



COSC 829 Advanced Virtual Reality and its Applications

Project Title: VR Social Club

Assignment Due: December 9th, 2021

Faculty: Dr. Sharad Sharma

Team Members:

Syed Ali

Contents

1. Goal and Objectives	3
2. Modeling	3
3. Functionalities	15
4. Unity3D Functionality	20
5. Avatar behaviors	21
6. Conclusion	22
7. References:	22

1. Goal and Objectives

1.1 Introduction

My anticipated project is to build a multiplayer game called VR Social Club. The game is like Rec Room where players will be able to join in and be able to hangout with their friends. In the game players will be able to choose between two scenarios, one of them will be in a university gym setting where players will be able to join in and participate in activities such as playing basketball or watching TV, just chatting. The second scenario will be at a camp setting where players will be able to hang out at the camp, go for a swim, hike, etc. Most of this game module is built in Unity and 3ds Max.

1.2 Target Audience

The main target audience are employees from various companies who are craving for social events like office parties which have been difficult to hold due to the current pandemic. Currently there are not many videogames in the market which have a socialized theme implemented to them. Companies like Xbox game studios, Amazon Lua, Google Stadia (when it existed), and Steam are actively seeking independent developers to make video games like these for their platforms.

1.3 Application Usefulness

Currently this game is in its beta state, so there are not many features implemented to the game such as the ability to play basketball or table tennis. In the future I am hoping to implement these features which will improve the player's overall experience.

2. Modeling

2.1 Planned Geometry

Assets for Home, World Outdoor and World School Gym scenes were imported from website called Turbosquid and were scaled down or up in 3ds Max. Similarly, other assets such as NPCs, table tennis tables, nerf guns, boxes, television sets, etc. were also imported from websites such as Turbosquid and CGTrader, and were either scaled down or up on 3ds Max. The terrain was built inside the Unity Engine. Figures below describe the building of these assets.

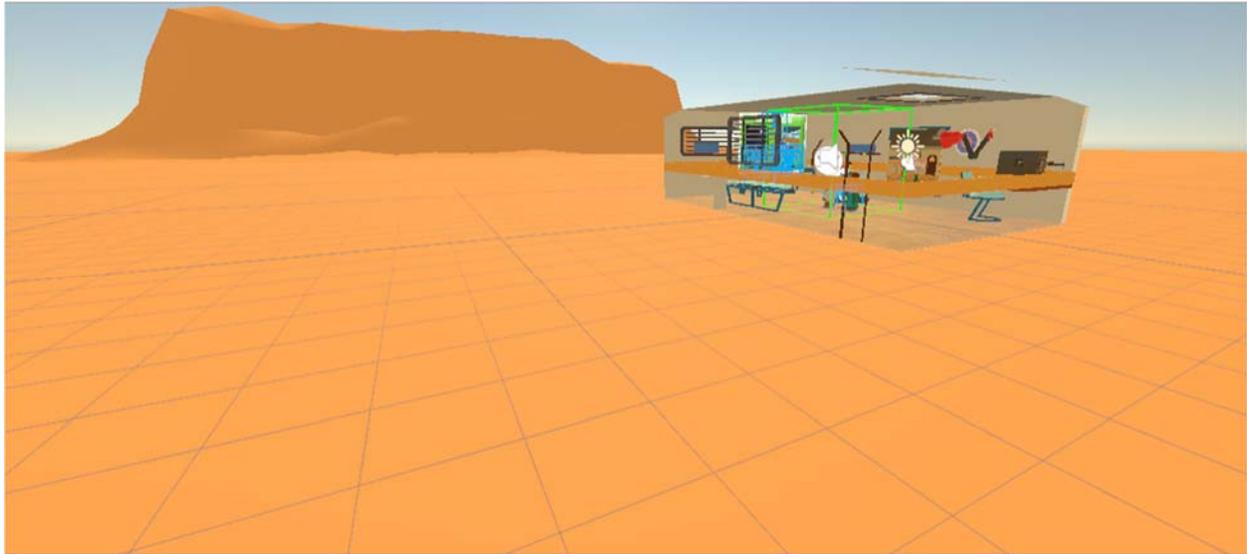


Figure 1: Terrain and Environment building inside Unity.



Figure 2: The building of UI canvas in Unity.



Figure 3: Table tennis model and stereo system imported from Turbosquid and scaled down in 3ds Max along with applying textures to the model in Unity.



Figure 4: Suited man asset imported from CG trader. The figure shows implementation of an animation state.

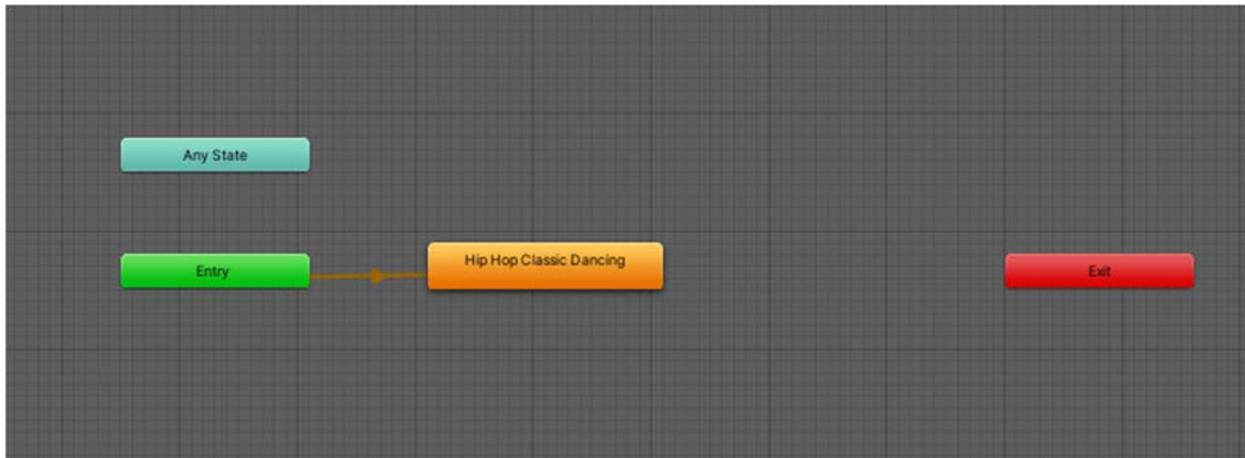


Figure 5: Animator controller for the suited man avatar and the parameters implemented to it.

