UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Hans Henrik Viinalass

MiChef – A 2D/3D Platformer Game

Bachelor's Thesis (9 ECTS)

Supervisor: Raimond-Hendrik Tunnel, MSc

Tartu 2020

MiChef – A 2D/3D Platformer Game

Abstract:

The thesis describes the development of a 2D/3D platformer computer game *MiChef*. The game was created in order to explore 2D/3D mechanics in video games. *MiChef* was developed with good software engineering and game development patterns in mind. The thesis gives an overview of the design and realization process. The game was *playtested* on potential users to assess the game quality and get feedback on user experience along with the 2D/3D concept.

Keywords:

Computer game, platformer game, software development, game development, Unity, game design, playtesting

CERCS: P170 Computer science, numerical analysis, systems, control

MiChef – 2D/3D platvormimäng

Lühikokkuvõte:

Lõputöö kirjeldab 2D/3D arvutipõhise platvormimängu *MiChef* arendust. Arvutimäng loodi eesmärgiga tutvuda 2D/3D mehaanikatega videomängudes. *MiChef* loodi häid tarkvara- ja mänguarenduse mustreid silmas pidades. Töös antakse ülevaade disaini ja realisatsiooni protsessist. Mängu testiti võimalike mängijatega, et hinnata mängu kvaliteeti ning saada kasutajakogemuse ja 2D/3D mehaanika kohta tagasisidet.

Võtmesõnad:

Arvutimäng, platvormimäng, tarkvaraarendus, mänguarendus, Unity, mängudisain, mängu testimine

CERCS: P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

Table of Contents

1 Introduction

Platformer games is a video game genre defined by having a player-controlled game character moving between platforms while avoiding obstacles [1]. This genre was popularized in the 1980s and is well known to this day. There are a lot of 2D and 3D, also 2.5D platformers, but not too many games intertwining both dimensionalities. Thus, a new 2D/3D platformer game *MiChef* (see Figure 1) was created in the process of this thesis project.

MiChef is a video game of a chef named Antoine, who continues to cook in a restaurant despite an ongoing world-wide pandemic[1]. The game goal is to collect ingredients in a restaurant's multi-storey kitchen to make gourmet meals for clients. Unfortunately, the virus has spread to his restaurant and our main character must manage the situation. There are no medical supplies left for the restaurant, however Antoine can masterfully swing his favourite pan for repelling enemies. The challenge will turn out to be filling the meal orders, because simultaneously fighting enemies and delivering meals to couriers on time is not a walk in the park.

Second chapter of the thesis starts with an introduction to platformer games and explains how *MiChef* fits into this category. It describes, where the inspiration came from and what helped to shape this game. Besides that, other 2D/3D games are discussed and compared to the created game.

The third chapter describes design choices and changes that happened during the development. It elaborates on *MiChef*'s background story, game objects and visuals. In addition to that, game mechanics and level designs are discussed from the perspective of game development theory.

The fourth chapter focuses on implementation. Both used and alternative technologies, that could have been used with suitable experience, are introduced at the beginning. It is followed-up by information about used programming patterns and the realization process.

Usability test sessions were held for evaluating the game quality, user experience and the 2D/3D effect. The test results and analysis can be familiarized with in the penultimate chapter. The questionnaire, recordings of the sessions, source code and more can be found under chapter Appendix.

[1] https://www.who.int/emergencies/diseases/novel-coronavirus-2019/events-as-they-happen

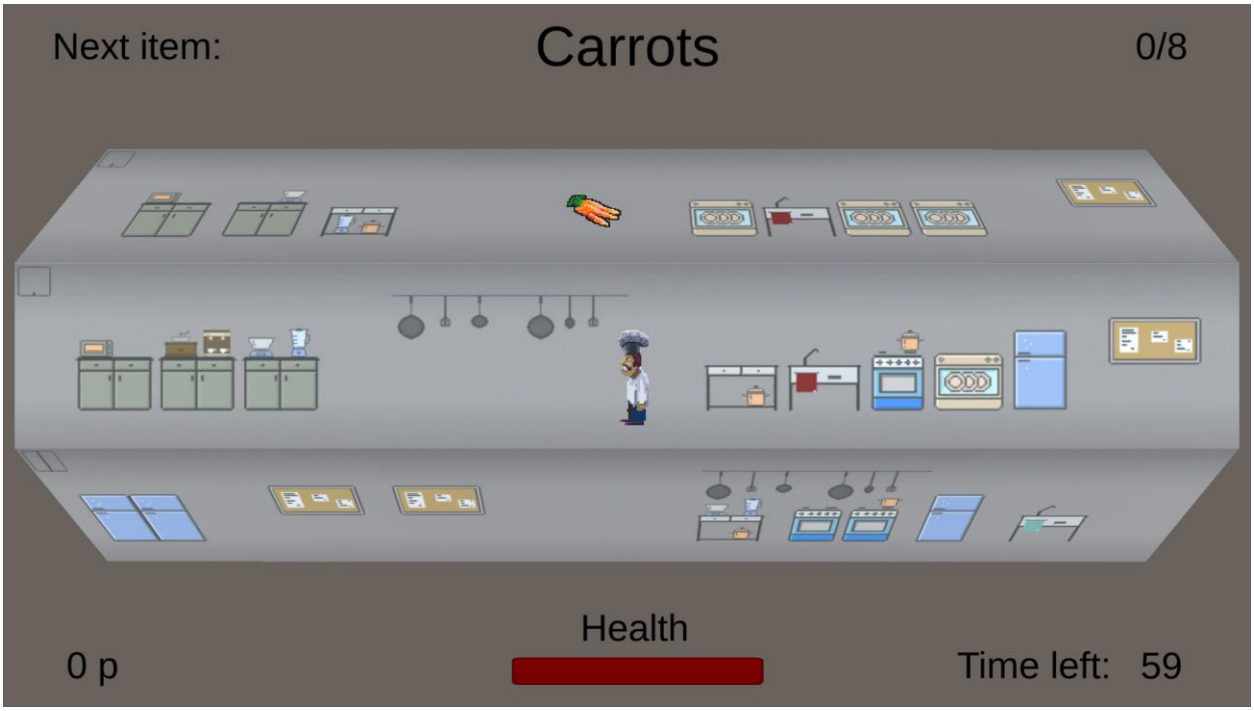

Figure 1. Screenshot of *MiChef*.

The created game with its many versions is available on the indie game distribution platform itch.io: https://hahavee.itch.io/michef as well as in the Appendix IV: Accompanying Files.

2 Background

It is important to know the background of *MiChef* to understand its design choices and development. Subchapter 2.1 gives an overview of platformer games history and describes *MiChef*'s connection with the genre. Subchapter 2.2 reveals inspirations, which influenced the created game. The last subchapter 2.3 discusses and compares 2D/3D effects used in other similar games.

2.1 Platformer Games and MiChef

Platformer games is a video game genre, which playstyle is characterized by moving a player-controlled character between platforms to reach an objective [2]. The gameplay typically involves having to dodge some sort of obstacles. The genre was popularized in the 1980s and has stayed relevant to this day [3].

Platformer games have been through different periods over their 40-year existence. The first games were single screen platformers. The players had to complete a task in each setting and were moved to a next scene upon completion. The first true platformer is *Donkey Kong*[2] (1981) by Nintendo. In that game players control a man, who hops over barrels and climbs ladders to rescue his love from a gorilla. Early platformers were developed single screen partly due to the limitations of the technology at the time.

The single screen era began declining a few years later after introducing side-scrolling games to the genre. Horizontal scrolling was quickly followed by multi-directional scrolling. *Super Mario Bros.*[3] (1985) by Nintendo and *Sonic the Hedgehog*[4] (1991) by SEGA were the biggest rivals and defining games of side-scrolling platformers.

[2] https://www.nintendo.com/games/detail/arcade-archives-donkey-kong-switch/
[3] https://www.retrogames.cz/play_005-NES.php?language=EN
[4] https://www.sega.com/games/sonic-hedgehog

First 3D platformers were released in the 1990s. They were preceded by 2.5D games, which fake a 3D perspective of the game world using 2D graphics. The game that revolutionized true 3D platformers was *Super Mario 64*[5] (1996) (see Figure 2 for a visual comparison of discussed games).

