Game Programming Laboratory

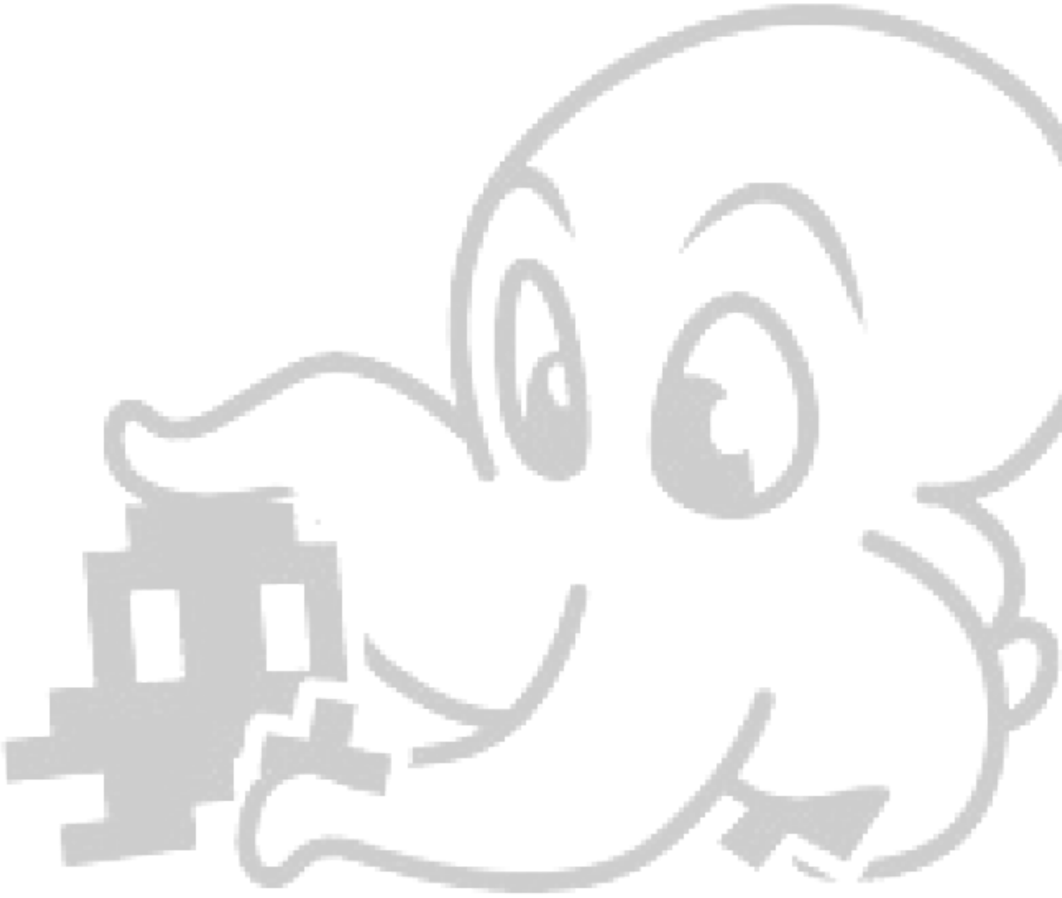
GRIMMETROPOLIS

„Attac & Protec“

Grimmetropolis

Manuel Serquet – Programmer, Designer, Visual Artist
Michael Giger – Producer, Programmer, Designer, Visual Artist
Tianyu Wu – Programmer, Designer, Visual Artist
Yelan Tao – Programmer, Designer, Visual Artist

Contents



Chapter 1. Formal Project Proposal

1.1. Game Description

1.1.1. Overview

The idea is a tower defense game with co-op multiplayer capabilities. Each player will control a character in a 3D top-down game world. The players will start at a castle which they have to protect. The players have to collect resources in order to construct buildings. Buildings have mainly two purposes: They help you collect more resources or defend against the enemies. The enemies will constantly come to attack the castle, buildings or the players. They will sometimes attack in smaller groups or in bigger waves. In the course of the game, the enemies become stronger and stronger. Therefore, you need to upgrade buildings or train your character. This also requires more resources which can be accessed when you expand around you castle. Eventually, you will lose. The premise of this game is to find the best strategy to defend against the enemy as long as possible.

1.1.2. Background Story

You are in the hands of some very powerful magical items. Protect them at all costs! With the power of the magical items, you can harvest the resources of the world and build a strong economy. But you are not alone! Enemies will come for you and the magical items! Work hand in hand in a team with Puss in Boots, Rapunzel, Cinderella and other famous fairy tale characters, each one of them with their own special quirks.

1.1.3. Design Decisions

Introduction

The game genre is mainly a tower defense game, but it also contains elements from a strategy and role play game. The visual style is kept simple with low poly art. The camera is looking from the sky down to the world like in a typical top down view in games like *The Legend of Zelda: Link's Awakening (2019).* The players start as a Grimm's fairy tale character in a world around a castle. The players can collect resources from the world and construct buildings. Buildings support the players at collecting resources or defending against enemies. The players themselves can attack the enemies as well. The goal is to defend the castle as long as possible. The game is over when the castle is completely looted.

World

The world is a finite map which is structured in square tiles. A tile can be clear or occupied by some natural structures. Natural structures include trees, big stones, rivers, hills or mountains. In the middle of the map is the castle. It will cover three by three tiles. Other buildings can occupy from a single tile to three by three tiles. As an initial step, only one map is planned. The reason for this is that people can compare their score with other players and optimize their strategy. On a later stage of the game development, new maps with interesting features can be played as well. Such features might be a castle surrounded by a river or at the border of the mountains which allow for new defense strategies during game play. A possible addition might be randomly generated maps. It might be difficult to create interesting maps by a random algorithm, but it is certainly considered as goal at a very late stage of game development.

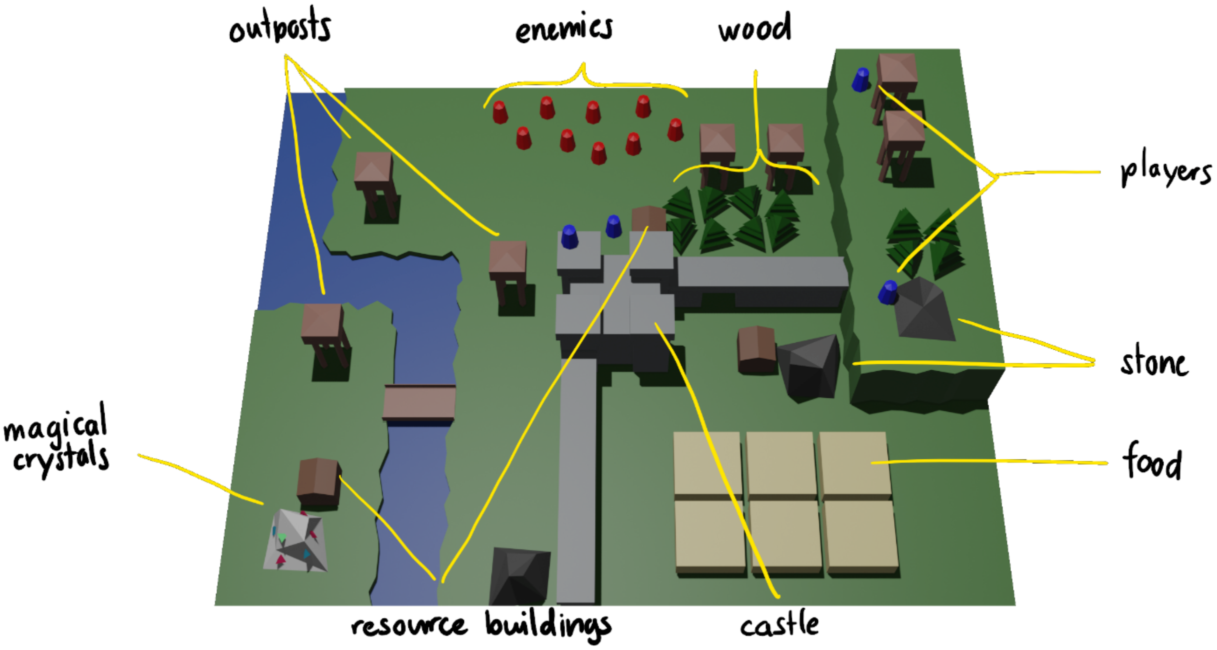


Figure 1: Truncated part of the game world. A possible base configuration during gameplay.

