Project plan: Tower Defense

Group 1

Henrik [redacted]

Antti [redacted]

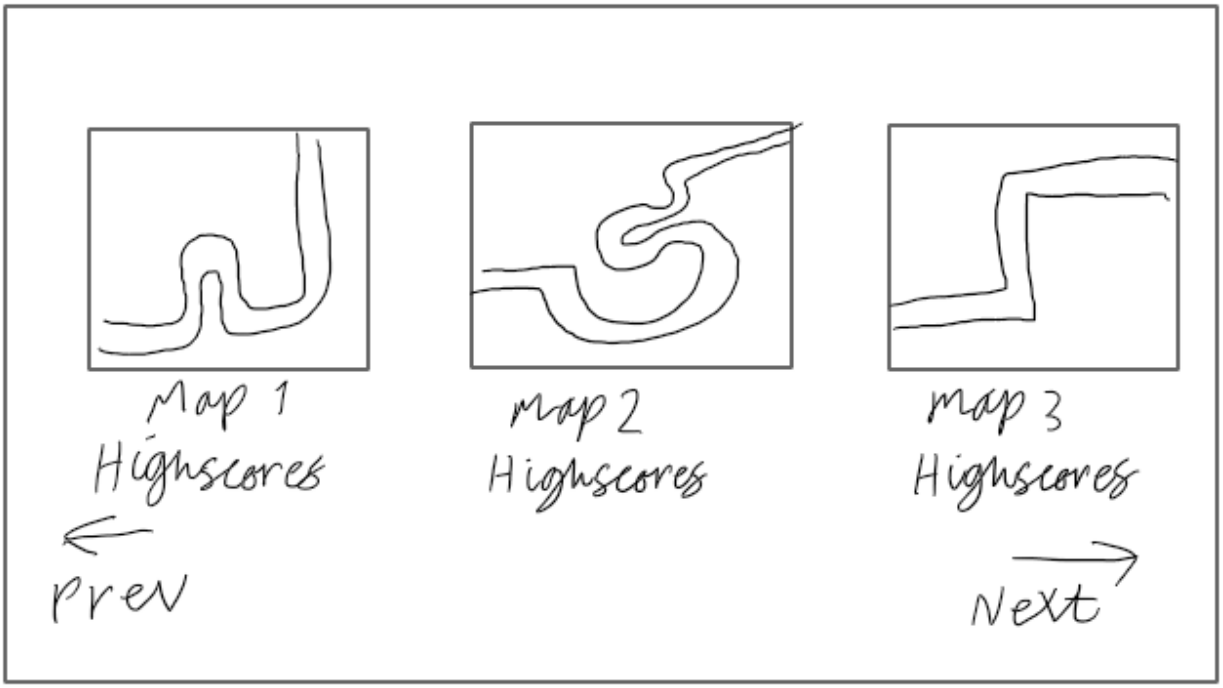
Kasperi [redacted]