Figure 2. Images of *Donkey Kong*[6] (single screen), *Sonic*[7] (side-scroller), *Penguin Adventure*[8] (2.5D), *Super Mario 64*[9] (true 3D).

Making video games has been made easier over the years with the help of rapidly improved *game engines*. Some indie developers are trying to revive the previous eras of platformer games by bringing new titles to the market [3]. The game *MiChef* developed during this thesis is an example of that. It is a single-screen platformer with *pixel art design*, similar to old school platformer games. The mechanic that separates it from older platformers is the 2D/3D game world.

[5] https://www.mariowiki.com/Super_Mario_64
[6] https://www.mentalfloss.com/article/63838/13-things-you-might-not-know-about-donkey-kong
[7] https://www.retrogames.cz/games/117/Genesis_01.gif
[8] https://images.generation-msx.nl/software_game/9a7bb843.png
[9] https://www.retroplace.com/pics/n64/ingames/97424--super-mario-64.png

2.2 Inspirations

MiChef is a keyboard only game, meaning that all in-game actions require keyboard input. This is designed so to make the game fully playable on a laptop, without requiring the use of a mouse. Many people with desk jobs want to change their environment ever so often and sit on a couch or lay on a bed while playing. Also, travellers have limited space and often than not they do not have sufficient room for using a mouse during transit. Some great video games combating this problem are single screen platformer game *Jetpack*[10] (1993) by Adept Software and side-scroller platformer game *Ditto*[11] (2014) by Nitrome (see Figure 3). Apart from inspiring the use of keyboard only controls for *MiChef*, they were also a part of the reason pixel art was used in the game design.

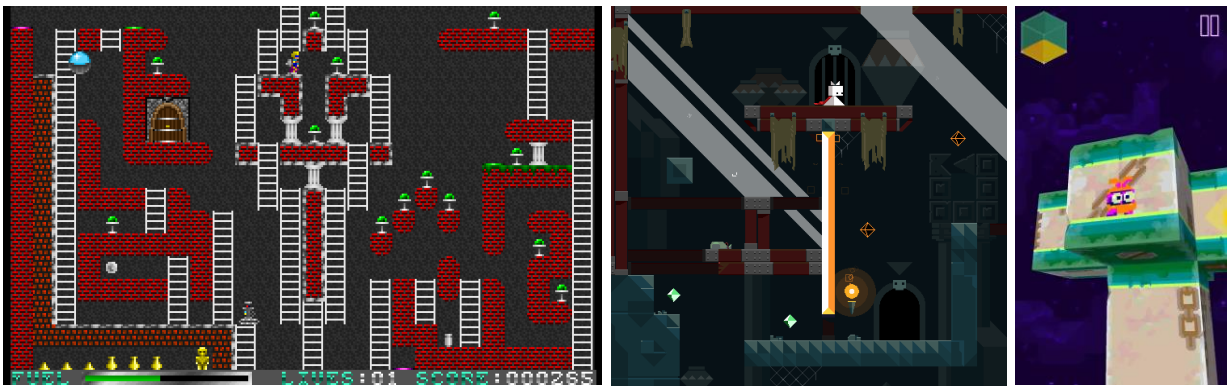

Figure 3. *Jetpack*[12] (left), *Ditto*[13] (middle), *Flat Pack*[14] (right).

The inspiration for the 2D/3D concept came from playing a mobile platformer *Flat Pack*[15] (2017) by Nitrome (see Figure 3). This mechanic is discussed more in depth in the next subchapter along with some other games.

[10] https://www.adeptsoftware.com/jetpack/
[11] http://www.nitrome.com/games/ditto/
[12] https://www.playdosgames.com/assets/screenshots/jetpack_2.png
[13] https://www.freeindiegam.es/wp-content/uploads/2014/03/ditto.png
[14] https://img.utdstc.com/screen/13/flat-pack-10.jpg:800
[15] http://www.nitrome.com/blog/articles/1424/

2.3 Comparison of 2D/3D Effects with Similar Games

Figure 4. Sprite folding.

Flat Pack is a mobile platformer game launched in September 2017. The player controls a bright-coloured 2D character traversing on the surface of 3D shapes. The iOS version of the game includes an *augmented reality* mode to present playable levels as if they were a part of the actual world. In comparison with *MiChef*, this 2D/3D effect is more refined, because the character sprite folds around the 3D corners. The character sprite sticks onto multiple faces (see Figure 4) when passing an edge of a 3D shape. This kind of realization of the effect is suitable for Flat Pack due to the game world being observable in 360 degrees horizontally and vertically.

Super Mario Odyssey[16] (2017) is a platforming game for the Nintendo Switch. The game is about the adventures of the famous video game character Mario. Majority of the gameplay happens in 3D. Nintendo made some sections of their game 2D and advanced the retro feeling of old Super Mario games even further by using 8-bit music in these sections. The dimension effect was cool and thus was similarly designed in *MiChef* (see Figure 5). The 2D sprites do not bend on 3D surfaces like in *Flat Pack*. The implementation is not unsettling because the transitions between 2D world faces are fast and the camera movement distracts players from the imperfection.

[16] https://www.nintendo.com/games/detail/super-mario-odyssey-switch/

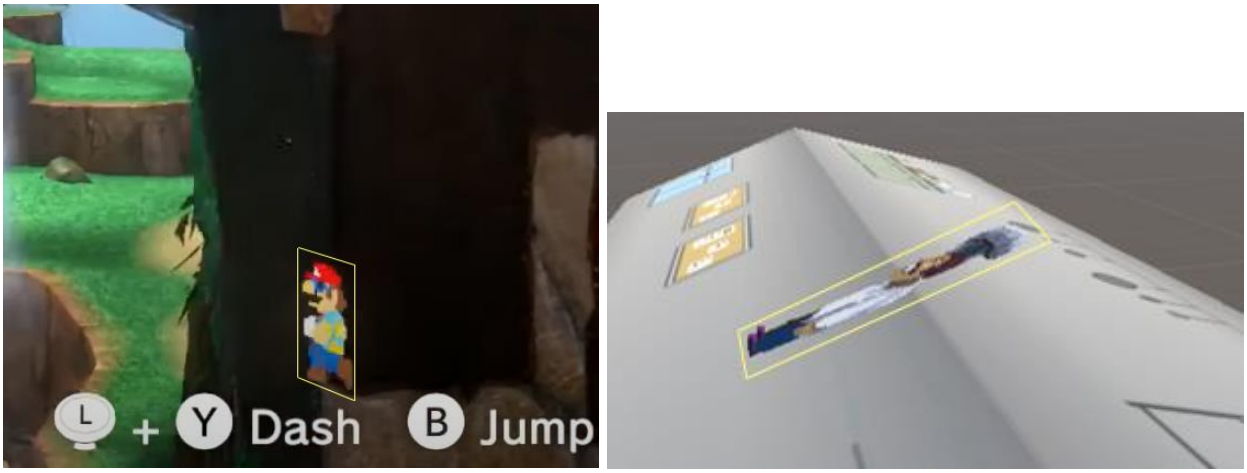

Figure 5. Screenshots of *Super Mario Odyssey* (left), *MiChef* (right). The yellow markings depict the sprite planes.

Perspective[17] (2012) released by Digipen has a different way of intertwining dimensions. This mind-bending puzzle platformer is all about finding a way to navigate around its world space. The 2D/3D effect is achieved by moving around and locking the camera view to create an environment for the character in order to maneuver platforms (see Figure 6).

Figure 6. *Perspective*.

[17] https://games.digipen.edu/games/perspective

Figure 7. Screenshot of *FeZ.*

A somewhat different manner of the 2D/3D effect usage was introduced in *Fez*[18] (2012) (see Figure 7). *Fez* is another puzzle platformer developed by Polytron Corporation. In this game, players adventure around the game world gathering cube shards. The game world of *Fez* exists in 3D space but is always presented to the player in two dimensions. The main character is always perpendicular to the camera and the whole world is orbiting around him. This was something I wanted to have in *MiChef*.