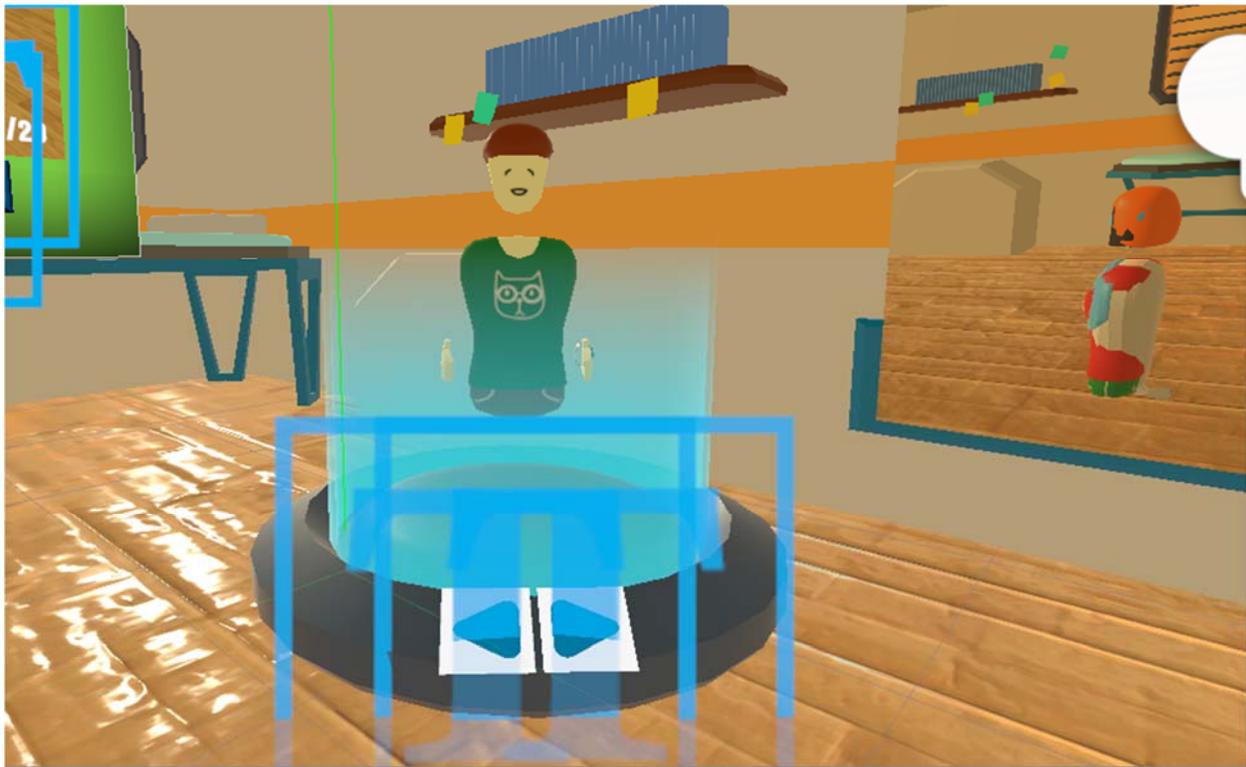


Figure 6: Avatar selection avatar asset imported from Unity. The figure shows implementation of avatar selection system.

2.2 Application Usage

As described in section 1.2, the main target audience are office employees. As the game begins, the player will first login using a unique ID, they can also login anonymously. Since this is a multiplayer game, multiple users will be able to login with a unique ID and interact with the environment. Once the user logs in, the user will have the ability to pick between two open-world environments which are namely the “World-Outdoor” and “Word-School”. Figures below describe the gameplay mechanics:

