Multi-user Evacuation in a mall

Fall Semester 2021
Aishwarya Reehl
Faculty: Dr. Sharad Sharma

Project Report

GOAL AND OBJECTIVES

The goal for this project is to create a Multi-User evacuation system in a mall. This game can be played individual or in a team. This would mean that either the user can create a game room and play there or join an existing room to play. In this environment the user can walk through the mall to see the layout. The user by default enters the fire and smoke environment where there is a lot of bombs, fire, smoke, and scared users.

The game is designed using Unity3D gaming engine and scripted in C#. The objective is to design and evacuation system where they can trigger evacuation and the people in the mall can hide to a safe spot.

In this project we are intending to design a multi-user evacuation which is very much real to the open shopping malls. This reason behind doing this project is with the increased number of attacks its really useful to have a virtual environment which helps in better planning evacuation.

This application is useful for anyone trying to understand and create evacuation system. The targeted audience is anyone who wants to design, learn or educate about the evacuation procedures/methods.

MODELING

The environment was created using Unity 3D platform and SketchUp.

The various elements like trees, grass, mountains, stones, water, and plane were designed using the terrains feature in Unity 3D as shown in Figure 1 below.

Figure 1 The load scene showing fireworks, fire, explosion, bombs, avatars and the terrain

There are many more elements present in the scene like patio furniture, cars in parking lot, mall structure with roads, gas station, electrical substation, police cars, fireworks from explosions, smoke etc as shown in Figure 2 and Figure 3 below.

Figure 2 Showing the Parking lot, church and cars

Figure 3 Showing electrical substation, grass, tree and water elements

The users can navigate in the scene using “WASD” controls, using mouse and keyboard.

For triggering evacuation, they can press the “G” on the keyboard as shown in Figure 4 below.

Figure 4 Showing the evacuation message and welcome message for the mall

The multi-user environment was implemented using Photon. This helps in creating multi-user environment. The number of Concurrent users i.e. CCUs configured are 20 as shown in Figure 5 below.

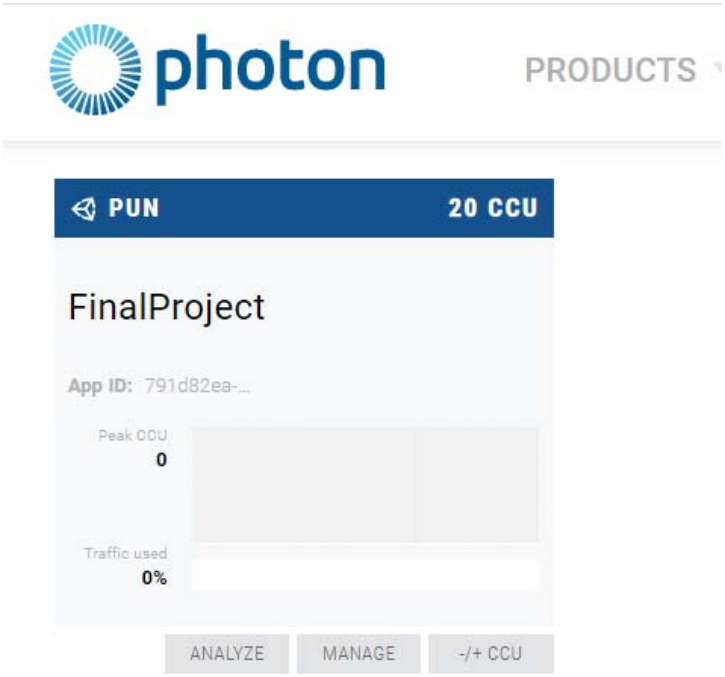


Figure 5 Showing Photon-PUN configuration

Figure 6 below shows how messages are transferred between Unity and Photon-PUN.

