Names: Roy Koganti, Alan Chang, Michael Chen, Ahmed Shalaby
Andrew IDs: rkoganti, alanchan, mechen, akshalab

Team Keepin' it Realtime

The main goal we want to accomplish is having a website with a state that all users can modify together in real time. They will all modify the same state, and view the same state, along with all the real-time modifications being made. Our goal is to make an interactive game that everyone will be able to play collaboratively, with all user interactions happening freely and contributing to the fun of the game. One thing is we want to set up different lobbies, with a distributed server system. Then, players can enter different lobbies. So if we have 4 lobbies, we have 4 different game states. This helps when we have a large number of players. Each user starts off with a character/starting state when they start the game. We are deciding on a 2D platformer game, where all users play together (racing to get to the end and working together to defeat enemies).

The Game:
It will be a 2D platformer game. It will be a race course, with obstacles and enemies attacking the players. Players will move together in a pack, with the main game camera focused on the leader of the pack. Any player character which falls too far behind and leaves the camera view, will end up dying, and respawns later once the pack passes a checkpoint. Players will be able to move their characters around using left and right arrow keys, and also will have a jump and shoot button. Jumping and shooting will both help the player to avoid obstacles and kill enemies. The player's score would be determined by how many metres the player has run without dying and how many enemies he has killed. We would have different stages too, where each stage would incorporate different kinds of obstacles, a different environment, and different enemies. At the end of each stage, we would have a sec screen that doesn't move. All players would spawn at the left side, and we would have a huge boss on the right side. The players would then work together to destroy the boss character. If a player dies from the boss, they will spawn again at the left side after their death timer runs out. Lastly, there would also be powerups available, but they can only be taken by 1 character, the first one to reach it.

One main point to note is that people won't need to sign up in order to use the website. As soon as they enter the site, any user, whether they are logged in or not, will be able to start playing the

game. They will spawn immediately at the center of the screen, near the leader. However, we will make registration available and optional. Being registered allows you to save your high scores for the game. Otherwise, you would begin with a randomized character each time you access the website again, with all your stats and scores reset. Other features we will have this site includes a chat-box feature, for people to chat with each other. Logged-in users will be able to chat with their usernames and those who are not logged in can still chat with a randomly generated handle.

We are considering using node.js instead of Django to make this website as based off our research on Django sockets, it was mentioned numerous times that they tend to be somewhat unstable. A large problem we are going to have to face is how to send information to each player. The issues of lag in a realtime game can make a lot of problems. We've talked about 2 different kinds of way of sending the game state information. We could store absolute coordinates for each player and send them many times a second to update the game but this seems like it will be very taxing on the server. The other way is storing inputs and having the client interpolate the movements.

Game Components:

We will try to host our web application using cloud hosting provided by Bitnami, which is in addition provided for free via the GitHub Student Developer Pack

Middleware (Socket connections) - *Ahmed*

This will coordinate real time game play by receiving inputs or positions from clients and updating their local game states by sending them the game state information stored on the server. We will use Node.js and socket.io.

Distributing Backend (Database / game state computations) - *Roy, Michael*

The core that will take information from the clients to calculate position of players and store information about players which can include things such as: user information, character stats, etc. Most player and game interactions that happen in the game will be computed and updated here.

Game State Design - *Ahmed, Roy*

This is the server side configuration of game state, or the “true” state. It includes the information needed to define the full state of the game as well as reactions to users’ inputs. It can be used to update client side states of the game

Live Updated Scoreboard (Ajax) - *Alan*, *Michael*

Live updated leaderboard of scores of people playing at the moment. It can refresh every 10 seconds or so and have different categories like current leader, all time leader, etc.

Save Personal Character State with Sign up - *Alan*

It is not required to sign up to play the game but if you want to save your scores and stats you can register for an account. Whenever you login again it will accumulate scores for all your plays. This will also allow for personal cosmetic customizations while an unregistered player will have a randomized character.

Unity game development - *Roy, Ahmed*

This involves making the actual game, with all the features involved. It would have several main features, such as giving the user control over his character, incorporating AI for the enemies and bosses, focusing the camera on the leader of the pack, designing the stages and having the players be able to move through the stages, having animations for the game, etc.

Chat Box - *Michael*

A chatbox that allows players to chat with each other in real time. All players can chat with each other, but logged in users have their username displayed while those who aren’t logged in will be given an anonymous username while their game session is still active with chats saved for the duration of a session.

UI/Website Design - *Ahmed, Alan*

We want the home page to be focused on jumping straight into the game. We want to have some sort of menu bar or some way of conveying the other parts of the site like login, account page, etc.

First Sprint

Project Owner - Roy Koganti (rkoganti)

Sprint Backlog:

- A simple 2D infinite side-scrolling game, where a player can control a character to keep constantly moving to the right, while avoiding obstacles
- Be able to run this game in real-time over the network using Socket.io
- Set up user accounts such that each user is given a random account when directed to the game page, to communicate with the game. However, they still won't be able to set up their own accounts or save their game state for the next time they log in.
- Set up a simple chat server, allowing different users to communicate over the network and have it synchronized.

We will all be contributing to each functionality but the main people in charge will be as follows. Roy will be making a basic game to use on the website first. Michael and Alan will set up a way for each user of the site to be assigned random user accounts. Ahmed and Alan will be in charge of using Socket.io and node.js in order to make the game networked and running in real-time. Lastly, Roy and Michael will be in charge of setting up the chat server, allowing different users to message each other over the network.