Player

The players can choose a character from the Grimm's fairy tales. When every player is ready, they will start in the world around the castle. A player is around half the size of a map tile. That means, two players can walk side by side when walking along a one tile wide path. A player can carry two items which he can use to do various tasks in the world map. Depending on the items the player holds, he can collect resources, construct buildings or fight enemies. The more a player does a certain task, the faster he becomes at doing it permanently. Therefore, a good team distributes the various tasks between the players. Every player has a unique skill which is dependent on the character he chose at the beginning of the game. Such unique skills might be for example pushing enemies away, jumping over stone walls or special magic abilities.

Resources

The map contains resources distributed in clusters all over the map. There are three common resources: wood, stone, and food. Wood can be collected from trees around the map and stone from stone quarries. These resources are used to construct buildings. They can be collected by the players or by buildings and are directly transferred to a common resource pool. The resources gradually regenerate after some time. This also means, a player cannot harvest infinitely much resources from a single deposit in a short time. The regeneration of a resource can be supported by buildings. Food can be harvested from farms, another type of building. It is required for certain types of buildings or as an upkeep. A rare resource is the magical crystal. It is only available further away from the castle. Therefore, it is not easy to harvest this resource. This resource is needed for special buildings or temporary buffs for buildings or players.

Buildings

Buildings can be categorized in two types of buildings: defensive and supporting buildings. The defensive buildings include a castle, outposts, stone walls and trenches. The castle and outposts attack close by enemies. Stone walls hinder enemies from walking freely around. Trenches offer more protection to the player from incoming fire. Supporting buildings include resource buildings, farms, hospitals, maintenance buildings, bridges and tunnels. Resource buildings give the player a passive resource income if build near resource deposits. Farms deliver food. Hospitals heal the players. Maintenance buildings repair other buildings or give them permanent buffs if they are close enough. These buildings all have a nonlinear upgrade tree. Bridges can be built over rivers to reach new areas of the map. Tunnels have the same use as the bridges but are used for mountains. The enemies will use these structures as well to reach the castle.

Items

In order for the players to do certain tasks, they need to equip themselves with the right items. The player can get their items at the castle and change it whenever they want to do so. At a later stage of the game development, an idea might be that the players start with some necessary tools but have to produce the other items at a special building. There are two types of items: weapons and tools. Weapons include swords, lances, shields, bows, crossbow or muskets. The weapons feel different to use and have different efficiency against certain types of enemies. Tools include items to harvest the resources: food requires a scythe, wood requires an axe, stone requires a hammer, magical crystals require a pickaxe. Buildings can be built with a trowel.

Enemies

The enemies will attack from any direction if the map layout does not hinder it. They usually attack in waves. Between the waves, the player has time to collect resources and construct buildings. The enemies will attack between the waves as well, but only in very small groups. There are going to be different types of enemies available: close combat fighters (sword fighter, lance fighter, beasts), ranged units (archer, wizard) and siege units (catapult, trebuchet, giants), maybe even flying units (crows). Depending on the base the players have built or the map layout, the enemies should attack cleverly. As an example, if the player is completely protected by stone walls, the enemies will first try to breach the wall with siege units before attacking with all their close combat units. After defeating enemy units, they will drop resources. An idea is also to introduce boss enemies which will come at certain waves. They will drop special items which the player can equip.

Tutorial

Whenever the players start a new game, another character spawns around the castle. He will talk about the first steps to build up a working economy and how to build a defense against the enemies. The player can skip the tutorial by killing him. He will drop some resources for the players to collect.

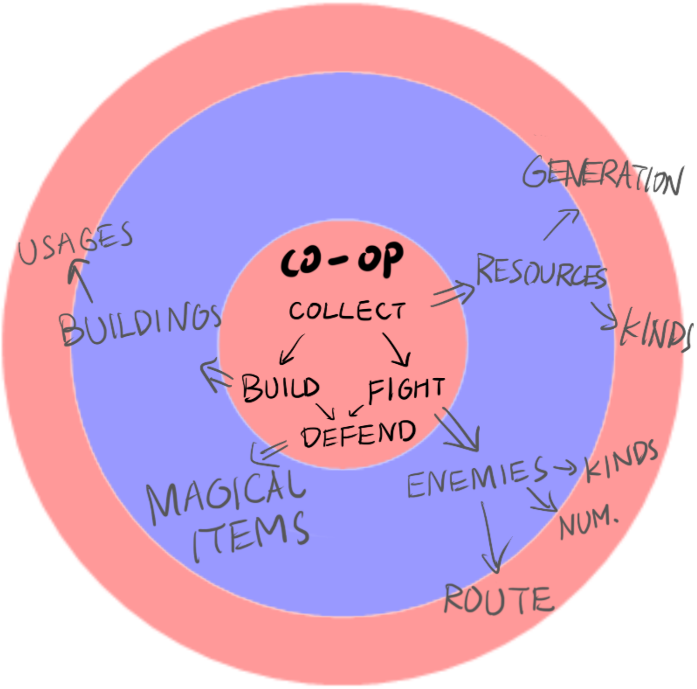
1.2. ‚Big Idea‘ Bullseye

Core Idea

Players cooperate with each other and defend the castle as long as possible by fighting against the enemies, collecting common and rare resources, constructing and upgrading buildings and improving skills.

Technical Innovation

Each player has his own special ability depending on the character from the Grimm's Fairy Tale. Different tools can be used to harvest resources and constructing buildings. The players need to expand to get more resources. An algorithm needs to be developed for enemy attacking routes.



1.3. Technical Achievement

In the tower defense genre, usually the enemies either take a predefined path or the shortest path to the base while ignoring towers. This leads to the incentive of placing walls to create artificial choke points and build defenses around them such that the enemies stay under fire as long as possible and run to their death.

The enemies however should act in an intelligent way using their knowledge of the map. They should avoid well defended choke points if they can simply take a longer path. Furthermore, they will retaliate against towers and try to breach walls where they are less defended.

1.4. Development Schedule

1.4.1. Layered Task Breakdown

1.4.1.1. Functional Minimum

- A single character can play the game.
- The map contains no natural structures apart from the trees.
- The only resource available is wood.
- The character can collect resources, build outposts and fight enemies without any items.
- A single type of close combat enemy will attack the castle by going straight to it.

1.4.1.2. Low Target

- Multiple people can play the game at the same time. The camera will zoom out so that all players are always visible.
- The map now also has more natural structures like big stones or rivers.
- Stone can be now collected as an additional resource.
- New buildings like resource collecting buildings and stone walls should be available.
- The outposts should have a simple upgrade tree available.
- Close combat weapons and tools have to be used by the players now.
- A single type of ranged unit and siege unit will appear in the waves.
- The enemies attack with more intelligence: If they are attacked by something, they counterattack.
- A simple game menu to start with.
- Add some music to the game.

1.4.1.3. Desired Target

- The player has some basic skills which they can improve.
- The players are unique characters with a special skill.
- The map contains now also hills and mountains.
- Magical crystals can now be collected.
- At least two enemies of each type except the flying type should appear in waves.
- Further buildings like the trenches, farms, hospitals, maintenance buildings and bridges should be available.
- All added buildings should have a unique upgrade tree.
- The player should also have access to ranged weapons like a bow.
- The tutorial character exists in the world map.
- The players have the possibility to pause the game.
- The game over screen should appear when the players lose.
- Creation of the trailer.

1.4.1.4. High Target

- All previously mentioned enemy types should be available
- Boss enemies are introduced which attack at certain waves.
- Boss enemies will drop very strong and unique items which the players can equip.
- All previously mentioned weapons should be now available.
- More maps are created which should offer very interesting strategies.

1.4.1.5. Extras

- Each player has their own camera view when the camera must zoom too far away to get all players on one screen.
- Map generation should deliver always new experiences while the playing the game.
- The game can be played online with other players.

1.4.2. Task List

In the following two pages, a list of all tasks can be seen. They are listed with the responsible person and the initial planned duration for this task.

