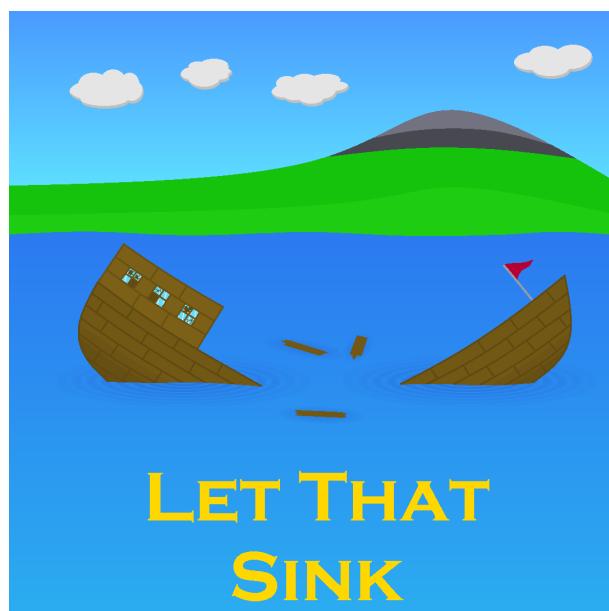




Game
Programming
Laboratory

Let That Sink



Lukas Egeling - Programmer, Designer

Rongxing Liu - Programmer, Designer

Yiming Wang - Designer, Visual Artist

Cedric Caspar - Programmer, Designer

Himanshu Joshi - Programmer, Designer

Pavel Pozdnyakov - Producer, Programmer, Designer, Composer

Contents

1 Formal Project Proposal	1
1.1 Game Description	1
1.1.1 Overview	1
1.1.2 Background Story	1
1.1.3 Design Decisions	2
1.2 "Big Idea" Bullseye	4
1.3 Technical Achievement	4
1.4 Development Schedule	5
1.4.1 Layered Task Breakdown	5
1.4.2 Task List	7
1.4.3 Timeline	9
1.5 Assessment	9
2 Prototype	11
2.1 Prototype Setup	11
2.2 Playing Experience	12
2.3 Findings and Conclusion	12
3 Interim Report	15
3.1 Progress	15
3.2 Challenges	18
3.3 Future Work	18
4 Alpha Release	19
4.1 Progress	19
4.2 Challenges	20
4.3 Future Work	22

Contents

5 Playtest	23
5.1 Playtesting Session	23
5.2 Questions and Comments	24
5.3 Design Revisions	25
6 Conclusion	27
6.1 Final Results	27
6.2 Experience	28
6.3 Personal Impressions	29

1

Formal Project Proposal

1.1 Game Description

1.1.1 Overview

The game belongs to the Tower Defense genre. The main objective is to destroy the enemy's fleet invading your territory. At the same time the enemy ships should be prevented from reaching a city, which serves as the players base. The ships are going through the river towards the city fighting along the way. The player uses weapons along the river to destroy them before they can reach it.

The game is designed in 2D and is meant to be played by a single player. One of the core features of the game is the background music dynamically adjusting to the scenery change.

1.1.2 Background Story

Years have passed since the people of Eldurheim fought their last battle. The abundance of food and other resources as well as peaceful rulers made traveling to foreign lands unnecessary. The rest of the world seemed to forget about Eldurheim surrounded by deep sea and vast mountains ranges. The centuries went by where the people of Elurheim lived in peace and quiet. From time to time wrecked ships with unknown weapons and materials were found on the coast. One day a fleet appeared on the horizon. A message was sent to Eldurheim to surrender to the unknown enemy. Now is up to you, the general of the army of Elureheim, to protect the land of your fathers from invaders.

1.1.3 Design Decisions

The core mechanics of this game is to protect Eldureheim and vanish the enemy's fleet - fulfilling the noble responsibilities of a general of army. To depict this scenario, we chose cartoon to feature the dynamic and exciting scenes.

The cartoon visual art depicts a grand castle (village) situated near the sea, with a large grassy area near to it, as the figure 1.1 shows. The castle is constructed of white stone and has several towers and battlements, giving it a majestic and imposing appearance. The sea is calm and blue, with numerous ships sailing on its surface. The ships are depicted with cartoonish details, such as exaggerated sails and masts, and are of various sizes and shapes, ranging from small fishing boats to large galleons.

On the grassy area near to the castle, there are several square units where cannons can be set up. The cannons are pointed towards the sea, indicating that the village is prepared for defense against potential attackers. The cartoonish cannons are depicted with large barrels and wheels, as shown by the figure 1.2. These powerful cannons are aimed at a queue of ships on the sea. In the same figure you can see that the ships are of varying size, color, and art, with some being small and nimble while others are large and formidable, indicating different efforts required to fire and sink them. The ships are designed to be visually appealing and distinctive, each one featuring unique characteristics that set it apart from the others.

As the game begins, the player set the positions of the cannons so that they aim and fire at the incoming ships. The scene is designed to be visually engaging, with the ships sailing towards the village's position in a steady queue, and the cannons firing back with thunderous blasts. The grassy ground and sea are depicted with vibrant colors and small details such as stones and trees, creating an immersive and dynamic environment for the player.

As the game progresses, the player must adjust their aim and timing to account for the varying sizes and speeds of the ships. The cannons and eventually other military resources feature different firing rates and abilities. The player can win, collect upgrade and customize these military resources throughout the game.

Overall, the scene is designed to be both visually striking and engaging for the player, with the combination of the cannons, ships, and grassy ground creating a memorable and exciting game experience. Despite the serious military preparation of the castle, the overall effect of the cartoon visual art is lighthearted and playful. The colors are bright and cheerful, and the cartoonish details give the scene a whimsical feel. The combination of the grand castle and the cartoonish ships and cannons creates an interesting contrast, adding to the overall appeal of the game.