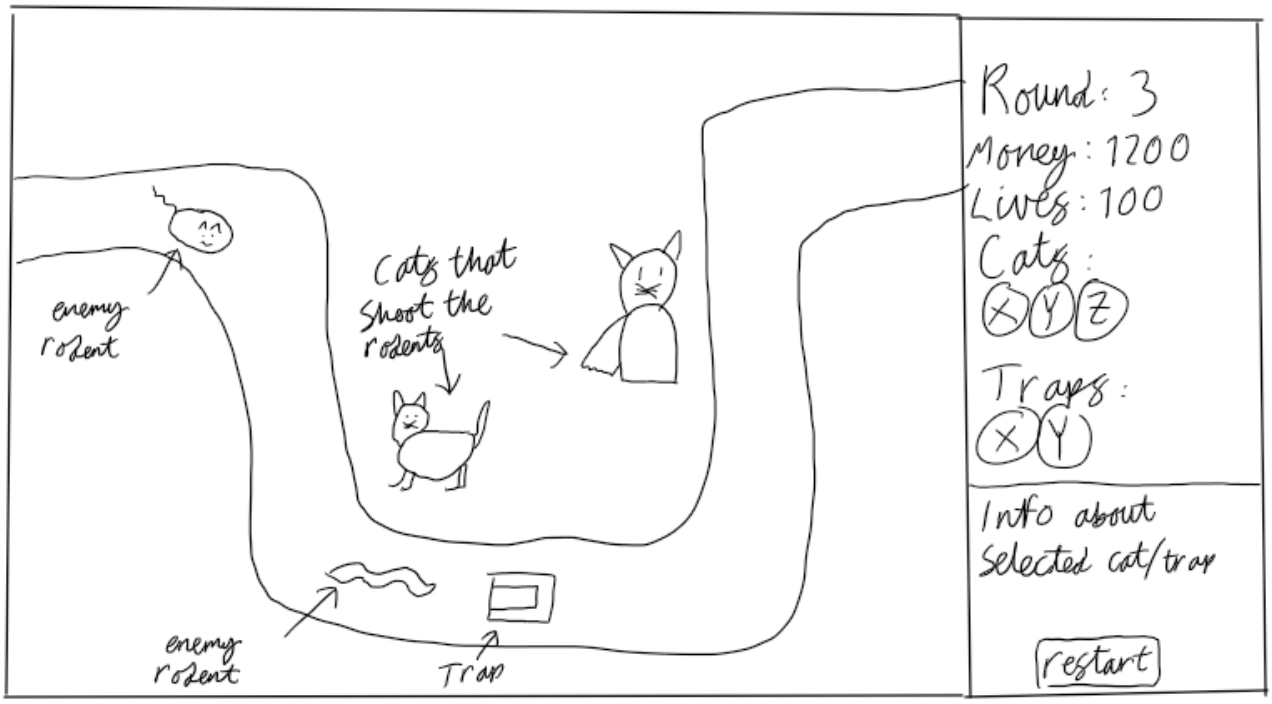
Otso [redacted]

Our project is the Tower Defense game. Our theme is cats who try to defend against mice and different rodents from passing through the path and prevent them from damaging the house at the end of the path.

Scope of the work

The game has a few different maps from which the user can choose as we can see from the rough sketch below. The game has multiple maps, some with branching paths.





The player has the view pictured above where they can place different cats that have different abilities and strengths against different types of rodents that have different hit points. Some of the enemies can be hit only by certain cats as the enemies have different abilities like camouflage. For example, only a sniper cat can kill camouflaged rats. The player can also place traps that are single use. If we manage to implement upgradable towers, there will also be a button to do so.

Below is a list of basic and advanced features that we will implement and a few advanced features that we'd like to implement if we have enough time and skills.

Features:

Basic features:

- Multiple maps, branching paths
- Different types of towers with different ranges and damage levels
- Game is over when player is out of hitpoints. Different enemies deduct different amounts of hitpoints when reaching the end of the path/the player's house.
- Money system, killing enemies gives money, placing/upgrading towers costs money
- Two modes: building towers (no enemies incoming) and a wave of enemies
- At least three different types of towers (planned in our UML diagram)
- At least three different types of enemies (planned in our UML diagram)

- Controlling the game by mouse: user can build/remove towers either between waves of enemies or during them
- Simple user interface that shows information such as resources, the current wave, and available towers and traps

Additional features that we plan to implement:

- Upgradeable towers
- More different kinds of enemies and towers
- Towers can be added, upgraded, and moved during the waves
- A list of high scores (stored locally, per map)
- Sound effects (Cats make noises when attacking and rats when moving and upon their deaths)
- Different attack and defense types (maybe enemies temporarily disable towers, tower that pushes back enemies, etc.)

