ELEC-A7151 - Object oriented programming with C++

Project Plan

Tower Defence 7
No roaches allowed!

Team Members:
Navid Ahmad, 796974
Zhong Xin, 875167
Narayan Ghimire, 876470
Martin Koskinen, 789567

Contents

Introduction

Game is about defending human settlements in the Mars from mutated alien cockroaches. Player's task is to fend off waves of enemy cockroaches from entering the settlement by strategically placing defensive towers along the path of attack. The game is lost when any cockroach reaches the end of path. The defensive towers have different types of bullets, with defined level of damage to cockroaches' health. It will be increasing tricky when player moves to higher level where the enemies have longer health and split function. The player will have default money in wallet to buy towers at the start. As more cockroaches are killed, more money will be stored in wallet as reward. Player can buy new towers in shop according to their prices and sell the towers to shop to get money back.

Project scope

Monetary system

Elon is the currency used in Mars based on the first colonizer Elon Musk. Elons are required to buy towers and are awarded for destroying cockroaches and selling towers.

Enemies

In this game, enemies are different type of cockroaches which try to reach the human settlement entrance. These cockroaches follow a non-branched path to the base. There will three different types of monsters which are listed below:

- Baby roach– A small cockroach with default speed. When killed gives 100 Elons.
- Male roach – A large cockroach with increased health and lower speed. When killed gives 200 Elons.
- Mother roach – The largest cockroach with same health as Male cockroach and default speed. However, when Mother roach is killed, it splits into 4 Baby roaches. When killed provide 100 Elons.

Towers

The towers are military defence structures developed by the Martian Defence Forces. There will be five towers which destroy the cockroaches trying to reach the human base. The tower types are described below:

- Turret gun tower – A semi-automatic turret with slow rate of fire. The turret shoots once at a time when the cockroaches are within the tower's standard range. This tower costs 10 000 Elons.
- Machinegun tower – A automatic turret with high fire rate. The turret shoots cockroaches within its medium range. This towers costs 20 000 Elons.
- Missile tower – A high damage missile projectile tower. This tower has long range shooting and damages the cockroaches within a blast radius. This tower costs 30 000 Elons.
- Ice tower – A ice gun tower which cause frost damage and slow down cockroaches within a low range. This tower costs 15 000 Elons.
- Flamethrower tower – A low range tower that shoots flame on the cockroaches. Cockroaches affected by the tower will start burning for a finite time and lose health every second they are burning. Costs 35 000 Elons.

Map and Level

Maps will have rocky Martian terrain with road which the enemies can follow. The player can put the defence tower next to the road however not on top of existing structures such as rocks or another tower etc. The maps will vary as the player advances each level. Each game level has a mixed set of enemies, a unique path for enemies to flow, and fixed amount of money at the start.

User Interface (UI)

The main window (Figure 1) will display the plot the of the game and once start is pressed the game window will open. The game UI (Figure 2) will show the available money and the tower shop menu on the top. The map will be visible where the player can buy and put the towers in the map. There will also be a "start wave" button which must be pressed for the wave to start.

