2023

LZSCC 306 Tower Defense

38553767

KAVUNCU, TOLGA (STUDENT)

TABLE OF CONTENTS

EXERCISE 1: USER ACTIONS

My implementation follows the requirements in the coursework document as closely as possible, In this document I will explain the parts that I deviate from the requirements and explain my reasoning behind them.

For the first part of the game implementation, I chose to use 60 frames per second instead of the 50fps in the requirements document, this was done because I felt like the 60fps mark felt smoother on the screen in opposed to 50. This however does not have a huge impact on the gameplay or other parts of the game itself.

Other than the frame rate, the other requirements are met. The game initially starts with the user having 100 gold (enough for 2 towers). The game start state is false before the player starts it. The tower chosen is shown as text and can be null (which has no impact on the canvas). The game starts with the player at level 1. It is increased at the first kill and then every 10 kills after that. The current player level decides how many monsters are going to be on the next wave.

The canvas is drawn with a light green background to simulate a grassy green area with a black path for the monsters to follow in it in a snaky pattern that utilizes most of the canvas space.

A tower class is implemented that has 3 different tower types; each tower has a different effect on the monsters. If the user has 50 gold and the location, they clicked already does not have a tower a tower will be created, and 50 gold will be taken from the players gold stash.

The towers are drawn using their draw method. They are constantly being redrawn in the main loop in the game. Each type has their own colors. Regular is blue, ice is white, and fire is red.

EXERCISE 2: MONSTER BEHAVIOUR

My implementation of the monster path is defining the corner locations of the path in the canvas, which helps me decide the shape of the monster path. And later on, in the monster path class the points in between the corners I decided are drawn and added into the monster path. A monster can follow the path with just the index of the current pixel the path it has. Since each path block is indexed the follow-up position for the next block is just 1 index higher. This makes it easier for me to implement monster movement.

Each monster has a position initially just outside the canvas. 100 health points and a monster path which they follow to reach their final position. The monsters are updated which refreshed their next position and their current health. The monsters are drawn as purple pixels that follow the black monster path that is pre-defined. Monster speed can also be changed for balance as it is just a variable in their class.

If a monster manages to get to the last index of the path. The player life is decreased by 1. If the player life goes below 0 the game resets and can be restart by the start button again.

Each monster kill gives the player 10 gold for their future purchases.

EXERCISE 3: SHOT BEHAVIOUR

The shot class implementation is very close the requirements document once again. Each shot has a type, a position, a goal which is a monster and color.

Towers have a cooldown of 1 second for each shot, which is 60 frames in my implementation. This is updated and counted down each iteration of the main loop of the game. During the time the tower is in cooldown. The towers are unable to shoot their targets.

The distance from the tower to the monster is being calculated for each tower in each iteration of the loop. There is a threshold distance for the towers range. If the monster is closer to the tower than the threshold distance a new shot is created and shot towards the targeted monster.

Shots are drawn as fat lines the same color as the type of tower that they are shot from. Each shot can deal different amounts of damage to the monsters according to their types. After they collide with their target monster the shots are deleted. If the monster is defeated at the same time the monster is also killed. If the shot is not enough to kill the monster they are shot again and again until their health reaches 0.

Ice towers deal less damage than regular towers do. However, they freeze the monster in place for 3 seconds. 5hp

Regular towers just deal damage, at a consistent rate. 10hp

Fire towers deal less damage than regular towers. However, they burn the monster over time in a stackable manner for 3 seconds. Meaning if the same monster is shot twice, it is burning twice. (8hp + 2hp per second during the burn).

EXERCISE 4: ADDITIONAL FEATURES

My tower defense implementation has 2 different additional features. One of them is the example given in the coursework specification document. Players are allowed to buy 1 life for 100 gold.

Another addition I made is an option to kill all current monsters for 1000 gold. This is a powerful ability that can save lives. That is why it is more expensive than the other one. This works by changing the state of every monster currently on the canvas to being defeated.

DESIGN FLAWS AND BUGS

My implementation of the tower defense game is playable but still flawed in some ways. I will write a few that comes to my mind, and I hope you can see my attempts on fixing these.

1. You can place towers on the monster path.
2. Only the tower on the top left corner of the canvas will function as intended, the rest of the towers on the canvas for some reason just refuse to shoot shots. They are actual towers as you can see them in the towers list, and you are not allowed to place towers on top of them.
3. The tower on the top left corner of the canvas moves when it defeats a monster. When it moves you can place another tower on the same pixel and that will also function as intended. So please while testing keep that in mind and add more towers to that corner on the green part.
4. The wave count increases the number of monsters that spawn. However, they are spawned on top of each other and are not visible. You can see it easier if you use an Ice tower on the top left corner of the canvas which freezes the monster that it attacks and lets the other ones that spawned with it move forward. This way you can see there are more monsters than one on each location.

This project was a good one to work on as this was my first time working with React, TypeScript, and JavaScript. And I am interested in seeing your feedback on my game. I hope you enjoy the game and can see that I tried to fix the bugs that I have encountered but at the end did not have enough time to deliver a completely polished tower defense game.