# Multiplayer Shooter

## Overview of the game

This is a multiplayer shooter game. Perspective is first person perspective. This allows us to keep players engaged in the game. Besides, the main tackling of this game is being a multiplayer game. Multiplayer feature makes the game competitive. For this game the communication between players is crucial. Latency is primarily foe. For high retention and enjoyable game feeling, the game should be superior on shooter mechanics and their engaged points with multiplayer gameplay features(Server and Client communication).

The requirements of the game are :

- First person shooter controller.
- Target hit with bullets(According to Game Mode).
- Scoreboard which will be displayed on all client's screens.
- Stun Bomb will immobilize the affected clients.

First person shooter mechanics are handled by basically encountering basic movement concepts. The main tackle of this concept is that the player should sync each player's position and rotation.

There is a game mode approaching in the game which will be changed within the interval. The GameMode decides the bullet that the player has to use to shoot targets. If players shoot targets with the correct bullet which the gamemode defines, they get 1 point. On the other hand, if players shoot a target with an inappropriate bullet, they will lose 1 point.

There is Scoreboard UI on the client screen but it should be updated with network data which contains all player's scores. Scoreboard should be updated after each target hit with the sum of hit.

Stun bomb can be produced within the interval which defines coolingTime in the game. Stun bomb should affect the player in a certain radius but player who plant this bomb. The approaching of bullet instantiation is followed to produce stun bomb on the server.

The game is not a massive multiplayer shooter game. There is a limit because players are described with color which is redefined. But it is extendable. Also this game does not have a lobby or similar solution. That is why this is a LAN game. To play this game together, you should be connected to the same ethernet connection.

As a game engine Unity is used for this project.
As a programming language C# is used for this project.
As a network solution Netcode for Game Object which is newly released is used for this game.

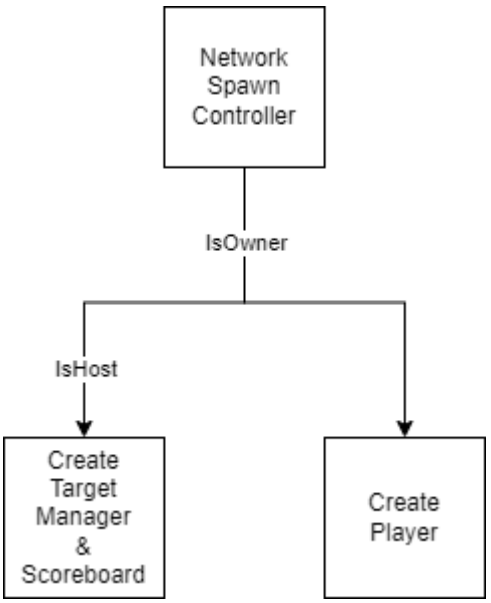
## How the game works

- NetworkSpawnController

This component is being spawned one on each client and server(host). The main responsibility of this object is handling Instantiation tackling the host and client side. For instance, some objects should be created just on the host but others should be created both

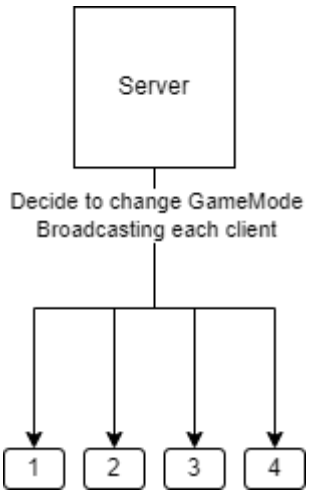
such as PlayerObject. At the beginning of the game, the player is asking whether client or a host. The NetworkManager which handles client communication creates this object. After that, this object starts to create players and environments.