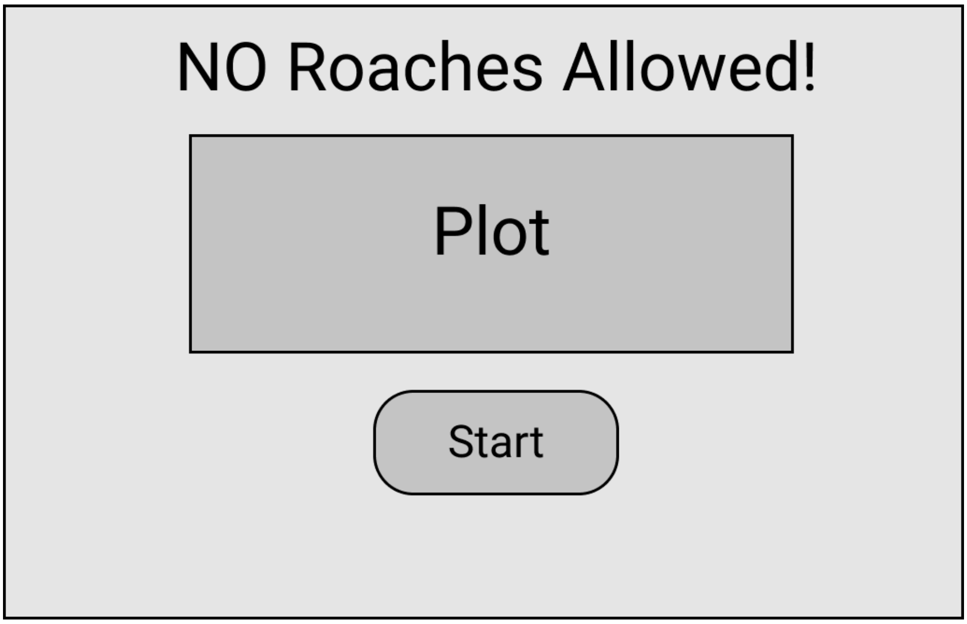

Figure 1. Main Window

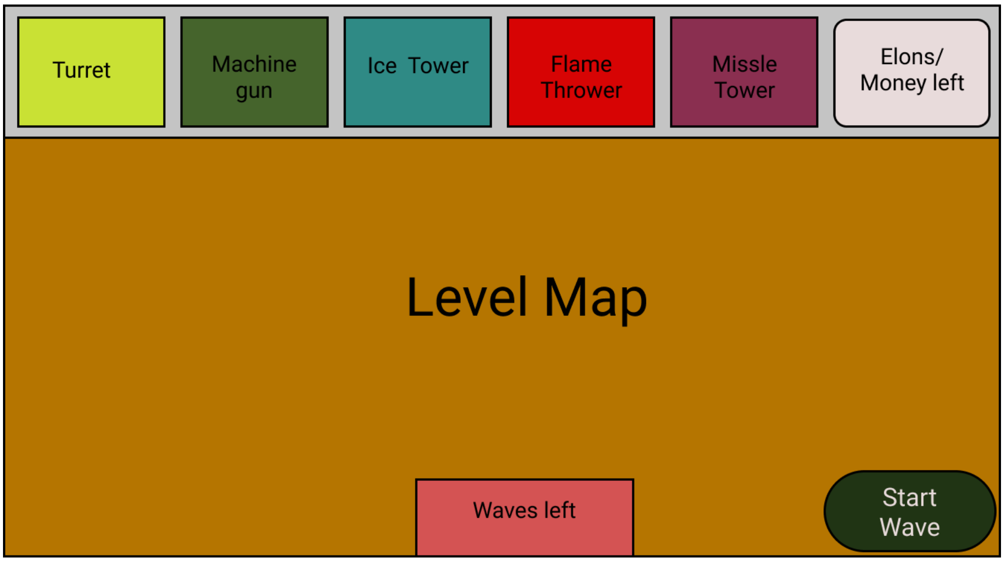

Figure 2. Game Window

High level software design

The following classes in the game

- Abstract class Cockroach
  - Derived class Baby roach
  - Derived class Male roach
  - Derived class Mother roach

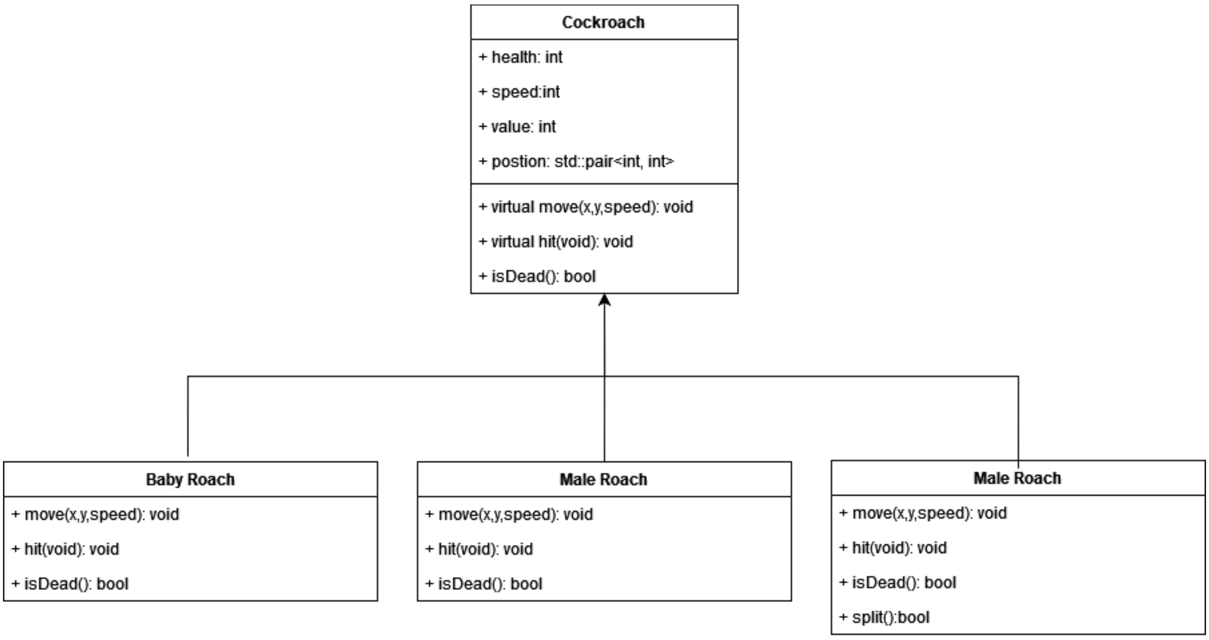

Figure 3. Cockroach Class

- Abstract class Tower
  - Derived class Turret gun tower
  - Derived class Machine gun tower
  - Derived class Missile tower
  - Derived class Ice tower
  - Derived class Flamethrower tower