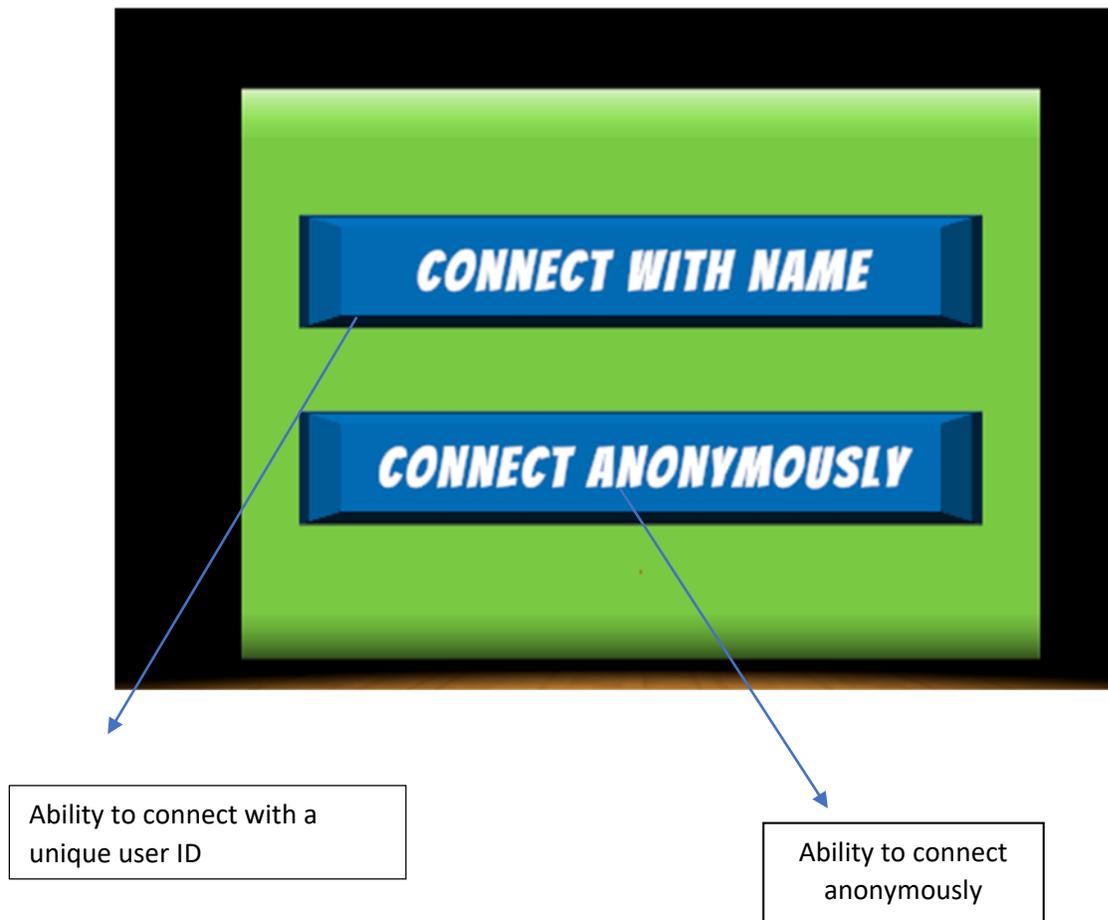


Figure 7: A figure describing the ability of the user login either anonymously or with a unique ID

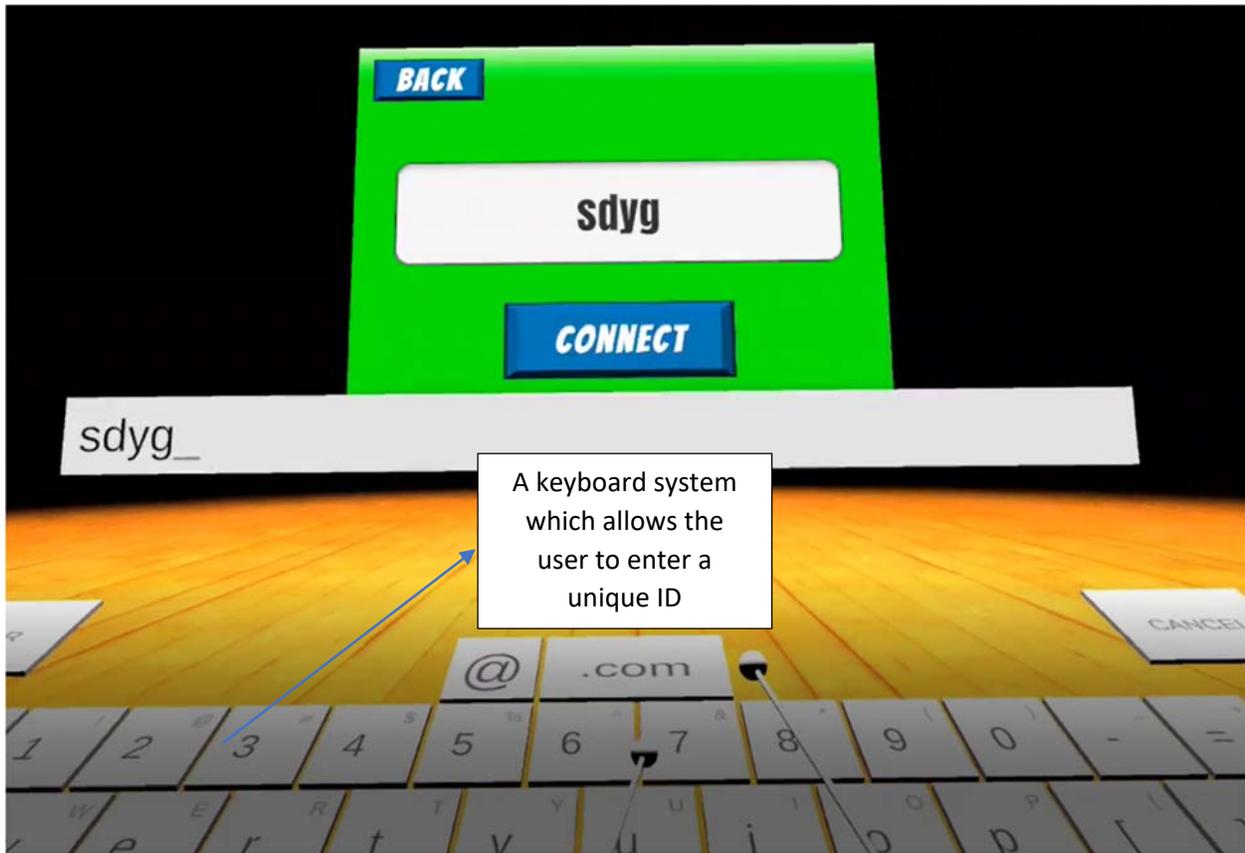


Figure 8: A figure describing the ability of the user to type in with a unique ID and connect to the environment.



A mirror allowing the user to look at the Avatar picked

The user can teleport anywhere within the Home environment

Certain objects within the environment will be interactable

Figure 9: A figure describing the Home Login scene after the user enters either anonymously or with a unique ID.