- **Creating Player**

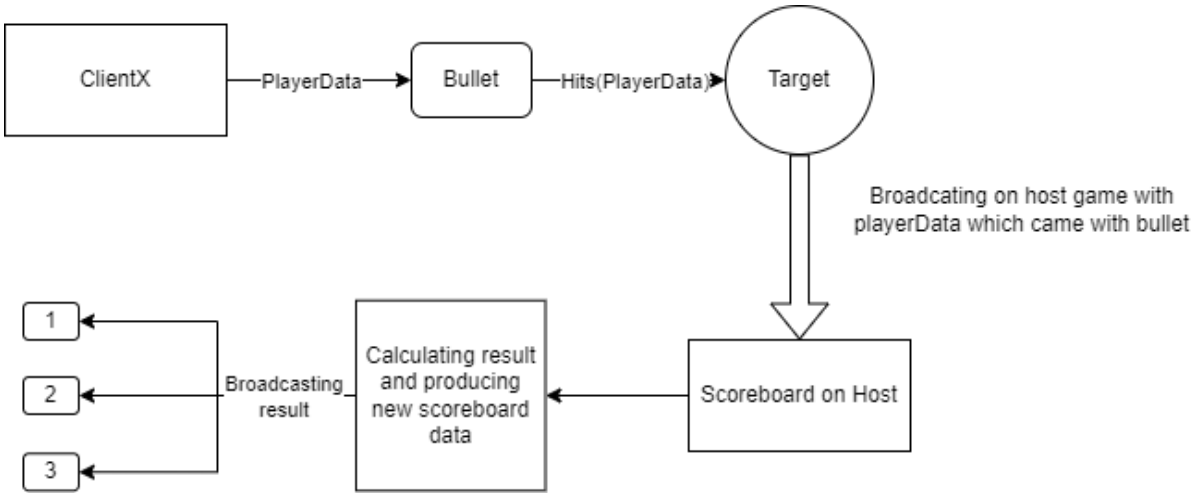



There is a diagram of how the game manages main network objects. We check if the client is the owner or not. Because we do not want to create two players for just one. If we are the owner we assume that the game is running on my device.
But on the other hand, TargetManager and Scoreboard are slightly different. We want them to be one in the game and belong to the host. Because we would not want to hit a target which belongs to Client7 as a Client1. We aim to hit the host's target with the spawning host's bullet.
The UI object creation differs at client and server point. Same UI object just belongs to the client so they do not communicate with the server such as the Player indicator which indicates the player's color and ID. Other UI objects are Scoreboard and GameMode displayer. These differ at collecting message points. Scoreboard is updatable for each player but GameMode displayer is being updated only by host.

