Tower Defense Project Plan

By: Aditya, Xiong, Dung, Nguyen

Game description:

Our project for the Object oriented programming course in C++ is a tower defense game. The goal of the game is to survive 10 rounds of enemies with increasing difficulty each round in order to win the game. The enemies will attempt to reach the end of the predetermined path and the goal of the player is to stop any of them from doing so. Failure of such will result in a loss. The player may achieve this through the building of towers in exchange of credits, which is earned through defeating enemies. Towers are also upgradable to deal with enemies in the later parts of the game. There are multiple types of enemies, towers and functionalities like health points, attack points and attack radius that will keep the player interested in the game and adequately increase the game's difficulty per round.

This is our intended look/UI for the game :

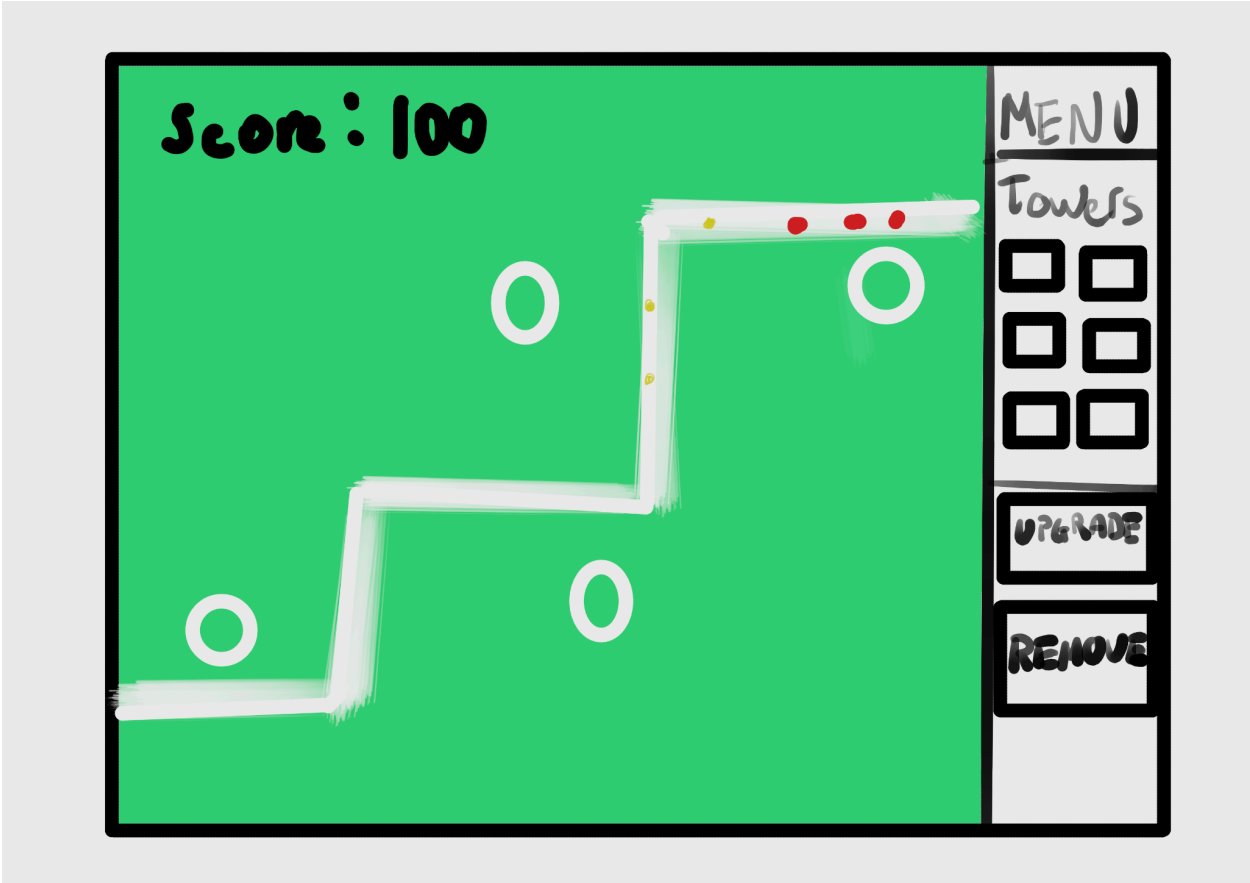

Fig.1 Intended GUI

As can be seen from the illustration, we also intend to include a score system. While not fully decided, we intend to use a timer-based bonus system alongside increasing the score as per damage dealt to enemies.