1	**FUNCTIONAL MINIMUM**		
1.1	Inital engine setup	Michael, Manuel	24h
1.2	Create map consisting of tiles	Yelan, Tianyu	16h
1.3	Player control	Manuel	16h
1.4	Assets (player, castle, enemy, tree) for the functional minimum	Yelan, Tianyu	16h
1.5	Player can collect wood as a resource	Yelan	32h
1.6	Fighting system with the player	Michael	24h
1.6	Very simple AI pathing aiming for the castle	Manuel	32h

2	**LOW TARGET**		
2.1	Player can build an outpost	Michael	24h
2.2	Implement multiple players	Manuel	24h
2.3	Adding new map features like big stones or rivers	Tianyu	32h
2.4	Add new stone resource	Yelan	16h
2.5	Asset creation for buildings (stone wall)	Yelan	16h
2.6	Asset creation for enemies	Yelan	40h
2.7	Asset for items (5 items)	Tianyu	32h
2.8	Asset for UI	Tianyu	16h
2.9	Adding new resource buildings	Michael	32h
2.10	Adding stone walls	Michael	4h
2.11	Adding UI for building's upgrade tree	Michael	32h
2.12	The player can take items with themselves	Manuel	16h
2.13	Adding items like swords and their behaviour	Manuel	16h

2.14	Adding tools	Manuel	16h
2.15	Add a ranged and siege enemy unit	Michael	32h
2.16	Enhancing enemy AI (counterattack)	Manuel	48h
2.17	Adding a simple game menu	Michael	32h
2.18	Testing out the game	Everyone	32h
2.19	Balancing the game	Everyone	32h
2.20	Add music to the game	Tianyu, Yelan	8h

3 DESIRED TARGET

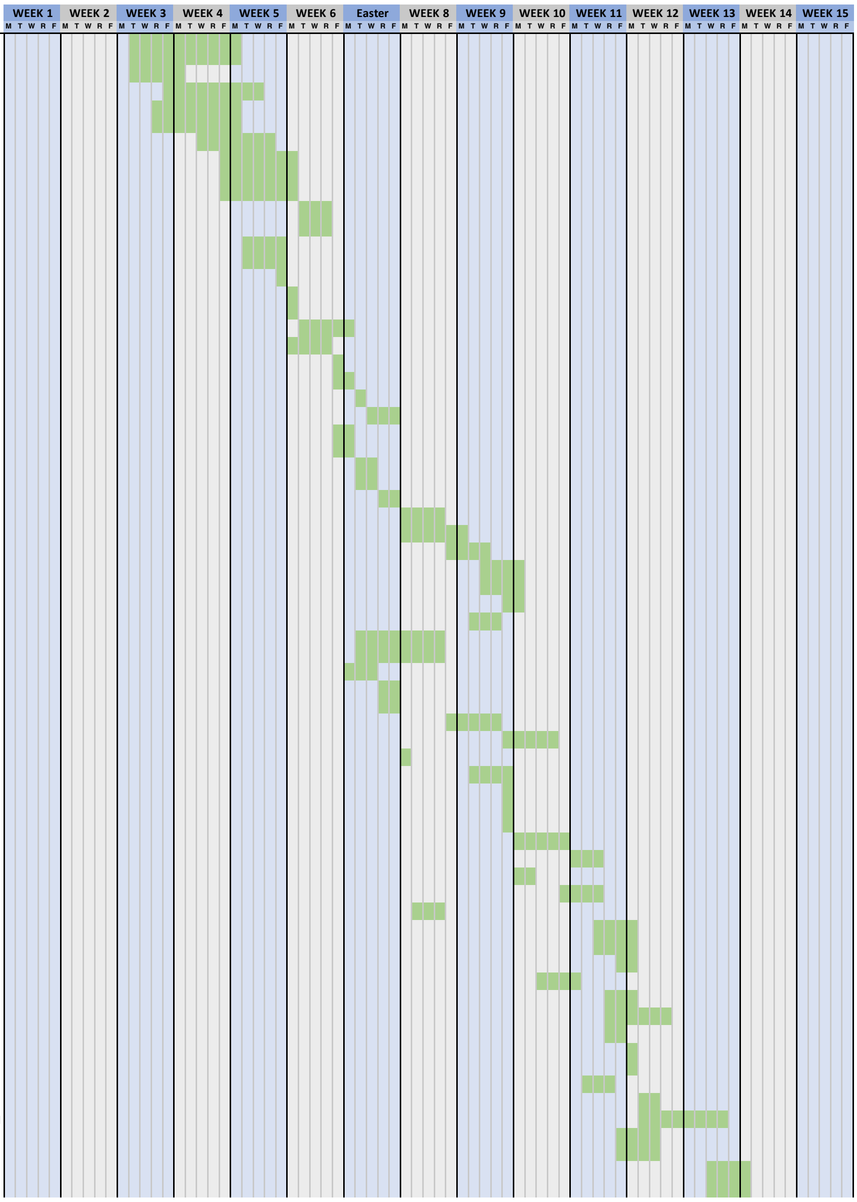
3.1	Players have a basic skill	Michael	24h
3.2	Assets and implementation of unique characters	Michael, Yelan	64h
3.3	Map contains hills and mountains	Tianyu	24h
3.4	Adding magical crystal as a new resource	Tianyu	12h
3.5	Asset creation for buildings	Michael, Yelan	40h
3.6	Asset creation for enemies	Yelan	40h
3.7	Asset for bow	Tianyu	16h
3.8	Add more enemy types (3 enemies)	Manuel	32h
3.9	Add more buildings: trenches, farms, hospitals, maintenance buildings and bridges	Michael	16h
3.10	Add functionality to buildings	Michael	40h
3.11	Add upgrade tree to buildings	Michael	24h
3.12	Adding a bow for the player	Manuel	32h
3.13	Adding pause menu	Michael, Yelan	32h
3.14	Adding a game over screen	Michael, Tianyu	24h
3.15	Testing out the game	Everyone	32h
3.16	Balancing the game	Everyone	32h
3.17	Create a trailer	Everyone	16h

4 HIGH TARGET

4.1	Add all enemy types (4 more enemies)	Manuel	32h
4.2	Asset for additional items	Tianyu	24h
4.3	Asset creation for boss and enemies	Yelan	48h
4.4	Add a boss enemy	Michael	16h
4.5	Add special items dropped by the boss enemy	Michael	8h
4.6	Adding different kinds of weapons	Manuel	24h
4.7	Add more maps	Tianyu	16h
4.8	Improve intelligence of the AI	Michael, Manuel	64h
4.9	Add some cool information to the game over screen	Manuel	32h
4.10	Testing out the game	Everyone	32h
4.11	Balancing the game	Everyone	32h

1.4.3. Timeline

1.1	Inital engine setup	Michael, Manuel
1.2	Create map consisting of tiles	Yelan, Tianyu
1.3	Player control	Manuel
1.4	Assets (player, castle, enemy, tree) for the functional minimum	Yelan, Tianyu
1.5	Player can collect wood as a resource	Yelan
1.6	Fighting system with the player	Michael
1.7	Very simple AI pathing aiming for the castle	Manuel
2.1	Player can build an outpost	Michael
2.2	Implement multiple players	Manuel
2.3	Adding new map features like big stones or rivers	Tianyu
2.4	Add new stone resource	Yelan
2.5	Asset creation for buildings (stone wall)	Yelan
2.6	Asset creation for enemies	Yelan
2.7	Asset for items (5 items)	Tianyu
2.8	Asset for UI	Tianyu
2.9	Adding new resource buildings	Michael
2.10	Adding stone walls	Michael
2.11	Adding UI for building's upgrade tree	Michael
2.12	The player can take items with themselves	Manuel
2.13	Adding items like swords and their behaviour	Manuel
2.14	Adding tools	Manuel
2.15	Add a ranged and siege enemy unit	Michael
	Enhancing enemy AI (counter attack)	Manuel
2.17	Adding a simple game menu	Michael
2.18	Testing out the game	Everyone
2.19	Balancing the game	Everyone
2.20	Add music to the game	Yelan, Tianyu
3.1	Players have a basic skill	Michael
3.2	Assets and implementation of unique characters	Michael, Yelan
3.3	Map contains hills and mountains	Tianyu
3.4	Adding magical crystal as a new resource	Tianyu
3.5	Asset creation for buildings	Michael, Yelan
3.6	Asset creation for enemies	Yelan
3.7	Asset for bow	Tianyu
3.8	Add more enemies types (3 enemies)	Manuel
3.9	Add more buildings: trenches, farms, hospitals, maintance buildings and bridges	Michael
3.10	Add functionality to buildings	Michael
3.11	Add upgrade tree to buildings	Michael
3.12	Adding a bow for the player	Manuel
3.13	Adding pause menu	Michael, Yelan
3.14	Adding a game over screen	Michael, Tianyu
3.15	Testing out the game	Everyone
3.16	Balancing the game	Everyone
3.17	Create a trailer	Everyone
4.1	Add all enemy types (4 more enemies)	Manuel
4.2	Asset for additional items	Tianyu
4.3	Asset creation for boss and enemies	Yelan
4.4	Add a boss enemy	Michael
4.5	Add special items dropped by the boss enemy	Michael
4.6	Adding different kinds of weapons	Manuel
4.7	Add more maps	Tianyu
4.8	Improve intelligence of the AI	Michael, Manuel
4.9	Add some cool information to the game over screen	Manuel
4.10	Testing out the game	Everyone
4.11	Balancing the game	Everyone