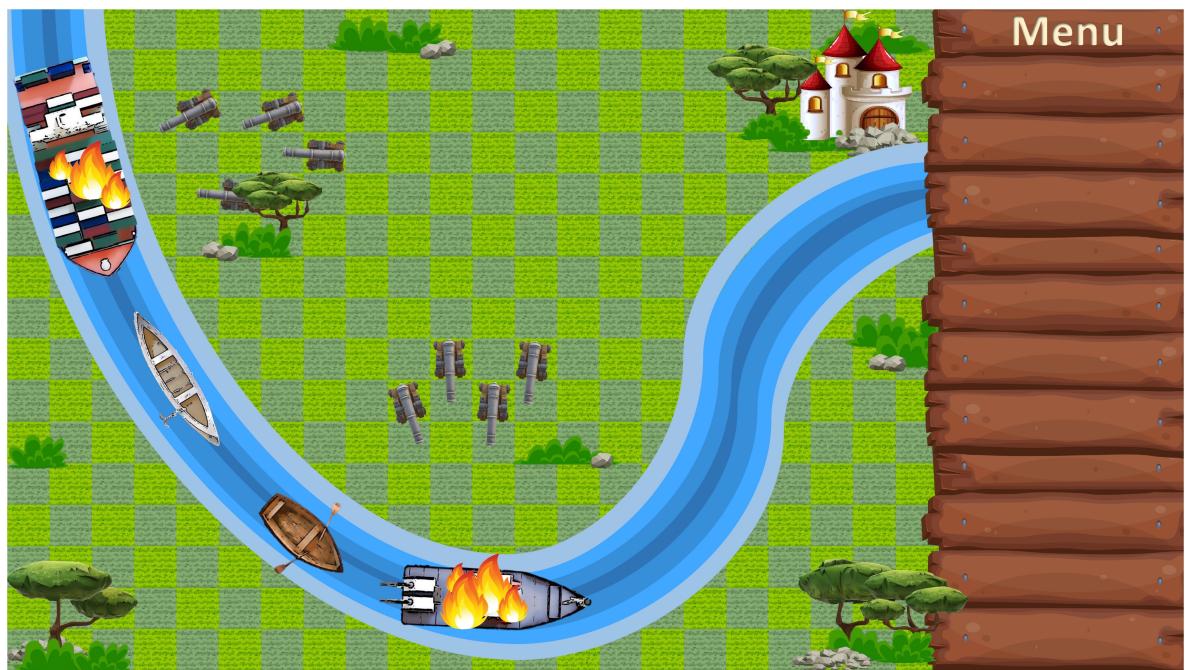
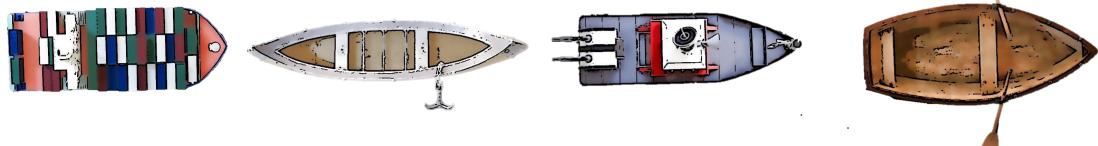
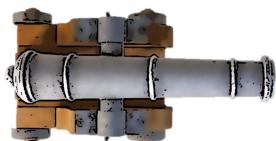


Figure 1.1: Map UI Design (Draft)

Ships (top-down view)



Cannon (top-down view)



Village



Figure 1.2: Elements UI Design (Draft, not fully listed)

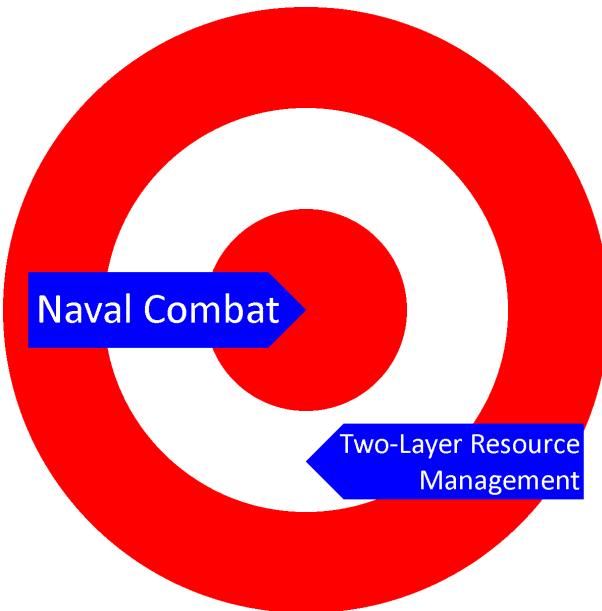


Figure 1.3: Big Idea Bullseye

1.2 "Big Idea" Bullseye

The core idea of the game is "Naval Combat". The main gameplay loop involves the use of various weapons to wreck ships invading your territory. There are different types of ships that vary in stats, have different resistances against certain weapons and can sometimes fire back. A battle is won once all ships are destroyed.

The main supporting component is "Two-Layered Resource Management". The game makes a distinction between battle resources and village resources. Battle resources are gained from destroyed ships and special workers and can be spent to build and upgrade weapons which can be kept for the duration of the battle. Leftover battle resources at the end of a battle are taken home and become village resources. Village resources can be spent on permanent upgrades that affect all future battles outside challenge mode.

1.3 Technical Achievement

Our targeted technical achievement is a dynamic soundtrack. At the beginning of each battle, the music starts off fairly calm with few instruments. As the enemy ships get closer to the outpost, more instruments start playing and the tempo increases, making the track more hectic, peaking when the outpost is directly under attack. If time remains, we will have unique music for different types of levels.

