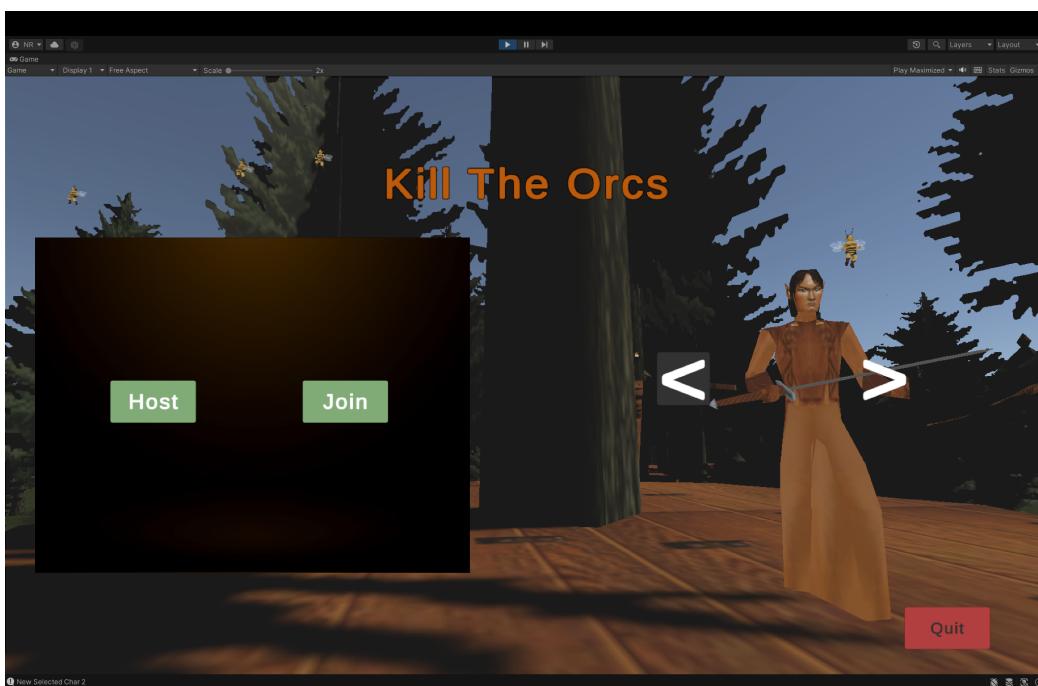
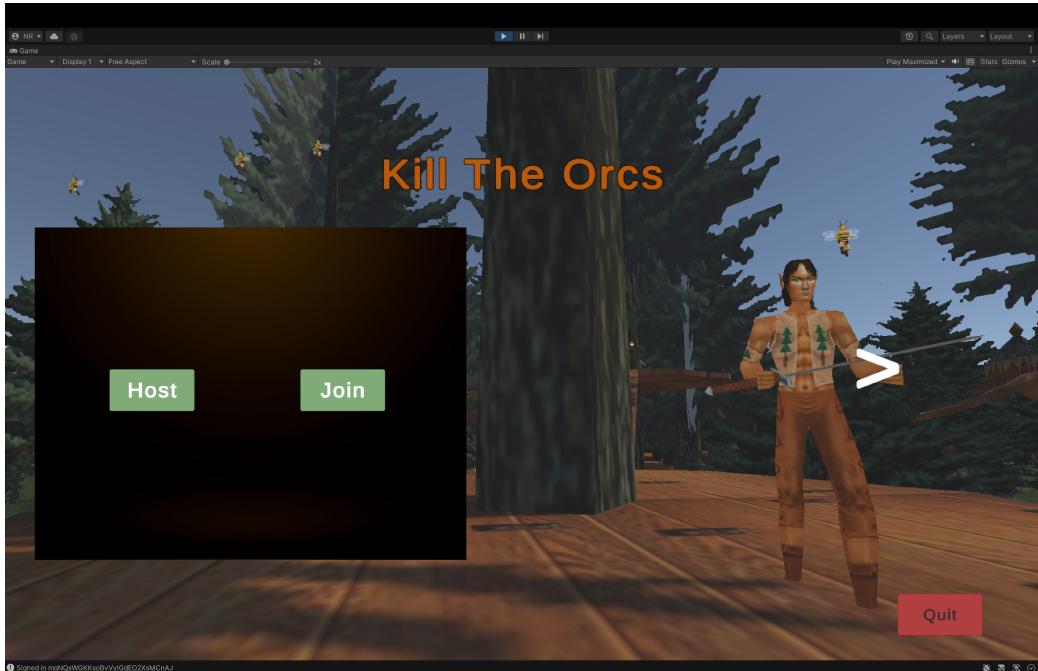