[18] https://fez.fandom.com/wiki/FEZ_Wiki

3 The Design

Agile is a software development approach characterized by the ability to create and respond to change [4]. The game was designed using this methodology. It turned out to be a great choice, because lots of things changed or were updated during the development of the project. Because of the uncertainty of the features and rapid change in ideas, most development was done using a *build fast, fix later*[19,20] start-upish approach. Following subchapters explain all the progress in a greater detail. Subchapter 3.1 tells the story of MiChef and its evolution throughout the implementation. Subchapter 3.2 gives an overview of gameplay. The last subchapter contains information about all visual aspects of the game.

3.1 The Theme

Schell has written in his book [5] that powerful themes create powerful experiences. Thus, a theme was chosen to enhance player experience. Following subchapters 3.1.1 and 3.1.2 unravel the backstory and its progression.

3.1.1 The Refined Theme

> Antoine works as a chef in Vendée region, France. He is a man with great talent for cooking and is known for his affection for high quality pans. After arriving at work one morning, it was supposed to be an ordinary day for him. But alas, the boss informed him that a deadly virus had begun troubling the country, and in fact, the whole world. The virus is called COVID-19. The owner of the restaurant told him to continue his work or else he would be laid off.
>
> That is exactly what Antoine does. He did not need to be told though, because he would rather die than let people live without his gourmet food. Unfortunately, the man possesses extremely conservative world views and has little to no trust in modern medicine nor does he approve of the established sanction standards. What does he do when confronted by the

[19] https://www.itprotoday.com/windows-10/release-it-now-fix-it-later-approach-software-and-hardware
[20] https://uk.reuters.com/article/us-tesla-quality-insight/build-fast-fix-later-speed-hurts-quality-at-tesla-some-workers-say-idUKKBN1DT0N3

virus? He grabs his most precious cooking pan and starts swinging it at the enemies who threaten him with the loss of his job. However, it is a challenging task to juggle fighting the virus and continue his baking. Can Antoine handle the pressure, or will he lose his position at the end of the day? It is time to find out.

The theme cannot be grasped in full detail by merely playing through levels. The gameplay of *MiChef* is centered around the backstory, but it must be familiarized with in the game menu in order to have a complete overview.

3.1.2 The Change of Theme

The original story was not intended to be COVID-19 themed. It is not meant to make fun of the virus. The inital story of *MiChef* was more or less the same but with the exception of the enemies, originally some sort of kitchen contaminators. It just happened to be that the designer linked with *MiChef* had previously drawn a virus. Because of the pandemic[21], this seemed like a good opportunity to create a lighthearted perspective with little changes to the original story.

3.2 Visuals

The first vision of *MiChef* was visualized in a notebook (see Figure 8). The sketch was the target to aim for whilst designing the game. It depicts the basic game mechanics, but some features were changed or improved during the process: 2D/3D effect, kitchen contaminators, attack mechanics and collectibles.

[21] https://www.who.int/emergencies/diseases/novel-coronavirus-2019/events-as-they-happen

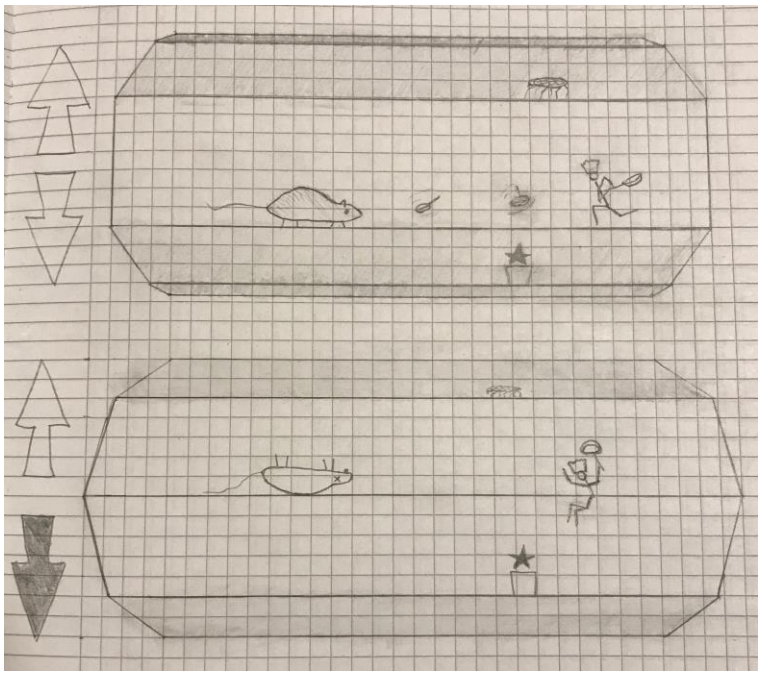

Figure 8. Page from the design notebook. Illustrates the 2D/3D mechanic.

Pixel art was chosen for this project because of the influence from playing other platformers and the belief that this form of art could be produced by the creator of the game. Nevertheless, during the realization process, a designer was brought onboard to make the game world feel more alive. Having an artist, especially for drawing characters, was a great measure to make the design more tailored to the game. It also cut the time it would have taken to draw everything independently.

3.2.1 Game Objects

Game objects are any objects that players can see or interact with [6]. Subsequent subchapters describe and illustrate the game objects of *MiChef*.

Player

The character is a male chef Antoine. His art concept was started by drawing stick figures (see Figure 9). Antoine's sketch model was replaced with a real asset after finding a designer from the Facebook group *APT GameGenerator*[22] – a game development group in Tartu. The designer used the created concept to draw two animations of the chef for this thesis: move and attack. Ideally there would also be idle and jump animations in case of no movement or switching between floors.

[22] https://www.facebook.com/groups/GameGenerator

These could be considered as future improvements. The chef's move animation is walking (see Figure 10) – duration 1333 ms. His attack animation (see Figure 11) is whipping a pan through the air. Original duration of this animation is also 1333 ms but it is sped up to 333 ms.

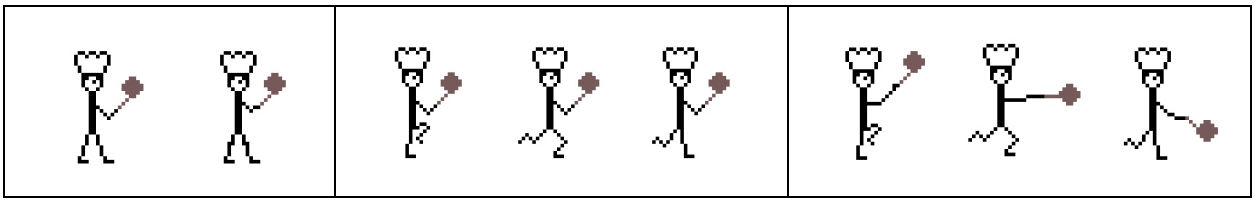

Figure 9. Chef legacy animation: idle (left), move (middle), attack (right).

Figure 10. Chef move animation.

Figure 11. Chef attack animation.

Enemy

Figure 12. The GrokeFigure 12.

Following the original idea, there was supposed to be more than one enemy. As can be seen from the game sketch at the start of chapter 3.2, some of the kitchen contaminators could have been mice and bugs. However, the created enemy sketch displays a different character. It is a hungry blob (see Figure 13), smoothly gliding on the ground moving towards the chef open-mouthed, wanting to eat him. The visual of the enemy sketch was inspired by The Groke (see Figure 12), a friendly but misunderstood character in the famous 1990s Japanese anime television series "Moomin". For clarity's sake, the eating part of the animation is not related to The Groke. The blob enemy character was discontinued after receiving the virus enemy design from the artist and thus the decision to make the game virus themed.

Figure 13. Blob move (top) and attack (bottom) animations.

