Tower Defense

Group: tower_defence_santeri_salmela_5
Mikhail Romanov, Aki Oura, Leo Teodosin and Kalle Lindgren

October 2023

Contents

1. Scope of the project

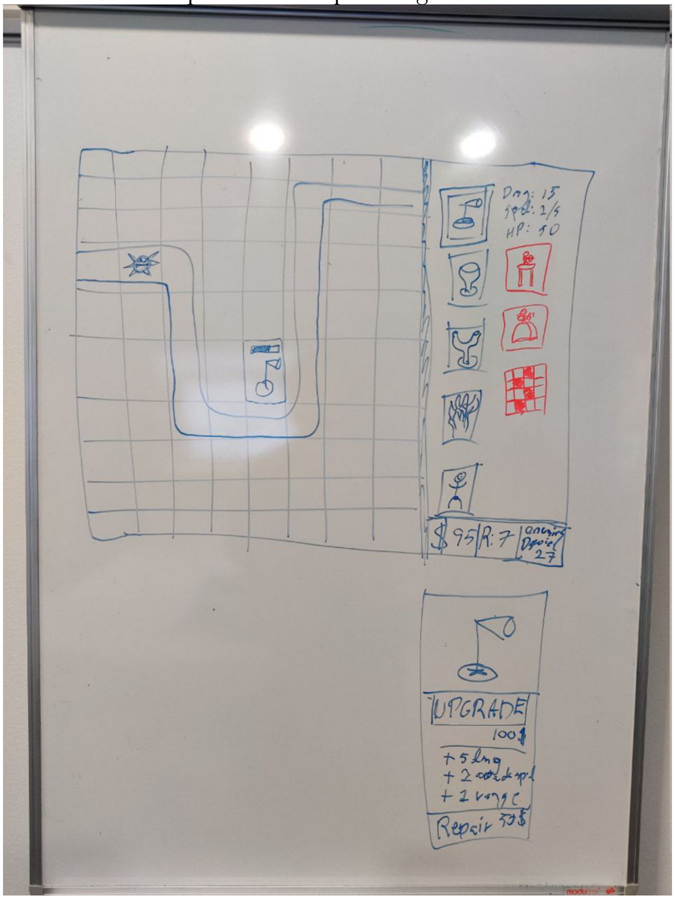
Our scope is to develop fun game that will have a lot of features. The game will have enjoyable graphics and sound effects. Also, the game will have a main menu for the user to navigate between levels.

The game will start with main menu where user can choose level to play or randomly generate fully new one. After that level will start and user will be given some fixed amount of money, and first round will start. Every round there will be some given number of enemies coming to field and users mission is to not allow them to get to the end of the path. Player will be given money for every enemy destroyed and for every round finished. User can win the game by defeating some fixed number of rounds or lose the game by allowing too many enemies to cross the finish line. After winning the game user will have option to continue game, then game will continue until player loses it.

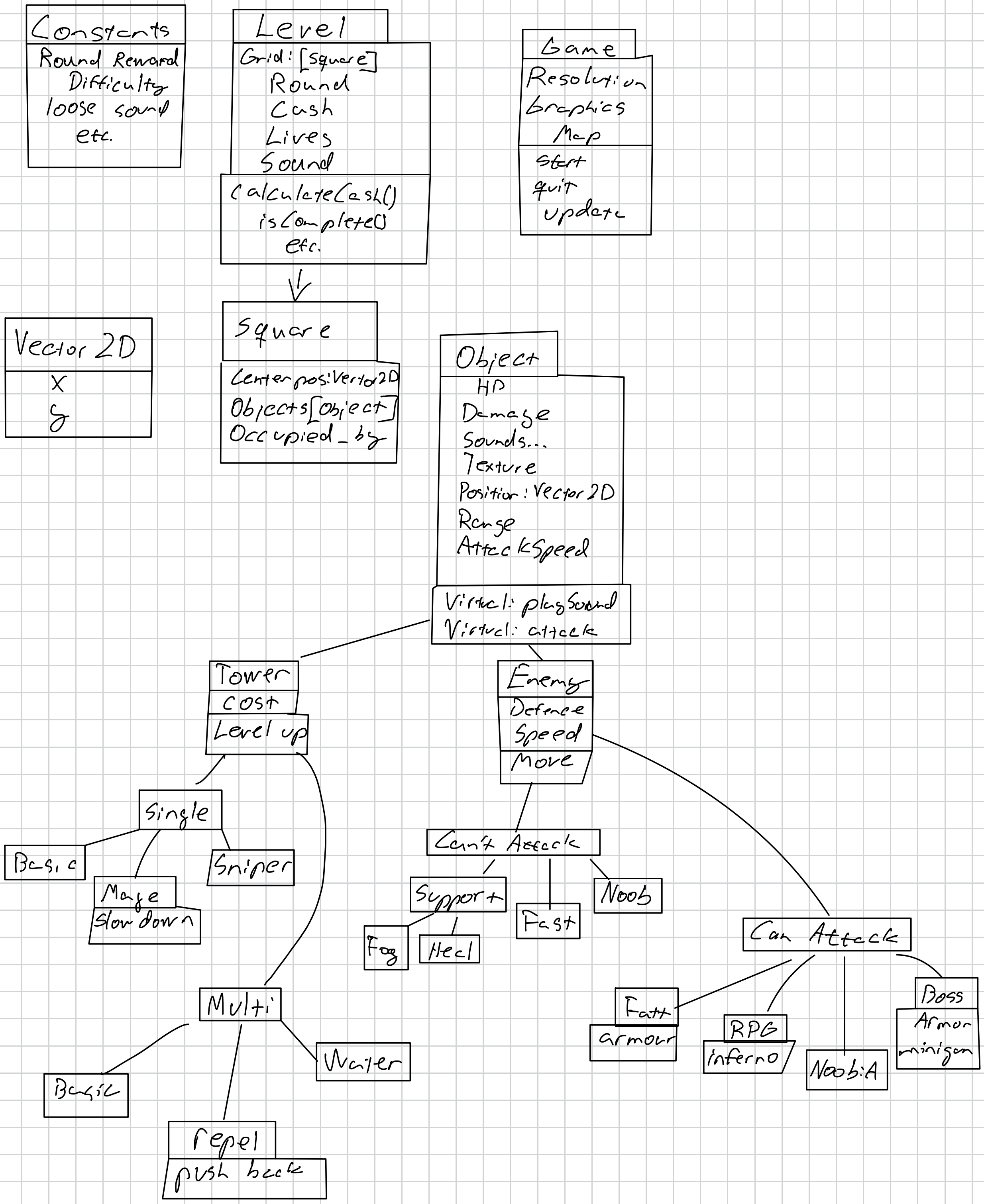
Planned features:

- Main menu
- Audio effects
- Visual effects
- Excellent graphics
- Pre-existing levels
- Random generated levels
- Multiple Tower types
- Multiple Enemy types
- High score
- User interface
- Mouse controls
- Multiple difficulties

Picture of the map that we are planning to make



2. High-level structure



Little description of enemies and towers:

Towers (single): Towers can attack only one target at the time
- Basic: Just basic tower without any special abilities
- Mage: Magic tower that slows down opponents and does extra damage against boss and fat enemies.
- Sniper: Sniper tower has really long attack range and does extra damage against boss and fat enemies.

Towers (multi): Towers can attack multiple targets at the time
- Basic: Just basic tower without any special abilities
- Repel: Tower pushes back opponents when attacking them, does extra damage towards support enemies
- Water: Tower that does extra damage against inferno(rpg) enemies

Enemies (can't attack): Enemies can't attack defenders towers
- Noob: Basic enemy without any abilities
- Fast: Really fast enemy that has only one HP
- Support (Heal): Support enemy that heals other enemies that are close to it
- Support (Fog): Support enemy that puts fog around itself that lowers nearby towers attack range

Enemies (can attack): Enemies that can attack defenders towers
- Noob: Basic enemy without any abilities
- Fat: Enemy that has armor on it that's make it really hard to kill. Deals extra damage towards water tower
- RPG: Enemy that has so kind of inferno ability that deals extra damage towards both basic towers
- Boss: Enemy that is some kind of final boss, it has a lot of power that makes it hard to kill and has some kind of minigun with it that makes a lot of damage towards every tower but towards water tower it makes extra extra damage

Game will consist of single Game-object, which manages rest of the game logic. It will consist of 3 simple methods, one of them starts the game, one quits and one of them updates.

We will have one level class that will have all moving parts stored in it. It will contain current cash and lives situation of the player. Also, class will contain matrix map squares that will contain all objects that are in the that square. Class will multiple functions as calculateCash and isComplete.

Square class will be class for storing all data in specific part of the map. Class will have variables x, y and list of all objects that are in it.

To store all towers and enemies we will have object class. Class will have multiple variables as HP and damage and few virtual functions as playSound and attack. Under object class there will be Tower and Enemy classes. Tower class contains all defense objects and will have variable cost and function level up. Enemy class will be same but for enemies and will have variable defense and speed.

We are planning to make map 10x10 size. We will have pre-existing maps that will be stored in .txt file and can be loaded from main menu. Also, we will have option to randomly generate map.

All basic parameters of the game such are resolution and graphics will be stored in the game class.

3. External Libraries

We are planning to use only one external library
- SFML or Qt

We are planning to use one of upper mentioned libraries, we haven't decided which one.

4. Division of work

Primary plan:
- Object class: Leo and Aki
- Level class: Kalle, Mikhail and (Leo)
- External libraries: Kalle
- Graphics: Aki and (Leo)
- UI / GUI: All

- Sounds: All

5. Planned schedule

Our plan is to start working simultaneously on object and level classes, because object class requires a lot of work and level class is required to test object class. In our opinion that requires three persons. Last person will start figuring out which external library we are going to use and how to connect it to out project. After finishing level class that person will help with object class. External library guy will then start developing UI and GUI. To stay in timing, we are planning to meet weakly to check on the progress and make changes in the plan if someone is not on time.

Plan week by week:

- Week 41 Planning
- Week 43 Finishing plan, starting work on the classes and getting to know external libraries
- Week 44 Basic class coding and implementing grid structure
- Week 45 Class coding and working on basic graphics
- Week 46 Connecting classes and graphics with the grid structure
- Week 47 Finishing touches / Documentation
- Week 48 Polishing up the game