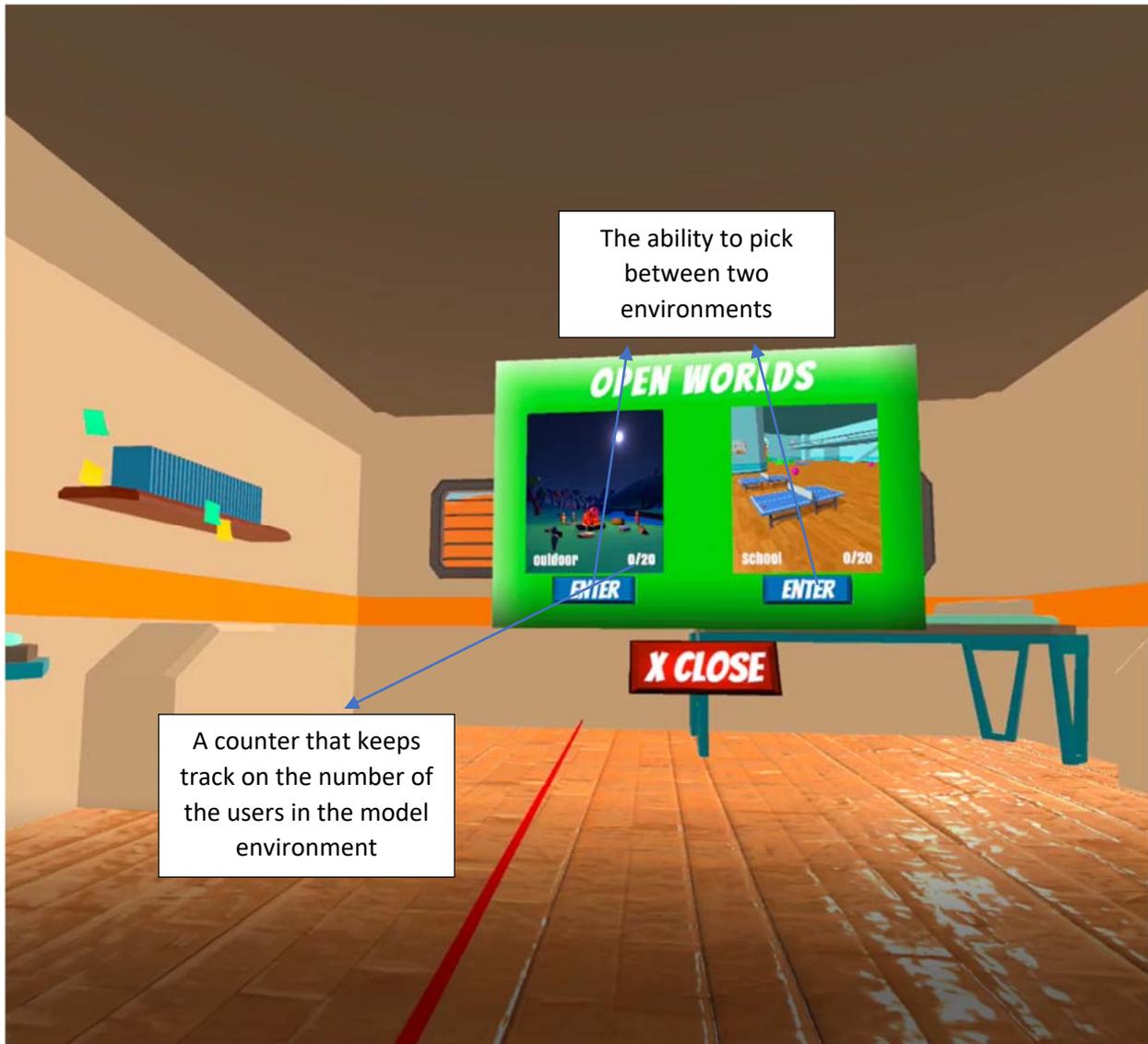


Figure 10: A figure describing the ability of the player to pick between two open worlds which are the World-School environment and the World-Outdoor environment.

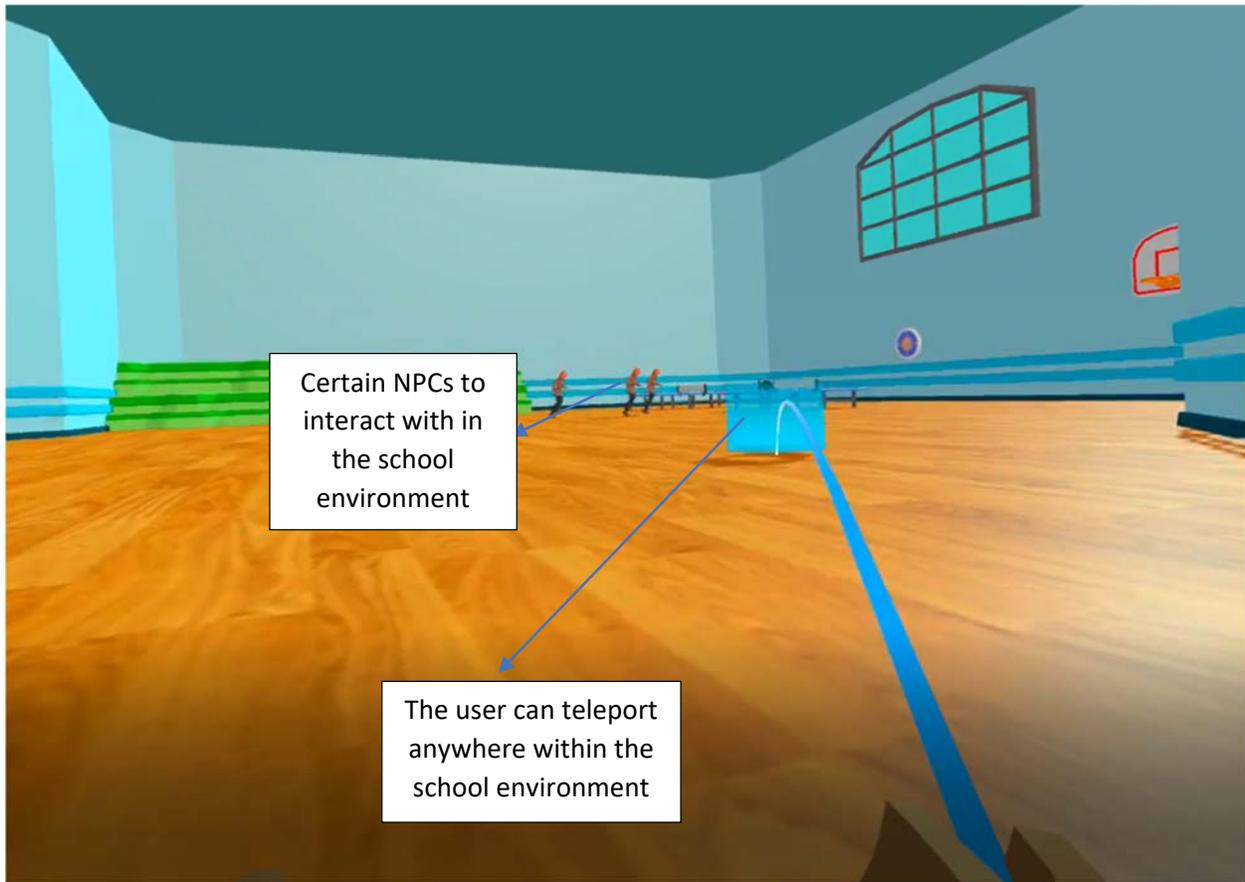


Figure 11: A figure describing the World-School environment.