Schedule plan & milestones:

Week No.	Implementation for the week
Week 1	We will implement the basic modules for the following - Tower class and its subclasses - Enemy class and its subclasses - Tower and Enemy classes which have an HP/XP/Special attack statistics system implemented - Map class We wish to have the basic structures of the program established by this week.
Week 2	- Graphics with UI to showcase actual gameplay - Sound effects - Implementation of rounds and pause between each round to allow user to change setup of towers, including upgrades and relocation We want a basic albeit working prototype of our program by week 2.
Week 3	- Ability to create custom levels (Level editor) - Ability to save highscores of game with a specific username - Upgradable towers We want to complete implementing advanced features into the program by week 3.

Week 4	- General code optimization and bug squashing - Optimizing gameplay satisfaction - Project documentation Our project should be practically complete by this week. Furthermore, we will create the project documentation PDF and explain the game's code structure thoroughly, alongside other topics mentioned.
Week 5	- Similar to week 4 - Project demonstration We will present the game and showcase its features and polished setup.

Program design:

The main classes of the game will include a tower class, an enemy class, and a map class. Each class is developed independently in the file with the same name.

Tower class includes towers that shoot at the closest enemy according to the path. Different towers have different ranges and different damage rates. More specifically, the tower class has the following subclasses:

- Plant type (basic tower, can only attack one enemy at a time with set damage)
- Fire type (damage over set time to multiple enemies)
- Water type (slows enemies in their path, does not deal significant damage)
- Bomb type (Attacks multiple enemies at the same time)
- Ground type (Blocks enemies in their path)
- Magic type (Gives other towers bonus health or attack points)

Enemy class includes various types of enemies that can and cannot attack towers, different enemies also have different passive, health, and strength. Enemy subclasses are:

- Plant type (basic enemy, can deal basic damage)

- Tree type (similar to plant type, but splits into multiple plant types enemies when its health reaches 0)
- Fire type (can deal damage over time to towers)
- Bomb type (can attack multiple towers)
- Magic type (Gives other enemies bonus health or attack points)
- Ground type (blocks towers from dealing damage)
- Water type (Makes towers only deal 50% damage as long as it is alive)
- Boss type (Can kill towers in one hit and has a lot of HP, but moves very slowly and has smaller detection radius)

Map class implements the function to read saved maps from a text file. This is how the 10 preset levels will be saved and played. The map design will be based on a 16x16 grid.

The UI of the game will be implemented in the main.cpp file. It shows the amount of currency the player has, and their score, alongside the main game that shows the tower's in a 16x16 physical grid attacking enemies. Users can track the tower's health and attack points and decide to upgrade/replace them as the game progresses. Al

We also will provide a level editor mode, where users can create their own enemy paths as per their wishes. They can load previously created levels from files and save newly created ones as well.

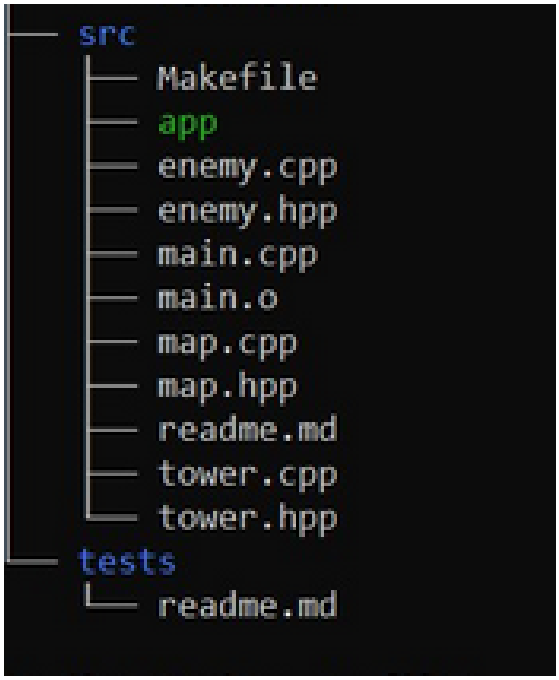

Fig.2 Main modules of the project

The project will make use of the Simple and Fast Multimedia Library (SFML) for graphics, audio, and processing mouse inputs.

Division of work:

This will be discussed and decided at the start of each week.

First week	
Name	Responsibility
Aditya Agrawal	Implementing the game UI
Dung Nguyen	Map class
Nguyen Hoang	Basic Enemy class
Xiong Xiong	Basic Tower class structure