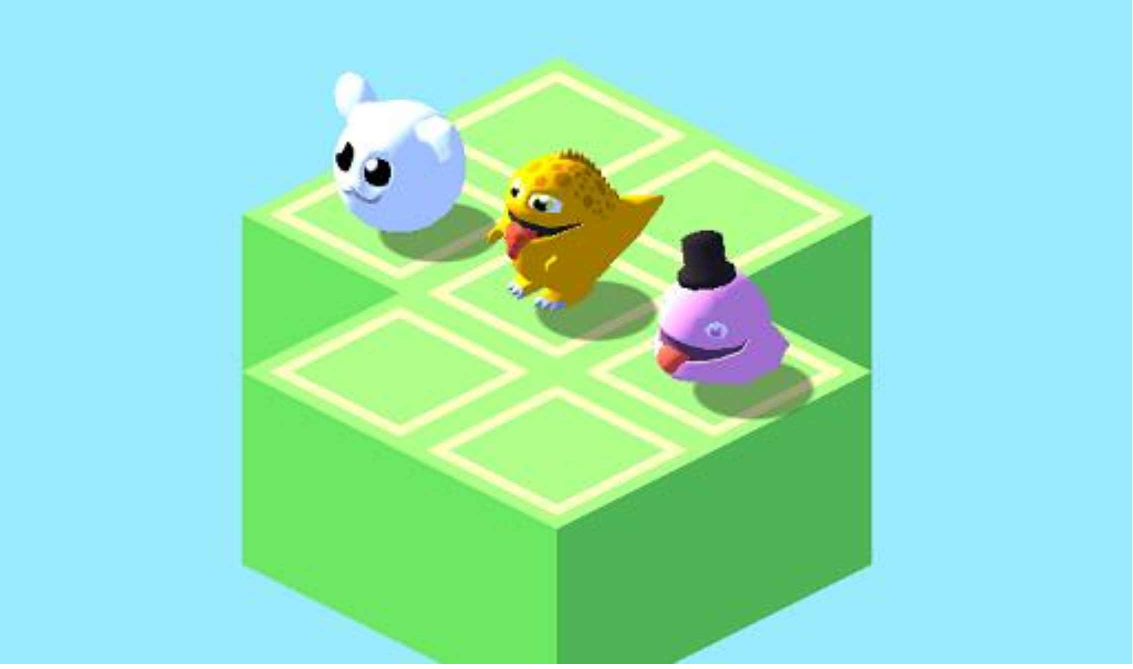


*FIGURE 15. The three character skins side-by-side as they appear in-game*

## 4.5 Objects

The game has two types of objects that the game character can interact with. Collectibles, which have a positive influence, and hazards, which have a negative influence. These two object types are introduced below.

**Collectibles** in this game are objects which the player can pick up to gain some sort of benefit or game currency. All collectibles are visible in Figure 16 below, and they are meant to be positive looking. An example is a green apple, which gives the player character a brief speed boost.

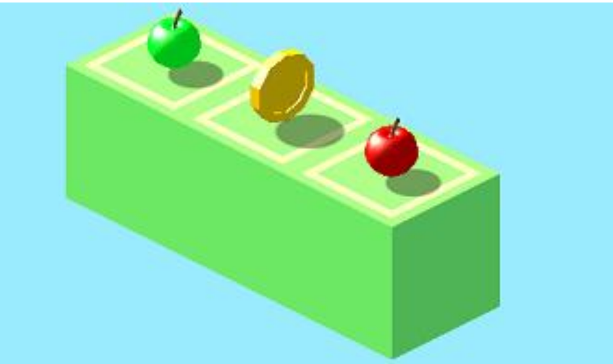


FIGURE 16. All collectibles currently in the game arranged side by side

**Hazards** are objects which can harm or somehow affect the player character in a negative manner. These hazards can slow the game character, make it jump to a random direction, and cause death. Every hazard is made to fit the game theme somehow. For example, a mushroom which slows the down the game character, fits well into the nature theme of the game. The mushroom, and two other hazards can be seen below in Figure 17.

