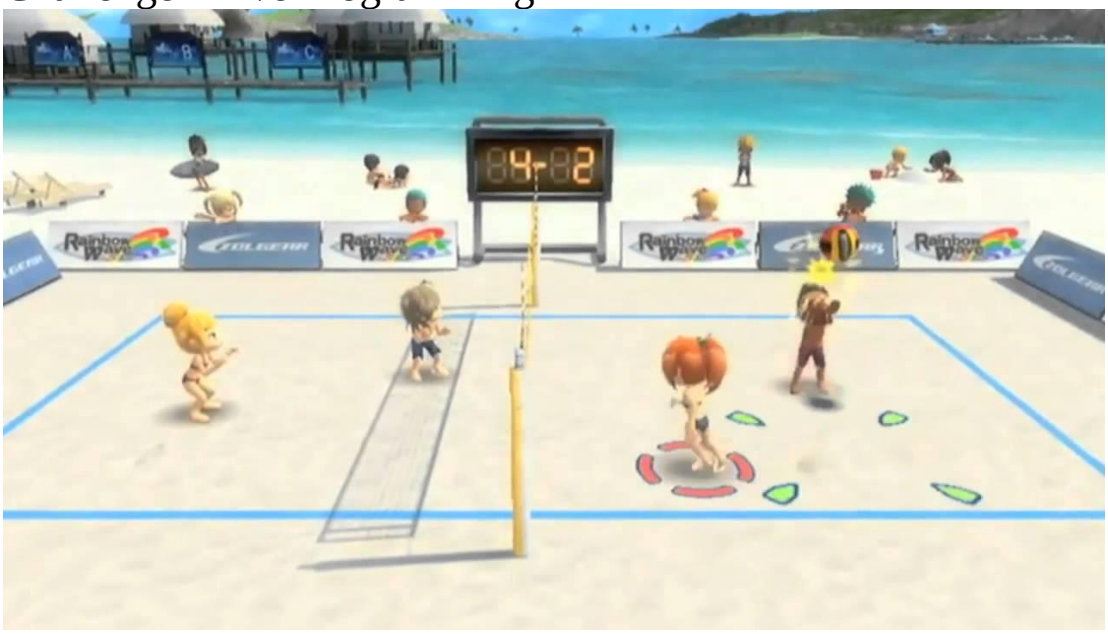
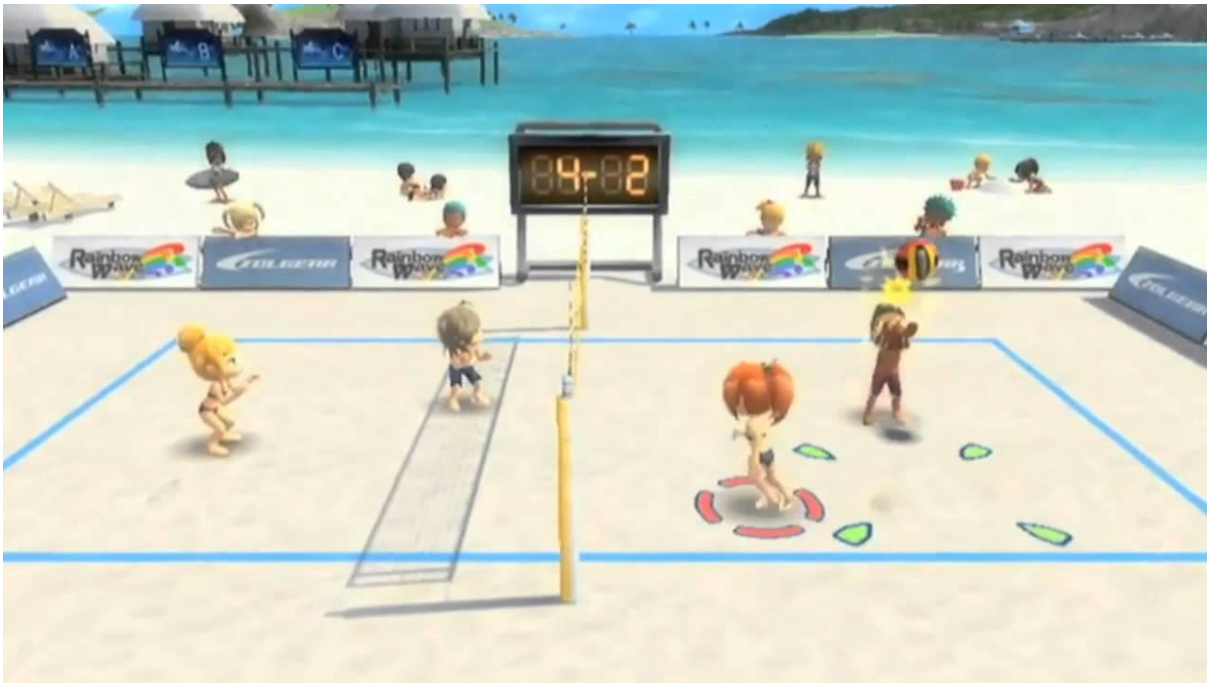
Challenge - Live Programming



(1h)

1. Create a plane and a sphere (volleyball) with gravity on the sphere.
2. Create 2 players (cubes).
3. Create a volleyball net (barrier) between the players and limit the area of the volleyball court.
4. Control the movement of the players (translation and jump)
    a. WASD for left player + Space (jump), Arrows for right player + Enter (jump)
    b. Limit the movement of player
5. Setup the physics when a player hits the ball.
6. To start the match, the ball spawns randomly in the volleyball court in the air (fixed height).
7. Add score when the ball hits the ground.
    a. In game scoreboard (like image above)
8. End the game and restart when reaching 7 points.
9. UI to start/restart match
10. Make a pulse highlighting where the ball would hit the floor when in the air
11. (Bonus) Make an AI that goes for the correct position of the ball and hits it.

Challenge - Beach Volleyball Multiplayer Cross Platform



Time for Challenge: 1 week (estimated 20h of active work)
Meeting after 1 week, 1h to discuss results

Develop a beach volleyball game which is multi-platform multiplayer. It is necessary to have at least **one player on each side** (not necessarily to have/control a team of 2). The gameplay must support at least 2 platforms from (only one version of each is necessary):

- Any standalone headset (e.g. Oculus Quest 2)
- Android phone
- IOS phone
- Windows/OSX/Linux

The main point of interaction of this gameplay is to make it **possible and in sync to hit the ball from one player to another**.
Plus:

- All the points from the live programming challenge and other volley interactions
- Player matchmaking and lobby
- Overall user experience/usability
- AI for single player (other team is controlled by the system)

Any assets and references from Unity Asset Store, official unity packages or any source from the internet can be used. 2 topics will be used to judge this challenge:

- Game play working for 2 platforms with at least 2 players.
- Codebase and project architecture

The **visuals are not important** for the challenge, you can treat it as a Proof of Concept for the gameplay and infrastructure, without having to care about shaders, characters, rigging,

animation, levels, and so on. **Usability is a plus** for the challenge, so designing controls for different platforms will have different inputs but their usability will not be the main focus of judgment. If the usability is not optimal you can use the meeting afterwards to explain why and possible ways of improving.

A local server can be used instead of publishing online. A README on how to test the unity project is mandatory.