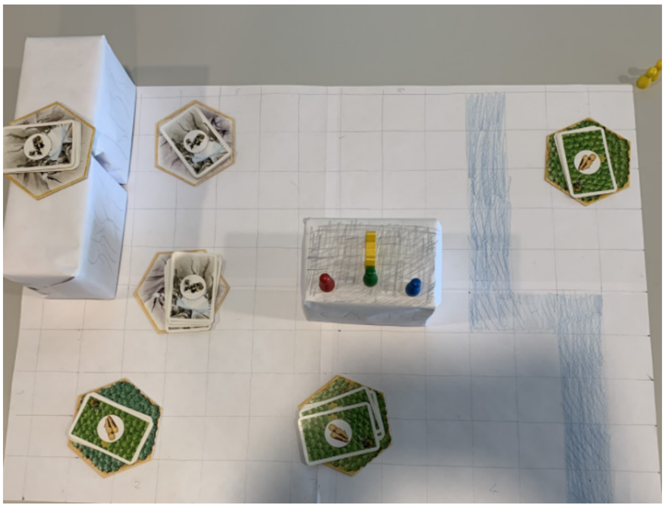


1.5. Assessment

Compared to conventional tower defence games, our game offers more varieties with the add-on RTS elements and multi-player collaborations. Players are free to choose different characters inspired by the Grimm's fairy tales with their unique abilities and fight together to protect the castle in a fantasy world. During the game, players have a lot of freedom in what they do. Therefore, the players can develop various strategies to fight against the enemies. The enemies of the games are designed to be more intelligent who can also make use of their knowledge of the map to avoid choke points, unlike taking a predefined or shortest path to the base in traditional tower defence games. This innovation adds more fun to the game, making the game experience more thrilling and challenging. The collaborative setting of the game makes our game a perfect fit for a multiplayer couch game with friends and may be of particular interest to strategy game addicts ranging from young teenagers to adults, seeking challenging and off-centre gameplay.

Chapter 2. Prototype

2.1. Prototype Setup



The board game consists of a map divided into a grid. In the middle of the map, a castle serves as the starting point of the players (red, green, blue). Besides the players, the castle also contains a magical artifact (yellow). On the top left side of the map, mountains can be seen. On the right-hand side, a river (blue) covers the whole right side. On the map, wood (green two by two tiles) and stone (grey two by two tiles) resource deposits are distributed.

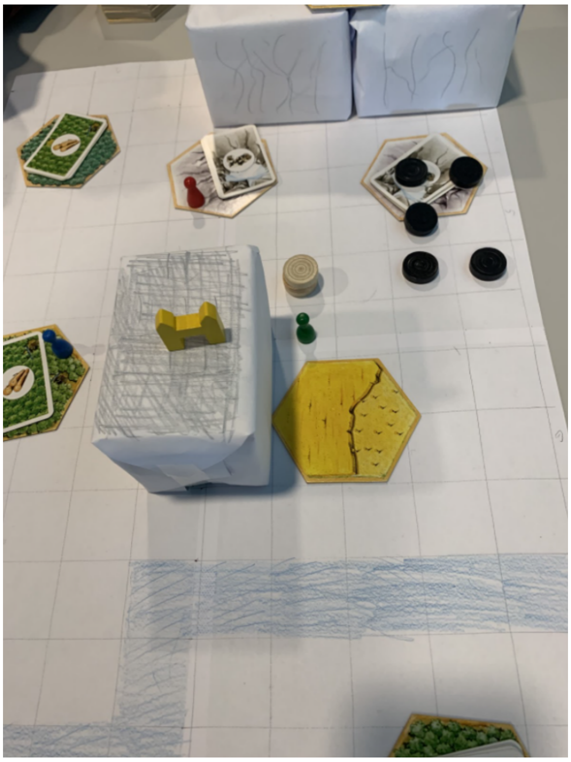
The board game has a turn-based game mechanic. The goal of the players is to defend the magical artifact from incoming enemies. The enemies will try to steal the magical artifact and escape the map with the magical artifact in their possession.

The players have various abilities in order to defend against the enemies. Players can walk two grid units per turn (not diagonally). All players can discuss their next move with each other and can play in an arbitrary sequence. The players can collect a single resource card if they stand on top of a resource deposit which will end their turn. Players can also attack enemies which stand right next to them (not diagonally) which will also end their turn. Players cannot walk over rivers, mountains, enemies and some buildings (stone walls, defence towers). They cannot end their turn on a grid cell where a player already stands. A player can be attacked three times by the enemy. After that, he cannot play anymore.

The players can also construct buildings. This action also ends the player's turn. There are various buildings: bridges cost two wood resource cards, farms cost two wood resource cards, stone walls cost two stone resource cards and defence towers cost one stone and one wood resource card. Bridges can be built over rivers to make new lands accessible. Farms are two by two buildings which produce a single food resource card every turn automatically. Stone walls serve as an obstacle for advancing enemies. Defence towers can shoot down an enemy once per turn if the enemy has a Manhattan distance of two grid units from the defence tower. They cost one food resource card per turn. If they do not get food, all towers will have a reduced shooting range of one grid unit. Stone walls and defence towers are non-passable buildings for players and enemies. Each building has three health points and can be destroyed by the enemy. Buildings can be repaired anytime (even when once destroyed) with a single wood resource. The player will have to stand next to the damaged building and end their move to do so.

Resources collected from the resource deposits or farms are stored in a resource pool which is always shared between all the players. Each resource deposit has five resource cards. The resource deposits will restore their resource cards every second turn until it has five resource cards again. This way, a player cannot overuse a resource deposit.

The enemies will come in waves. Every fifth turn, the enemies will start their wave at a random location at the border of the map. In the first wave, five enemies will spawn. In every consecutive wave, the number of enemies increases by half of the amount from last round. This way, waves become more and more difficult. Enemies can move only one grid unit per turn. If they want to attack, they must stand right next to the building or player (not diagonally). This will also end their turn.



Here is an exemplary moment in the board game: The green player has built a farm (yellow, below the green player) and a defence tower (white, above the green player). The enemies (black) are advancing from right to left. The defence tower will be able to shoot down the lower left enemy of the advancing wave. The blue and red players are collecting resources.

2.2. Playing Experience

Overall, the playing experience of the game is challenging yet interesting. Although simulating such a complex interaction in a turn-based game mechanic is a bit tiring and time-consuming process, the game turned out to be playable with only some slight adjustments. During the process, we realize that the balancing of the game is crucial and need to be fine-tuned in the implementation. Because the map we created for the physical prototype is relatively small, the enemies came with a large amount can have more advantages, making it very difficult to defend. We are also quite happy to find out that the collaboration and coordination between players are the key to defence longer towards the enemies, which is exactly the aim of our game. As the resources are distributed in the entire map and enemies can come from any direction, players need to actively communicate with each other to work together and find the best strategy to fight against the enemies. This makes the game challenging and not boring at all playing with friends.

2.3. Findings and Conclusion

The main takeaway from the prototype is that properly balancing the game is a challenging but important aspect of the game. One of the larger balancing issues was that the scale of the physical prototype was too small. The map itself needs to be bigger while both the players and the enemies need to move faster around the map. The towers should also cover a larger portion of the map. Another balancing issue was the food production. Due to the somewhat larger size of the fields and the small map, we quickly ran out of farm space to sustain a larger number of towers, a one to one ratio of fields and towers proved to be too tight. The fields either need to be smaller or provide more food per field. However, we need to keep in mind that the map will be bigger at the same time. A third issue was that the time between two consecutive waves was too short. There was barely any time left after defeating a wave to build defences for the next one.