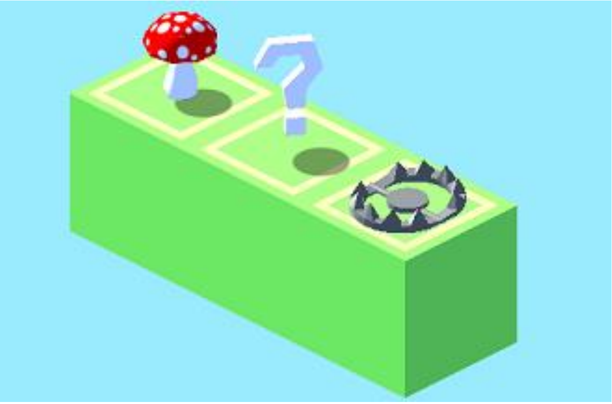


*FIGURE 17. All hazards currently in the game*

# 5 CONCLUSIONS

The aim of this thesis process was to develop a mobile game with the game studio Kaamos Games. The game needed to be a free-to-play action mobile game with an orthographic camera, constructed using 3D elements, built using the Unity game engine and targeted specifically for the Android platform.

The tasks for the development were programming, prototyping, design and working with the game engine. The whole development team had a few years of experience in game development, and there were no issues that could not be solved. Everyone in the team provided equal contributions towards developing the game. One of the most important things observed during the development process was that good communication is very important in a collaborative working environment. The team communicated well about issues regarding this development, and made the process a lot more efficient. In retrospect, a few things could have been done differently. If the game had been planned more carefully in the pre-production stage, there would have been no need to make as many design changes in the production stage, and as a result the whole game development process would have been finished earlier.

This thesis process has given me a better understanding of game design, and it is something I am sure I can utilize in game projects in the future. Additionally, my understanding of the Unity game engine and programming skills have improved immensely.

The current version of the product is well-made and hopefully it will have some success. It can certainly be used for marketing purposes. The game has been showcased in multiple game development conventions where investors have shown interest in it. Kaamos Games continues to develop the game and intends to find a publisher for it. All requirements for the development were fulfilled.

# REFERENCES

Clark, O. 2014. Games As A Service : How Free to Play Design Can Make Better Games. Cited 19.6.2016, https://oamk.finna.fi/Record/nelli17_oamk.3710000000086533.

Edwards, R. 2006. The Game Production Pipeline: Concept to Completion. Cited 27.9.2016, http://www.ign.com/articles/2006/03/16/the-game-production-pipeline-concept-to-completion?page=2.

FramesPerSecond, Unify Community Wiki. Cited 25.6.2016, http://wiki.unity3d.com/index.php?title=FramesPerSecond.

Free-to-Play: Basic Game Loop Theory, Both Guns Blazing. Cited 23.6.2016, https://bothgunsblazingblog.wordpress.com/2013/03/02/free-to-play-basic-game-loop-theory.

GameObject, Unity. Cited 19.6.2016, https://docs.unity3d.com/Manual/class-GameObject.html.

Lazzaro, N. 2004. Why We Play Games: Four Keys to More Emotion Without Story. XEODesign. Cited 9.11.2016, http://www.xeodesign.com/xeodesign_whyweplaygames.pdf.

Määttä, M. & Nuottila, E. 2016. Opas peliliiketoimintaan. Oulun ammattikorkeakoulu. Cited 23.9.2016, http://urn.fi/URN:ISBN:978-951-597-137-1.

New mobile game statistics every game publisher should know in 2016, SurveyMonkey. Cited 7.11.2016, https://www.surveymonkey.com/business/intelligence/mobile-game-statistics/.

Sommer, T. 2013. The Global Mobile Games Landscape (Infographic). Cited 4.6.2016, http://www.applift.com/blog/mobile-games-market.html.

Video game development, Wikipedia. Cited 4.6.2016, https://en.wikipedia.org/wiki/Video_game_development.

Werbach, K. 2016. Lecture 37 - 7.4 Activity Loops. Video. Cited 9.11.2016, https://www.coursera.org/learn/gamification/lecture/EOCUG/7-4-activity-loops.