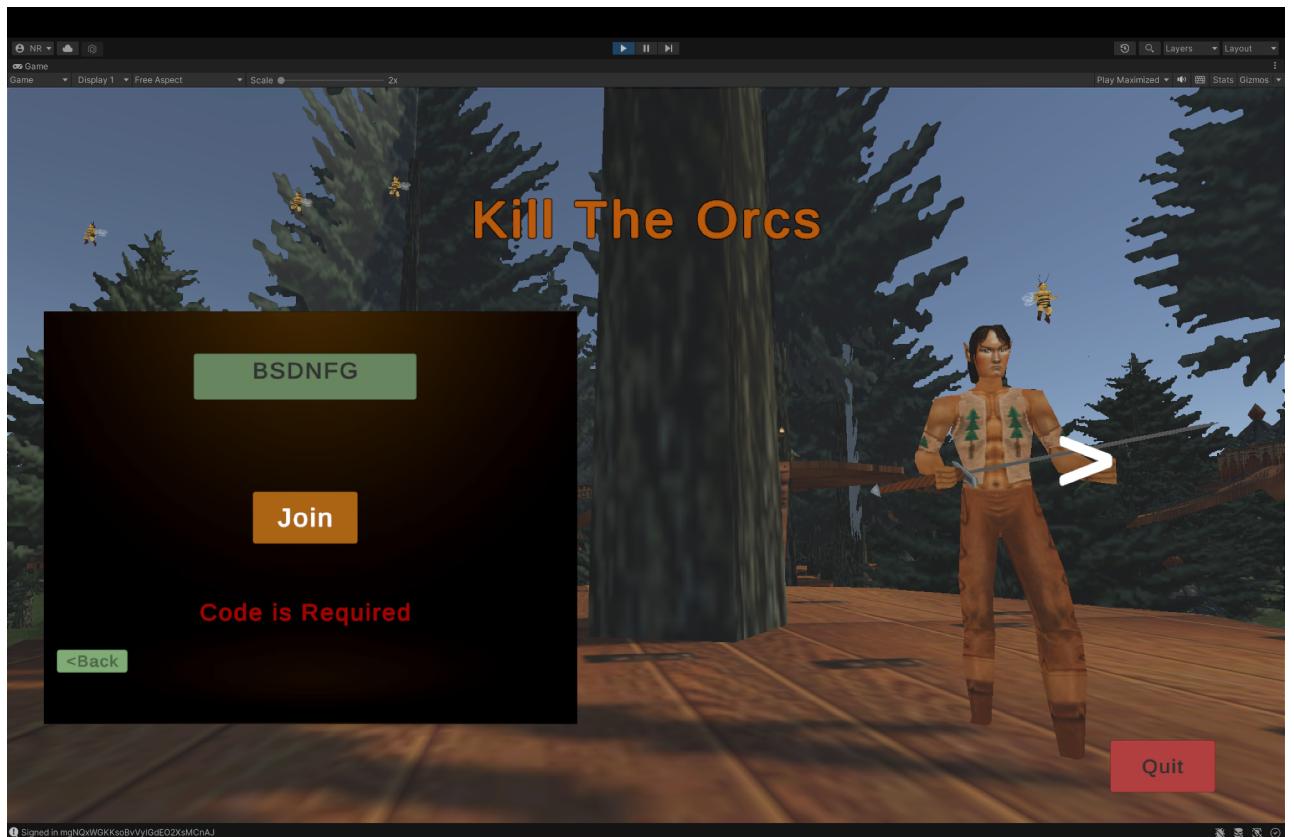


