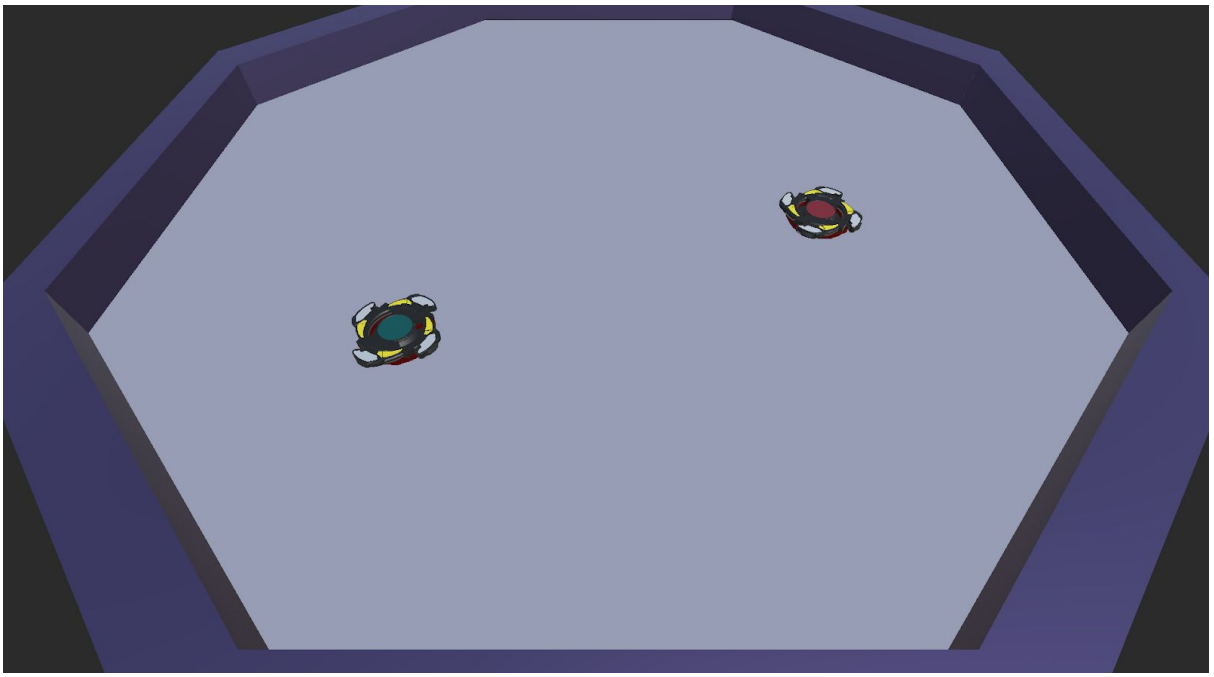
Document technique
Spiner Octogon Arena

Context

I made Spiner Octogon Arena with Unity and Photon bolt for my Networking module.



Game concept

Spiner Octogon Arena is a multiplayer game where each player controls a Spiner in the Octogon Arena and have to bump his opponents as strongly as he can.

It is playable on PC, each player controls his spiner's movements by accelerating with WASD keys, and can collide with walls and opponents.

Lobby

A player can create a named session from the main menu, active sessions are displayed on the main menu and can be joined by clicking on the corresponding button, by joining a random session, and by typing the name of a session you're looking for.

This works with Photon Bolt methods for starting servers and clients, and a loop to search a specific session by name.

Players instanciation

Everytime a player joins a session, the server will spawn his Spiner and allow the player to control him, by giving him Control of this entity, which is done using bolt methods for entities.

However the server keeps ownership of every entity, so only the server can determine the actual state of the game, which makes it an authoritative game.

Game state

Bolt uses the state feature to spread properties of entities on the network. It's used to share any value of an entity that is relevant to other players, like its position or health.

The state of an entity can only be modified by its owner, that's how the server keeps control over the game.

The server and clients being players alike, I used an abstraction with PlayerObject class that allows me to know if they're a server or a client while considerating them as players.

Bolt uses the commands feature to allow users to send inputs to the

server and run them on their game instance. So clients can simulate the game state as soon as they give an input, but if their resulting state is different from the server's state, they receive a correction from the server by setting their game state to the last valid state from the server.