1.4 Development Schedule

1.4.1 Layered Task Breakdown

Minimal Viable Product

Game Mechanics	Design
<ul style="list-style-type: none"> - Place & upgrade weapons on the map - Waves of ships (simple health bar) - Weapons automatically shot at ships - Destroyed ships drop materials (wood & metal) - Game over condition: Ship reaches the city 	<ul style="list-style-type: none"> - Single 2.5D Map with river system to city - Simple ship designs - Two weapon designs & simple fire animations

Low Target

Game Mechanics	Design
<ul style="list-style-type: none"> - Switch soldiers between weapons & city defense (like command points) - UI for game stats & weapon options - Multi round game play - Weapons firing modes 	<ul style="list-style-type: none"> - Simple sounds for events - Generic soundtrack - Multiple degrading stages for ships

Desired Target

Game Mechanics	Design
<ul style="list-style-type: none"> - Persistent upgrades in village - Weapon visibility to target - Specialized workers with buffs - Splitting skill tree with weapon specialisations 	<ul style="list-style-type: none"> - Village design - Different ship types with weaknesses - Gods buffs & ability - Soundtrack adjustable to the scenery

1 Formal Project Proposal

High Target

Game Mechanics	Design
<ul style="list-style-type: none">- Ship can switch lanes- Xp system for number of soldiers- Ships attack back at weapons	<ul style="list-style-type: none">- Village buildings level dependent look- Boss Ship as final hurdle- Soldier Management system- Couple maps with defined enemies- Improved mixing and detailed orchestration for the Soundtrack

Extras

Game Mechanics	Design
<ul style="list-style-type: none">- Build paths along river allowing access to weapons- City can acquire new land down river for defenses	<ul style="list-style-type: none">- Procedural maps on creation of game save- Random events on map

1.4.2 Task List

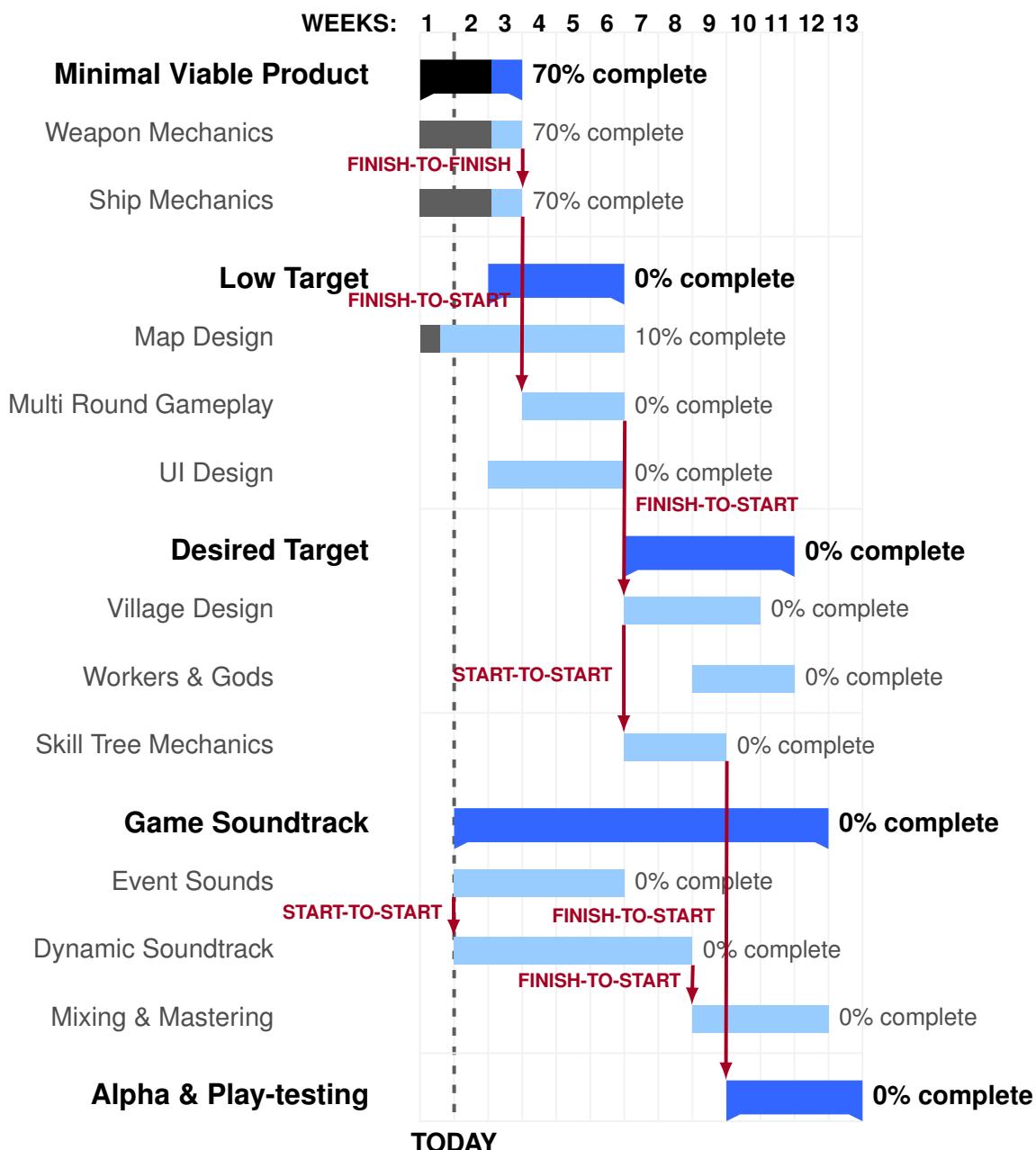
Nr	Task	Layer	Assigned	h	Priority
1	Create first map	Minimal	Yiming+	12	high
2	Design simple ship sprites	Minimal	Yiming+	10	high
3	Make ships run down river	Minimal	Lukas	3	high
4	Design both weapons	Minimal	Yiming+	8	high
5	Make weapons placeable & upgradable	Minimal	Lukas	3	high
6	Weapons shoot at nearest ship	Minimal	Lukas	3	high
7	Game over condition	Minimal	All	4	high
8	Materials from destroyed ships	Minimal	Lukas	6	high
9	Weapons firing modes	Low	All	5	mid
10	UI design	Low	Yiming, Lukas	10	high
11	Village design	Low	Yiming, Lukas, Cedric	16	high
12	Switching soldiers	Low	All	4	mid
13	Simple sounds for events	Low	Pavel	8	high
14	Generic soundtrack	Low	Pavel	24	mid
15	Multiple degrading stages of ships	Low	All	14	mid
16	Multi round game	Low	All	6	high
17	Persistent upgrades in village	Desired	All	6	mid
18	Workers from village with buffs	Desired	All	8	mid
19	Soundtrack depending on game	Desired	Pavel	24	high
20	Weapon attack only visible enemies	Desired	All	10	mid
21	Gods buff & abilities	Desired	All	8	mid
22	Skill Tree	Desired	All	8	high
23	Different ship types with weaknesses	Desired	All	12	high
24	Make everything look good	Desired	All	30	high
25	Ships attack back at weapons	High	All	10	mid
26	Soundtrack mixing and adjusting	High	Pavel	24	mid