#1 Character Selection Screen:

- This is the first scene which gets loaded when the game starts.
- In this screen the player can select the player avatar for the game and can host or join the game.
- The player can select the host/ join the game.



- If the player selects the host, the main game scene gets loaded and host code is displayed on the canvas.
- If the player selects the join, the player can input the code provided by the host.
- The player can quit the game using the Quit button.



#2 Main Scene:

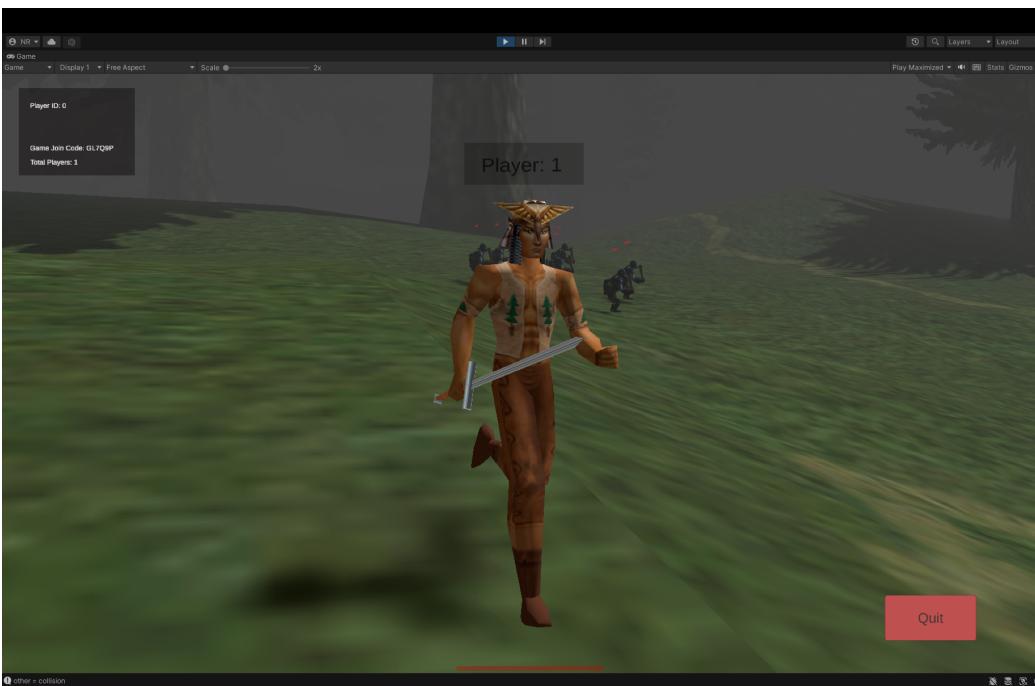
- The players whoever joins the game will get spawned into the scene and players can kill the orcs.



- Spawning of the player objects will use the Netcode for Game objects framework provided by un



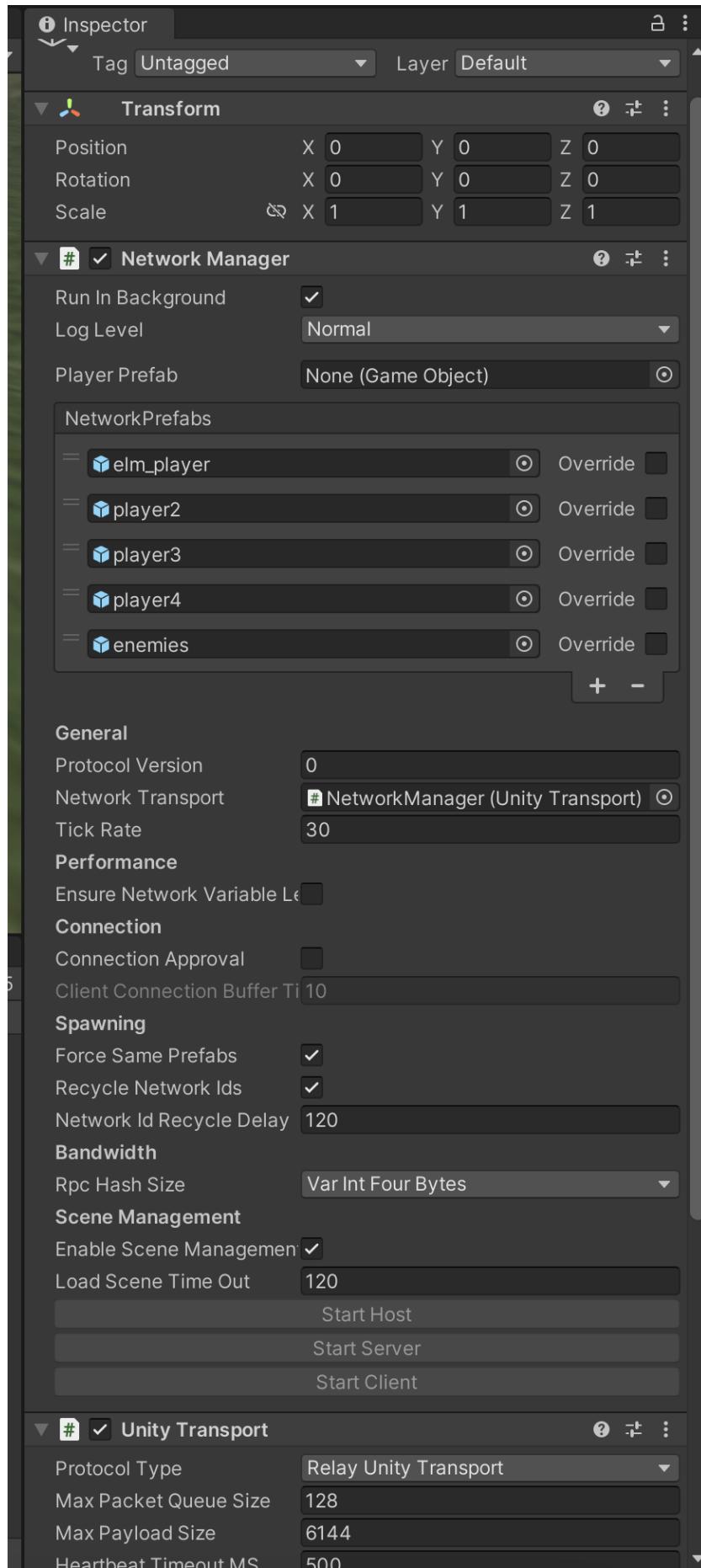
- Players can walk, run, jump and combat.