2.3 Programming

Several programs were written in C# which provided specific competences within the game. The list of programs created along with their brief descriptions are provided in Table 1:

SCRIPT NO.	SCRIPTS	DESCRIPTION
1	AvatarHolder	This script allows the user to see the avatar’s hands in the VR head along with the movement of avatar’s head, body, and hands by calling on script 8.
2	AvatarSelectionManager	This script allows the user to switch between avatars while in the Home Scene.
3	LoginManagerEditor	This script allows the user to connect to the Photon servers.
4	RoomManagerEditor	This script allows the user to pick a specific open world they are interested in exploring.

5	HighlightVoice	This script allows the user to see if there is a communication between the two avatars.
6	MicPermissionHelper	This script allows the user to either enable to disable microphone option.
7	VoiceDebugUI	This script is a test script and allowed the user to see if there is indeed communication happening between the two avatars.
8	ActionBasedControllerManager	This script allows the user's avatar to fluidly move around the environment, such as the movement of head, body, and hands. This script came with the XR Interaction Toolkit installation.
9	AvatarInputConverter	This script allows the user to get avatar movements based on controller inputs. This script came with the XR Interaction Toolkit installation.
10	ButtonInteraction	This is a test script that ensured that the controllers are working properly.
11	LocalPlayerUIManager	This script allowed the user to go back to the Home Scene.
12	LoginManager	This is a test script that allowed the user to check if there is a connection established between the photon servers.
13	LoginUIManager	While in the Login scene, this script allows the user to either connect either with a name or connect anonymously.
14	Multiplayer VR Constant	This script contains constants that allows the user to implement them in script 18.
15	MultiplayerVRSynchronization	This script allows the user to synchronize head, body and hand movements when in different scenes. This script was taken from Photon's GitHub page.
16	NetworkedGrabbing	This script allows the to grab objects when connected to photon servers while in World-School and World-Outdoor scenes.
17	PlayerNetworkSetup	This script allowed the user to teleport to different areas. The script also allowed for synchronization of teleport-based movements when it different scenes.
18	RoomManager	This script is main script that takes in scripts 9-17. It basically allows the loading of scenes, movement of avatars when a certain room is selected, teleportation, item grabbing and so forth, while connected to Photon servers.

19	SpawnManager	This is a test script that allowed to user to test if the player can spawn to different areas while connected to the photon servers.
20	UI_InteractionController	This script allowed the user to enable or disable user interface based on Oculus controller inputs. This script came with XR Interaction toolkit installation.
21	VirtualWorldManager	This script allows the user to see how many users are currently on World-School and World-Outdoor scenes.

Table 1: The list of programs created along with their brief descriptions.

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using Photon.Pun;
5  using Photon.Realtime;
6  using TMPro;
7
8  public class RoomManager : MonoBehaviourPunCallbacks
9  {
10     private string mapType;
11     public TextMeshProGUI OccupancyRateText_ForSchool;
12     public TextMeshProGUI OccupancyRateText_ForOutdoor;
13
14
15
16
17     // Start is called before the first frame update
18     void Start()
19     {
20         PhotonNetwork.AutomaticallySyncScene = true;
21
22         if (PhotonNetwork.IsConnectedAndReady)
23         {
24             PhotonNetwork.ConnectUsingSettings();
25         }
26         else
27         {
28             PhotonNetwork.JoinLobby();
29         }
30     }
31
32 }
33
34 // Update is called once per frame
35 void Update()
36 {
37 }
38
39
40 #region UI Callback Methods
41 public void JoinRandomRoom()
42 {
43     PhotonNetwork.JoinRandomRoom();
44 }
45
46 public void OnEnterButtonClicked_Outdoor()
47 {
48     mapType = MultiplayerVRConstant.MAP_TYPE_VALUE_OUTDOOR;
49     ExitGames.Client.Photon.Hashtable expectedCustomRoomProperties = new ExitGames.Client.Photon.Hashtable() { { MultiplayerVRConstant.MAP_TYPE_KEY, mapType } };
50     PhotonNetwork.JoinRandomRoom(expectedCustomRoomProperties, 0);
51 }
52
53

```

Figure 12: An example of RoomManager script created in Visual Studio.

```
NetworkedGrabbing.cs x RoomManager.cs
NetworkedGrabbing
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using Photon.Pun;
5 using Photon.Realtime;
6
7 public class NetworkedGrabbing : MonoBehaviourPunCallbacks, IPunOwnershipCallbacks
8 {
9     PhotonView m_photonView;
10
11     Rigidbody rb;
12     bool isBeingHeld = false;
13
14     private void Awake()
15     {
16         m_photonView = GetComponent<PhotonView>();
17     }
18     // Start is called before the first frame update
19     void Start()
20     {
21         rb = GetComponent<Rigidbody>();
22     }
23
24     // Update is called once per frame
25     void Update()
26     {
27         if(isBeingHeld)
28         {
29             rb.isKinematic = true;
30             gameObject.layer = 11;
31         }
32         else
33         {
34             rb.isKinematic = false;
35             gameObject.layer = 9;
36         }
37     }
38
39     private void TransferOwnership()
40     {
41         m_photonView.RequestOwnership();
42     }
43
44     public void OnSelectEntered()
45     {
46         Debug.Log("Grabbed");
47         m_photonView.RPC("StartNetworkGrabbing", RpcTarget.AllBuffered);
48
49         if (m_photonView.Owner == PhotonNetwork.LocalPlayer)
50         {
51             Debug.Log("We do not request ownership.Already Mine.");
52         }
53     }
54 }
```

Figure 13: An example of NetworkGrabbing script created in Visual Studio.

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using Photon.Pun;
5
6
7 public class MultiplayerVRSynchronization : MonoBehaviour, IPunObservable
8 {
9
10     private PhotonView _PhotonView;
11
12
13     //Main VRPlayer Transform Synch
14     [Header("Main VRPlayer Transform Synch")]
15     public Transform generalVRPlayerTransform;
16
17     //Position
18     private float _Distance_GeneralVRPlayer;
19     private Vector3 _Direction_GeneralVRPlayer;
20     private Vector3 _NetworkPosition_GeneralVRPlayer;
21     private Vector3 _StoredPosition_GeneralVRPlayer;
22
23     //Rotation
24     private Quaternion _NetworkRotation_GeneralVRPlayer;
25     private float _Angle_GeneralVRPlayer;
26
27
28     //Main Avatar Transform Synch
29     [Header("Main Avatar Transform Synch")]
30     public Transform mainAvatarTransform;
31
32
33     //Position
34     private float _Distance_MainAvatar;
35     private Vector3 _Direction_MainAvatar;
36     private Vector3 _NetworkPosition_MainAvatar;
37     private Vector3 _StoredPosition_MainAvatar;
38
39     //Rotation
40     private Quaternion _NetworkRotation_MainAvatar;
41     private float _Angle_MainAvatar;
42
43
44     //Head Synch
45     //Rotation
46     [Header("Avatar Head Transform Synch")]
47     public Transform headTransform;
48
49     private Quaternion _NetworkRotation_Head;
50     private float _Angle_Head;
51
52     //Body Synch

