Multiplayer Fight Game - Documentation

Rasmus Göransson

Game Design

When we got the assignment my first idea was a fighting game. Two games that immediately came to mind and were a big inspiration were *Super Smash Bros* and *Stick Fight: The Game*. Both games are fast paced fighting games where you fight, win or lose and then start again.
I had a few ideas of what I wanted in the game, some of which I got implemented, others I would put in if I were to continue on this project. A few examples being health, a variety of attacks and power ups.
What I ended up with is a more simplistic version of *Super Smash Bros*. You have a stage, one big main platform and three additional platforms in the air, which you can pass through. Two or more players spawn in the middle and fight to push/knock the other player(s) off the stage. If you fall off you will touch a death barrier and get teleported back to world origin. Players can also taunt each other and emote using emojis.
Ofcourse, if I were to continue on this to make it a finished product, I would, apart from the previously mentioned ideas, also implement general stuff such as a lobby, player score, player spawn points and a win/lose screen.

Implemented Network Features

Apart from the UI, everything is done through a single player script. If I were to expand on this project I would most likely move stuff such as emoting and attacking to separate scripts but for the scale of this project it seemed easier to simply have it all in one place.

Additionally, almost everything is managed with Server Authority. If the player wants to do an action it tells the server what it wants to do and the server updates the action for all players.

I started with just getting some basic player movement synced. Moving left and right along with jumping. This is done by updating the player's rigidbody velocity with the direction the player wants to move in and adding an upward force to its rigidbody to jump. Attacking is done in a similar way to jumping, it does a raycast when the player does an attack and if the raycast hits another player it adds a force to that player's rigidbody launching it in the direction of the attack.

When it came to implementing the pass through platforms I used an in-built component *(PlatformEffector2D)* in unity for one way pass through. When I wanted to go through from the top I simply changed the layer of the player wanting to pass through to a separate layer the component ignored.

The Taunt feature is a simple player animation on the player sprite that plays when the player presses a button. I just made sure the player sprite has a network animator component and told the server when I want to play the animation.

Lastly, the emote feature, the single thing that has some client authority. The player tells the server which emote it wants to display and the server then tells all clients what should be displayed and the clients individually update correspondingly. The reason for this will be explained in challenges.

Challenges / Solutions

I didn’t have too many challenges, my most recurring problem was things not syncing, and a lot of the time it was because I was trying to change stuff locally which the server wouldn’t allow leading to unsynced actions or simply actions not executing. This was almost always fixed by making sure I only sent instructions to the server and letting the server do the changes. Like I mentioned previously regarding the emotes I decided to use some client authority. This was because I couldn’t get the server to sync the emotes played, I’m not sure why or what I did wrong but since this was something less important compared to movement and attacking I decided to give the clients some authority and simply display the emotes themselves, which solved the issue.

My Overall Experience

Overall, I found this assignment incredibly insightful and fun. I had been hesitant to get into multiplayer because it seemed quite difficult and complex, and I’m sure it can get a lot more confusing and messy on larger scale projects but for this it wasn’t that bad. If I decide to expand on this project or just make a new one I will definitely try to structure some things better and try some more ways to synchronize different things.