1 Formal Project Proposal

Nr	Task	Layer	Assigned	h	Priority
27	Xp system	High	All	5	mid
28	Couple maps	High	All	16	low
29	Boss ship variation	High	Yiming	9	mid
30	Ship lanes Ai	High	All	8	low
31	Village buildings look level dependent	High	Yiming	16	low
32	Visible soldier management system	High	All	12	low
33	Map with procedural fillings	Extra	All	14	low

* People assignments not binding

1.4.3 Timeline



1.5 Assessment

The main strength of the game will be its unique way of handling in-game currency and player health. Instead of a single generic currency ("money", "gold", etc.) used to buy towers and one or more different ones for permanent upgrades, our game uses the same set of resources to craft in-battle and permanent upgrades. In addition, rather

1 Formal Project Proposal

than being able to let a fixed number of enemies reach the end before losing like in most TD games, our game ties the player's health to the way troops are distributed. This gives the player a strategic choice to either assign more troops to operate weapons, allowing ships to be battled more efficiently, or assign more to defend the outpost, allowing more enemies to reach it before losing.

One of the coolest aspects of the game is the way resources are used. Rather than buying weapons from an unknown source, you are crafting them using the resources themselves. Additionally, resources brought to the village are used to prototype new and improved weapons, hire and level up workers and sacrifice to the gods, giving some more context for the resources.

Our game is intended to appeal to those who enjoy strategic singleplayer games. It is not intended to give instant gratification for a short duration, but rather to keep people playing for a longer period of time by taking a few tries to work out functional strategies for each level. The game is intended to simulate some real-life aspects of naval combat without being too focused on historical accuracy. This and the above points are meant to make the game's design more coherent by making the game mechanics fit into each other. Our criterion of success would be players playing the game to the end and maybe even grinding resources to fully level up all permanent upgrades to reach 100% completion.

2

Prototype

In our physical prototype, we demonstrate some basic mechanics of our tower defence game (or rather any tower defence game in general). It includes enemies, weapons, resources, map, path and destination. Through this demonstration, people that are not familiar with tower defence game will understand its concept. Unfortunately, one of our key mechanics, soundtrack, cannot be incorporated into a physical prototype.

2.1 Prototype Setup

Our physical prototype consists of the following:

- A level map with size of 4 pieces of A4-paper. The design is shown in Figure 1.1 of the first chapter. All of our demonstrations will take place on this map.
- Four stick men drawn on small pieces of paper. They represent soldiers, which is part of the key game mechanics. They will also play an important role in our recorded demonstration.
- Weapons represented by matchsticks. Each requires one soldier to operate.
- Enemy ships represented by two toy ducks and two matchboxes. A standing matchbox represents a boss ship.
- Resources represented by coins. Each "ship" will carry some resources and by destroying the "ship" we can harvest the resources on it.

2.2 Playing Experience

The game play can be represented mainly by the seven steps below. In the beginning we have one coin on the top left corner denoting the resource we have. We also have four soldiers standing by in the castle. Then we spend one resource building our first weapon and let one soldier operate that weapon. When the first enemy ship arrives, the weapon attacks it and destroys it. We collect the resource dropped and use it to build the second weapon. Next, more enemies are coming and we have to be prepared for that. If not there could be some enemies passing through our defence and threatening our castle. In addition, we have two demonstrations for the manpower management, one winning case and one losing case. In the winning case, we deployed all four soldiers to operate the weapons (this is not possible in the real game because we need at least one soldier defending the castle). With four weapons we can destroy most of the enemy ships except the boss ship which is much stronger. In this case we quickly switched the soldiers back to defence the incoming attack. In the end, three soldiers were sacrificed to defeat the boss. However, if we are not careful and let one enemy sneaks in, we would lose soldiers unnecessarily. Our defence would be vulnerable to future attacks which leads to a losing result.

2.3 Findings and Conclusion

While making this prototype, we found that the concept of a tower defence game is very straightforward and can be easily demonstrated with simple tools. That is why it is very easy for a player to get the hang of a tower defence game.

However, when we played the game, we realized that it is not easy to time the flow of ships and weapon firing perfectly. We either destroyed the ships too quickly or too slowly and let the ships reach the castle. It is also difficult to come up with a suitable winning case and a contrasting losing case that demonstrates manpower management properly. These findings suggest that the balancing task of a tower defence game could be extremely challenging, especially given the fact that things get tricky when the setting is still simple. We do not want the game to be too easy nor too difficult. Hence, in the future development stage we might have to invest more time adjusting various hyper-parameters.

2.3 Findings and Conclusion



Figure 2.1: Initial status.



Figure 2.2: First enemy comes.



Figure 2.3: Enemy hit by weapon.



Figure 2.4: Enemy destroyed.



Figure 2.5: More enemies and more weapons.

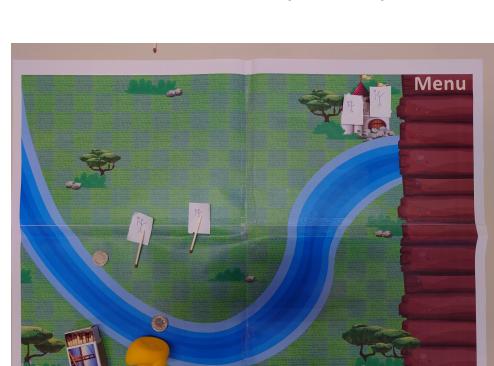


Figure 2.6: Enemies destroyed.



Figure 2.7: One enemy passes through.