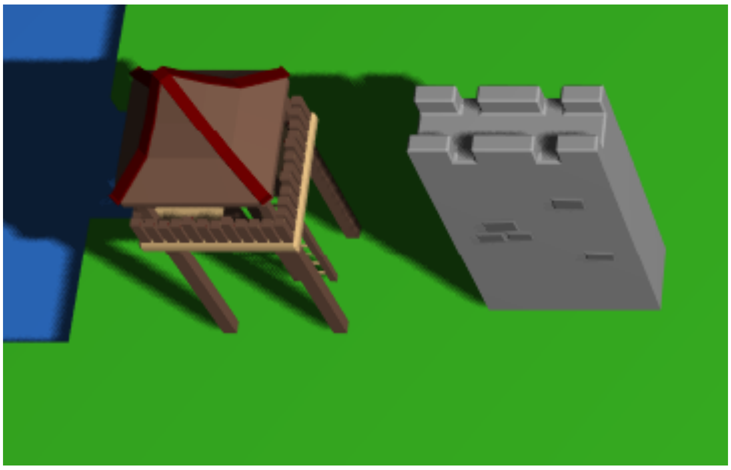
Chapter 3. Interim Report

(Max 5 pages)

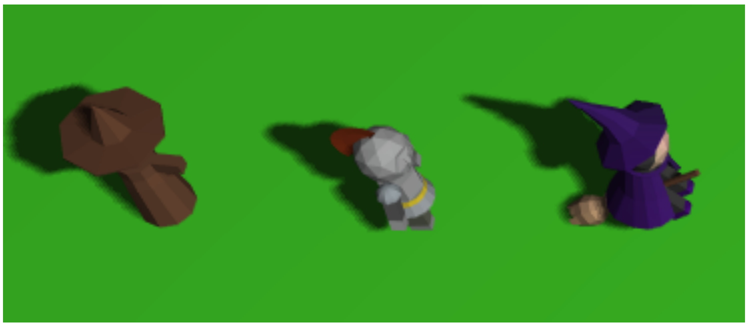
3.1. Progress

The game can be played with 4 players. Each player can take various items in their hands. Currently, swords, pickaxes and axes are available to players. Depending on the item, they can be used to collect certain resources or attack enemies.

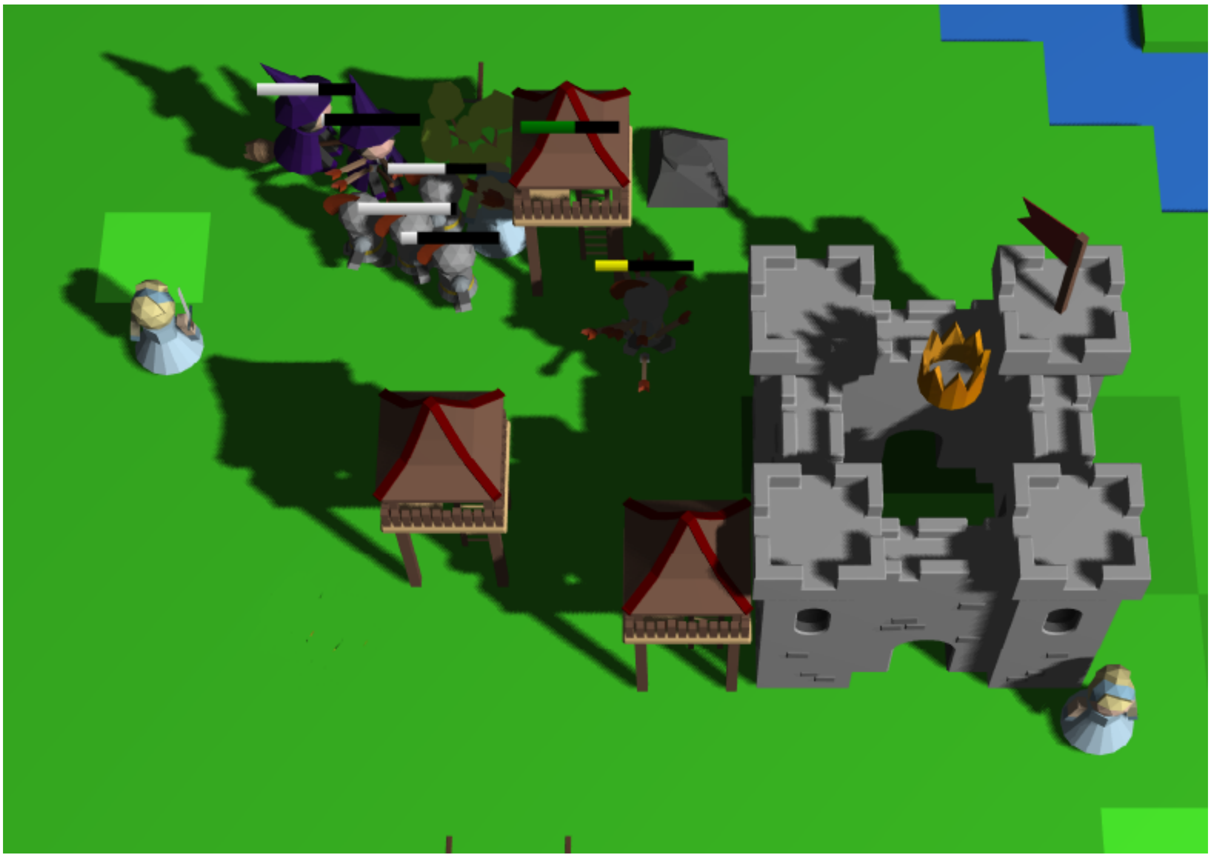
There are currently two types of buildings which can be constructed: walls and outposts. Walls have the ability to block the way of enemies with their high health. Outposts have a bit less health but can shoot projectiles at enemies.



There are three types of enemies: the witch which attacks from afar, the knight who directly attacks the building and the catapult (not with the right model yet) which can outrange outposts. Enemies try to find the shortest path to the castle which is located in the middle of the map. Depending on the location of the outposts, they try to circumvent them if there are too many of them. Because catapults can outrange outposts, they generally attack outposts first.



Enemies spawn in waves. After every wave, the amount of enemies increases slightly. This leads to a game which already requires certain strategies in order to survive and protect the castle.



3.2. Challenges

Overall, our game is on track and we manage to stick to our development schedule. So far, we have not encountered big challenges which are not doable. However, we do notice that the balancing of the game is crucial to our game. During our initial test, we realized that the enemies were too powerful to defeat. Therefore, we tried balancing the game by tuning different configuration parameters in the following test rounds. The process could take a lot of time, indicating that we should reserve more time for the final tuning. In addition to game configurations, we also revised the keyboard mapping for functional keys to ensure better playing experience with laptops.

3.3. Future Work

Menu

We will add a menu screen which allows the players to get familiar with the rules and keyboard commands before starting to play.

Magical artifact feature

The magical artifact to be protected needs more details. The enemies will try to steal the magical artifact and escape the map with the magical artifact in their possession.

Special abilities and special characters

We will design and model more fairy tale characters, each having its own special abilities dependent on the character. (For example, jumping over barricades, higher moving speed, magic abilities, etc.)

More enemies

We will design and model more enemy types, e.g. a boss enemy who is harder to kill or can cause bigger damages to the buildings.

Outpost upgrades

The outposts can be upgraded in terms of damage, area damage, attacking priority and attacking range.

Resources

We plan to introduce the farms feature and add possible resource depletion. In other words, the towers need upkeep and will no longer be functional when there is not enough food resource.

Buildings

We will design and model more types of buildings. Such as farms for food, resource buildings for faster resource collection and resource regeneration, bridges for river crossing etc.

Increase map size

The current map is very limited and players can easily reach its boundary. We will increase the map size and introduce more landscapes.

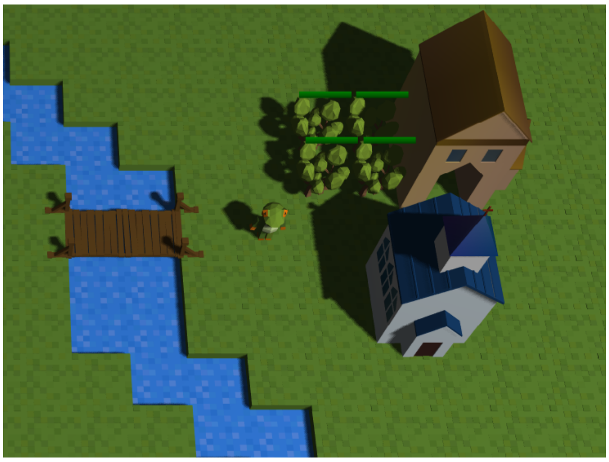
Music

We will find suitable music and sound effects for our game.

