

# Multiplayer Assignment

Axel.Larsson

---

## Introduction

**Hello, my name as it says above is Axel. This is my written assignment as part of the course. I decided to use components from other projects I made and are working on. These components are to clear:**

**The dungeon generator with the PlaceFinalpoint script, RoomBehavior and the prefab.**

**The object pool originally is from the same project but has been altered to work with multiplayer components.**

**Any assetpack used are within the asset pack folder with exception to the particle effect who i could not move.**

**My assets that I worked first hand on are in the Muh\_stuff folder.**

## Documentation

When I first started this project I thought it was a bit of a short time to single handedly make a game so I used assets from another project I made to flex in another course. At first I was unsure how Rpc were supposed to be used. The first script I made was the movement script and it shows in retrospect. The understanding I worked under was that all data a client can use to perform "Foul Play" was to be kept client side not server side (heard a story where a game had health on client side and the players would play in windowed mode and shake it to make themselves immortal). One of the things I left for later was the animator where I was not able to change animations which I do exclusively by code and I didnt find

---

---

the time to go back and fix it. I only use the animator controller so save and name the animations. Adding the spawn/despawn to the object pool made it feel like I had an easier time than my classmates since I only needed to make a serverRpc to call the object pool manager and not worry about spawn and despawn in different places and scripts. The object pool I tried to save the creator of the object in the item as owner of the object but since I was not sure if that puts the client in control of where the object is (no need to make targeting projectiles a possible hack) I ended up saving most of the code in the script but not fully implementing it. It was intended to be the way the projectile knows who fired it and prevent friendly fire. But I did in fact over scope with this a little bit and ended up having to cut that part. Same as I ran out of time when I was making the Health Bar I got it to update across all clients but it was updating based on the last current health to take damage and I tried to separate them and ran out of time in the process and currently am trying to merge 2 different github branches so I get all in 1 branch if I figure it out I'll try putting em in the "Main" branch.

## Conclusion

Knowing what to place on the server or Host is very important and I think what type of a game decides what those are. If you make a PvP game then keeping data in the hands of a server is very important to keep a fair match going like health, damage and the like. But if you're making a peer to peer game then the host will be able to cheat and none else and it's harder to make the game fair. A PvE game there is no need for such concerns and you can unburden the server of a lot of responsibility and try to prevent cheats no longer matter if at all. Unburdening the server/ host is important to not break it and risk lagging a game, hence keeping things like models and sound on a client base is more logical but once again it is up to the game if those sound effects are an important part of a PvP game like in CS go or valorant keeping them server side matters and makes a difference in Valheim it does not.