Quinn Roemer & Logan Hollmer

Dr. Daryl Posnett

CSC 133 MWF 9-9:50am

March 27, 2020

Tower Defense - Initial Report

We decided to program the Space Defense game as our project this semester. This game design features a space themed esthetic that will be demonstrated by the maps, enemies, and towers. We started implementing this game by defining a GameEngine class that will be used to form the backbone of the project. This controls the flow of the game and the interaction. In addition, we are using a GameWorld class as a container for all of the game objects and a GameView class to handle the drawing of every object on the screen.

We started implementing some of the features of this game by programming the Drone enemy type. We plan to use a factory design pattern to create different aliens with different stats and resistances. This will benefit us by allowing the use of a shared data structure for all enemy types. Simplifying collision detection and movement. Aliens will move according to a movement strategy that will be chosen during the instantiation of the alien object. The chosen strategy will be based on the enemy type. The fastest will be the Drone followed by the Soldiers and Behemoths respectively. The map will be held inside of a dedicated game object. This will define the background to be used, the path to be followed by the aliens, the location of the base you are trying to protect, and the initial spawn point of the aliens. As of today, we currently have the Drone enemy defined, and spawning at a certain location on a map that is drawn at the beginning of our draw loop. This will be modified to conform to the aforementioned standards as we develop the game. All of these mentioned methods inherit from an abstract GameObject with the aliens further extending an abstract Movable class. The movable class has a heading and a speed which is used to control how the objects move. We also decided to implement the “lowest” level tower which was the “plasma turret”. When a tower is built it is given a default set of stats based on the tower type, these stats are held in a container that is owned by the tower. We plan on using a state pattern in the future to update the state of the tower to represent the tower being upgraded. Right now the plan is to use a factory pattern to create the towers. A tower also holds the type of projectile it has. We have also created a “projectile” class that extends movable. When a tower “shoots” a projectile object is created and moves towards the specified location. A projectile will be removed if it exits the range of the tower or if it hits an alien. Upon collision with an alien, it will deal a specified amount of damage to the alien that was hit. Finally we plan on implementing sound using a strategy pattern.

Another important aspect of the game is the user interface. We are not using the java button classes, rather each “button” on the screen is a drawn rectangle with an image. The towers will also have “invisible” rectangles drawn over them so they act like buttons to display information when touched. We have implemented an observer pattern that waits till there is a touch event and checks to see if any of the buttons or towers were clicked. If so, the appropriate action is taken. We currently have two buttons in our game, a play/pause button and a button to place a tower. When the tower button is clicked then a tower will be placed at the next location the user touches.