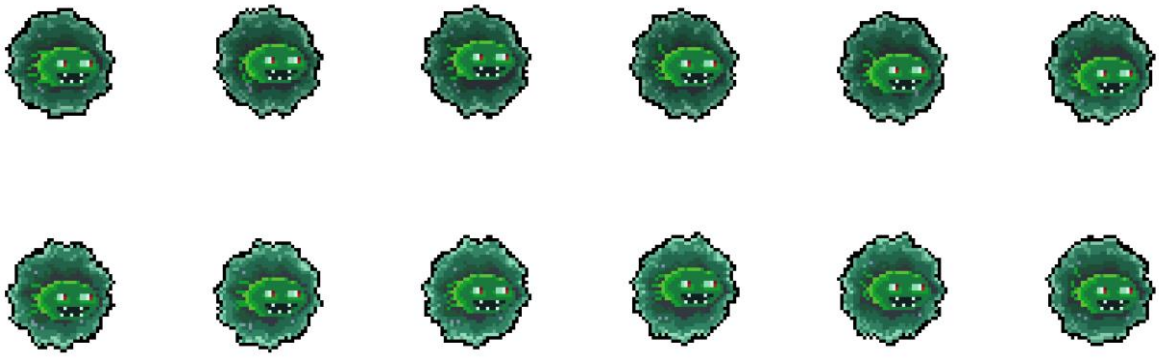

Figure 14. Virus move animation.

Figure 15. Virus attack animation.

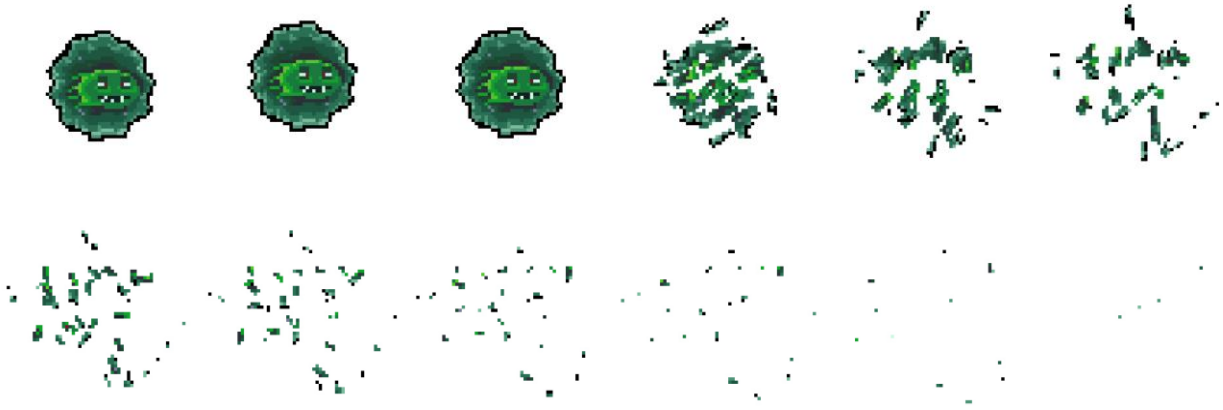

Figure 16. Virus death animation.

The currently implemented enemy is a virus. There was a need for the designer's existing drawing to be tweaked a little bit for the game's purpose. The virus seems to be coming from another dimension to cripple anyone in its path. Portrayal of the portal to another dimension is not equally understood by people and can be falsely interpreted. For example, the supervisor of this thesis thought that the enemy resembles an unharvested cabbage. The virus has three animations: move (see Figure 14), attack (see Figure 15) and death (see Figure 16).

Recipe Ingredients

Figure 17. Ingredients

The purpose of the main character is to fulfill food deliveries. In order to do that, they need to collect different ingredients (see Figure 17) from the game world. All of the ingredient sprites but one were made by the designer. A drawing of ginger from an open-source pixel art package was added as an extra item. The items are all 32×32 pixels in size. The ingredients appear raw, as they should be, for making it obvious that a meal will be prepared from them. The dark outlines help the ingredients stand out more clearly from the background.

Building

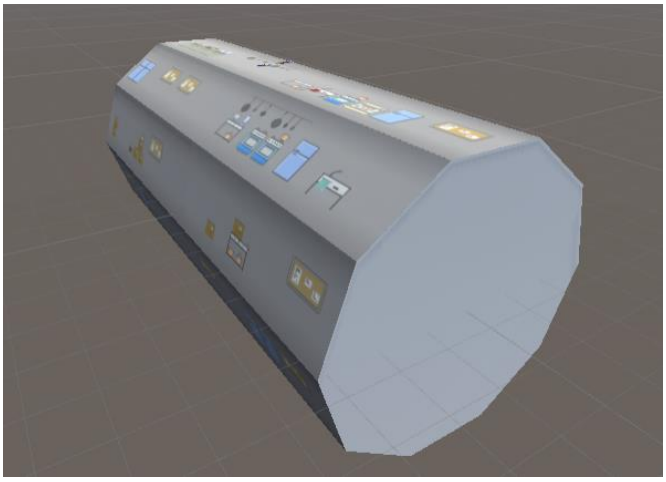

Figure 18. The building.

The game environment, a restaurant kitchen is a *decagonal prism* (see Figure 18) designed by the author of the thesis. This ten-storey kitchen has repeating elements on every floor, but each one is somewhat different. All floors are colored light gray with dark gradient near the flooring and ceiling. The gradient is applied in order to stimulate lighting. The kitchenware items have dark outlines for sharpness but appear a bit dim. The reason for this is to separate floor design elements from game objects.

Themed Menu

Completing a level makes a themed menu (see Figure 19) appear. It congratulates players on the achievement and gives an overview of the score gathered from the level. There are two buttons at

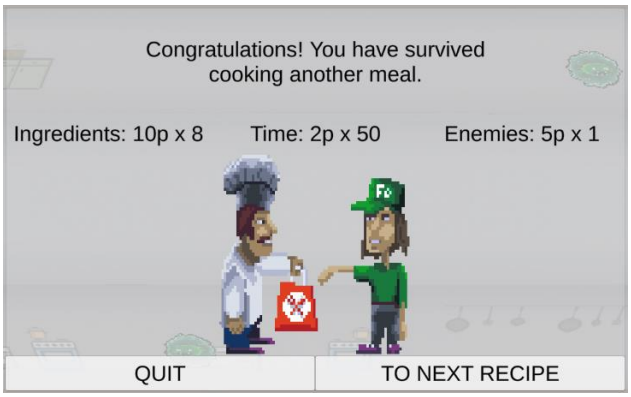

Figure 19. Menu pop-up upon level completion.

the bottom of this menu to let players continue their journey or quit the game. The drawing of the chef handing out a prepared meal to a courier is necessary to deliver the story to players.

3.2.2 Effects

Three visual effects (VFX) are implemented in *MiChef*. One of these is a *particle effect*, which triggers every time when players collect an ingredient. This VFX is a short burst of stars in the location of a collected ingredient (see Figure 20). Feil and Scattergood have written in their book [7] that adding particle effects is one of the fanciest ways to create ambiance in a game. They mention also that well-used particle effects can immerse players in the game world. That is why this effect is applied in *MiChef*.

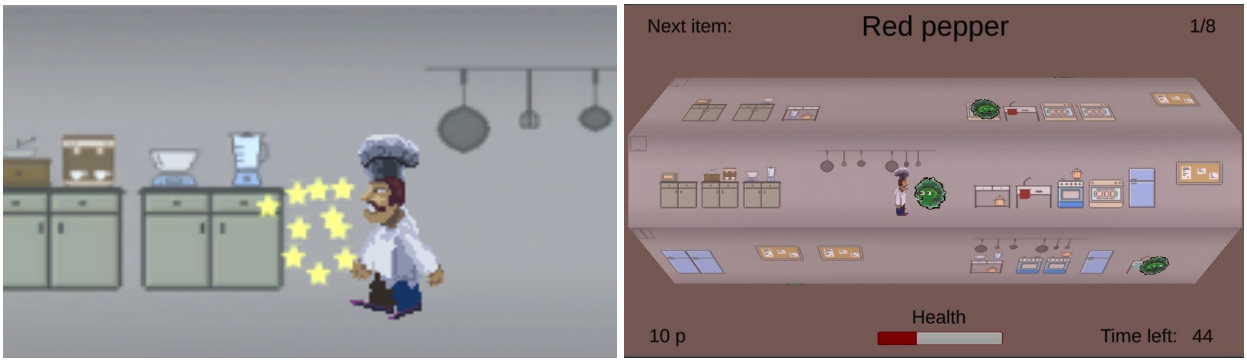

Figure 20. Stars particle effect (left) and red vignette (right)

Another common effect in games is the *vignette*. Alan Thorn has addressed vignettes in his book [8]. He explains that vignettes convey tension and panic. In *MiChef*, a flashy red vignette is used to signal players about the low health of the character (see Figure 20). This VFX is bound together with a camera shake effect in order to enhance tension and panic. Jonasson and Martin demonstrate in their talk [9], that camera shake means power. However, in *MiChef*, this effect displays the opposite: lack of power. It shows that players are beginning to lose control over their character.