```

Figure 14: An example of MultiplayerVRSynchronization script created in Visual Studio.

3. Functionalities

There were several key functionalities added to the game that allowed the player to feel more immersed into the environment and they are provided below:

3.1 Vision

In terms of vision the game has the following features:

- Terrain with grass, trees, and hills when in World-Outdoor scene
- A school gym containing table tennis, basketball hoops, nerf guns, tv, and audio device when in World-School scene
- A trailer containing a mug, mirror, bed, and work desk when in the Home Scene
- NPC Avatars
- First person character with real time body movements

An example figure below summarizes the vision aspect of the game:

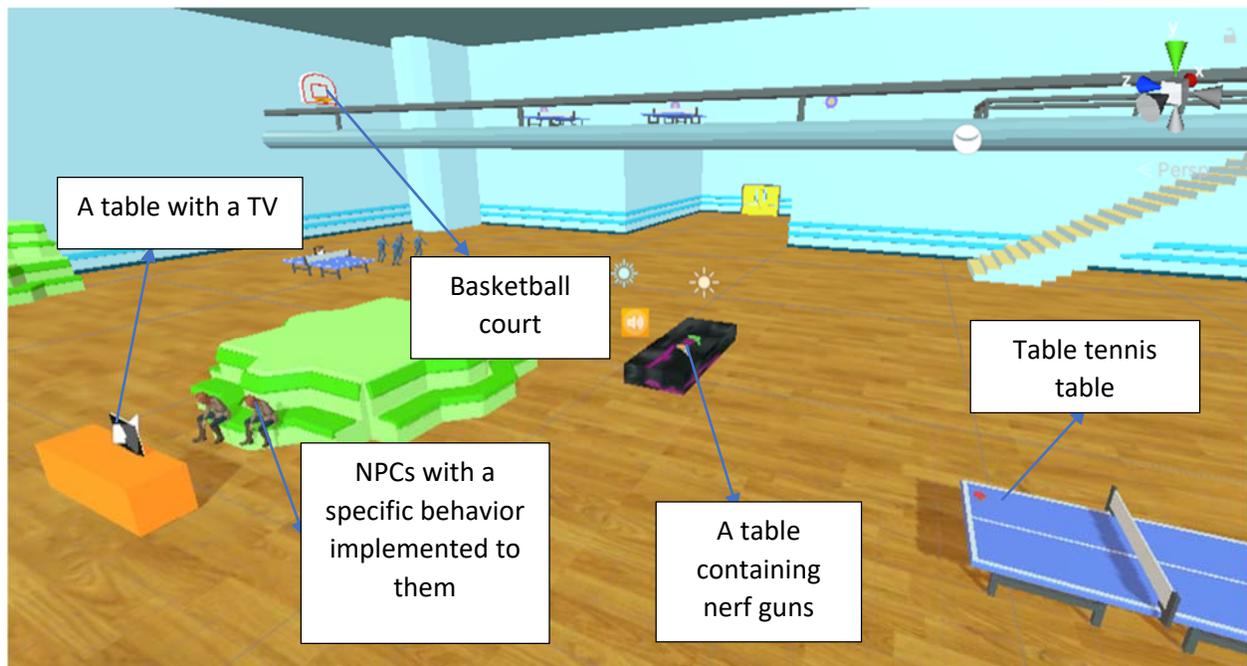


Figure 15: Example screenshot demonstrating the visual aspect of the game when World-Outdoor.

3.2 Sound

In the login scene, there is a music playing in the background. In the home scene there is music playing in the background along with a sound of an analog clock. While in World-School scene there are sounds of the TV along with songs being played on the audio devices. While in the World-Outdoor scene there are sounds of the campfire, along with conversations between several NPCs, along with sounds of crickets and wind.

3.3 Animation

There are several animations associated with NPCs, which include dancing, talking, and laughing animations. An example animation of an NPC is shown on the next page:

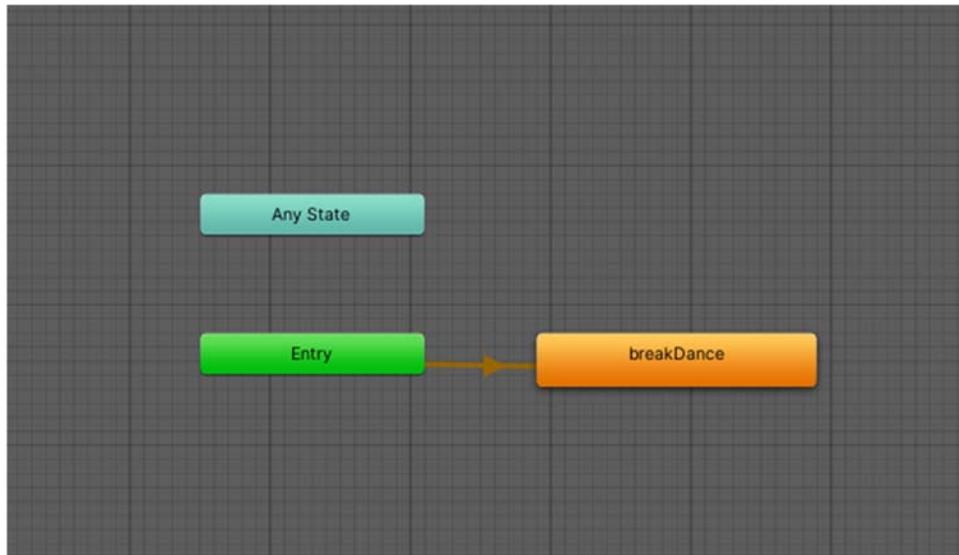


Figure 16: Animation path for a random NPC.