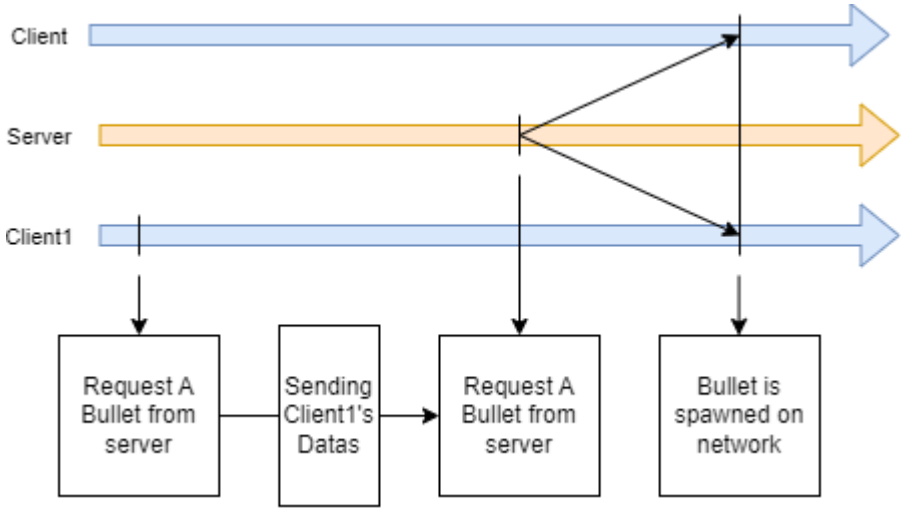
- **GameMode Workflow**



- **Scoreboard Workflow**

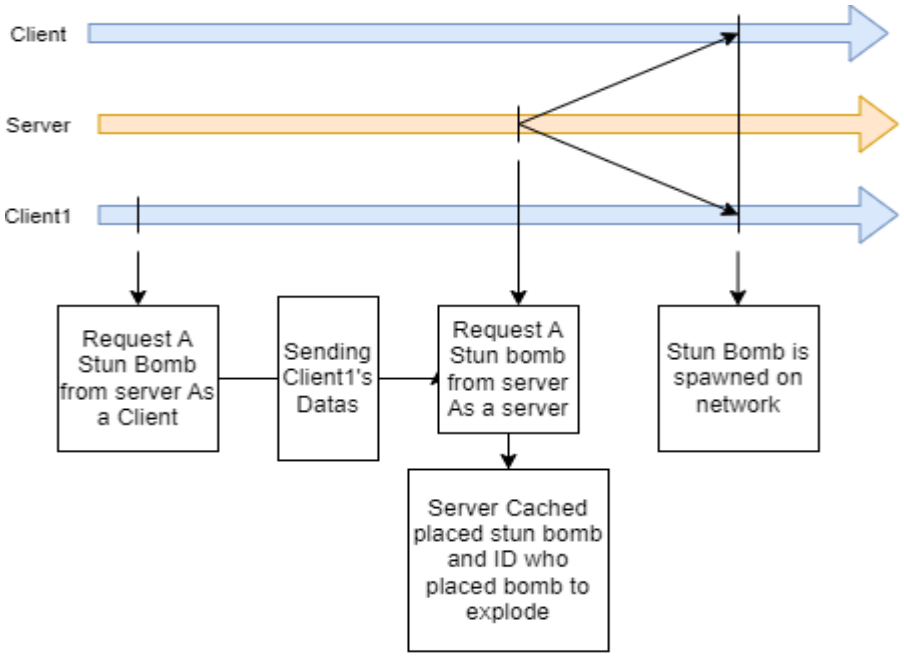



- **Creating A Bullet**

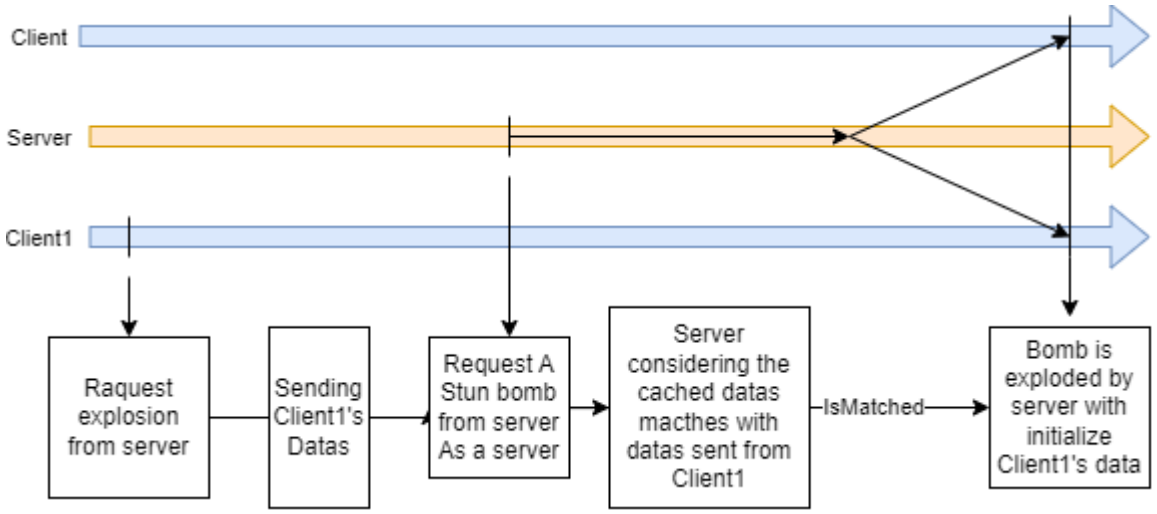



To create bullets we decided to use spawn only server approaching. Because targets are being spawned on server only and it could be cheaty for players. This is a trade-off and accuracy is selected. Also this is in the Known Issues.

- **Plating A Stun Bomb**



- **Explode A Stun bomb**




Here, we have different approaches. As we cache bomb and ID on server, we could not do explosion operation on client. We had to send a message to server that client wants to explode a bomb if there is its bomb. If the data which client sends and cached data are matched, the server allows an explosion.

Then, the exploded bomb send a message to server with position and player data which belongs to who exploded by. Server says that an explosion happened at that position and with that data. Clients take that information and consider whether the explosion affects or not.