Additional features that we would like to implement:

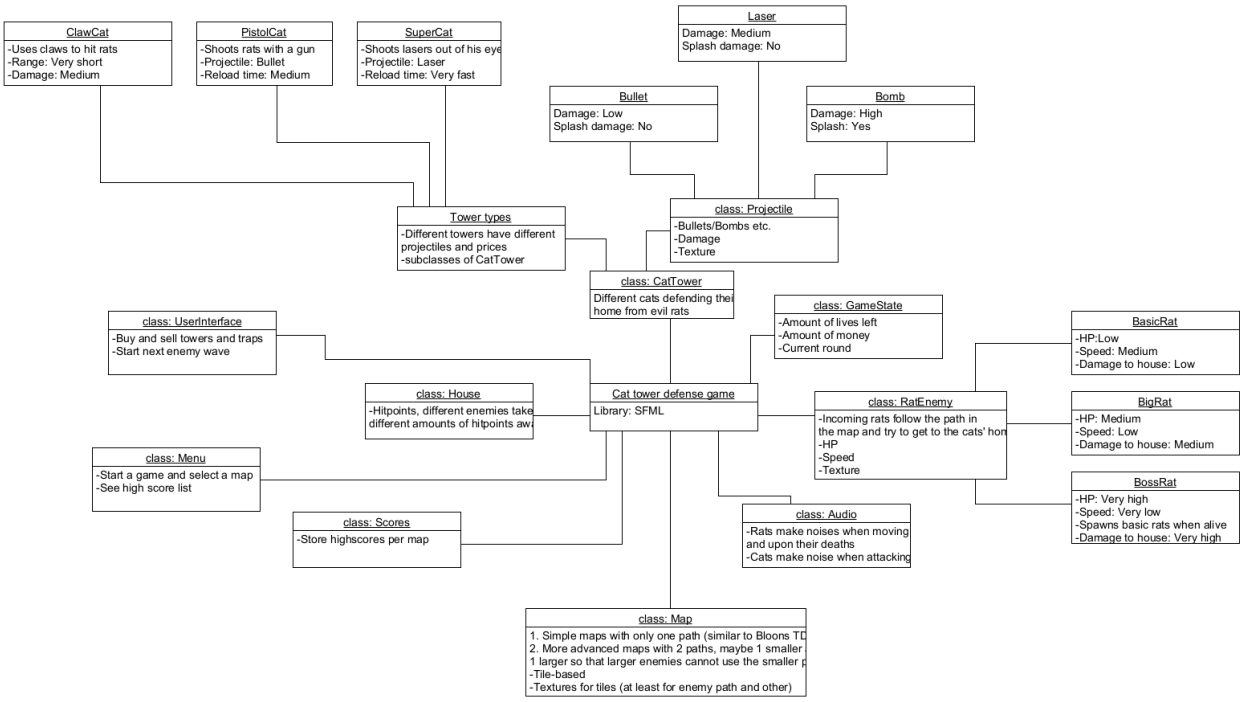
- Game pause menu, which allows saving progress, quitting the game and going back to main menu

High-level structure of the software

We'll create:

- Main classes for the cats (towers) and for the mice and rats (enemies).
- Projectile class for different kinds of weapons like arrows and bombs.
- Class for the "house" that the cats are protecting from the rodents.
- Classes for storing the towers and storing the enemies.
- Game state class that stores the amount of money and lives left.
- Class for the level map.
- Class for storing high scores.

The classes and their relations are depicted in the picture below:



Planned use of external libraries

We'll use the SFML library to create the GUI and sound effects. We chose this library because it's listed in A+ and according to our research it seems to be more beginner friendly than Qt or SDL. We're not aware of us needing any other libraries at this point, SFML seems to be quite fully featured.

Division of work

All the project members will create the base code together, which includes the game engine. This is because it's important that everyone understands how the base code works. We will probably divide the work a little bit in the base code stage too after we have some working content and have a better idea how to utilize the SFML library. After that we'll share the work as follows:

Kasperi	Base code. Enemies and storing the enemies.
Henrik	Base code. Graphics and sounds. Resources class.
Otso	Base code. Main menu screen. Game level class.
Antti	Base code. Towers and storing the towers.

Planned schedule and milestones

Base code working (We have a basic GUI that works, player can place towers and play a wave): Week 47

Basic features working (Graphics, multiple maps, sound effects): Week 48

Advanced features: Week 49

Project final commit to git: Friday, December 09, 2022 at 23:59