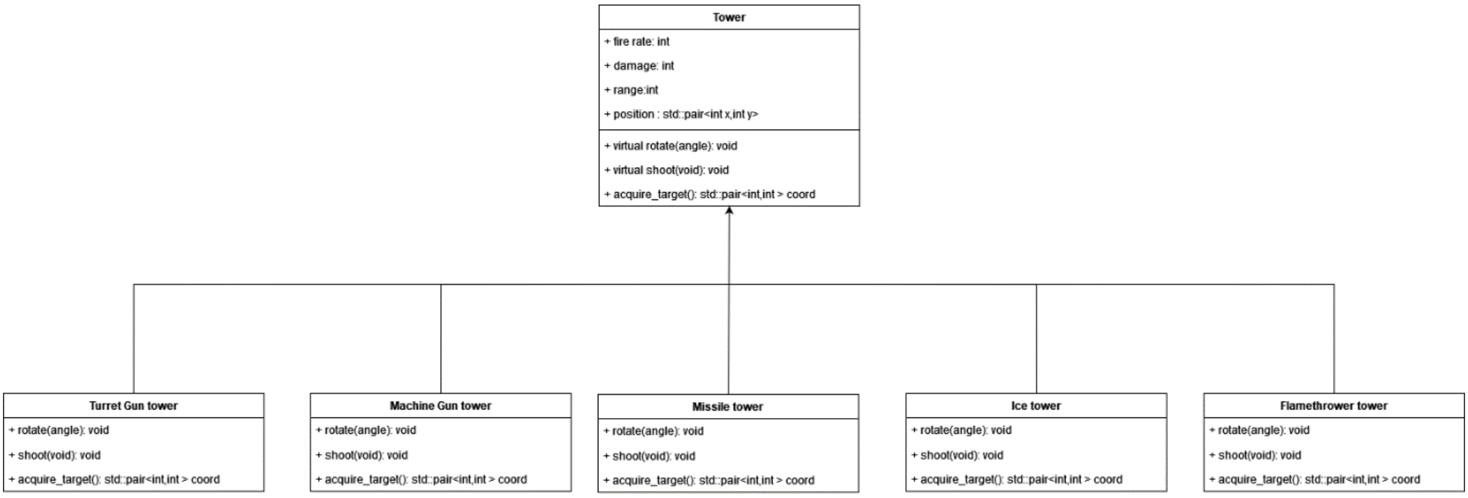
Tower
+ fire rate: int
+ damage: int
+ range:int
+ position : std::pair<int x,int y>
+ virtual rotate(angle): void
+ virtual shoot(void): void
+ acquire_target(): std::pair<int,int > coord
Turret Gun tower
+ rotate(angle): void
+ shoot(void): void
+ acquire_target(): std::pair<int,int > coord
Machine Gun tower
+ rotate(angle): void
+ shoot(void): void
+ acquire_target(): std::pair<int,int > coord
Missile tower
+ rotate(angle): void
+ shoot(void): void
+ acquire_target(): std::pair<int,int > coord
Ice tower
+ rotate(angle): void
+ shoot(void): void
+ acquire_target(): std::pair<int,int > coord
Flamethrower tower
+ rotate(angle): void
+ shoot(void): void
+ acquire_target(): std::pair<int,int > coord

Figure 4. Tower Class

- Wallet class
    - Contains the current balance of the player's money. Has methods of add and take for money.

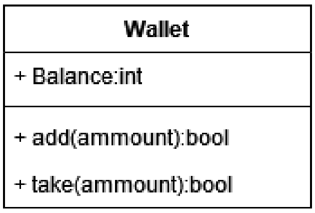

Figure 4. Wallet Class

- Shop class
    - Has methods for buying and selling towers. Once user buys a tower, a new tower object will be created. If user sells a tower, the tower object will be destroyed.

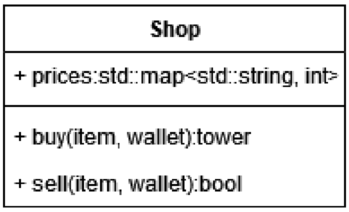
Shop
+ prices:std::map<std::string, int>
+ buy(item, wallet):tower
+ sell(item, wallet):bool

Figure 6. Shop Class

- Map class
    - Contains for example information about background or paths. (May not be necessary)
    - Could have information about available spaces where towers can be placed.