Figure 17: Break dance animation implemented to an NPC.

3.4 Interactivity

There are several interactable objects within the environment that allows to grab and interact with them. These objects can held by both hands. These objects will be seen by other players as well. Once the user picks an item, the photon network gives the user holding the object a unique ID allowing it to synchronize the view with other players as well. An example screenshot demonstrating the interactable object can be shown below:



Figure 18: A figure describing the ability to interact with the environment.

3.5 Sensor

There are several sensors in the game. There are proximity sound sensors placed in the environment. For example, as the player approaches the TV in the World-School environment the audio becomes clearer and louder and as the player moves away from the TV the audio sound dissipates as the player moves away from the TV. Similarly for the audio devices, as the user approaches an audio device the sound becomes clear and louder, and as the player moves away from the audio device, the audio sound dissipates over time.

3.6 Avatars

There are several NPCs in the game with different animations. In the World-School scene there are several NPCs with a dance animation associated to them, along with several NPCs having a laughing animation associated to them. In the World-Outdoor scene, there are several NPCs having conversational animations associated with them. A screenshot below shows an example of an NPC with a specific animation:



Figure 19: NPCs in the World outdoor environment. Talking animation is implemented to these NPCs.

3.7 Multiplayer Functionality

This game has a multiplayer functionality which allows up to 20 players to be in the environment and interact with each other at the same time. Voice synchronization has been implemented to the game which allows players to communicate with each other and perform various activities between each other. Figures below describe the multiplayer aspect of this game:



Figure 20: First person character in the World outdoor environment. This is from the perspective of an NPC. As I press the W, A,S,D keys the first person controller will be able to move around the environment.

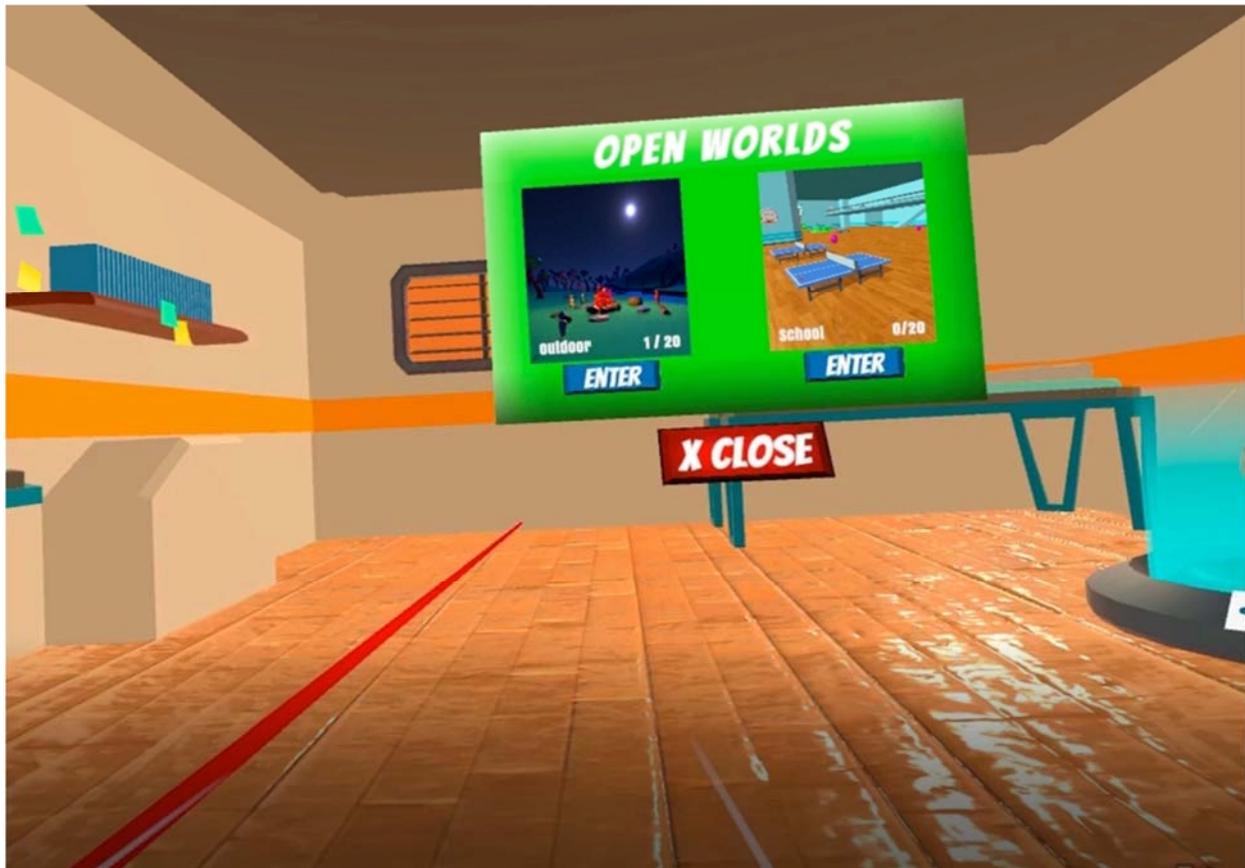


Figure 21: An example describing the multiplayer functionality. As you can see one player is present in the environment.

4. Unity3D Functionality

- **Lights:** In terms of lights, they vary across different scenes, for example the Home and World-School scene has indoor lighting, whereas the World-Outdoor has night lighting, and lighting coming out from the campfire.
- **Controller functionalities:** There are controller inputs for each action conducted by the player. For example, the A key on the right-hand controller is used for grabbing objects in the environment, whereas the A key on the left-hand controller is used for enabling the UI. The left-hand controller joystick is used to move around the environment whereas the right-hand joystick is used to teleport around the environment.
- **Proximity Sensor:** Each Audio devices have proximity sound sensors associated with them, and they increase or decrease in volume as the user moves closer to or away from the audio device.
- **Avatar and Avatar Animation:** There are 10 – 15 avatars, with most of them having repeated animations from dancing to laughing to talking animations.

