Team: Potato Power
Alitquan Mallick, Arif Roktim, Shakil Rafi
Period 4

Final Project Proposal

Tower Defense Game

Our final project is a tower defense game. We will use processing for our GUI and stacks, queues, and quadtrees for other features.

A tower defense game is a game where the user places “towers” in designated locations along a path with the intent of preventing enemies from reaching a destination. The different towers attack enemies using a variety of weapons, each tower and weapon having its own advantages/disadvantages.

Features and Implementation Details

Towers:

- Different types of towers (ex: single target, splash damage, long range)
- Player can buy upgrades for their towers
- Targeting:
    - The towers will use a queue and a stack to choose which enemies to target. Enemies in the stack will receive priority over those in the queue for targeting enemies
    - Enemies will be enqueued onto the front of the queue based on which enemies come into range of the tower first
        - Use quadtrees to efficiently retrieve the enemies closest to the tower
    - User will have the option to manually override the tower’s target. The enemies will be pushed onto the front of a stack in the order in which they were targeted. The tower will target the object at front of the stack. The enemies will be popped off the stack when they die

Enemies:

- Enemies will spawn according to the order of the spawn queue. Depending on the level and difficulty, the frequency, order, and strength of enemies will vary.
- Different enemies will differ in attributes such as speed and health
- Some enemies will have several states or phases. An example is: An armored enemy becomes a frost resistant enemy when his hp reaches a certain threshold. Then after reaching another threshold, his speed increases.

Efficient Collision detection with quadtrees:

- Main reference
- Checking to see if every object is colliding with every other object is very expensive in terms of runtime efficiency.
- Instead, use a QuadTree to narrow down the number of objects that are checked. Use the concept that items on the opposite sides of the screen can't be colliding

Extra features if we have extra time:

- Level designer
- Scoreboard
- Cheats menu

Minimum Viable Product:

- Player can place towers on a map
- Enemies move along the fixed path(s) toward their destination
- Towers target and shoot nearby enemies
- Enemies collide with and are damaged by projectiles
- Player loses is 3 enemies reach end of map
- Player wins if all enemies have spawned and are dead