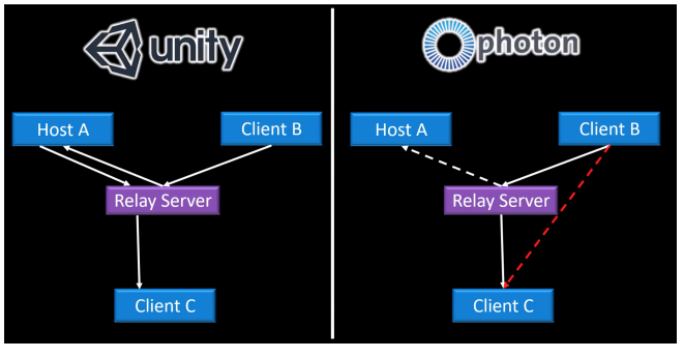


Figure 6 Architecture diagram for Unity and Photon

For programming I have used C# language. For example, as shown below in Figure 7 the Player.cs script below shows how the police car used in evacuation scene is hidden in initial load and when the evacuation is triggered the cars come into the scene.

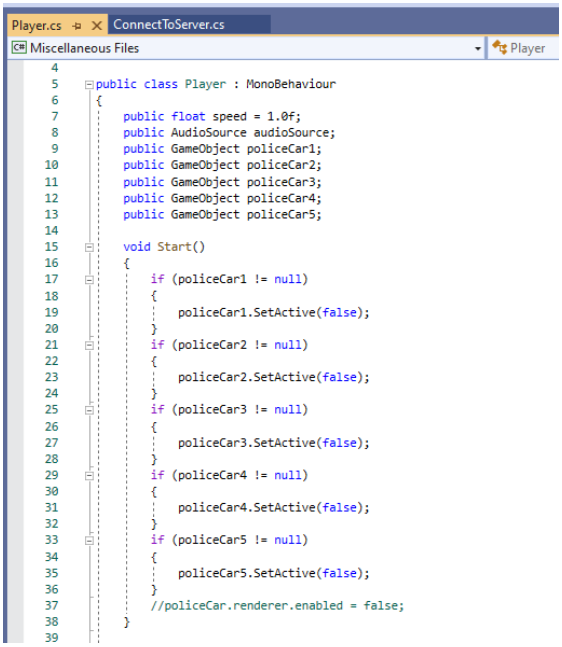


Figure 7 Player.cs script

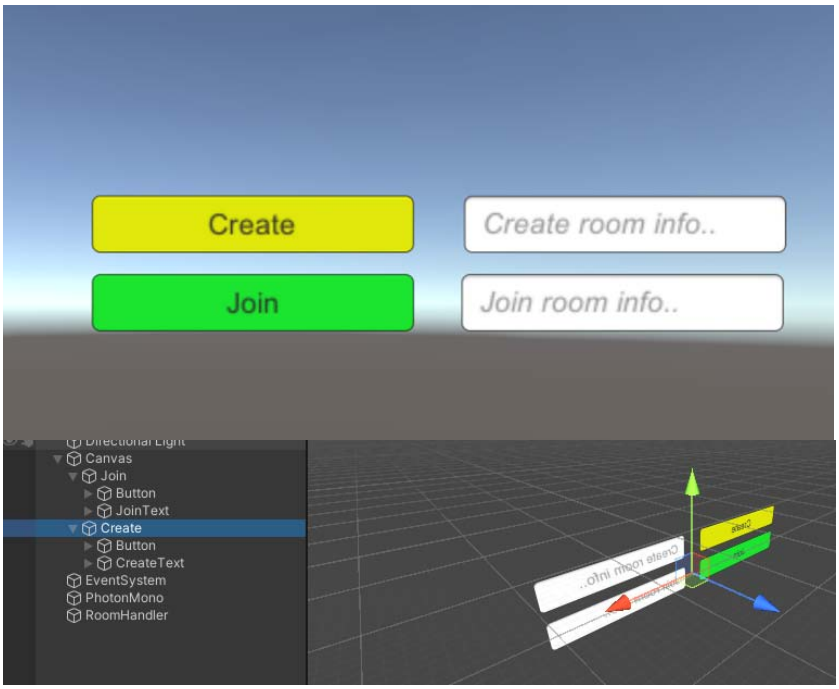
FUNCTIONALITY

INPUT:

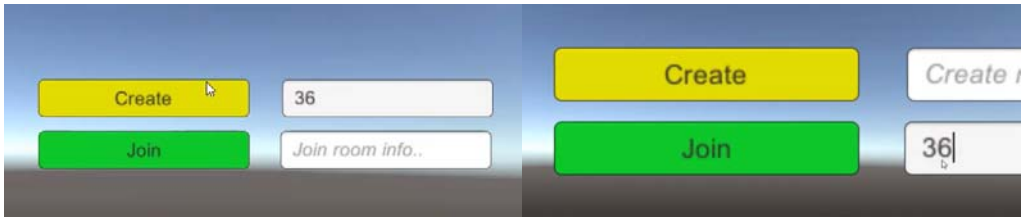
In this project we have used input for control. To turn left and right, the player may use A and D or the left and right arrow keys. To go front should use W key and go back use S key or top and bottom arrows.