Chapter 4. Alpha Release

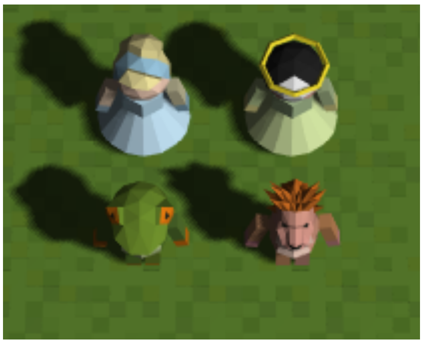
4.1. Progress

There are several fundamental changes to the game. New buildings are introduced. The first new building is the bridge which can be built over waters. This will make paths to certain resource deposits shorter, but will also enable the enemy to come to the castle quicker. Another building is the hospital (or church). Hospitals heal player in close proximity. Then, there is the resource building which automatically collects resources close by very slowly.



In order to balance the game and to make players go further away from the castle, there are much less resource deposits close by the castle. Resource deposits also can be depleted when the players harvest too many resources. They will regenerate resources after some time. Also, the walls are almost never used. Therefore, they are cheaper and quicker to build.

There are new character models. Finally, players can be easily differentiated. They do not have their special ability yet:



Enemies have an improved behaviour. They try to act in groups. They steal the crown from the castle as soon as they can and try to escape the map from where they initially spawned. The players can kill the enemy carrying the crown and get it back to the castle. If the enemy reaches the border of the map, it should be be “Game Over”. This is not yet implemented.

4.2. PROGRESS

@Note: Explain what has proved to be harder (or easier) than expected. What design revisions have you made to your game as a result of what you've learned with the implementation?

Discuss the implementation challenges you faced. Were there aspects that you wanted to build but were unable to do so?

Implementing an AI that appears to be smart to the player proved to be harder than expected. Modelling the map as a graph where the edges represent different actions, and weighting said edges not only according to distance, but also according to how beneficial a move is, allows to efficiently find the optimal path to a location. While this approach allows to easily add individual behaviour, e.g. avoiding outposts, it doesn't help with determining what the ideal target location is for the enemies and doesn't take into account what other enemies are doing at the same time.

Coming up with an efficient way to determine the best target location proved to be harder than expected, as there are a large amount of potential scenarios for which our algorithm needs to find a good solution. In the end we decided to partition the enemies in groups and use a couple of simple rules to make the enemies appear smarter.

We also planned to make the enemies move in a formation until they arrive at a player building. However, a seemingly simple task like moving in a formation proved to be harder than expected, as there are several hidden implications. What should happen if the used path isn't wide enough? At what speed should the formation move and rotate such that everyone can keep up? How do we handle obstacles in the way of the formation? Since the feature itself was only of visual nature, we decided to drop the enemies formation for now.

Another challenging part is writing efficient code. While computing the shortest path on a graph can be done efficiently, doing it every update cycle for every enemy quickly leads to performance issues as the amount of enemies increases. We still need to come up with an elaborate way to mitigate this issue.

4.3. PROGRESS

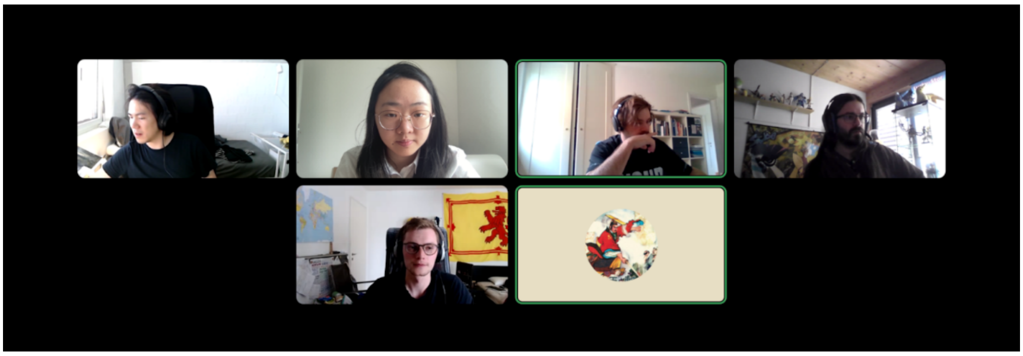
1. **Menu**
   We will add a menu screen which allows the players to get familiar with the rules and keyboard commands before starting to play.
2. **Special abilities and special characters**
   We have designed more types of characters, and we will add special abilities to each of them. (For example, Cinderella can make enemies friends, Snow White can attract enemies, Beast can have a huge attack boost, Frog can jump over buildings, etc.) In order to avoid Disney IP issues, we will try to design and model our characters according to the description in the original Grimm's fairy tale.
3. **Outpost upgrades**
   The outposts can be upgraded in terms of damage, area damage, attacking priority and attacking range.
4. **Music**
   We will keep looking for music and sound effects for our game.
5. **Fix bugs**
   We need to fix some bugs, such as players shouldn't be able to pick up the crown.

Chapter 5. Playtest

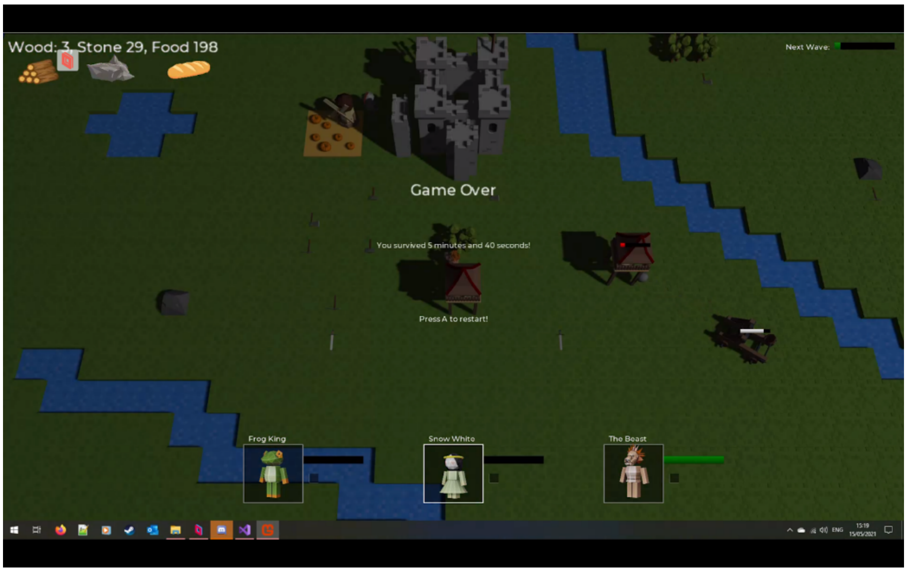
(Max 5 pages)

5.1. Playtesting Session

For the playtesting, we decided to ask several of our friends to test our game. We then organized two playtesting sessions, one with four players and another one with two players, and let them play the game over parsec and discord without playing ourselves.



For the playtesting itself, we first showed the players a cheat sheet displaying the controls, then let the players play our game without telling them anything else. We decided to tell the players the controls in advance since we planned to have them explained ingame too. We took notes of what the players did while playing. After the first playthrough, we asked them if they were confused about anything and made sure they understood the controls of the game.



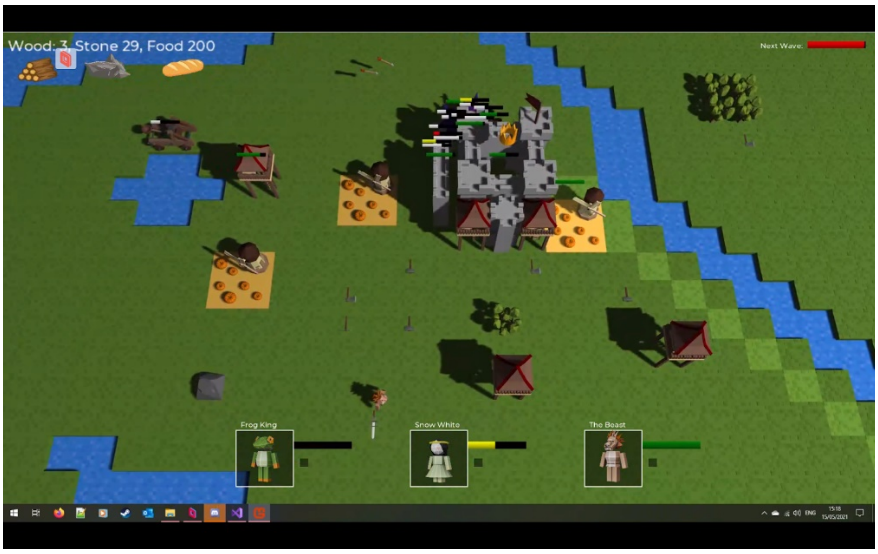
Wood: 3, Stone 29, Food 198

Game Over

You survived 5 minutes and 40 seconds!

