LinusJernström Netcode Assignment

Dodgeball

A top-down multiplayer arcade game for 2+ players.

Players face off in a circular arena. Each Player has [3] hit points, henceforth called HP. An object known simply as the Ball is the game's focal point.

One Player arbitrarily starts with the Ball.

The Player holding the Ball is designated the Thrower, and is visibly indicated as such. The Thrower can throw the Ball, which bounces off of the arena's walls as well as Players, based on the angle of impact. Each bounce increases the speed of the Ball. If a Player other than the Thrower is hit by the Ball, their HP is reduced by 1 point.

The Thrower has a timer of [5] seconds to throw the Ball, and if they fail to do so, a **Timeout** occurs. When a **Timeout** occurs, possession of the Ball is given to the Player that was closest to the Ball at the moment of **Timeout**, excluding the previous Thrower. This Player then becomes the new Thrower. A **Timeout** also occurs after the Ball has been airborne for [5] seconds.

While not holding the Ball, Players can attempt a Catch. To complete a Catch, the Player must initiate the Catch within [200]ms before being hit with the Ball, and they must be facing toward the Ball. After attempting a Catch, there is a [1] second cooldown before it can be attempted again.

When a Player reaches 0 HP, they are knocked out and removed from that round of play. The last Player standing wins the round. A match is made up of [5] of rounds, with optional tie-breakers.

Development

This assignment was very tough for me, having never done netcode like this before. After setting up simple player movement and rotation scripts that relied on Unity's NetworkTransform script, the latency was immediately obvious. The ball, having high velocity, was even worse. On the client's side, it appeared as though the ball bounced way too soon, and it was difficult to catch the ball since the player had to guess when they would be hit. Therefore, I think I'll make my own script which features client side prediction and reconciliation. The limited scope of this assignment means I will have to pursue that in my spare time after turning the project in.

The player has authority over their own movement, while authority of the ball stays with the server most of the time. That changes only in the instant that a player catches the ball, as they gain ownership of the ball before destroying it.

I ran into an annoying issue that was caused by my movement not being physics based. My players started phasing through walls. Since I didn't want physics movement, I had to improvise. So, I made my own collision detection for movement and rotation, which works well but is not performant.

I didn't manage to make the play again button work, so each round must be alt+f4'd out of. My unique ids also stopped working, so a lot of stuff is very broken right now.