MENU:

I have created a menu where the user can chose to either create a new room or join an existing room. The users can enter the room info using keypad and in the input areas as shown in Figure 8(a). Once they click “Create” a new room is created. If they click “Join” the join an existing room as shown below in Figure 8(b).



(a)



(b)

Figure 8 (a)Showing the menu option with input options (b) Showing the create and join room options

AUDIO:

The audio is triggered when the control reaches near the cube as shown in Figure 9 below.

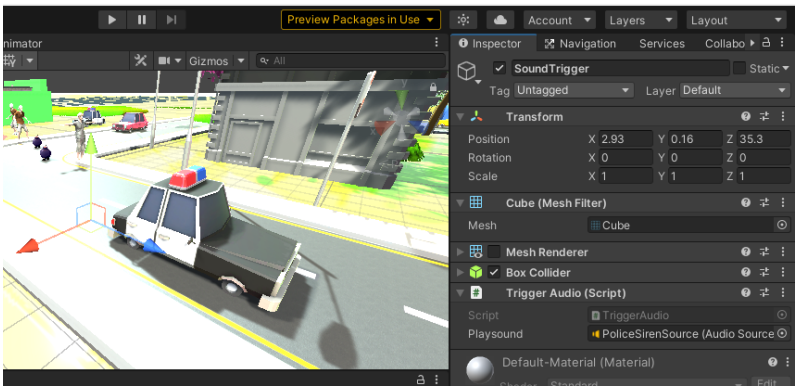


Figure 9 Showing the trigger audio script applied near the police cars

I created the triggeraudio.cs script that starts the siren sound as shown in Figure 10 below. This is applied throughout near all police cars.