Press A to restart!

We then let them play a couple of rounds while continuing to take notes. After those rounds we then had a discussion with the players about the game and asked them additional questions.



5.2. Questions and Comments

How was your experience?

The overall look of the game is kind of boring. With the continuous progress of building new structures, the map becomes more and more interesting though. The game was hard to play, because the UI was not delivering enough information or did not notify the player when certain things happened in the game. A typical problem was also that tool were misplaced behind other buildings. This made it very difficult to find them in a later stage of the game. There should be a border which surrounds the map instead of having a black background.

Some play testers expected a wave counter and also that waves would only start when the previous wave is done (similarly to CoD Zombie).

How were the controls?

For players with controllers, everything seemed fine. A small complaint is that the build menu should be opened with the button X or Y instead of RB. Keyboard players had a harder time. The biggest issue was that players had problems to look into the right direction when placing a building. Also, switching between the buildings in the build menu with the keyboard keys Q and E seems to be not intuitive.

How was the difficulty?

Playing this game with two players is very difficult. It requires the players to plan ahead. Otherwise, they will lose the game very quickly. Even with the initial resource bonus for two players, it did not balance well enough. For three players, the game was very well balanced. It was very interesting to find out new strategies and to survive longer and longer with each new playthrough.

Did you encounter problems?

The main issue was that there is no good explanation on how to start the game. A tutorial was required. Other problems were (as already mentioned) that the tools could be misplaced behind some buildings. Additionally, people did not know what some of the buildings do. Another issue was that players tried to run away from the enemy hordes with the crown as they did not do anything against the player. This way, the enemies could never still the crown.

What did you like about the game?

The idea of enemies trying to steal crown and get away with it seems to be an interesting idea. Additionally, the cooperative gameplay aspect is very interesting. People have to talk with each other to develop new strategies to defend against the hordes of enemies.

What did you not like about the game?

Generally, the UI was to weak to notify the players when something important happened. This includes placing tools behind buildings, players taking damage and the crown being stolen. Additionally, after a certain point in the game (if the map is almost completely filled with buildings), there is nothing special to do anymore.

5.3. Design Revisions

After the playtesting we decided on several design revisions to our game which we plan to add in the future. Most of them concern the user interface and feedback we provide to the player.

First we decided to add a small timer before starting and restarting the game. It happened a couple times that the game was accidentally started before all players choose their character. To avoid this we decided to add a small timer with a countdown and allow players to cancel said countdown. The same will be added before the game restarts after the game over screen.

Another set of revisions are about telling the players relevant game information. We decided to add a small tutorial to explain the basic gameplay. Furthermore we plan to add warnings for different in-game events (start of a wave, castle is attacked, player is attacked, crown is being stolen), as players often failed to notice those and were surprised when they suddenly died. Additionally we plan to add a visual clue for the effective radius of different buildings, as right now it is unclear what resources are automatically collected and how close a player has to be to the hospital to get healed. The menu overlay at the top of the screen also needs to be updated such that it contrasts with the game scene more as the players didn’t notice it. The resource upkeep will also be added to the menu overlay.

Finally there are also a couple of small revisions to the gameplay itself. Players tried to drop items on top of another to swap them and decided to implement said feature. We are also planning to add a way to revive players and for players to have a sprint/dash ability. We will also have to review the button mapping for the build menu, as players tried to open it by holding the button pressed.

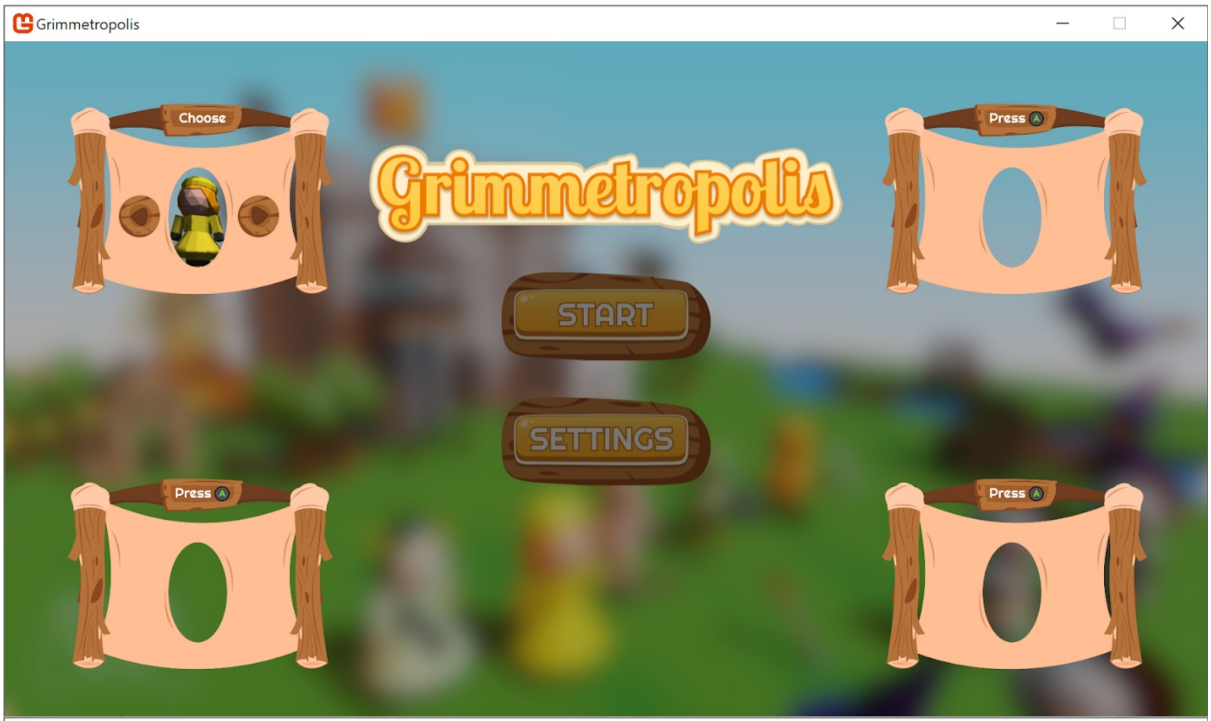
Chapter 6. Conclusion

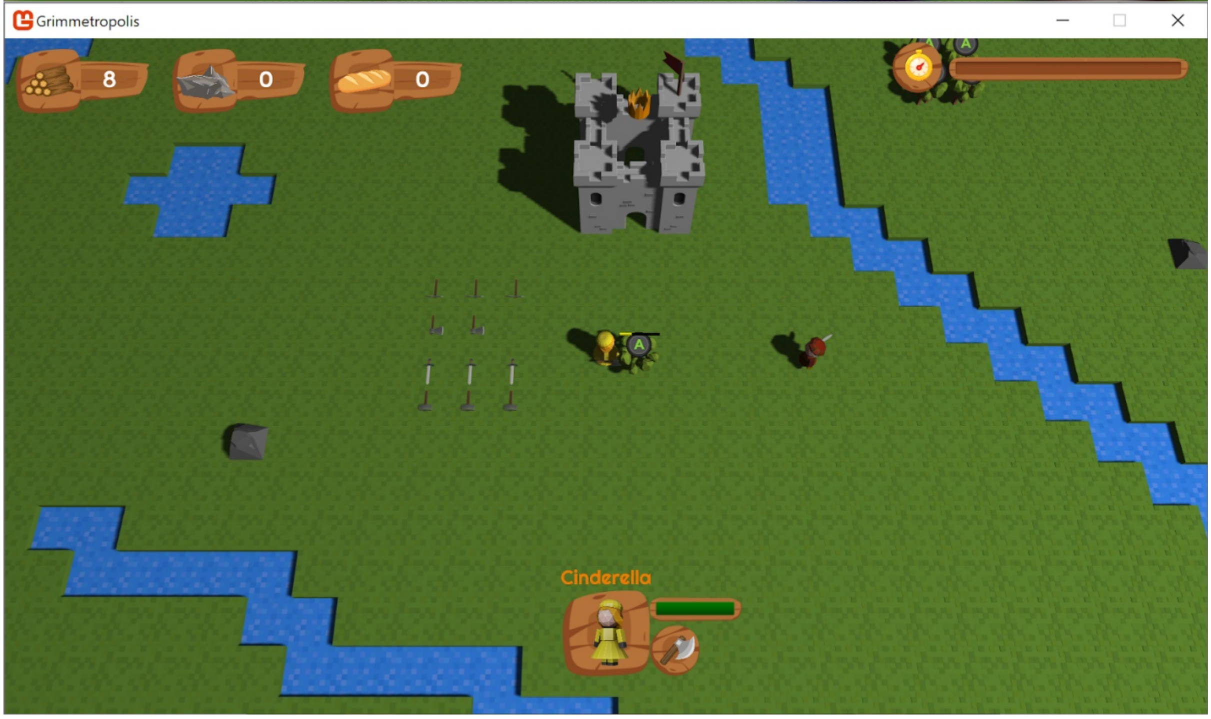
(Max 5 pages)