3

Interim Report

3.1 Progress

In this section we will give a brief description of how the players can interact with the game. People will understand the layout and basic operations in the game so that they will be able to play the game with no difficulties. In addition, we will also reflect about our current progress and make plans for the future.



Figure 3.1: Start of a game. On the top right corner are the options of pause, slow, continue, speed up and setting. Next we have the resource amount and troop size shown below, followed by three basic weapons. At the bottom we have the option of abandoning all weapons.

We have finished layer 2 and some features in layer 3, including splitting skill tree with weapon specialisations and soundtrack. Here we will provide some screen shots to demonstrate how players can interact with the game. There are a few actions players can take, including building a new weapon, upgrading a weapon, dismantling a

3 Interim Report

weapon, abandoning a weapon, abandoning all weapons and focusing on a ship. As shown in Figure 3.1, players can choose to build one weapon out of three by clicking on it, moving it to a place on map where the placements are all green (3.2) and place it there. It will then take a few seconds for the construction to complete after which the weapon will start shooting at enemy ships.



Figure 3.2: All green means players can place the weapon.



Figure 3.3: Otherwise they have to place it somewhere else.

Next, players may click on the weapon to upgrade it, abandon it or dismantle it. Abandoning the weapon will switch the trooper that operate the weapon to defend the village. The weapon will stop shooting and it needs to be reclaimed to be able to shoot again. Dismantling the weapon will return the troops and a portion of the resources used to build the weapon. Again all such operations take a few seconds to finish.



Figure 3.4: Upgrade, abandon and dismantle the weapon.



Figure 3.5: Reclaim the abandoned weapon.

During the battle, players may want to focus on a specific ship that poses the biggest threat. To do this, they can simply click on the ship and choose the option "Focus this ship" on the bottom right as shown in Figure 3.6. Since focusing the ship does not add extra boost to weapons, this option has no cool-down.

There is a wide variety of weapons and ships. For weapons, there are three basic types of weapons, namely crossbow, catapult and cannon. Each basic weapon can be upgraded to an enhanced version, then it can be upgraded into three types of more powerful weapons. For example, crossbows can be upgraded into piercing crossbows which shoot piercing arrows across all ships on its path, or fire crossbows that light ships on fire, or split crossbows that shoot multiple arrows. Finally there will be one



Figure 3.6: By clicking on the ship, players can see the stats about the ship including its health, troop size and resource dropped. They can also let all weapons target this ship to destroy it quickly.

final upgrade to enhanced version of the specialised weapon. The full hierarchy of crossbow is shown in 3.7.

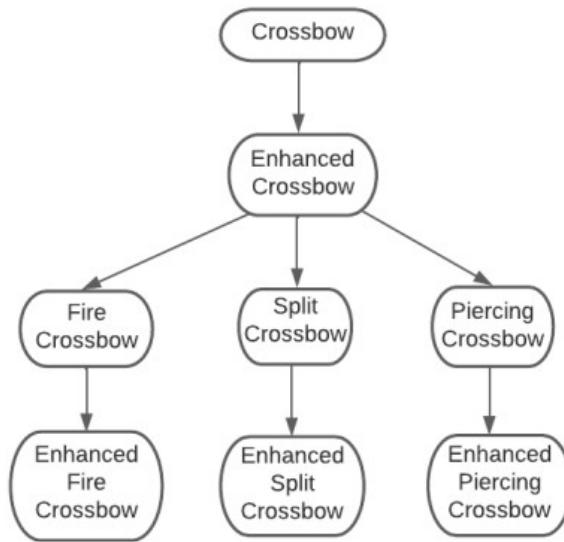


Figure 3.7: Hierarchy for crossbow.

For ships, we have basic wood ships, enhanced wood ships, sprint ship with high speed but low health, iron ship, armored ship and heavy armored ship. In addition to the normal ships, we also created two types of boss ships, a slightly weaker which will be deployed in the middle game and a stronger one which will be deployed as the final boss. It has a lot of HP but it moves very slowly. A lot of effort was made to balance the game. Overall, if the players set up the weapons properly, there will be no difficulty winning the game.

3.2 Challenges

The major challenge that presented itself, is the creation of assets. While implementing additional features comes more easy, the time needed to create all animation frames and styles is bigger than anticipated. In addition we have trouble deciding whether to use existing assets as matching styles and potential copyright stand in the way. This all reflects in the progress made in designing the map(s) where after many iterations the result is still not satisfactory. Therefore in simple terms the hardest challenge remaining is "making the game look good".

Other challenges come with the soundtrack. Difficulties arising when creating the soundtrack is by the fact that both, the music and the game has to be created in parallel. Although in the usual workflow the soundtrack would be one of the last things added to the game, imposed time restrictions made it necessary to work on both simultaneously. This resulted in the need of guessing, how the end version of the game will look like. The current sounds for the cannon shots, for example, were based on the assumption of a real time shooting speed of a canon. This resulted in the very muddy sound given the much faster shooting speed in the game and, therefore, the need for the replacement.

3.3 Future Work

The bulk of the most pressing work still needed is in creating various menu screens. The village mechanics as well as the map selection and general settings menu. In addition, the asset and map generation needs to go further.

The music score to the soundtrack is mostly completed. The score now needs to be recorded, which will be done consecutively during the following month. Some already existing sounds needs to be adjusted to the current game mechanics. New sounds for all the new weapons currently added to the game or planned to be included in the game have to be created.

4

Alpha Release

In this section we will describe the extra work done after the interim demo. Several improvements have been made including UI and level design. The game now looks more complete. We will also talk about some challenges we faced and plans on future work.

4.1 Progress

At this stage, we have not met all the desired targets but we do have some high targets implemented. This is due to some misevaluations of the target difficulty and importance during the proposal stage. Some targets seem to be more difficult yet less important than others when we really dive into some detailed implementation of the game. For example, we have not achieved the desired targets "Specialized workers with buffs" and "Gods buffs & ability" but we managed to achieve the high targets "Ships attack back at weapons", "Boss Ship as final hurdle" and "Couple maps with defined enemies". Next we will explain the improvements we have done after the interim demo.