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4
5   public class TriggerAudio : MonoBehaviour
6   {
7       public AudioSource playsound;
8       void OnTriggerEnter(Collider other)
9       {
10          playsound.Play();
11      }
```

Figure 10 TriggerAudio script

ANIMATION:

There are various animations used in the project. I created an animation for the welcome text using animator as shown in Figure 11 below.

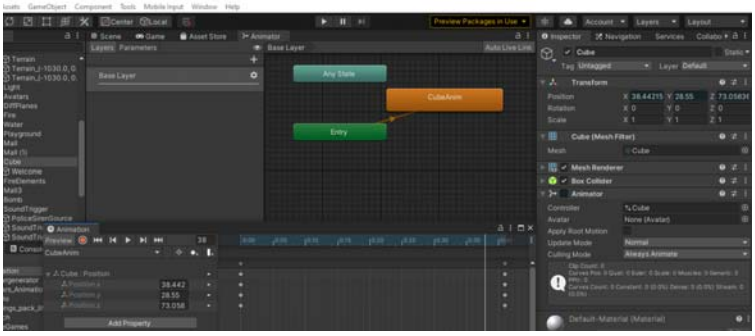


Figure 11 The animator created for animating the welcome text box

For Avatars I have used various animations like walking in place as shown in Figure 12, covering ears, a lady hit with bullet as shown in Figure 13.

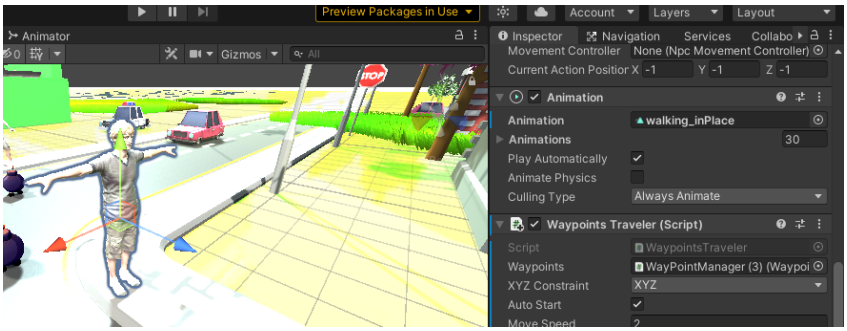


Figure 12 Showing the walking in place animation used for the avatar

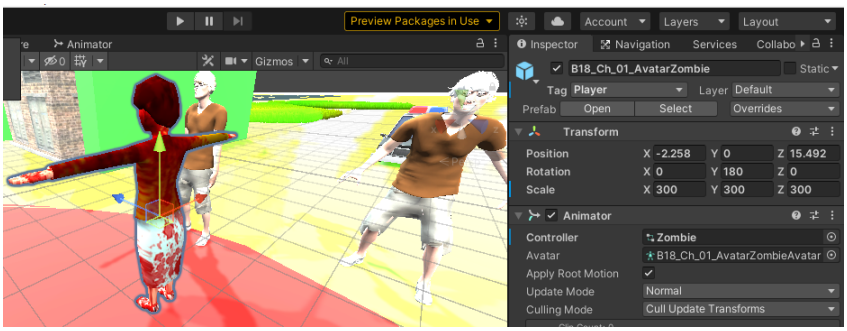


Figure 13 Showing the zombie controller used

WAYFINDER:

I have used wayfinder to make avatars walk in a predefined path as shown in Figure 14. For many avatars I have used wayfinder points as shown below and applied it to the avatar later as shown in Figure 15 below.

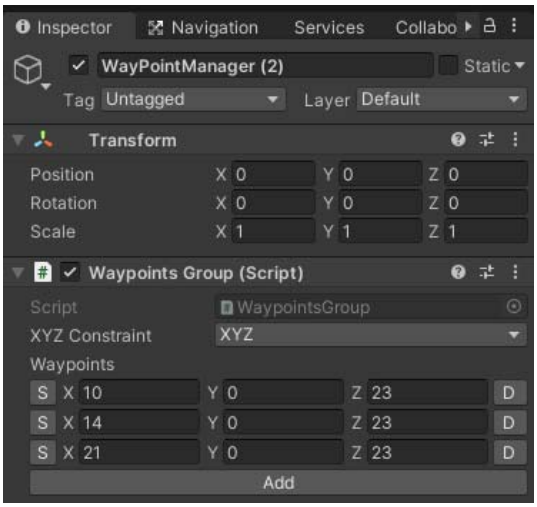


Figure 14 Waypoint finder points

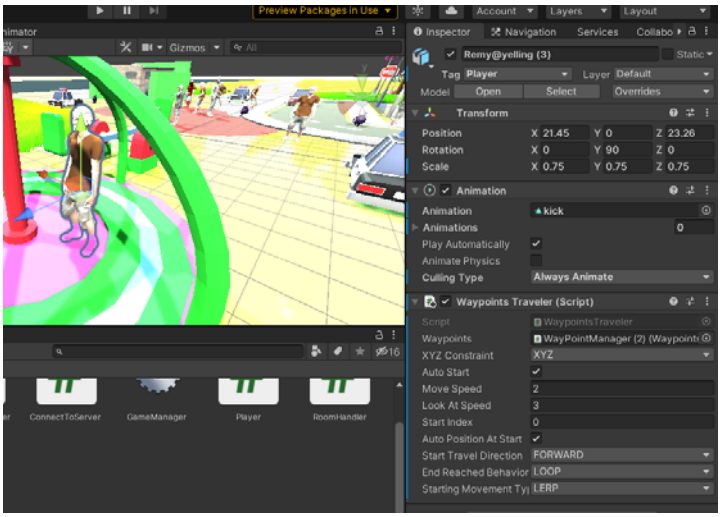


Figure 15 Showing how to apply the wayfinder to the avatar

COLLIDER:

The collider is used to initiate the trigger when the First person controller enter the box collider area as shown in Figure 16 below. At this time the health bar also reduces as shown in the Figure 17 below.

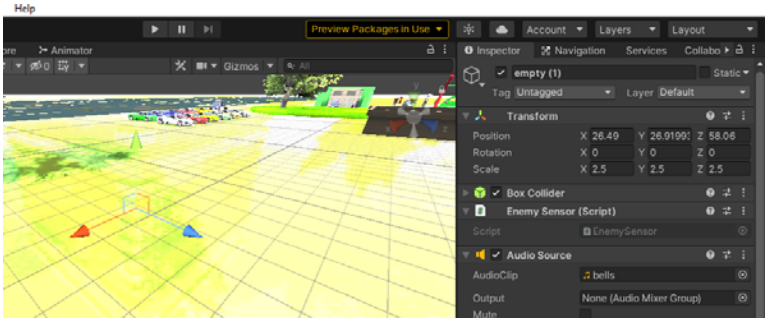


Figure 16 Showing the box collider for Enemy Sensor

HealthBarScript.cs

Miscellaneous Files

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4   using UnityEngine.UI;
5
6
7   public class HealthBarScript : MonoBehaviour {
8       Image healthBar;
9       float maxHealth = 100f;
10      public static float health;
11
12      // Use this for initialization
13      void Start () {
14          healthBar = GetComponent<Image>();
15          health = maxHealth;
16      }
```