3.3 Gameplay Design

Without gameplay, there is no game. Subchapter 3.3.1 describes in-game interactions, which is crucial for gameplay. There it is explained what things take effect and how they happen in the game. Subchapter 3.3.2 reveals the level design of *MiChef*.

3.3.1 Mechanics

The primary goal of each level is to catch all the ingredients for the level's recipe. With this requirement fulfilled, players can progress to the next level (see Figure 22). For a recipe item to be collected, players must locate the item by switching between floors and touch it by simply colliding with the physical space of the item. The physics of the building could appear confusing to some players because there is no limitation in moving around floors. Players can switch between the lowest and the highest floor by going up or down one floor. This movement could have been restricted or one floor of the building could have been replaced with a roof. Nevertheless, it was decided that the game world will be designed to be repeating in a vertical direction.

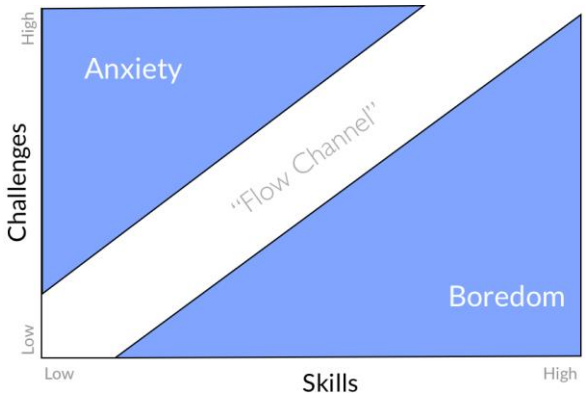

Figure 21. Flow chart.

Simply catching ingredients does not feel enough to make the game challenging. Without challenge, the gameplay results in boredom (see Figure 21). That is why there are viruses who impede the chef from reaching his goal. The enemies have a single *hitpoint*, meaning they are killed by a single attack of the chef. Viruses initially spawn one per each floor a few moments after a level begins. They are tied to the floor they are spawned into and cannot move to other floors unlike the chef. Viruses move horizontally between the walls and wait impatiently for the chef to arrive. The enemies stop their action and start moving toward the chef when they sense him being on the same floor by their line of sight. When being in continuous contact with Antoine, they drain his health. The chef dies if he has no health left and the game ends as he can no longer accomplish the task at hand (see Figure 22).

There is another lose condition for making the game more challenging and trigger an alternative ending. Each level has a timer, signaling players that they have a finite time to complete the task. This gives players an indication if they should either hurry with the recipe or take their time fighting enemies. If players do not manage to complete a level in given time, the game is over (see Figure 22). There are two reasons to fight viruses, as this is not the requirement for completing the game: killing them grants players a short-term speed boost and adds extra points to the score.

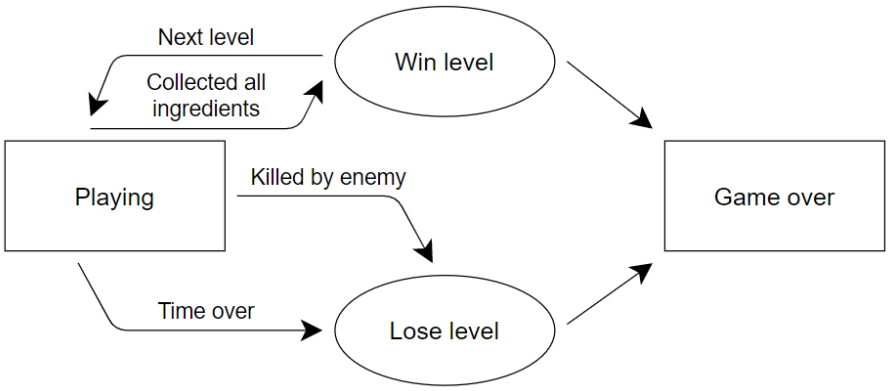

Figure 22. Game flow chart.

Score is gathered in three ways: collecting an ingredient gives 10 points, killing an enemy adds 5 points to the score and 2 points are given for each second left to complete a level. This mechanic makes the game competitive. To get a high score, players have to find balance between collecting recipe items as fast as possible and bashing enemies.

3.3.2 Level design

MiChef consists of five levels ranging from 1m to 1m 40s in duration. No more levels are made for the minimal viable product (MVP). Five levels are enough to get feedback from playtesters to point out the strengths and weaknesses of the game. Levels in *MiChef* are designed to increase gradually. In each next level, players have to collect two more items than in the previous level for advancing. Except in the last level only one item is added to the list of ingredients. In addition, enemies are made increasingly faster after level 2. Players are given ten seconds extra time per level to combat this increased difficulty.

Ideally, there would be more features to include in level design, e.g. side quests or boss fights. However, due to the size of the thesis project, there was not enough time to implement them.

4 Implementation

It is important to talk about the implementation, which realized the game design. Games are one of the most difficult software types to implement [10]. That is why there are game engines and game patterns to ease the development. Subchapter 4.1 declares the technologies used for creating *MiChef* and points out alternatives that could also have been used. Subchapter 4.2 describes the realization process including game and software development patterns.

4.1 Technologies

This game is mainly built on four technologies, excluding development dependencies. A game engine was chosen for faster and simpler development. A scripting language was picked for writing the game logic. 3D modelling required a computer graphics software toolset, and pixel art a raster graphics editor.

Unity[23] is the game engine used for this project. There are four main reasons behind this choice. First and foremost, Unity is an open-source game engine, free for students and personal use if received revenue or funding is less than $100K in the past 12 months[24]. Furthermore, Unity is a well-known game engine with a large userbase [11]. This is helpful, because a lot of trial and error can be avoided by searching for help online. Lastly, Unity features cross-platform development – excellent for making *MiChef* playable on many systems. These requirements narrowed alternatives down to Unreal Engine[25] and Godot[26]. However, Unity had an advantage because of the author is experienced with only this game engine.

Games can be created without scripting, but developers are much more limited by not using any [11]. C# is an industry-standard language similar to Java and C++. Unity supports this programming language natively[27]. Without any other options, C# had to be used for the

[23] https://unity.com/
[24] https://store.unity.com/#plans-individual
[25] https://www.unrealengine.com/
[26] https://godotengine.org/
[27] https://docs.unity3d.com/Manual/CreatingAndUsingScripts.html

implementation together with Unity engine.

Blender[28] is an open-source 3D computer graphics software toolset. It is used in *MiChef* for modelling the building, a decagonal prism. 3D models of the restaurant with floor designs are imported to Unity for utilizing the result achieved in Blender. Alternatives to Blender were SketchUp, Autodesk Maya, Wings 3D and Natron to name a few[29]. Again, choosing Blender has to do with the author's prior experience.

In-game pixel art for characters, animations and floor designs done by the author were created with Adobe Photoshop CS6[30]. There are many open-source alternatives to that, Gimp[31] and web-based Pixilart[32] for example. Adobe's tool was selected for the premium experience and variability of its toolkit.

4.2 Realization

Implementation of *MiChef* follows agile methodology principles, likewise to design. This is partly because of limited experience of the game's creator. Creating the game gave a great deal of knowledge about various ways to implement features. Throughout the project, many mistakes were made, especially at the beginning. Those mistakes were gradually improved upon until the end.

