NETWORK PROGRAMMING ASSIGNMENT -1

KIRAN GEORGE GAURDIAN
GitHub Link

Overview

- The game is a multiplayer top-down shooter game that supports a minimum of 2 players and max 4 players
- Movement can be done using WASD keys
- Shooting can be done using the left-click for mouse
- Emote texts can be done by pressing space bar, which would vary depending on whether the player is alive or dead
- Whoever is left standing at the end of the game wins

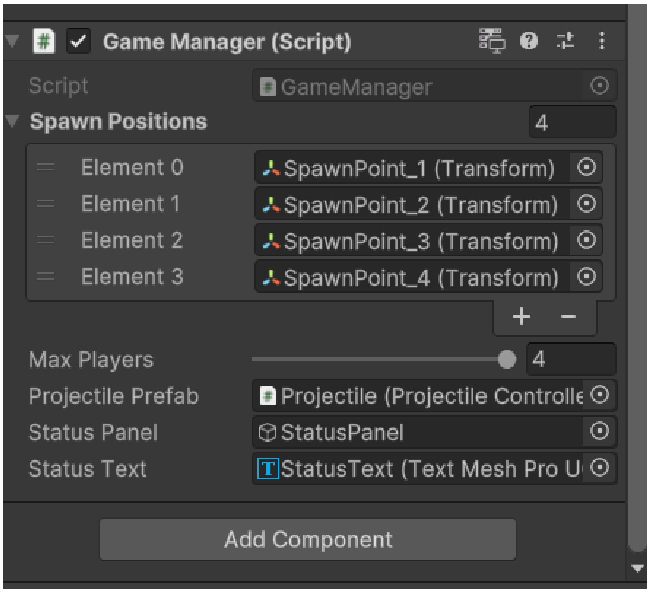
Implementation Details

Game Manager

- The Game Manager instantiates the game services and acts as the Service Locator.
- Data such as how many max clients can be in the game and spawn positions are stored in the Game Manager.
- Game Manager handles server-authoritative spawning of projectiles.

- Game Manager also handles appropriate game-state events such as the waiting state for more players and playing state.
- Game Manager is responsible for player initialization using RPC calls when the game starts.

Below is a screenshot of the GameManager script component:



Game Manager (Script)

Script	GameManager
Spawn Positions	4
Element 0	SpawnPoint_1 (Transform)
Element 1	SpawnPoint_2 (Transform)
Element 2	SpawnPoint_3 (Transform)
Element 3	SpawnPoint_4 (Transform)
Max Players	4
Projectile Prefab	Projectile (Projectile Controlle
Status Panel	StatusPanel
Status Text	StatusText (Text Mesh Pro U

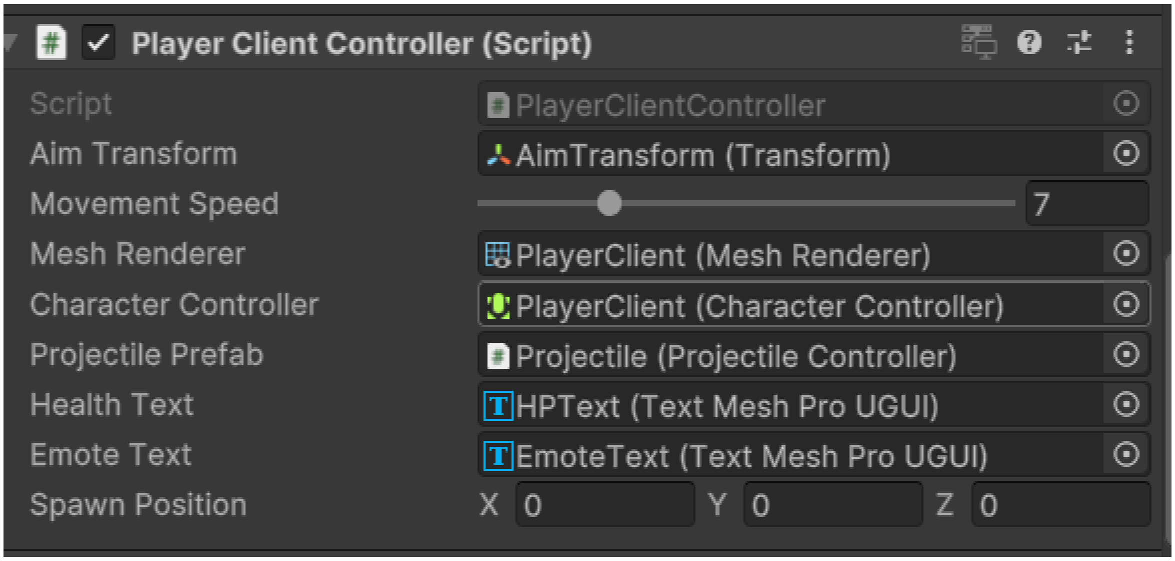
Player Client Controller

The Player Client Controller is the controller script for the player.

- Movement is done utilizing the Character Controller component.
- For shooting, it uses ServerRpc calls for requesting the server to spawn a projectile.
- Emotes are done by using a ServerRpc call followed by a ClientRpc call to reflect emote changes across all clients.
- Spawn Positions are supplied from the Game Manager before the controller is enabled. During the enabling process, the controller

uses this info to go to the respective spawn positions before enabling the Character Controller since movement is not server-authoritative.

Below is a screenshot of the PlayerClientController script component:



☑ Player Client Controller (Script)

Script	PlayerClientController
Aim Transform	AimTransform (Transform)
Movement Speed	7
Mesh Renderer	PlayerClient (Mesh Renderer)
Character Controller	PlayerClient (Character Controller)
Projectile Prefab	Projectile (Projectile Controller)
Health Text	HPText (Text Mesh Pro UGUI)
Emote Text	EmoteText (Text Mesh Pro UGUI)
Spawn Position	X 0 Y 0 Z 0