Figure 17 Showing the HealthBarScript

SENSOR:

The enemy sensor is used for the First person controller (FPSC) which triggers whenever the FPSC enters the collider area as shown in Figure 18 and 19 below. Like in this project, I am using it while entering the electrical substation.

Figure 18 FPSC entering evacuation area

EnemySensor.cs | Player.cs | ConnectToServer.cs

Miscellaneous Files | EnemySensor

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4
5   public class EnemySensor : MonoBehaviour {
6
7       public void OnTriggerEnter(Collider other)
8       {
9           HealthBarScript.health -= 20f;
```

Figure 19 Enemy Sensor script

FIRST PERSON CONTROLLER:

The First-Person Controller (FPSC) is a soldier/militant who has a gun in his hand. The initial scene is loads with the soldier entering the scene and throughout the game the soldier stays in front of the camera as shown in Figure 20 below. I have added Photon scripts to the FPSC so that it can communicate with the FPSC in the scene as shown in Figure 21 below.

Figure 20 FPSC entering the scene

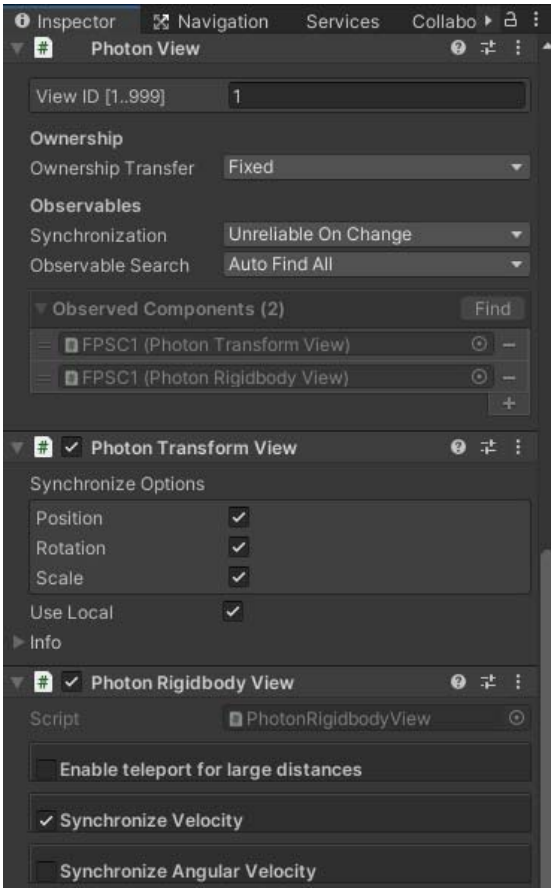


Figure 21 Photon scripts for FPSC

AI:

When the event is triggered, all the avatars will go to the electrical substation automatically as shown in Figure 22 below.

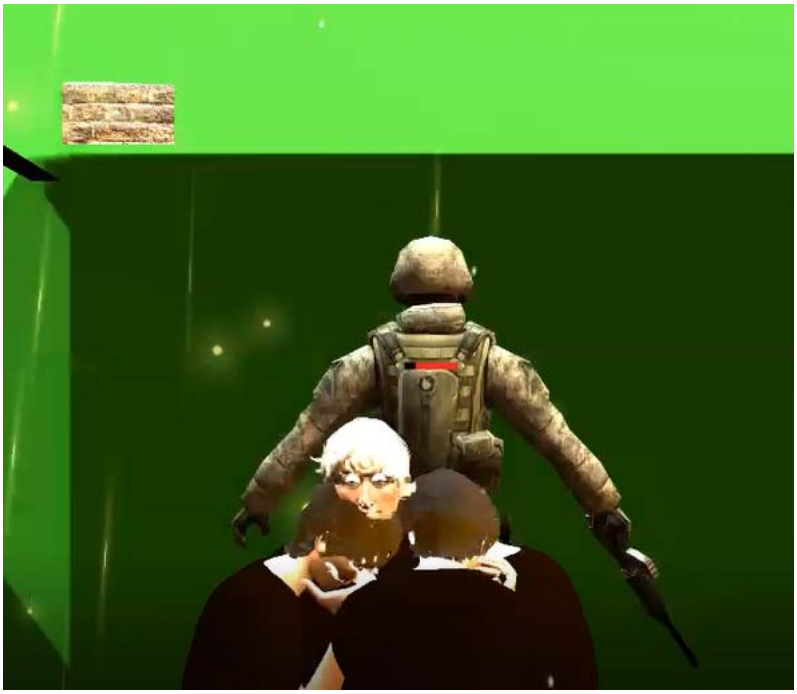


Figure 22 All avatars are gathered in electrical substation when evacuation is triggered

The Police cars will come with the evacuation event and audio source is of siren is triggered as shown in Figure 23 below.

Figure 23 Showing Police cars entering scene in case of evacuation

MULTIUSER:

For this project I have used Photon Unity Network (PUN) to create multiuser environment. Using Photon helps as its on cloud and connecting to it easy. As seen below in Figure 24 the

Photon is connected using the APP ID created in PUN website and these connections help in creating multi user in the scene.

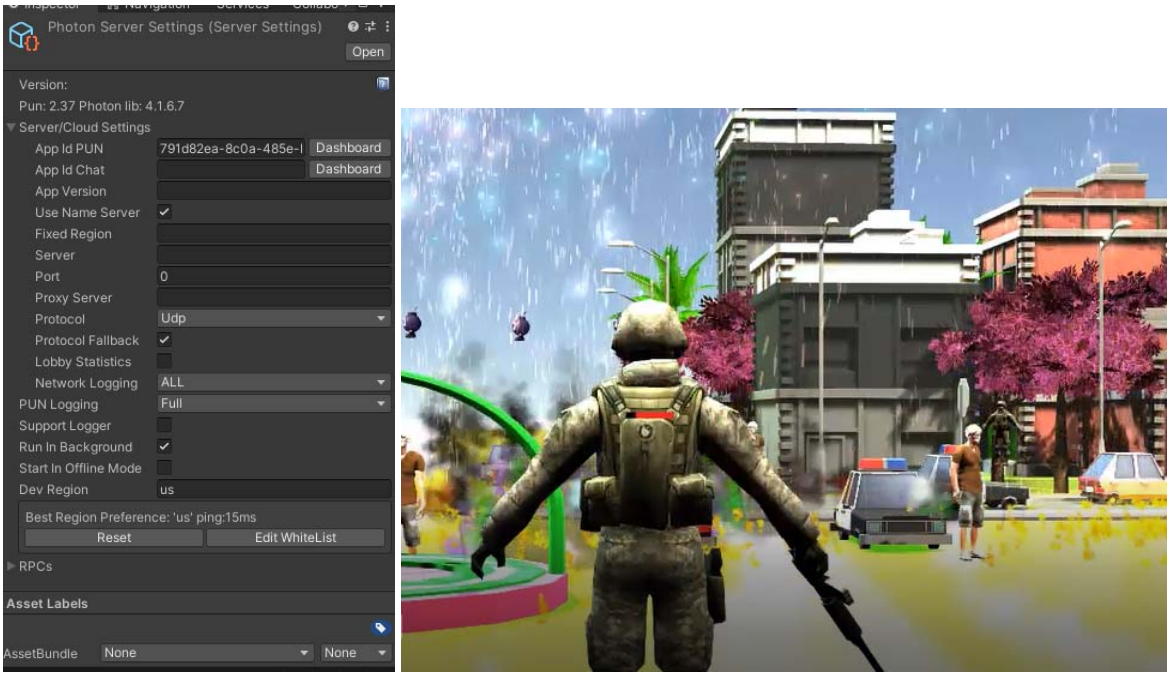


Figure 24 Photon connection and the second FPSC from the other scene

The script ConnectToServer initiates the connection to the Photon and allows multiuser environments as shown in Figure 25 below.