Aside from that, some things were correct from the beginning. Code comments should answer why something is done and its purpose instead of how it is done, which can be understood from reading the code [12]. The developed code follows this principle and is a sign of good documentation. Self-explanatory and consistent use of naming convention is also a favourable code trait. A. Hunt and D. Thomas have written that variable names should be well chosen and meaningful. Developers write the code a few times but read it hundreds of times [12]. It can be deduced that this recommendation applies not only for variable names, but everything in code repository, file and folder names included. It is important to use relevant names for clarity. It saves a lot of

[28] https://www.blender.org/
[29] https://alternativeto.net/software/blender/
[30] https://www.adobe.com/products/photoshop.html
[31] https://www.gimp.org/
[32] https://www.pixilart.com/draw

headache for anyone catching up with a project and makes the code maintainable. These good practices were followed during the development of *MiChef*.

Another essential basis for clean code is the single responsibility principle (SRP) [13]. Writing clean and maintainable code means that files should not be thousands of lines long. The code may work fine, but it is not sustainable. It puts a limit on further development as *code smells* can easily turn into unworking code by not dealing with them. *MiChef* had problems regarding that topic at first but code smells were gradually removed by refactoring, so that each script in the game would serve a single purpose.

S. McConnel has written in his book [14], that unused parameters in code correlate to more errors. This idea can be broadened to code in general. The less there is, the easier to read and develop it. He has listed clearing unused code as an important step in code code clean up. There was unused code at times during the development of *MiChef* due to uncertainties. However, untouched code was removed as soon as its necessity for future development became irrelevant.

In *MiChef*, a singleton[33] object is used to hold game info. The singleton pattern means that there exists a single instance of a class, which is globally accessible [15]. This object is where player score is held during *MiChef* game time. Aside from that, it serves a purpose in the game to store level design information. This is where level times, ingredient amounts and enemy speed for each level is read from other scripts.

For keeping the main character fixed to the prism-shaped environment, Unity's gravity system is used via the Rigidbody[34] component. There is a script attached to the building object that handles attraction. In order to keep the character from spinning clockwise and counterclockwise, the chef's rotation is frozen on some axes by the Rigidbody component constraints.

Building rotation was initially achieved by using coroutines[35]. Though, this method had problems. The rotation was precise when floors were switched slowly. But errors in rotation degrees started to appear when floors were switched fast. These small errors added up and the method had to be changed. The current method used in Update() method is shown below.

[33] https://csharpindepth.com/articles/singleton
[34] https://docs.unity3d.com/Manual/class-Rigidbody.html
[35] https://docs.unity3d.com/Manual/Coroutines.html

```
if (rotateTimeLeft <= 0)
        {
            if (Input.GetAxisRaw("Vertical") == 1)
            {
                targetRotation *= Quaternion.AngleAxis(rotationDegrees, Vector3.up);
                rotateTimeLeft = duration + 0.1f;
            }
            if (Input.GetAxisRaw("Vertical") == -1)
            {
                targetRotation *= Quaternion.AngleAxis(rotationDegrees, Vector3.down);
                rotateTimeLeft = duration + 0.1f;
            }
        }
        // apply rotation
        transform.rotation = Quaternion.Lerp(transform.rotation, targetRotation, 10 * smooth *
Time.deltaTime);
```

The rotation time for switching a floor is 0.4 seconds. Additional 100 ms is added after the rotation to lock the floor in place for a brief moment before allowing further vertical movement between platforms.

Figure 23. A spawner object.

Spawner objects are used in the middle of each floor for two purposes: spawning enemies and ingredients. Spawner objects are located very close to floor surfaces (see *Figure 23*), because enemies and ingredients do not use gravity. They are not pulled toward the building. Both are spawned as children of the spawner object inheriting their parent's position in the game world. The difference between the two is that ingredients are spawned with random horizontal offset from the middle.

Despite following good programming patterns, the realization is not perfect. There are still code smells that require attention if the project is to be continued. With the experience gained from this project, lots of mistakes can be avoided in future developments.

One code smell is mixed up logic. Some programmers prefer sticking logic to scripts, some to game engines. *MiChef* spreads the logic between Unity and C# scripts. Both methods have their pros and cons, but one should be chosen for simplicity. However, some aspects were not decided on. For example, it is tedious to give parameters to components in Unity. First, an object needs to be dragged to the parameter's location. This alone is not too annoying, but if anything needs to be reset in the object's component, then everything in that component resets, including parameters. In contrast, finding and using parameters via scripting is string-based. That means no strong-

typing[36] is involved. If an object's name is changed, then all components using that parameter break. That is why there are parts of logic shared between Unity and C# in *MiChef*.

Another code smell is regarding spawner objects. Placing spawners close to floor surfaces is done manually. This is not perfect for further development. If another building is to be added, then spawners need to be attached and tweaked manually, again. A more elegant solution would be of a great use in there.

[36] https://whatis.techtarget.com/definition/strongly-typed

5 Testing

Testing was conducted to assess the game quality and get feedback on user experience and the 2D/3D concept. Subchapter 5.1 explains the testing strategy and gives a brief overview of the chosen participants. Subchapter 5.2 displays the test results and draws conclusions from received feedback.

5.1 Methodology

It is known that testing on five people is the best option for maximum benefit-cost ratio [16]. That is why feedback for *MiChef* was gathered by conducting five playtesting sessions. Those sessions were conducted on Internet messaging platforms that allow screen sharing in order to monitor the gameplay. The participants are acquaintances of the game creator's friends. They were chosen for receiving neutral feedback that could not have been obtained by testing the game on close friends.

Players were told to learn the objective of the game by navigating around the menu. After brief contact with game instructions, they could begin playing. Playtesters were recommended to talk about their impressions and upcoming questions during playtime. However, they were assured that talking could have also been done in-between levels or whilst in pause mode. During playtime, it was observed how well the testers understood the game and what they focused on. Some hints were given to testers at times they had obvious difficulties. A questionnaire (see Accompanying Files) was given to them afterwards.

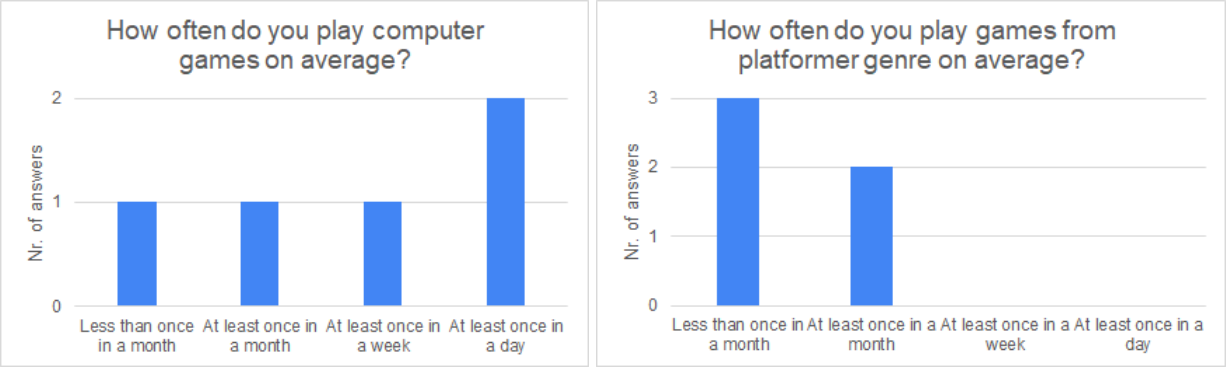

Figure 24. Testers background with computer games.

“How often do you play computer games on average? How often do you play games from platformer genre on average?” It turns out that most testers are actively playing computer games. However, platformers are not too popular among them (see Figure 24). This means these players are not too accustomed to platformers and might have more difficulties playing them. Taking account only the time spent inside the application, the shortest test session lasted little over 11 minutes and the longest close to 21 minutes. Recordings of test sessions can be found in the Appendix V: Test Sessions.

5.2 Results

Figure 25. Overall experience.

"How do you rate your overall experience playing *MiChef*?" All in all, players had a good experience with *MiChef* (see *Figure 25*). Testers claim that the game is challenging and not overly complicated. Though, some of them had trouble understanding the objective. These testers focused a lot more on killing viruses than collecting ingredients. Only after a while did they realize by themselves or after being given tips, that collecting ingredients is the main goal. Perhaps a solution would be to display the objective on the screen of the first level if an ingredient has not been collected in a fixed time.

