Assignment 3
Design Document

Thaisnang Reang — Mayank Singh

May 8, 2018

Abstract

In this assignment, we have developed a turn based multiplayer game RoboWarZ which is a modified version of the game pocket tanks. The objective of the game is to destroy the other robot by depleting their health bar. We have provided various offensive and defensive tools to each player for the game.It can be played in singleplayer or multiplayer mode.In multiplayer mode each player will have an unique login ID and they can chat with each other during the game.

1 Contents

2 Mainmenu Design

The game will start with loginscreen it has an option to register if no account was previously made.After a successful login the mainmenu screen will come.

2.1 Mainmenu

The mainmenu will have the following options:-

2.1.1 Create Game:

To host an online session

2.1.2 Join Game:

To join others session

2.1.3 Play offline:

To play with your friend on a single laptop

2.1.4 Exit:

To close the game and move back to login page

2.2 Weaponshop

Before the match starts each player can select upto 8 tools from 10 provided.It is achieved through the following functions:-

2.2.1 Weaponshop()

This function loads all the images of the tools and displays all in an defined order.

2.2.2 Weaponselector()

This read the incoming inputs and loads the selected tool in the inventory of the player.

2.2.3 AIweaponselector()

This function selects the tools of the A.I. It selects the weapon which has a possibility to countering the weapons selected by the player.

3 Robot Design

Each turn the player will be given two moves by default. Some tools will take two moves to use and some will take one.A player can save upto 3 moves max per turn.

3.1 RoboLoad()

This function loads the spritesheet of the robot and the images of the selected tools.

3.2 Status()

This function calculates damage and update health and armour of the robot.It stores all the buffs and debuffs that are currently on the robot.

3.3 Inventory()

This function display all the available tools to the player.The player can select one of the displayed tools and fire it i.e-call the appropriate weapon function along with proper parameters. Once a tool is used it removes it from the inventory.

3.4 Movement()

This function will calculate the current position of the robot, the available moves to the player and update the position according to the buttons pressed.

4 Terrain Design

4.1 TerrLoad()

This function loads the images of the buildings, roads and the background.It places the buildings of different heights randomly in a selected range.

4.2 Terrmod()

This function calculates the changes in the terrain ,which will come due to the different tools use by the players, and update the terrain.

5 Offline-Play

With this feature two players can play together on a same laptop turn by turn

6 Offensive and Defensive tools

6.1 Parameters()

This function will calculate and update the power and angle according to the incoming inputs.

6.2 Tools

6.2.1 Projectile()

This function takes the angle and power from the parameters() function and calculate the path of the projectile.

6.2.2 AgniVI()

This weapon is a missile which will go to a target location and high damage in a small area.

6.2.3 TacticalNuke()

This weapon is a nuclear warhead which will go to a target location and do medium damage in a large area and knockback all the obstacles including the player.

6.2.4 AcidRain()

This weapon is a projectile which will explode mid-air, rain down acid on a target area,which decreases movement speed of the robot and do constant damage. Acid will permanently reduce armour of the robot and do corrosion damage.

6.2.5 DefensiveShield()

This gives the player a buff, reduces incoming damage and prevents all kinds of debuff.

6.2.6 ShockBomb()

This projectile goes to a target location and do a create a small AOE shock field which disables the robot for a turn.

6.2.7 ElectricalWall()

This projectile goes to a target location and creates an electrical wall which disrupts any projectile that passes through it.

6.2.8 StealthMine()

This projectile goes to a target location and gets planted there if the robot comes within a certain radius of it will explode and knockup the obstacles including the player.This projectile will only be visible to the user not the opponent.

6.2.9 DischargeTrap()

This projectile will go to a target location and plant itself if the player comes within its radius then it deletes two moves from the robot.

6.2.10 RemoteScout()

This projectile goes to a target location and scan the area. If the enemy robot is within that certain area it gives away its exact position and suggest on what angle and power to choose in order to hit the target.

6.2.11 RepairBot()

This tools repairs the robot and restores health and armour of the robot and remove all debuffs.

7 Server

7.1 Login

On opening the game login window will come where player has to write his username and password. Player will enter his username and password then his data will be sent to our database(i.e Google Firebase) if the provided username and password matches with the data present in database then user will be able to play game else an error will come indicating incorrect username or password. With login window there will be an option to sign up where player can make his profile and his provided data will get stored in database.

7.2 State Syncing between Players

To make sure all clients are in sync, the simplest way is to let client send update to server in a fixed interval. Clients will send data to server in every x ms(depends on the speed of internet) of time and server will update the data and will send the data to all the clients.
For this feature to perform on Realtime basis we have used Google Firebase. Data from hosted player and joined player is exchanged with the help of firebase

data from both the side is sent to database and then it gets retrieve from there in realtime.

7.3 In-Game Chat

7.3.1 Text Chat

We will implement this feature with the help of nodejs and in nodejs we have used Socket.io, Express. It allows the real time communication between the players connected. This WebSockets technol- ogy is fast as lightning.Socket.IO is a JavaScript library that helps improving work with WebSockets. The main idea behind using Socket.IO is the ability to send and receive any events with any data. It can be any object as well as a binary data.

7.3.2 Video and Audio chat

For this feature we have used WebRtc service using PeerJs, which provides realtime communication between players visually.