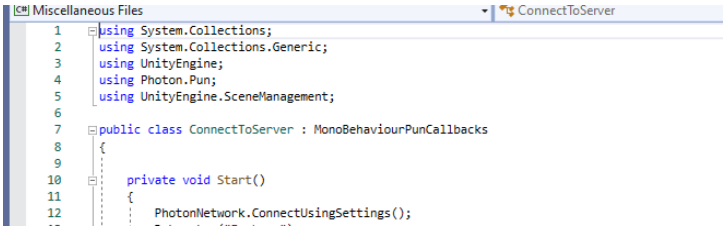


Figure 25 Showing Photon connecting scripts

There are many other elements added in the scene like Fire elements and fireworks as shown in Figure 26 and 27 below.

Figure 26 Showing Fire elements

Figure 27 Showing Fireworks

AVATARS

There are many avatars used in the project as shown in Figure 28 below with multiple animations used.

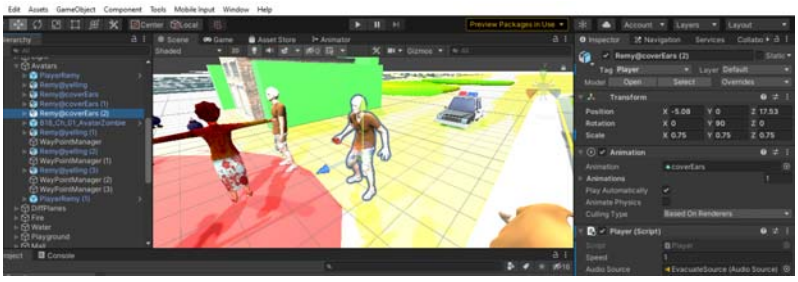


Figure 28 Showing all avatars used in the project

Animators for Avatars are as shown below covering ears as shown in Figure 29, Yelling as shown in Figure 30 , Walking as shown in Figure 31 and Waving as shown in Figure 32.

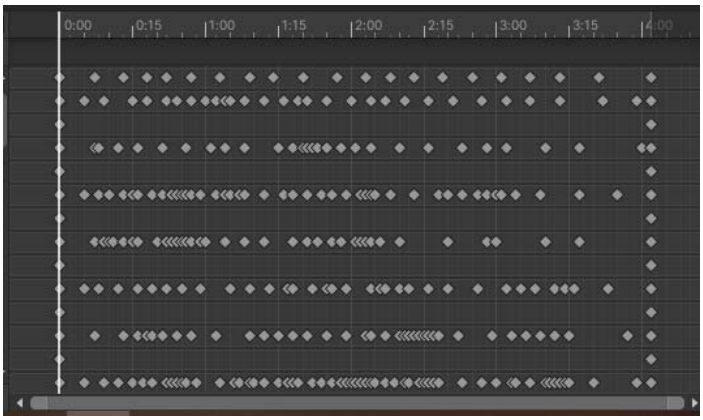


Figure 29 Showing covering ears

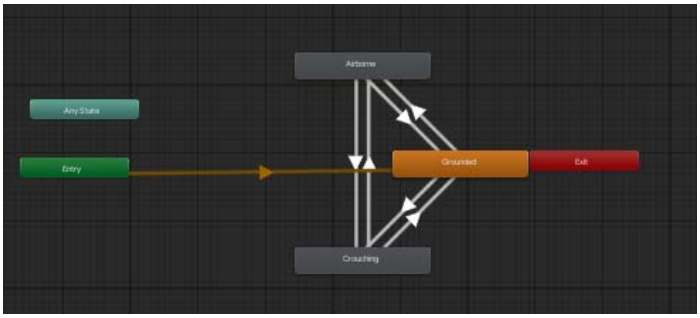


Figure 30 Showing yelling animator

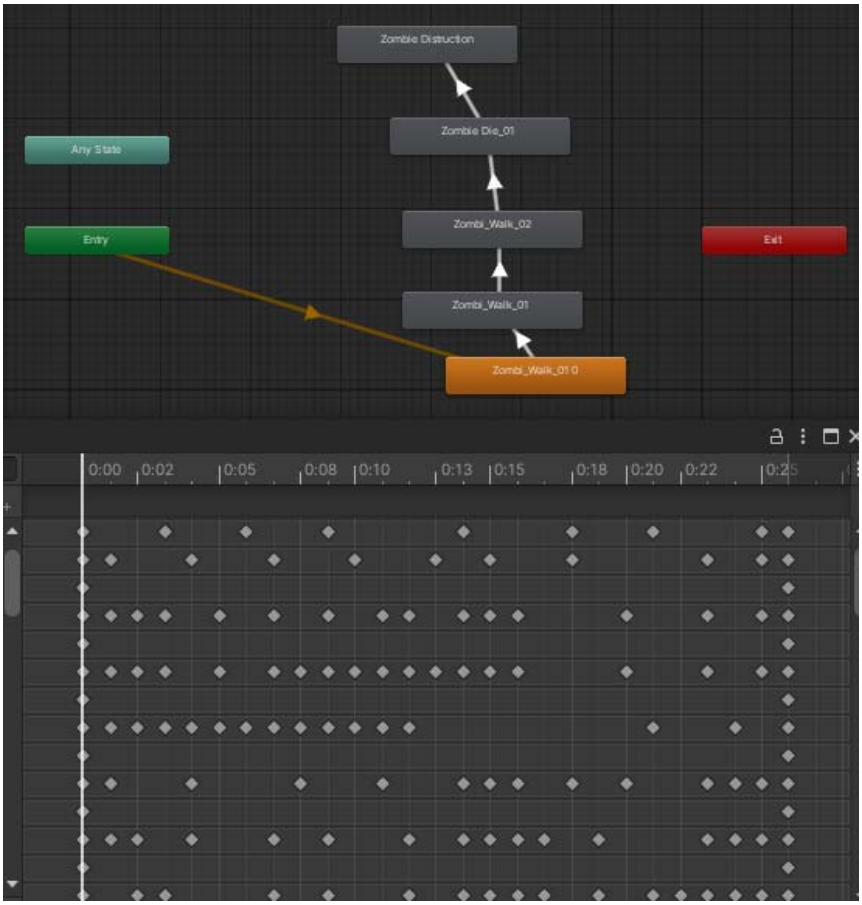


Figure 31 Showing Walking in place Animator

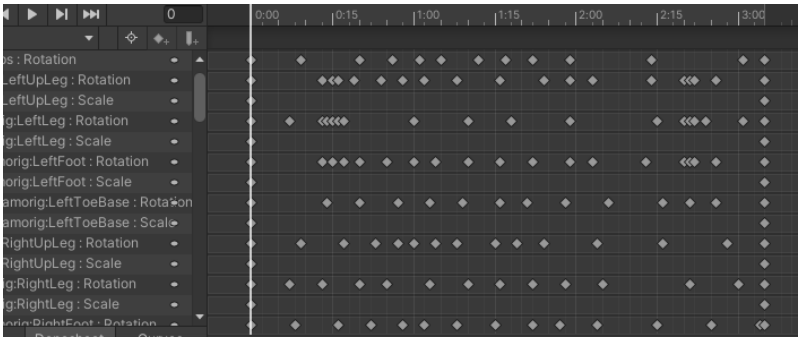


Figure 32 Showing Waving animation

I feel this application is extremely useful in understanding evacuation models for any structure. I also feel virtual reality is the best option because it gets the scenario as close to real life as possible.

CONCLUSION AND FUTURE SCOPE

In this project I created a multiuser evacuation for the mall. The users are in the mall and when the event was triggered, all the avatars go to the safe place i.e. the substation. The police

enters the scene in evacuation so that the users feel safe. I have used Photon component to create the multi-user environment. In the future, I will add more avatars in the project and then create multiple scenes.

ACKNOWLEDGMENT

I would like to thank Dr Sharad Sharma for his continuous support and guidance through the class and project. I would also like to thank the Computer Science department at Bowie State University for the materials and support provided.

I would also like to mention that the system architecture diagram was used from the website *https://www.raywenderlich.com/1142814-introduction-to-multiplayer-games-with-unity-and-photon* which helped me in understanding the architecture better.