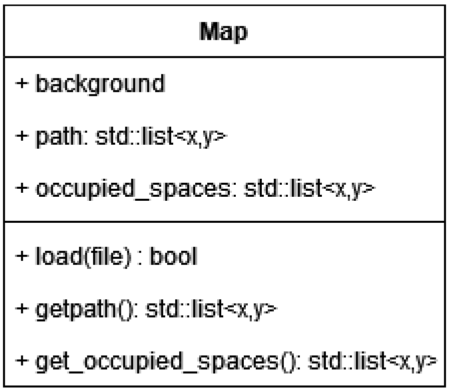
Map
+ background
+ path: std::list<x,y>
+ occupied_spaces: std::list<x,y>
+ load(file) : bool
+ getpath(): std::list<x,y>
+ get_occupied_spaces(): std::list<x,y>

Figure 7. Map Class

- Sound class
    - Contains background music, soundtrack of firing/explosion, killing, tower rotating & implementation and wallet charging & paying.

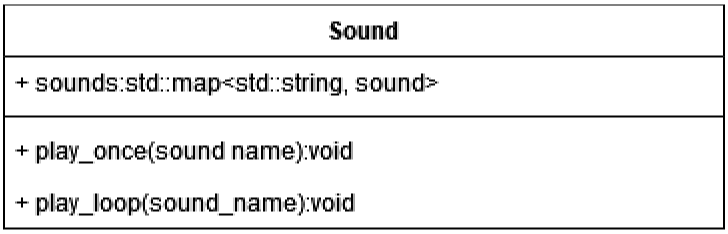
Sound
+ sounds:std::map<std::string, sound>
+ play_once(sound name):void
+ play_loop(sound_name):void

Figure 8. Sound Class

- UI class
    - Contains all visible elements in game windows: player's wallet, maps, paths, towers, cockroaches, sound e.g.

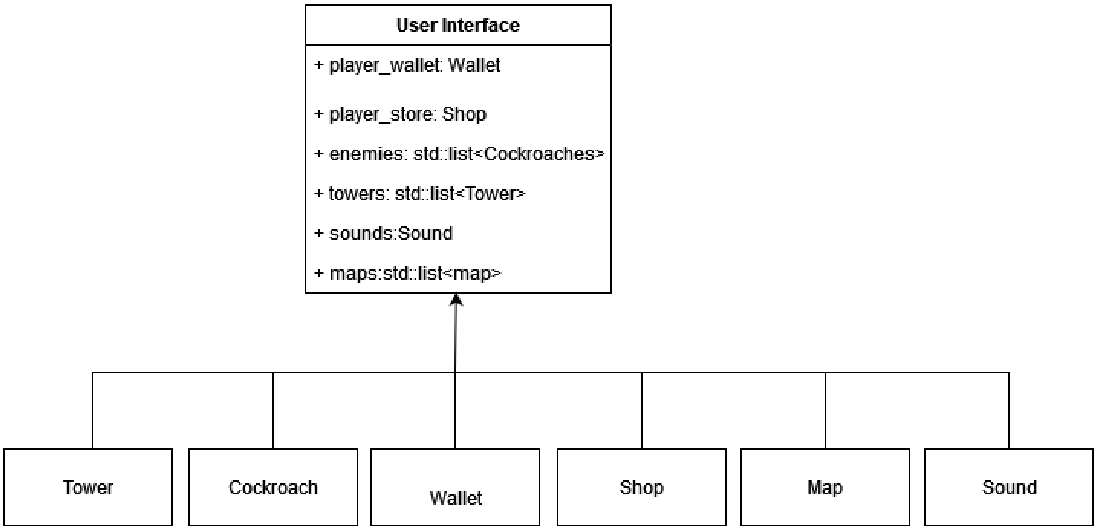

Figure 9. User Interface Class

Planned use of external libraries

Libraries

- QT (Graphics and game mechanics)
- Multimedia libraries e.g SNML

Work Division

Note: Initial division of work might chance after getting started and the full scope of the project becomes clearer.

- Graphics and Mechanics – Navid, Martin
- Physics – Narayan, Martin
- Models – Navid, Martin
- UI – Xin, Narayan
- Sounds – Xin, Narayan

Planned Schedule

- Week 1:
  - Project planning
  - Setup development environments
- Week 2:
  - Setup main UI
  - Scenery (road, rocks, etc)
  - Models (cockroaches, towers, projectiles etc)
- Week 3:
  - Physics/Movements
  - Shape interaction
- Week 4:
  - Game Mechanics
- Week 5:
  - Testing & debugging
  - Sound Effects
  - Generate different type of maps (according to the game level)
- Week 6:
  - Documentation