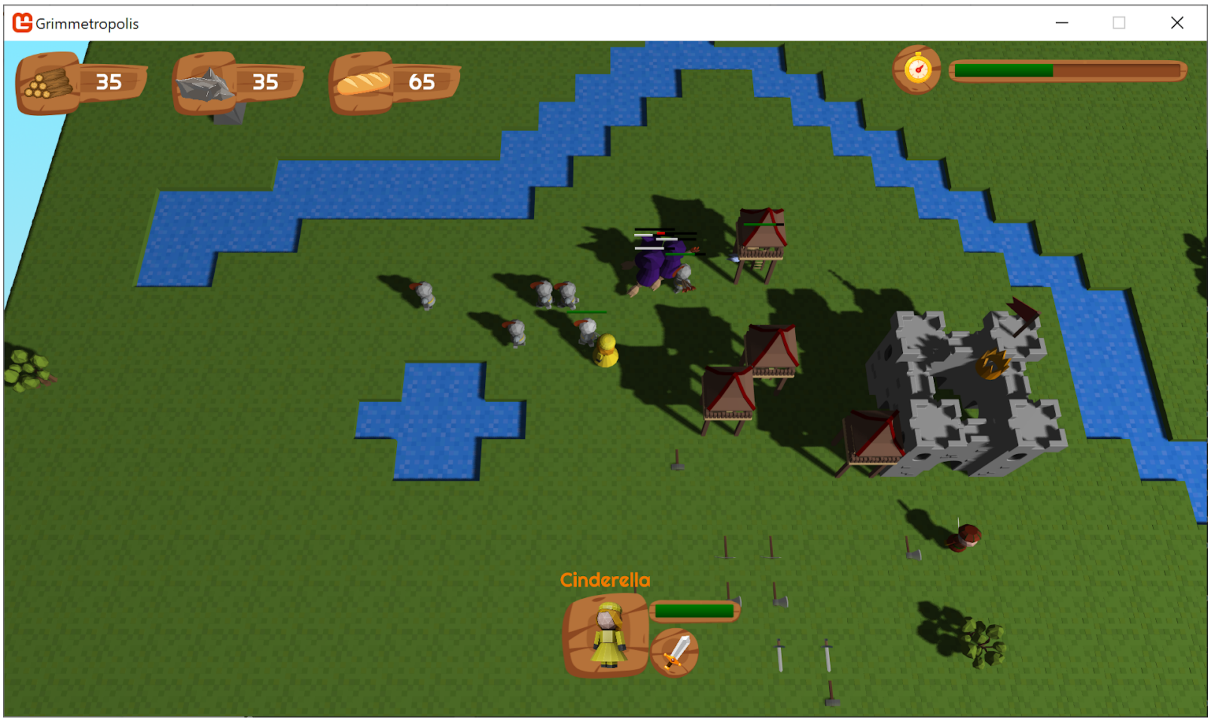
6.1. Final Results

There are several new features for the game: We added a new main menu, added a resurrection menu, some quality-of-life improvements (mostly regarding controls), updated a new GUI and a tutorial for newcomers.

Here are some screenshots to illustrate the new features:







Aside from visual improvements, the controls were slightly changed. As an example, tools can be swapped directly. Another thing is, that if the player wants to pick up items, the items need to be beneath the player instead in front of them. There were further small changes which improve the game experience.

6.2. Experience

Overall, we enjoyed our time during the class and have really learned a lot from the project. The final realization of the game deviates quite a lot from our initial design. This is mainly because we had a lot of ideas at the beginning and wanted to add many game mechanics; however, it is obviously not feasible to achieve everything we planned. Though we tried to plan the development schedule as detailed as possible, the estimation of time for some tasks are not exact. We turned out to be stuck at certain points and some of the tasks appear to be harder than we initially thought. Therefore, we struggled sometimes to strictly follow our schedule. One of the biggest challenges we had is the balancing of the game. As there are many game mechanics involved in our game, the balancing of the game turned out to be very crucial yet time-consuming. It sometimes required a lot of reiterating and play testing among ourselves; otherwise, players could be easily overwhelmed by the game. Another important aspect that we learned from our overall experience during the class is to focus more on one core game idea and keep polishing around it. It is also meaningful that we plan wisely so that the game can easily scale out.

6.2. Personal Impressions

What was the biggest technical difficulty during the project?

The biggest technical difficulty was implementing the AI of the enemies such that it works fast enough on large maps with a large number of enemies. We had to reimplement some part of the AI a couple of times, as some assumptions we used to improve the performance did not hold anymore when we added additional features. To further improve performance, we also decided to add caching for the path computations, which came with its own set of challenges and bugs.

What was your impression of working with the theme?

We found that having a story heavy theme limited our creativity more than it helped. While there are a lot of Grimm's fairy tales, each one comes with its own characters and world they live in. We felt that choosing a single story to focus would limit the possibilities, as our game would then need to somewhat fit inside said story. In the end, we decided to not focus on a single story and interpret the theme more openly. However, we would have preferred if the theme itself was less story heavy and more open.

Do you think the theme enhanced your game, or would you have been happier with total freedom?

While we were not quite so happy with this year's theme, we still think that a theme is necessary, as it helps with coming up with early ideas. Maybe giving several themes and letting the students decide which theme to choose from could be a valid alternative.

What would you do differently in your next game project?

Partway we realised we had too many ideas we wanted to implement. So, for our next game project, we would rather limit the number of features we plan to add and focus on a couple of important core ideas and flesh those out. Additionally, we want to do more playtesting, both internally and externally.

What was your greatest success during the project?

We consider the balance we found between performance and the computations needed for the AI to be our greatest success. In the later rounds of the game, there are a lot of enemies roaming around the map, and we need to compute a target destination for each enemy as well as a valid path for them to go there. Those computations need to be fast enough, otherwise the increasing wave size would quickly lead to lag and make the game unplayable. To achieve this, we first decided to partition the enemies in groups and decide the target destination per group and not per enemy. The pathing itself is still done individually. However, computing the full path every update cycle for each enemy individually is not feasible for a large number of enemies, so we compute the paths only if the target destination changed or some time has passed since the last path-computation.

Are you happy with the final result of your project?

Overall, we had too many ideas we wanted to implement and could not really meet our expectations in the end. There was not enough time to implement everything we wanted. Despite this, we are still happy about the core mechanics that are available.

Do you consider the project a success?

Rather than a success, we see the project more as an experience. We have learned a lot during this semester, not only pertaining to how to develop a game, but how to approach such a project and we are confident that our next project will be more successful.

To what extent did you meet your project plan and milestones?

While at the beginning of the project we were still on track with the project plan, we later realized that we had a lot of different game mechanics in our game. This made it rather difficult to expand quickly and to add additional features. It took us overall more time than initially planned, which led us to drop some features from the project plan so that we could instead polish the core gameplay and meet the deadlines. Overall, we felt we needed more time.

Did you like using MonoGame?

We find that using MonoGame is a good choice in the context of this course. Since it only provides a framework, it forces you to implement most features yourself, and in doing so helps you understand the very basics of a game engine and greatly improves the learning experience. However, it takes a lot of development time before we can begin with implementing the actual game. For future works, we would rather start with a game engine like Unity.