#3 Multiplayer - Netcode For GameObjects:

- To run the multiplayer game we made use of Netcode for GameObject concepts.
- This can be achieved by using NetworkManager, NetworkAnimator, NetworkObject, NetworkTransform components.
- This Game is Hosted in the unity servers by using **Relay Server** in Unity Gaming Services.
- We have configured the Relay Server for this project in the unity dashboard.



```
◆ RelayManager ▶ JoinRelay(string joinCode)
6     using Unity.Services.Relay;
7     using Unity.Services.Relay.Models;
8     using Unity.Networking.Transport.Relay;
9     using Unity.Netcode;
10    using Unity.Netcode.Transports.UTP;
11
12    public class RelayManager : MonoBehaviour
13    {
14        public static string gameCode = "";
15        private async void Start() {
16            try
17            {
18                Debug.Log("Replay start method");
19                await UnityServices.InitializeAsync();
20
21                AuthenticationService.Instance.SignedIn += () =>
22                {
23                    Debug.Log("Signed in " + AuthenticationService.Instance.PlayerId);
24                };
25                await AuthenticationService.Instance.SignInAnonymouslyAsync();
26            }
27            catch (AuthenticationException e) {
28                Debug.Log("## "+e);
29            }
30        }
31
32        public static async void CreateRelay() {
33            try
34            {
35                StartSceneController.hostCodeStr = "";
36                Allocation allocation = await RelayService.Instance.CreateAllocationAsync(3);
37                string joinCode = await RelayService.Instance.GetJoinCodeAsync(allocation.AllocationId);
38                Debug.Log("Join Code - " + joinCode);
39                StartSceneController.hostCodeStr = joinCode;
40                gameCode = joinCode;
41
42                RelayServerData relayServerData = new RelayServerData(allocation, "dtls");
43
44                NetworkManager.Singleton.GetComponent<UnityTransport>().SetRelayServerData(relayServerData);
45
46                NetworkManager.Singleton.StartHost();
47            }
48            catch (RelayServiceException e) {
49                Debug.Log(e);
50            }
51        }
52
53        public static async void JoinRelay(string joinCode) {
54            try {
55                Debug.Log("Joining Replay with " + joinCode);
56                JoinAllocation joinAllocation = await RelayService.Instance.JoinAllocationAsync(joinCode);
57
58                RelayServerData relayServerData = new RelayServerData(joinAllocation, "dtls");
59
60                NetworkManager.Singleton.GetComponent<UnityTransport>().SetRelayServerData(relayServerData);
61
62                NetworkManager.Singleton.StartClient();
63
64                gameCode = "N/A";
65            }
66            catch ( RelayServiceException e){
67                Debug.Log(e);
68            }
69        }
70    }
71
72 }
```