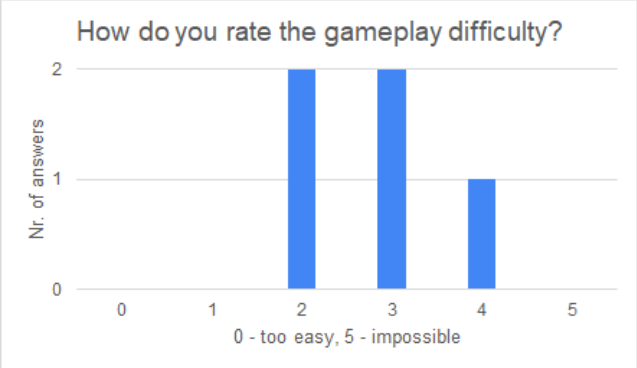

Figure 26. Gameplay difficulty.

"How do you rate the gameplay difficulty?" Testers' ratings of the gameplay difficulty can be summarized as moderate (see *Figure 26*). One tester who did not manage to finish the game after several attempts reached level three before quitting. Some testers explained that the game was hard at first, but further levels turned out to be easier after getting a grip on the concept. That is why one participant requested adding more obstacles to the game. One player mentioned that time restriction was the main factor in determining gameplay difficulty. Some of the respondents expected to see a final boss in the last level and argued that this and other features could be implemented to increase the difficulty. It was noted that enemies spawning unexpectedly created panic and was the main reason for losing. It should have been expected that this sort of level design has too fast of a learning curve and gets boring quickly. Players' suggestions on this matter are on point and should be taken into account in case of future improvements.

“How do you rate the visual effects?” Unfortunately, this topic did not receive expected answers despite a description under the question. Thus, a graph is little of use. By analysing the answers, it seems that the testers rated visuals and graphical user interface instead of particle effects, red vignette and camera shake for this question. Interestingly, a bug was found in two sessions regarding the vignette. It appeared that this VFX had far too much opacity and disturbed gameplay.

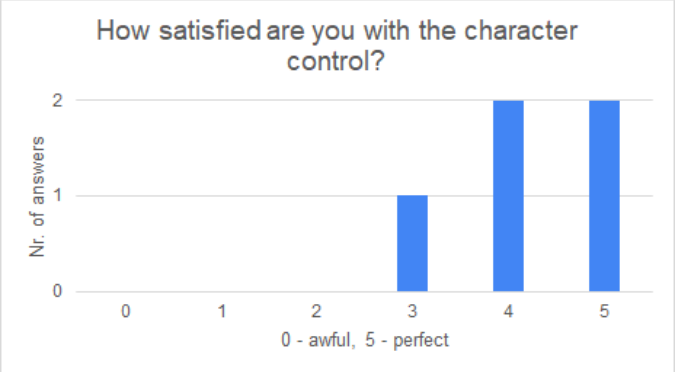

Figure 27. Character control.

“How satisfied are you with the character control?” Playtesters are satisfied with character control (see Figure 27). The controls are considered intuitive and easy to handle. There is a comment from a tester suggesting making the attack movement a bit slower in order to make the game more difficult. There are few ways to control the character, thus the implementation could not have gone very wrong. There may be some imperfections like not having a jump animation, questionable character acceleration speed or animation timings. Fortunately, the testers did not mind these problems too much.

“How do you rate the aesthetics? How do you rate the graphical user interface?” Testers are mostly on the same page with the aesthetics and graphical user interface part (see Figure 28). They are satisfied, but there is room for improvements. Some of the testers suggest adding distractions to the building design and more animations overall. Players agree that menu styles differ from in-game style and expected more consistency, that means pixel art theme to be used everywhere. Most of them think that buttons and text could be aligned better, and a matching font would help with the theme. This is definitely an issue that was overlooked by the creator of the game. The menus did not get deserved attention and were handled as secondary features. Nonetheless, the menus are easy to navigate and fulfill their primary role.

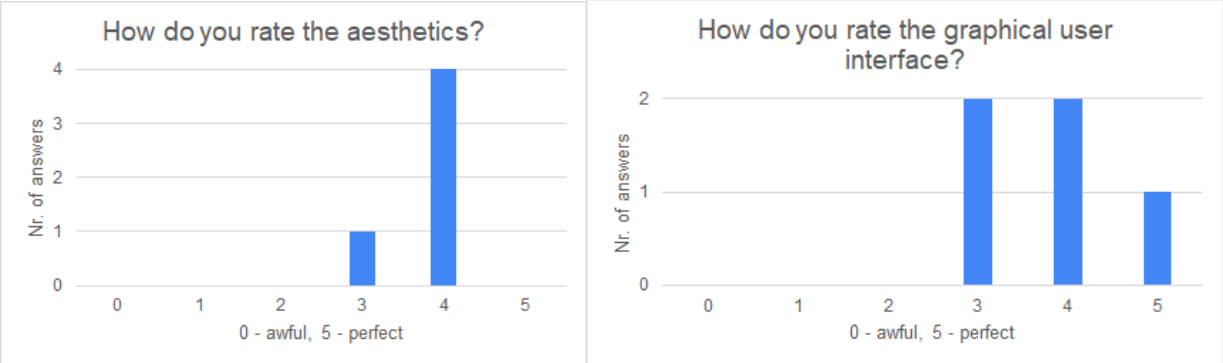

Figure 28. Aesthetics (left), graphical user interface (right).

"Can you name any other games you have played that use 2D/3D concept? What are your thoughts on their 2D/3D implementation?" Three testers claim to have played other 2D/3D games. This means that those testers had comparison when rating the implementation of the concept in *MiChef*. Those other games played are *Super Mario Odyssey*, *The Pedestrian*[37] (2020), *Fez* and *Little Big Planet*[38] (game series). Feedback shows that the participants think highly of those games. However, *The Pedestrian*, a seemingly 2D/3D game is considered to be a 2.5D game by the creators.

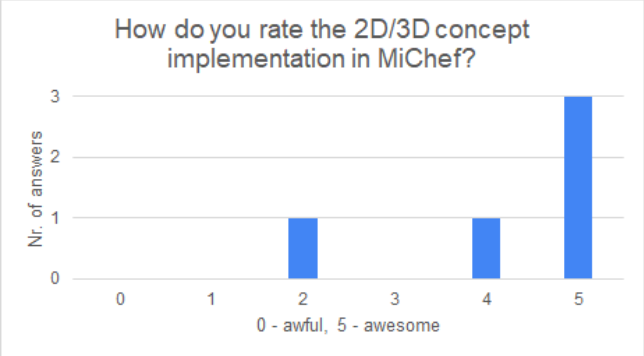

Figure 29. 2D/3D concept.

"How do you rate the 2D/3D concept implementation in MiChef?" Majority of the respondents feel that the 2D/3D concept in *MiChef* is clever and fun (see Figure 29). However, one tester is confused with the theme and does not see reason for the chef to be working in a decagonal prism. Another player commented that the game does not feel like it is 3D. The theme of a chef working in a ten-storey kitchen can be justified with mentioning that it is a huge restaurant or a kitchen-warehouse. This should have been added to the lore. It can be understood why some think that the game does not feel as 2D/3D as it should. This is probably because the building does not rotate sideways. The rotation mechanics should have been examined and discussed more thoroughly in early design stages.

[37] https://www.skookum-arts.com/
[38] https://littlebigplanet.fandom.com/wiki/LittleBigPlanet_(series)

“How could the 2D/3D concept be improved?” The participants gave wonderful feedback regarding the 2D/3D concept. There is a request to have the camera more zoomed in order to make the enemies not so easily visible. One tester answered that the game should look more like the picture (see Figure 30) to enhance the 2D/3D effect. Also, a player wrote that the theme should better justify usage of the concept. Last but not least, a valuable proposal was made stating that players should be given another axis to move for an even better realization of the concept. All of the offered ideas are great. Rotating the building or moving the camera around it sideways could have made the 2D/3D effect much more pleasing.

