Group 2

Design Document For:

Project 5

Index:

Game Overview

Summary

[Title] is a isometric multiplayer wave based survival game where the goal is for four players to work together to survive as many waves of enemies as possible. Players choose one of four characters to control, all of which are dependent on others for success.

Gameplay Overview

Waves of enemies will spawn and head towards the closest player. Each wave will hold X more enemies than the last wave (min Y at lvl 1) It is the job of the players to survive as many waves as possible by working together and taking advantage of each characters unique abilities.

Server Info

[info on how server works here]

Client Info

[Basic info on Client here]

How to play

Once in game, the control scheme is simple. [up] to move upwards, [right] to move to the right, [left] to move to the left, [down] to move downward, and [action] to use your characters unique action. Each Character has one unique action which can be used by the [Action] button. The Fighter attacks in a wide arc, slaying multiple enemies in one blow, the Archer shoots arrows from afar, the Medic will heal an ally one point of health, and the Mage will slow enemies down. Working together is the only way to survive the game.

[Gameplay Activity Flowchart Here]

Features

Server Connection

Client Connection

On connection with the server, the current list of players is acquired. The Lobby (Character Selection) screen will be loaded.

[Lobby Screen Activity Flowchart Here]

After a character has been selected by all four players, the server will send the clients information on what level information to load. Once loaded, the client will send a signal to the server confirming that it is ready to play

Once that information has been received by the server from all clients, the game will begin. See Gameplay.

Gameplay

[Gameplay Action Flowchart Here]

Waves

Waves will begin at (X) enemies and will increase by 1.5 times (rounded up) for each wave cleared

[Wave Action Flowchart Here]

Wave Math

1.5 x num enemies

Powerups

Power ups occur every (x mins || x enemies killed) they stay until they are picked up by a player at which point their effect is applied to that player

Health Pickup

Heals player by x health points

Damage Pickup

Doubles player damage for (x seconds || x attacks)

Rendering

Phaser.io is used for the frontend and will handle the rendering of the game.

Client Lobby View

The lobby will have the four characters in idle animation, below them will be buttons which will select that character & make that character unselectable by other clients. There will also be a messaging section where clients may communicate [Milestone]

[Lobby Screenshot Here]

Game View

The game view will render the background, enemies, players, and powerups. These things will be rendered in a loop using information from JSON objects.

The JSON objects of the current client will be updated by that client, and sent to the server. Other JSON objects will be updated by the server. If the JSON objects are not updated between render loops, enemies/players will continue their animation and stay in place until an update is made.

If a client loses connection with the server completely, or exits the game while in the game view, the player character will be killed.

[Gameplay Screenshot Here]

2d-Isometric

Animation

Sprites

Collision Detection

Characters

There are four playable characters. The Archer, The Fighter, The Medic, & The Mage. Each of them rely on the others to be effective, as they can only do one thing each.

Archer

The archer moves fastest of the four adventurers. [It (Until Sprites are found)] is the only adventurer able to harm enemies from far away. The Archers arrows go until they leave the boundary of the screen.

The Archer will aim in the direction of the mouse, [Action] will fire a shot. Once fired, another arrow cannot be shot until the fire & reload animation is finished.

Fighter

The Fighter has the most health of all the adventurers, though the heavy armor comes with a penalty to speed. The Fighter is able to attack many enemies in a wide arc in their front. As such, the fighter is the only adventurer able to defeat multiple enemies at the same time.

The Fighter will attack in the direction they are facing by using the [Action] button. Until the attack animation is finished, another attack cannot be made.

Medic

The Medic is the only adventurer who cannot harm the enemies. Do no harm, extends to even the dead it seems. The medic can heal the other adventurers by X amount of health. Once this is done, X seconds must pass until it may heal again.

Mage

The Mage, while unable to directly wound the enemies, can slow them down enough to allow the other adventurers to safety defeat them.

Enemies

The enemies are many, but weak. Until they are all defeated, they will be attracted to the closest adventurer.

Zombie

One hp, easy to kill, but many.

User Interface

Overview

Gameplay VIew

Menus

Connection Menu

Resolution Menu

Character Selection Screen

Help/Control Scheme

Music & Sound Effects

Overview

Technical

Server

Client to Server Codes

Code Name	Simple Description	Called When
Exit_Game(Player Object) (Jay)	If Client Exits Game whilst Playing, the game continues, but that character dies. If Client Exits Game whilst in lobby, Removes Player from Player List	Called when Quit Button Pressed by Client
Class_Selected(Player Object) (Nick)	Tells Server that calling Client wants to play as chosen class If player already has a class chosen, it removes player from previous class and adds to new chosen class	Called when client selected class to play as
Game_Loaded() (Jay)	Sent to the server to tell it all game assets are loaded and calling client is ready to play	Client has loaded all assets for the game and is ready to play

Client

Server to Client Codes

Code Name	Simple Description	Detailed
Player_Joined (Jagjit)	Tells clients a new player has joined the server	Called when someone joined, and has been initialized by the server
Player_Left(Player Object) (Jagjit)	Tells clients a player has left the game	Called when someone leaves the game and the server is

	If in Game, a small label shows this before fading If in Lobby, message will be posted on board [MILESTONE]	informing clients
Update_Selections(Json Array) (Jagjit)	Updates available & unavailable classes for clients to pick from	
Load_Game() (Tigran)	Tells clients to load specific files	When all characters chosen, begin game at first level
Begin_Game (Tigran)	All clients have loaded the game properly, begin play	Server has received a game_loaded call from all clients

JSON Appendix

Player JSON: {

“player_id” = [*value*]

“character” = [*value*]

“xpos” = [*value*]

“ypos” = [*value*]

“health” = [*value*]

“Speed” = [value]

“current_state” = [enum]

“class” = [value]

}

https://www.dynetisgames.com/2017/03/06/how-to-make-a-multiplayer-online-game-with-phaser-socket-io-and-node-js/

Sending (Our Player Json)
Receiving and updating (Enemies & Other Players)

```
update()
{
Read server file
Render
Check for user input
Update our file
}
```