The first improvement we have done is extending one level to five levels. The first two levels are very easy and we designed it in such a way that as long as players know the basic operations, they should be able to pass the level easily. The third level starts to become more difficult as we have increased all ships' HP by 20%. As a result, if the player makes a mistake in resource allocation or weapon placement, some ships will slip through the defence and threatens the castle as shown in 4.1. Nonetheless, we still give a high tolerance for inaccuracies so that players can pass the game unless they make some very poor decisions.

The fourth and the fifth level are where the fun begins since we introduced boss ships



Figure 4.1: Example of bad strategies and resource management.

that can fire back at weapons and create more challenges for players. In level 4, ships' HP are increased by 40%, which makes the opening stage much more difficult. On the other hand, the level becomes longer and players will have enough resources to upgrade the weapons. They can explore different functionalities from different high-tier weapons and choose the one most suitable for the game. In general, we are not expecting players to win level 4 and 5 in one pass. They will need a good understanding of different weapons, ships, resources as well as the map itself in order to make the best decision. In level 5, ships' HP are increased by 50% and there is almost zero tolerance for inaccurate decisions. As of level 4, the opening stage is definitely one of the most challenging tasks for players, but once they are able to progress to the middle stage, they will form enough fire power with abundant resources. Then there will be a small boss ship which drops gun powder (essential resource for high-tier weapon upgrades). In the end, there will be five small bosses followed by one big final boss. Together these ships can deal a lot of damage to the weapons and players have to decide whether to repair the weapons and which weapons to repair. We expect players to be able to kill all the previous ships and if they play really well, they will be able to kill the final boss ship. Otherwise they always have the option to abandon weapons, switch soldiers back to castle defence and counter the threat of the final boss. 4.2 gives an example of the game progressed to the very end.

The recordings of the soundtrack continued. In addition to the one music track recorded before the interim demo, two other tracks were recorded: background music for the boss ship fight as well as calm music for the initial stages of the game.

4.2 Challenges

With most of the functionalities implemented, the biggest challenge now is balancing, which is notorious for a tower defence game. The challenges we are facing are three-



Figure 4.2: Final boss ship appearing in the end. Weapons with no health bar are destroyed.

fold.

Firstly, we need to make sure that all weapons have equal importance in the game, no weapon will outperform others substantially and if that is the case, then we will either nerf it or increase its cost to make it less accessible. For example, during the interim demo we found that the explosive catapult was so powerful that it would kill any enemy instantly and building one such weapon will win the game easily. Therefore, we cut its damage by a significant amount. Now it is not that strong anymore but it is still an effective crowd control weapon.

Secondly, we need to make sure that there is a smooth progress in the level. There should not be too many weak enemies nor too many strong enemies. The enemies level should grow from weak to intermediate and finally there will be some strong bosses appearing at the end of a level. At the same time, the enemy ships will drop enough resource for the player to build or upgrade weapons and be able to face stronger enemies. However, they should not drop too many resources as that will give players too much fire power and the game will become boring (though one of our game mechanics involves troops to operate the weapon which serves as a limiting factor for excessive resource).

Thirdly, we need to make sure that there is a smooth transition from one level to another. As mentioned in the previous section, the difficulty increases from level 1 to level 5. Nonetheless, players should not feel that some level suddenly becomes very difficult to play which will discourage their interest. This task is less difficult than expected because we have decomposed it into five small tasks. After balancing one level, we can inherit most of the configurations to the next level. In addition, earlier levels give fewer resources and fewer weapons can be built or upgraded. We just need to focus on balancing some specific weapons and expand the scope gradually.

4.3 Future Work

The soundtrack recordings take more time than anticipated and are planned for the next weeks. Music for intense stages of the game as well as sounds for various new weapons have to be recorded among others.

5

Playtest

5.1 Playtesting Session

We invited some of our friends for playtesting. Each member will take note on 2-3 friends and collect feedback from them. Since it is difficult to find a time suitable for everyone, we decided to organize the playtesting sessions individually.

During the sessions, we would first let them play through the whole game without explaining anything to them. We wanted to find out how much tutorials we should add to our game so that the players will neither be bothered by excessive amount of information provided nor be discouraged by unclear instructions and lose interest to the game. We understand that different players have different level of expertise in playing tower defence game, so we did not set any time limits for them. Strong players who finished the game earlier will give feedback first while weak players can take their time playing the game slowly.

Also a couple one-on-one game sessions were conducted. Since the game is single-player, observing them alone is crucial to see whether the game can hold their attention without additional human interactions.

For the feedback session, we started by asking each other some basic questions regarding various aspect of the game, then they will give us some precious advice on how to improve the game. We also prepared different questions for strong and weak players to make sure that our game in the end will appeal to both groups.

5.2 Questions and Comments

Here are a few questions we asked:

1. **Q:** Do you have any difficulties understanding the gameplay?
A: After exploring for a while, we are able to figure out most of the gameplay. The concept of tower defence games is not anything new and complicated, therefore functionalities should not be a problem.
2. **Q:** Do you think the weapons, ships and resources are balanced?
A: During later stages when there are many armored ships, the power of fire ballista and fire cannon diminishes drastically, which make them unfavorable. In addition, they can hardly make use of sniper cannon and piercing ballista in the map. It would be a good idea to design a map that emphasizes the use of these two weapons. The power of other weapons are balanced very well. For example, although the "crator creator" is extremely effective in crowd control, it has a high cost both in terms of resources and in terms of troop size. Therefore one cannot afford too many "crator creator". Besides, it is not effective when dealing with bosses, so one should also consider other options instead of solely relying on it.
Ships are also very balanced as the amount of resources they drop is roughly proportional to their strength. There is no sudden big gaps in ships' strength, except for bosses, which is expected and players are ensured enough resources to build weapons that can destroy the bosses.
3. **Q:** Do you think there is a smooth progression from level 1-1 to 1-5?
A: Generally yes. 1-1 to 1-3 are fairly easy. 1-4 starts to get a bit tricky because of the miniboss. 1-5 is definitely difficult even for experienced players. They need to explore the power of various high-tier weapons to be able to make good decisions about weapon placements and upgrades. The feedback arose that all the waves can be solved with almost the same strategies. Here the creation of different maps and better balancing is important.
4. **Q:** Are you comfortable playing with the mouse?
A: Yes. Generally tower defence games do not require good mouse skills like some MOBA or RPG games, though one can easily panic in the middle stage of 1-5 and missclick some buttons.
5. **Q:** Do you like the game UI?
A: Overall the feedback of the various UI elements was positive. Besides minor nitpicks, like not enough information on upgrades and destroyed weapons, the navigation was clear and not an issue. All playtesters found their way around the game quickly. One subquestion ask was, whether missclick security was necessary, meaning adding a "Are you sure" button. The answers were fifty-fifty.
6. **Q:** Any other suggestions?
A: There should be a level number appearing in each level so that players can know which levels they are playing.