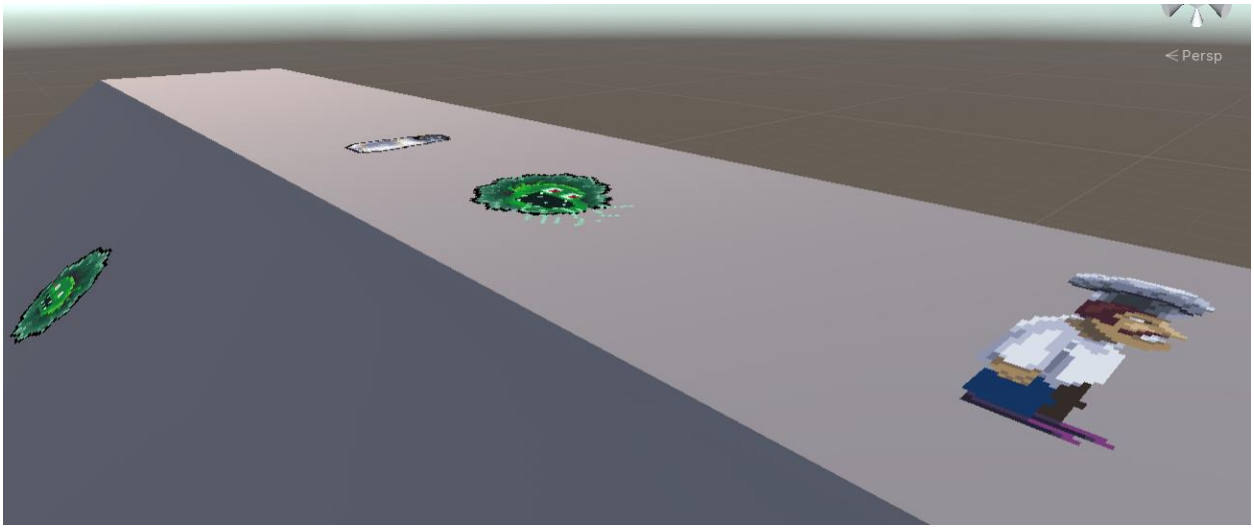

Figure 30. Picture in the questionnaire.

6 Conclusion

This thesis was about designing a minimal viable game with a unique 2D/3D mechanism. For that, four 2D/3D concept usages from some rather well-known platformers were discussed and compared to one another. Being inspired by those games, *MiChef* was designed and implemented accordingly.

A 2D/3D platformer game *MiChef* was developed as a result of this thesis. The game was designed to follow good software engineering and game development patterns. The project is filled with explanations behind design choices and development patterns in general, of which the most important was agility. The development had to keep up with the process of constantly emerging ideas. For example, this is how the theme was tailored to depict a recent world event that began in the second half of the development. The realization process is the most important part of the thesis. It is revealed how exactly *MiChef* shaped into the game that it is. In addition to that, shortcomings of the project are brought into light to be avoided in the future. Some of the shortcomings worth mentioning are the scarcity of features, conflicts between the game engine and scripting and lastly, too much need for manual tweaking of the game world.

Five usability test sessions were held for evaluating the game quality, getting feedback on user experience and the 2D/3D concept. It turned out that the participants are mostly satisfied with the result but indicate that there is still room for growth. One idea worth implementing as an improvement would be allowing further movement to enhance the 2D/3D concept. It is not certain if *MiChef* will be improved upon with user feedback taken into account. Nonetheless, the criticism is extremely useful for future projects in software development. Although the game is far from perfect, the author is content with the culmination considering his previous limited experience with game design and development.

References

[1] Klappenbach M. What is a Platform Game? 2019. https://www.lifewire.com/what-is-a-platform-game-812371 (08.05.2020)

[2] Hosch W. Electronic Platform Game. 2009. https://www.britannica.com/topic/electronic-platform-game (08.05.2020)

[3] Nodwin Gaming. The Evolution of Platform Games In 9 Steps. 2017. https://www.redbull.com/in-en/evolution-of-platformers (08.05.2020)

[4] Agile Alliance. Agile 101. https://www.agilealliance.org/agile101/ (08.05.2020)

[5] Schell. J. The Art of Game Design – A Book of Lenses. United States of America, Burlington: Morgan Kaufmann Publishers. 2008.

[6] Game Design Novice. Game Objects. 2012. http://gamedesign.wikidot.com/game-object (08.05.2020)

[7] Scattergood M., Feil J. Beginning Game Level Design. Thomson Course Technology. 2005.

[8] Thorn A. Game Development Principles. Cengage Learning PTR. 2013.

[9] Jonasson M., Purho P. Juice It or Lose It. 2012.

[10] Blow J. Game Development: Harder Thank You Think. Queue. 2004.

[11] Gamedesigning. The Top 10 Video Game Engines. 2020. https://www.gamedesigning.org/career/video-game-engines/ (08.05.2020)

[12] Hunt A., Thomas. D. Pragmatic Programmer, The: Form Journeyman to Master. Addison Wesley. 1999.

[13] Martin R. C., Martin M. Agile Principles, Patterns, and Practices in C#. Prentice Hall. 2006.

[14] McConnell S. Code Complete, Second Edition. United States of America, Washington: Microsoft Press. 2004.

[15] Nystrom R. Game Programming Patterns. Genever Benning. 2014.

[16] Nielsen J. Why You Only Need to Test with 5 Users. 2000. https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/(08.05.2020)

Appendix

I. Glossary

1. **Augmented reality**[39] – an enhanced version of reality created by the use of technology to overlay digital information on an image of something being viewed through a device
2. **Code smell**[40] **–** surface indication that usually corresponds to a deeper problem in the system
3. **Decagonal prism**[41] – ten-sided geometrical object with two identical ends and flat sides
4. **Game engine**[42] – software that provides developers with the necessary set of features to build games quickly and efficiently
5. **Hitpoint**[43] **–** amount of damage a character can withstand before it is defeated
6. **Particle effect**[44] – emitting vast number of tiny particles
7. **Pixel art design**[45] – art form where images are created and edited at the pixel level
8. **Playtesting**[46] **–** method of quality control by letting testers play a game to point out flaws
9. **Vignette**[47] – a filter that can be applied to images and videos

[39] https://www.merriam-webster.com/dictionary/augmented%20reality
[40] https://martinfowler.com/bliki/CodeSmell.html
[41] https://www.mathsisfun.com/definitions/prism.html
[42] https://unity3d.com/what-is-a-game-engine
[43] https://www.wordnik.com/words/hit%20point
[44] https://link.springer.com/chapter/10.1007%2F978-1-4302-3814-0_9
[45] https://www.techopedia.com/definition/8884/pixel-art
[46] https://www.techopedia.com/definition/27197/playtesting
[47] https://www.gamasutra.com/view/feature/3537/persuasive_games_videogame_.php?print=1

II. Launch Guide

System requirement:

- 16:9 display resolution
- Operating system: Windows

Follow these steps to launch the game:

1) Extract the accompanied files.
2) Open the folder “/Build”.
3) Run the file “MiChef.exe”

III. Project Repository

The assets and source code of *MiChef* can be found at https://gitlab.com/h.henrik97/michef.

IV. Accompanying Files

The archive file containing the accompanying files has the following structure:

- **/Build** – folder containing the game and its necessary files
- **/Source** – folder containing the source code and project files for the game
- **/Testing** – folder containing the questionnaire and responses

V. Test Sessions

Session 1: https://youtu.be/ZzG8Q_pjYXQ

Session 2: https://youtu.be/FUa9tiWb0mM

Session 3: https://youtu.be/4qXYRHa3a18

Session 4: https://youtu.be/Pj7hu3X85C0

Session 5: https://youtu.be/iBDndA72NLQ

VI. License

Non-exclusive licence to reproduce thesis and make thesis public

I, **Hans Henrik Viinalass**,
(*author's name*)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

MiChef – A 2D/3D Platformer Game,
(*title of thesis*)

supervised by **Raimond-Hendrik Tunnel**.
(*supervisor's name*)

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Hans Henrik Viinalass
08/05/2020