Essa Saeed

Mr. Schattman

ICS 3UI

Jan. 24, 2014

Design document for the computer game *Tower Defense*

# Game description:

Tower defense is a strategy defense game. In tower defense, players use resources to purchase and build defensive turrets in strategic locations around a set path which enemies follow to get into the player’s territory. The objective of the game is to set up all your defenses such that no enemy force can proceed past the turrets into your territory.

To begin, the user utilizes his starting resources to purchase a first turret, and place it where he sees fit. Then proceeding to the game, waves of enemies will try to get past your defenses and into your territory. Player starts with:

* $100(value subject to change) to use in order to purchase a turret
* Multiple locations to build turrets
* 3 lives(value subject to change)

Two building rules:

* The turrets may not overlap
* The user may not build anything on the path

Game ends when:

* User wins when all enemies have been defeated
* User loses when he quits or loses all lives

# Sketch of the game screen:

The game screen will look something like this.

|  |
| --- |
| C:\Users\Owner\Desktop\untitled.jpg |

# Instructions for play:

To open the game, open the Python file **TowerDefense.py** and push F5.

You will see a background with a grid overtop, and a path running through the background. The enemies follow the path, and when placed, the turrets defend your territory.

To place a turret/tower, drag the turret’s icon from the bottom of the screen (the shop), to where you want to place it. The range is shown surrounding the turret to tell the user where turrets should be placed for optimal efficiency, the range indicates how close an enemy unit needs to be before the turret attacks. So long as the turret is off the brown/orange path, it should be placed where the mouse click is released and the correct amount of money should be deducted from your total.

The enemies will come in waves and the user will have to strategically place turrets in order to take out the enemies in the most effective way possible. If an enemy gets into the user’s territory, one life/heart will be lost, if there are no hearts remaining, the game will end and a “Game Over” screen will appear.

To quit, press the **“Q”** key on the keyboard.

Warning: Clicking **Quit** permanently erases the current game position, as there is no save function.

# Pseudocode algorithm for the runGame() procedure:

Draw the player’s screen with path #green for buildable locations, brown/orange for the path, and gridlines each 50 pixels apart

Player is allowed to place his/her turrets as enemies begin to spawn

Enemy waves follow the path towards the user’s territory

**While** player still has at least 1 life **and** user has not quit: #will allow the game to continue until it is no longer possible

Player chooses a location to place turret #if they want

Player’s total money decreases #amount depends on the type of turret placed

Determine if enemy gets within turret range #to know whether or not to shoot and deal damage

**If** enemy is in range of at least one turret:

A projectile is fired #from the corresponding turret to the corresponding enemy unit

Reduce that enemy’s health by 1 #or another value that corresponds to the type of turret

Enemies continue moving #they don’t ever stop until they have been eliminated

**If** that enemy’s health = 0:

Reduce the computer’s enemy wave count by 1

Destroy enemy unit

Increase money by a set amount #depending on the type of enemy

**Else**:

Enemy units continue to move

Continue to check if any enemies are within turret range #if they are then re-do the above

Continue to check if any enemies have gotten into your territory

**If** enemy unit is in your territory: #When an enemy gets into your territory one life is lost on screen

Lives count reduced by one

**If** LivesCount = 0 **or** PlayerQuit == **True**:

Show “Game Over” screen #User loses

**If** EnemyUnits= 0:

Show “Congratulations” screen #User wins