It would be very helpful to add in a reminder beside weapons that are eligible

for an upgrade. As players, they do not know the upgrade costs very well, so if there is no such reminder, they have to click on each weapon to check for their upgrades which is very inconvenient. Furthermore, there could also be some reminders beside weapons that are destroyed as healthbars are not very intuitive.

In conclusion, most of the feedback lies in the category of quality of live improvements and further balancing of the game. There is still much work to be done to make our game more user-friendly.

5.3 Design Revisions

The major missing things like proper tutorial, visual differences between game object statuses and more information visible to the player are brought into focus. To accommodate the feedback, for example damage numbers and better resource cost displays will be added to the game.

Additionally, more variety with the enemies and their interaction with the weapons was desired. But in the scope of the lecture and the small remaining time, the focus will lay on creating a more finished and polished game. Since most of the arising problems during playtesting were due to missing parts of the game, like proper weapon unlocks and resulting loss of game balance, for the final stretch of the development many smaller improvements will be implemented.

Design-wise the game was accepted well, except the background map of the levels. One part is that there is variety missing but the creation of more maps is already a to-do for us. The art style of the latest iteration of the existing maps did not resonate with the playtesters and will be reworked.

Finally, the music system stated as one of our bullseye points is missing entirely. But the music is at last fully composed, recorded and ready to be integrated into the game.

6

Conclusion

6.1 Final Results

The final result is a singleplayer naval tower defense game with a Norse setting. The goal of each level is to destroy all incoming ships. These ships drop five different resources, which are used to build three types of weapons and various upgrades. The player has a limited number of troops to operate weapons and defend the castle. When a ship reaches the castle while insufficient troops are there, the battle is lost. Leftover resources after a won battle are brought to the village and can be spent on permanent weapon and god unlocks.

Some of the game's interesting implementation details are hidden in the backend and not apparent to the player. For one, the game runs in constant length ticks in which various game processes are handled. Since these ticks don't align with the framerate, each frame runs as many game ticks as can fit into the deltatime and ends with a visual-only tick in which objects may move based on the remaining time but don't interact with each other. This system ensures that the game speed and framerate don't affect the gameplay. In contrast, games like Super Meat Boy are dependent on the framerate, so a lag spike may cause Meat Boy to phase through walls. Another interesting implementation detail is that the common stats of weapons and ships, as well as the level layouts are defined in json files. This makes balancing a more streamlined process, as it can be done mostly by changing just these files.

The game's graphics were almost completely overhauled since the alpha release, much of which was done between the jury and final releases. This includes the creation of all three final map backgrounds, different sprites for ships and their modifications, six distinct animation frames for each weapon (one for the idle state, two for shooting, two for building and one for the broken state), some particle effects and several tweaks to

6 Conclusion



Figure 6.1: Title screen

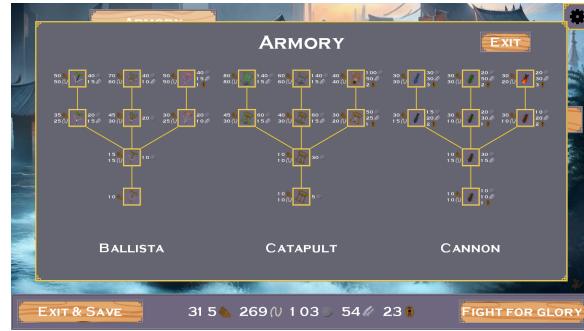


Figure 6.2: Armory, all weapons unlocked



Figure 6.3: Pantheon, all gods unlocked



Figure 6.4: Gameplay featuring final boss

the UI.

The final soundtrack consists of 7 pieces of music as well as short music snippets for winning and losing the game. The set of sounds includes various weapon shots, target hits and a button click.

All of the music except for one track and all of the sounds were added to the game after the alpha release. A background track used in the alpha release was remixed. All of the sounds from alpha release were redesigned.

Various gameplay improvements were added in the later stages. The second and third stages with five waves each were added in this period (the five waves of the first stage were already implemented, but waves 2-4 were only accessible through debug keyboard shortcuts). The village, including the armory, pantheon and level select were added. The gods were made accessible in-game with unique boosts to their associated tier 3 weapons, activated abilities and ways to charge these abilities. A difficulty slider was implemented to make the game more accessible (with easier difficulty reducing resources earned as a tradeoff). The final boss was made to spawn other ships during the last wave of stage 3.

6.2 Experience

The creation of "Let That Sink," a village defense game, was an intricate process, harmonizing a medley of ideas into a unified design. This journey was a test of both

our creativity and technical skills, and while demanding, it resulted in a game that exceeded our original expectations.

Our initial design ideas were significantly realized in the final game, following a successful melding of diverse team ideas into a shared vision for "Let That Sink." Notably, a major technical challenge involved game balancing, particularly between the alpha and jury release stages. This underscored the necessity of effective group communication, as a minor lapse had escalated the issue.

While our development schedule acted as our guiding roadmap, we faced some unexpected hurdles. The creation of visual assets experienced a slowdown due to our initial choice of visual tool, yet these deviations did not significantly sway us from our planned timeline.

Project structure elements like playtesting substantially contributed to our progress. They allowed us to identify unforeseen UI issues, enhancing the alpha release. However, we realized that incorporating feedback channels, like questionnaires, early in the design phase could have optimized this process.