- Environment: There is sky, moon, grass, trees, and a lake added to the terrain when in the World-Outdoor scene. In the World-School scene there are basketball courts, table tennis court, nerf guns, TV, and audio devices as shown in figure 15.
- Audio: The login scene has a unique music playing in the background whereas the home scene has its own unique music being played in the background along with a sound of an analogue clock. The World-School scene has different audio devices playing different music. The World-Outdoor scene has sounds of fire crackling along with sounds of crickets and wind.

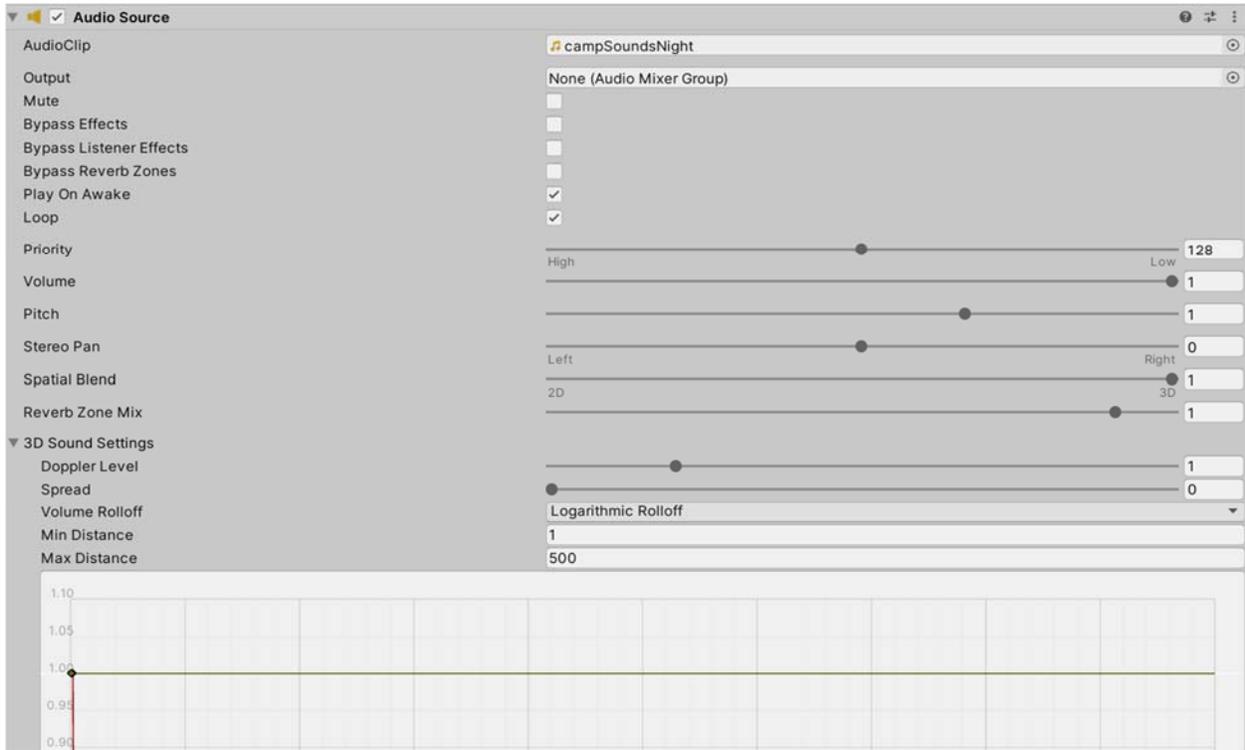


Figure 20: An example of an audio source added to the World Outdoor scene.

5. Avatar behaviors

The NPCs have different behaviors that range from dancing, to laughing, to talking. An example of an NPC with a specific behavior is outlined in figure 21:



Figure 21: NPCs with a laughing animation associated to them.

6. Conclusion

As mentioned in the goals and introduction section of this report, this game module is currently in its beta state. There is still a lot more work that needs to be done. I am hoping to implement more features such as the ability to play basketball or table tennis when in the World-School environment, and in the World-Outdoor environment I am hoping to implement the ability to go for a swim or hike somewhere within the environment. I am also hoping to improve the overall rendering of the environment to make it more immersive

As discussed in the target audience section of this report, currently there are not many games in the market that have a socialized theme implemented to them, which is one of the reasons why companies like Xbox game studios are seeking independent developers to help make these genres of games for their platforms.

7. References:

Assets:

- <https://www.turbosquid.com/3d-models/3d-trailer-house-model-1386729>
- <https://www.turbosquid.com/3d-models/3d-office-table-items-model-1198135>
- <https://www.turbosquid.com/3d-models/3d-old-dirty-bed-pillows-1412086>
- <https://www.cgtrader.com/3d-models/interior/other/school-gym-c43dca01-ee0c-424c-989f-622e5cbfb14c>
- <https://www.cgtrader.com/free-3d-models/sports/equipment/pingpong-table>
- <https://www.cgtrader.com/free-3d-models/interior/living-room/smart-tv-c075ac00-05c7-42ab-8665-aed7b0d575b5>
- <https://www.turbosquid.com/3d-models/subwoofer-woofer-3d-model/527825>

- <https://www.turbosquid.com/3d-models/isometric-style-model-1318759>
- <https://www.turbosquid.com/3d-models/3d-model-school-old-chair-1541002>
- <https://www.turbosquid.com/3d-models/toy-water-gun/775253>
- <https://www.turbosquid.com/3d-models/3d-man-business-suit-t-pose-1518255>
- <https://github.com/BlackStartx/VRC-Gesture-Manager>

Animations:

- <https://www.mixamo.com/#/>

Code:

- <https://github.com/AwesomesauceLabs/vrkb-manual>
- <https://docs.unity3d.com/ScriptReference/SceneManagement.SceneManager.LoadScene.html>
- <https://github.com/quintesse/PlayoVR>
- https://github.com/emotitron/Simple_Addon_for_Pun2
- <https://github.com/mtaulty/SharingPUN/tree/master/PhotonTestProject/Assets/Photon%20Unity%20Networking/UtilityScripts>
- <https://github.com/lyuma/Av3Emulator>
- <https://github.com/julce/April-Tag-VR-FullBody-Tracker>