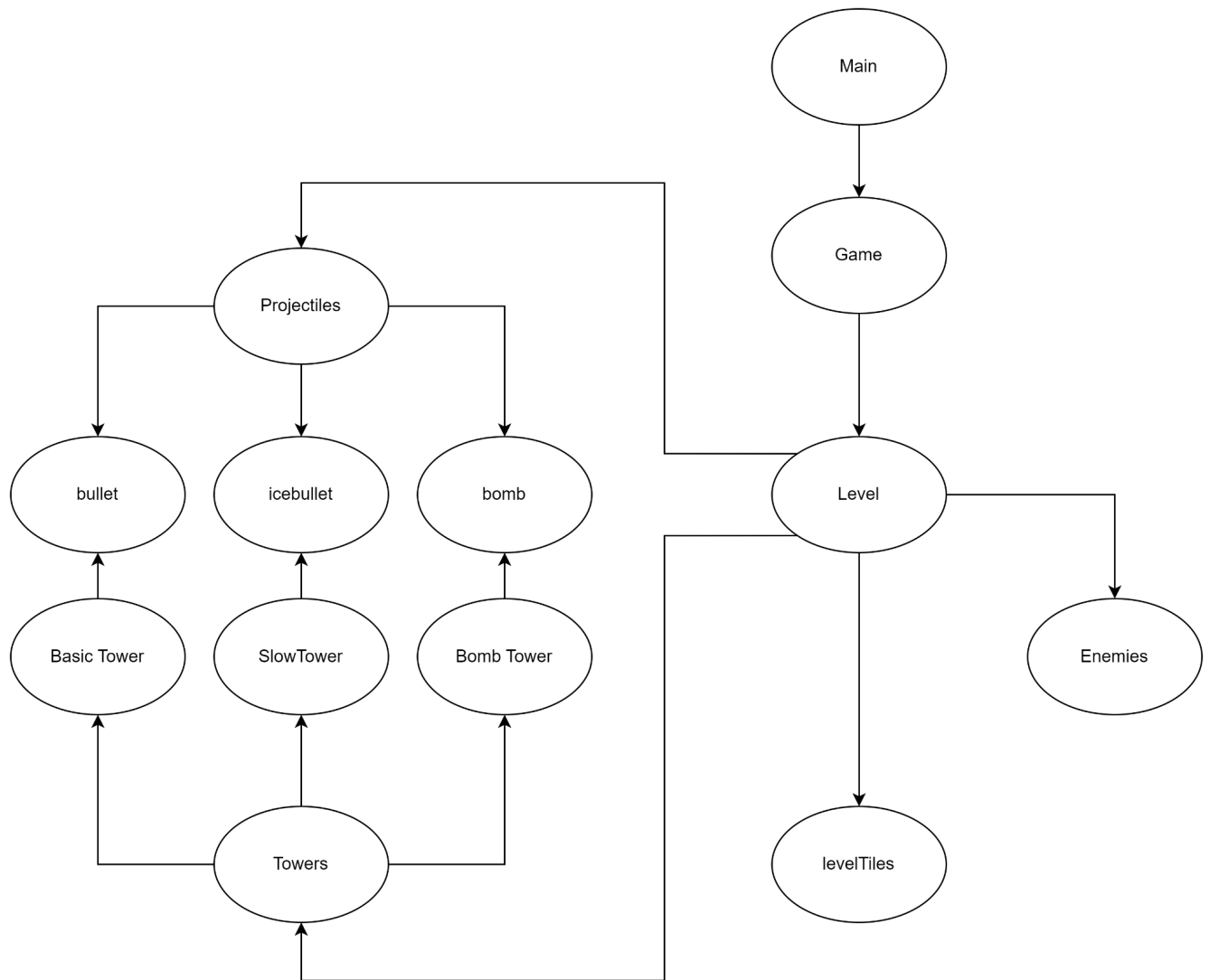


## 1. Overview what the software does, what it doesn't do? (this can be taken/updated from the project plan)

A tower defence game is a simple game where players are faced against waves of enemies trying to reach some goal, usually the player's stronghold or simply the end of the map. The player can build "towers" alongside the path the enemies are taking, using some form of currency. These towers would then hinder the enemies progression by either destroying them or effecting them in other ways, such as slowing them down.

## 2. Software structure: overall architecture, class relationships (diagram very strongly recommended), interfaces to external libraries

The game itself is an instance of the Game class. Each level is an instance of the level class. The level uses the enemy, tower, projectile and leveltile classes. All of the tower and projectile classes inherit from SFML's CircleShape class while enemies and leveltiles use the Sprite class of SFML. The class structure is shown in the diagram below. More inheritance figures can also be found in the Doxygen generated documentation file.



### 3. Instructions for building and using the software

Please ensure that CMake and g++ compiler has been installed on the computer. Then running CMake to compile the project, finally, run the MyGame file. We recommend using Visual Studio Code when building the game. Before building the game make sure that the image and sound files have correct file paths in the code. You might have to make some changes depending on if you are using Windows or Linux. If the game fails to open, you might want to try to comment out the lines related to the sound files in the level.cpp file (lines 91, 92, and 360).

### 4. How to compile the program ('make' should be sufficient), as taken from git repository. If external libraries are needed, describe the requirements here

When using VS Code, open the folder that includes the CMakeLists.txt. This should prompt you to choose a compiler. Choose the g++ compiler. Afterwards, the program should configure itself. If this does not happen or you need to reconfigure the program, click on the terminal

tab on the top toolbar and choose Run Task... and then CMake: configure. To build the file, again choose Run Task... but choose CMake: build instead. This will create a build/ folder that includes the executable MyGame. The program requires the SFML library to function, though you should not need to download the SFML library to your own computer as CMake fetches the required files from a GitHub repository.