Looking back, "Let That Sink" is a symbol of our creativity, resilience, and collaborative effort. Each challenge encountered presented an opportunity for growth, instilling invaluable lessons for future game development endeavors. Despite various hurdles, our commitment to the project resulted in a final game that aligns with our initial vision and beyond, making this journey a truly rewarding learning experience.

6.3 Personal Impressions

Did it meet your expectations? Are you happy and proud of your game? Do you feel there wasn't enough time or that the schedule was too compressed?

Our journey creating "Let That Sink" has culminated in a game that exceeded our expectations and fills us with pride. The final product, featuring 15 diverse maps and a unique soundtrack, was accomplished within the challenging timeframe of a 14-week semester, a testament to our dedication and efficiency. However, the compressed schedule did present some issues. The brief period between the playtesting phase and the jury release limited our ability to fully integrate valuable feedback received. Furthermore, the necessity of creating a physical prototype, while not overly time-consuming, seemed less relevant for our tower defense game, and we believe that time could have been spent more effectively elsewhere. Despite these challenges, we delivered a game that surpasses our initial vision and serves as an enriching learning experience. We suggest future iterations of the course consider a more flexible schedule, perhaps adding 2-3 weeks post-playtesting, to allow better integration of feedback into the final product.

What was the biggest technical difficulty during the project?

The most formidable technical hurdle encountered during the development of "Let That Sink" was maintaining game balance. This intricate task involved meticulous adjust-

6 Conclusion

ments to game mechanics to ensure an engaging level of difficulty and progression. Our strategy involved defining common statistics of weapons, ships, and level layouts in JSON files. Modifying these files streamlined balance alterations, thereby mitigating the complexity of this challenge.

Another notable technical detail revolved around the game's operational mechanics. Our game operates on a system of constant length ticks, where each frame accommodates as many game ticks as possible. This unique design, independent of the framerate, ensures that game speed and framerate do not impact gameplay, thereby circumventing potential issues such as unexpected behavior during lag spikes.

What was your impression of working with the theme? Do you think the theme enhanced your game, or would you have been happier with total freedom?

Working with the designated theme provided a beneficial framework that guided our design decisions and streamlined our creative process. The theme additionally lent a clear structure to the course, facilitating easier categorization of different games and enhancing the overall course experience. Nevertheless, a set theme may restrict creativity if it doesn't align with a team's passions. For future courses, we suggest a compromise - a core theme for guidance, but enough flexibility for teams to tailor their themes to align more closely with their interests. This approach would maintain the benefits of a unified theme while also allowing space for individual creativity, thereby enhancing the overall game development experience.

What would you do differently in your next game project?

In future game development projects, our approach would focus on several improvements. We would foster an open exchange of ideas within the team at the outset, particularly around technical issues and debugging, to enhance efficiency and stimulate innovation. User testing would be incorporated earlier in the process via questionnaires to inform design choices. We also plan to dedicate more resources to creating captivating visual content from the project's inception. We will carefully select UI tools that align with our agreed art style and test them extensively before committing to one. These strategies, learned from our experience with "Let That Sink," aim to improve the development process and the overall quality of our future games.

What was your greatest success during the project?

Our greatest success during the project was multifaceted, stemming from various components of the game development. This includes mastering the implementation of ship movement trajectory and innovating in visual representation, such as simulating a realistic catapult projectile. Another success was manifested in our creative efforts, particularly in crafting detailed sprites for weapons and ships, and composing an original soundtrack. Each accomplishment, from technical proficiency to aesthetic creativity, demonstrated our commitment to excellence in crafting "Let That Sink," enriching the overall player experience.

Are you happy with the final result of your project?

We are immensely proud of the final result of our project, "Let That Sink." Our diverse game ideas coalesced into an engaging game that exceeded our initial expecta-

tions, an accomplishment that testifies to our team's effort and creativity. The positive feedback from our pilot playing session further affirmed our satisfaction with the final product. Overall, this project was not only about creating a game, but also a valuable journey of shared learning and growth.

Do you consider the project a success?

Undeniably, the development of "Let That Sink" was a resounding success. The final product exceeded our initial expectations, providing us with valuable learning experiences. Beyond creating a playable game, the project enriched our understanding of game development, honed our problem-solving skills, and fostered effective team collaboration. These invaluable lessons are invaluable for our future career trajectories.

To what extend did you meet your project plan and milestones (not at all, partly, mostly, or always)?

Our team adhered to our project plan and milestones for the most part. We encountered a slight delay in visual asset creation due to the mismatch between our initial tool selection and desired visual effects, identified during the playtesting phase. Despite receiving abundant constructive feedback from playtesting, time constraints restricted full integration of these suggestions into the jury release. However, demonstrating resilience, we managed to realign our project trajectory with the established milestones, ensuring successful project completion within the proposed timeline.

What improvements would you suggest for the course organization? (Perhaps in D1 evaluation)?

Reflecting on the organization of the course, we suggest several improvements that could facilitate smoother project development. One suggestion includes providing jury release date earlier, which would better aid in project planning, especially for implementing post-jury changes. Additionally, expanding the duration between playtesting and the jury release could offer more room for game refinement. We also propose facilitating role-specific cross-group dialogues to enhance inter-group learning and knowledge exchange, particularly after witnessing the compelling visual effects achieved by other teams. Finally, organizing an experience-sharing session with past project teams could provide invaluable insights into potential challenges and solutions, including the importance of early-stage user testing. These adjustments could enrich future cohorts' learning experience, foster collaboration, and empower practical project management.

Did you like using MonoGame?

Our game development journey with MonoGame was highly rewarding. As we embarked on the beginning stage of our game project, MonoGame offered a user-friendly platform that was well-matched to our needs. Its structure catered perfectly to our project's size and scope, providing a solid foundation to construct our game. This made MonoGame an ideal choice projects which are relatively small in scale. The simplicity and utility of the framework were key factors that aided in streamlining our development process. However, it's worth noting that MonoGame's constraints, such as lack of support for multi-player mode over a network, must be taken into account while choosing it for a game development project.