CS135601 Introduction to Programming (II)

MiniProject 2

1. Goal

In the project2, you are asked to extend the Tower Defense game and learn the following skills:

- Understand Allegro5 game development process.
- Get familiar with the OOP concept and the code structure.
- Using Allegro5 and C++ to develop a game.

2. Problem Description

In this game, you need to place the turrets to destroy all the enemies through the enemy wave to win the game.

The game consists of two main components:

1. Enemies: Enemy waves are defined in the enemy*.txt file in the Resource/ folder
2. Turret: Placed by the player.

In the playing scene:

- Press key 0-9 to change the speed multiplier.
- Press Q , W to perform a quick select on different armies.
- Press M can mute / unmute the bgm.
- Click on the empty spaces in the map to place the selected turret.

The rule of turrets placement:

- Should not be placed on the path where the enemies pass by.
- Turrets cannot be placed where another turret has already been placed.

If you finish the hackathon, all the functions should work properly.

3. Code Requirements (Add new features)

1. Add a new enemy. (1%) (Demo video)

a. The enemy has high HP which can resist attack for a while.

b. When the enemy is destroyed, it will generate a dice enemy which is already implemented in hackathon.

2. Add a new turret (1%) (Demo video)

    a. The icon of the turret will not show on the right bar, and it cannot be placed directly. The turret can only be generated by placing two machineGunTurret at the same place.

    b. It shoots 2 bullets at once, with higher damage than the machineGunTurret.

    Place two machineGunTurret in a same place

    Upgrade to the new turret

    Shoot 2 bullets at once

3. Add a new turret with new bullet (1%) (Demo video)

    a. The turret won't track enemies and attack them. Instead, the turret creates bullets that move in a circular orbit, causing damage and disappearing when hitting enemies.

    b. The movement of bullets can be either clockwise or counterclockwise.

    c. There are initially 4 bullets, and they will be reloaded only after all bullets have disappeared.

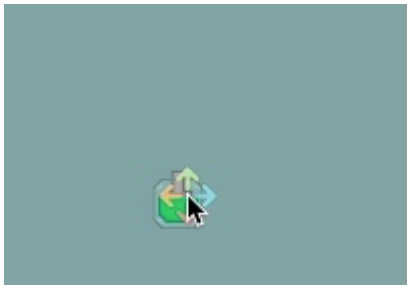
    d. When the turret is removed, the bullets should disappear as well.

e. Bullets should not be generated when the turret is moving.

New turret (bullets move in a circular orbit)

4. Add a new object - shovel (1%) (Demo video)
    a. The shovel can remove the turret and return half of the money.
    b. The shovel preview will be removed when clicking it at the right bar (same as turrets).
5. Add a new object - shifter (1%) (Demo video)
    a. The shifter can move the position of a turret.
    b. Placing a turret using the shifter will not deduct money.
    c. When the moving turret preview is clicked at the right bar, it will be directly removed with no money returned.
    d. The shifter preview will be removed when clicking it at the right bar (same as turrets).

Shifter on a turret



The placed turret becomes movable

6. Do extra functions that are not mentioned in the above. (1%)

Please use your creativity to do what you want. If you don't have any idea then you can implement the following examples:

EX1: Create an enemy boss which can generate enemies periodically .

EX2: Create a spell which can enhance the attack speed and damage of the turrets in the spell effect region.

4. DEMO

Make sure you finish all the requirements above and demo the game to TA to get the points of miniProject 2. Otherwise, you might receive a points deduction.

Same as the hackathon, you have 10 minutes to demo the work, you need to show all the requirements above to get points of miniProject 2. Notice that TAs will score only by your demonstration, so make sure you prepare before the demo. Otherwise, you might get points deduction for missing showing some of the requirements.

[NOTE] If there's still some parts of the hackathon that you have not demoed, make sure you also demo them to get the points of the hackathon.