## 5. How to use the software: a basic user guide

The software is launched by executing the executable MyGame that can be found in the /build folder after first building the software with CMake. When the software is launched, the player is presented with an opening screen where they can choose between five different levels. Clicking on a level tile launches that level. In the levels the player can buy towers by first clicking the wanted tower icon on the right side of the screen and then clicking anywhere on the map. If the player clicks on an already placed tower the tower will be upgraded. When the player is ready they can click the start wave button to start the spawning of enemies. At any point the player can also click the back button to go back to the level select screen. If any enemies reach the end of the level, the game is lost. The player wins when all enemies have been killed.

## 6. Testing: how the different modules in software were tested, description of the methods and outcomes

The testing of the software was done simultaneously with the making of each file. The person who made a file or made any changes to a file always checked afterwards that everything worked as intended. Most of the testing was done by playing through the game repeatedly.

## 7. Work log: This might be a simplified/restructured version of the weekly meeting notes file.

During the meetings each participant shared their current progress, discussed any problems they faced and planned what they will be doing during the next week. Any changes to the original division of work in the plan were also discussed.

### # Meeting 11.11.2022 14::00

**\*\*Participants\*\*:**

1. Veeti Jaakkola
2. Kun Ren
3. Olavi Kiuru

### ## Summary of works

Everyone worked together to finish the plan.

### ## Actions

1. Veeti and Kun will work on making a working game screen
2. Olavi will work on a basic tower class.

## **## Project status**

Project plan finished. Next step is to start familiarising with the SFML library.

### **# Meeting 18.11.2022 14::00**

**\*\*Participants\*\*:**

1. Veeti Jaakkola
  2. Kun Ren
- (Olavi away at a conference.)

### **# Meeting 25.11.2022 16::00**

**\*\*Participants\*\*:**

1. Veeti Jaakkola
2. Kun Ren
3. Olavi Kiuru

## **## Summary of works**

Kun has made a first version of an enemy class and fixed some issues with the game and level classes. Olavi has made two more tower classes (bomb tower and slow tower) and a projectile class that includes bullets and bombs.

## **## Actions**

1. Kun will look into how to get the enemies moving along the path.
2. Olavi will look into placing towers with the mouse.

## **## Project status**

We are able to draw to the screen enemies, towers and projectiles. We are also able to update their positions.

### **# Meeting 02.12.2022 14::00**

**\*\*Participants\*\*:**

1. Kun Ren
  2. Olavi Kiuru
- (Veeti away at military service.)

## **## Summary of works**

Kun has gotten the enemies to follow the path. Olavi has made it possible to place towers by clicking. We also moved most of the game logic to the level class.

## **## Actions**

1. Kun will look into deleting enemies after they hit 0 HP. The player should also receive money when an enemy dies.
2. Olavi will look into making the bomb projectile hit many enemies.

## **## Project status**

The enemies now move correctly and the towers can shoot them. The UI still needs to be updated to include money, lives and instructions on how to place towers. We are close to a fully functioning first level. Adding other levels after that should not be too difficult.

## **# Meeting 09.12.2022 14::00**

**\*\*Participants\*\*:**

1. Kun Ren
2. Veeti Jaakkola
3. Olavi Kiuru

## **## Summary of works**

Kun has made multiple different enemy types. Kun has also added waves and different maps. Olavi has added the range attribute to towers. Veeti has made some sprites for the GUI. Olavi and Veeti worked together to add a level select screen, upgradable tower and a functioning GUI.

## **## Actions**

1. The deadline is in two days. Next everyone will work together to make sure the Git has all the necessary files in the correct format and place.

## **## Project status**

We have a functioning game. Only thing left is to write the documentation and any other files still missing from the Git.

## **8. Detailed description of division of work and everyone's responsibilities**

Olavi was responsible for making the towers and also handled most of the making of the GUI. Veeti was responsible for many of the sprites and made the initial map building tool. Kun was responsible for making

the enemies and he also made the wave system and the additional maps. Olavi and Kun worked together to build the level and game classes.

**9. For each week, description of what was done and roughly how many hours were used, for each project member.**

07-13.11.2022:

Everyone: Worked on the plan and familiarized themselves with SFML, approximately **10h** each.

14-20.11.2022:

Olavi: Made the first tower class. **12h**

Veeti: Struggled with SFML libraries not working on his computer. **8h**

Kun: Made the Game class and the first Enemy class, the enemy can move by pressing keyboard **13h**

21-27.11.2022:

Olavi: Made the rest of the tower classes. Also made a Projectile class that the towers shoot. **12h**

Kun: Made the enemy follow the path by predefined route, and refractory the level and levelTile class and add some functions. **10h**

Veeti: Got SFML libraries working. Created the level and leveltile classes. Created the function of building and rendering level layouts. Created the initial Sprite work and made the first levels. **12h**

28.11-04.12.2022:

Olavi: Made it possible to place towers. Made the bomb tower do aoe damage. Changed most of the pointers to shared pointers to improve on the memory management of the game. Moved most of the game's update logic to the level class together with Kun. **12h**

Kun: Add three more types into the Enemy class, and add a simple GUI text to show the money. **10h**

Veeti: Was vibing in a forest. (Military Service) **164h**

05-11.12.2022:

Olavi: Made the GUI for the game, added sound effects, and added a starting screen with the help of Veeti. Added ranges to the towers and fixed many bugs. Wrote parts of the documentation and compiled a Doxygen file of the project. **18h**

Veeti: Created additional artwork for GUI, assisted Olavi via pair programming, worked on documentation and other assistive work. Created the audiofiles used **16h**

Kun: Made the enemy follow the path automatically. Added a simple enemy spawn logic. Made the level and map read from files. **16h**