# Case Study 2 – 3D Multiplayer Race

**Case Study 2:** A 3D multi-player obstacle race game for desktop. The game is played across multiple timed rounds, after the end of which players are eliminated. The game itself focuses heavily on fast reaction times to complete the levels, so any lag is unacceptable. Each game can have up to 40 players playing at the same time.

## Feature Deliverables:

* [X] 3D game
* multi-player game (network-based or web-based) – support up to 40 players
* [X] deploys to desktop
* [X] multiple rounds within a game
* [X] timed rounds
* [X] players eliminated after each round
* [X] involve fast reaction times

## Design elements:

* [X] menu item to host or join a game
* [X] menu item to choose a racetrack/scenario
* [X] menu item to configure scenario settings
  + [X] how many rounds
  + [X] how many players continue to the next round or are cut from field
* [X] racetrack(s) with obstacles (animation too)
* [X] player controlled element (marble)
* [X] Timer on the race and high score tracking

## Technical challenges to learn:

* making a multiplayer game
  + testing the game with multiple players
  + creating an AI to simulate some players
    - [X] Simple
    - advanced

# Wireframe - Game Flow

Quit

Join a race

My Name

My look

Title

Main   
Menu

Host a race

Round Count

Scenario

Loser filter

Play  
laps

Lap/  
Race  
Results

Play  
race

# Server/Client messaging flow

|  |  |
| --- | --- |
| **Server source** | **Client source** |
|  | Set player visual preferences |
|  | Request to create a new race (host) |
| Start server |  |
| Create a new race |  |
| Start the new scene |  |
| Create new player, spawn to clients |  |
| Apply client’s preferences to a new player |  |
| Give control of player to client |  |
|  | Client 1 takes control of new player |
| Start the timer delay for additional clients to join |  |
|  | Client 2 player UI prefs, request to join the race |
| Create new player, spawn player to clients |  |
| Apply client’s preferences to a new player |  |
| Give control of player to client 2 |  |
|  | Client 2 Takes control of new player |
|  | Player movements sent to server |
| Player movements propagated to clients |  |
| When a player finishes, set finish time |  |
| When all finish, set game state to review |  |
|  | Client gets state change & displays review screen |
|  | Client clicks next, and requests to be put into next lap |
| Once first request is made, reset lap and scene |  |
| Add all qualified players to next scene |  |
| Distribute control of correct players to clients |  |
|  | Clients gain control of ‘new’ players |
| When lap complete, disconnect everyone |  |
|  | Client returns to main menu |
|  |  |

# To Do Notes

Trello board: <https://trello.com/b/tpM0ZOIl/marble-race>

# Networking Notes:

[X] MLAPI - for networking

[X] Project Clone – for dual development and testing client and host

* networked object is each sphere player / AI
* moving track pieces as networked objects?