

Multi-User Active Shooter Response

Overview:

The goal for this project is to create a Multi-User Active Shooter Response game with Unity3D gaming engine. This game can play individual or cooperative. You can hold a game play by yourself or you can join other people's game. In the game you can walk through the campus, you can talk with the avatars, you can drive the car. When you trigger the event, gangster will appear both in the building and outside the building. The players need to find all the gangster in 60 seconds and destroy all the enemy. When they destroy all the enemy or after 60 seconds the event will end. When the gangster appear all the other avatars will go to the gym and the police will stand outside the gym to protect the people who are in the gym. For this report, it will cover the defined of the environment, and how to do the multiplayer, animation, user interaction, sensors and the AI functions. This game can be played across multiple platforms.

Environment:

This game is set in a campus. It had a gym, buildings, ground. All these 3D modules were downloaded from Unity Asset Store. The 3D modules were shown in figure 1(a) and 1(b).



Figure 1(a)

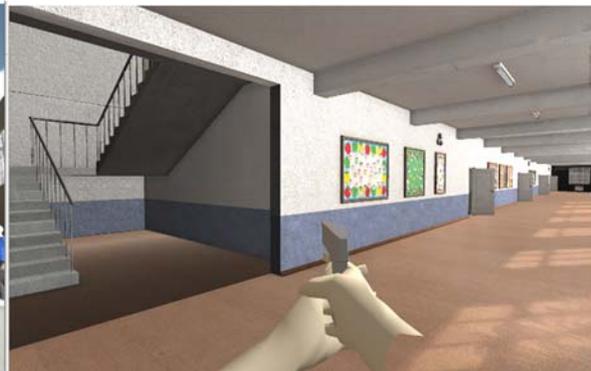


Figure 1(b)

From the figure 1 we can see that, the environment has a lot of stuff, trees, light, avatars. In this game it has four different kinds of avatars. It is shown in figure 2. Figure 2(a) is the enemy who will attack the campus in the game. Figure 2(b) is the police who will protect the students. Figure 2(c) is the teacher and figure 2(d) is the student. In this game we totally has 4 enemies. To enhance immersion in the campus, I added shadowing for the models. For lighting, my project uses the directional light for ambience.

Audio is added to three objects in this game. When we walk in the game we can hear a sound, when we trigger the event an alert sound will appear. When we fight with the enemy a blasting sound will be played. When we want to reload the bullets, a sound will play. These audio sources can help to increase immersion when they play the game.



Figure 2(a)



Figure 2(b)



Figure 2(c)



Figure 2(d)

In this game I had a driving model for the users. The user can drive the car around the campus. Figure 3 is the car I had for this game.



Figure 3

Input:

For this game, it relies on keyboard and mouse input for control. To turn left and right, the player may use A and D keys. To go front should use W key and go back use S key. To jump may use the space key. The players may use F key to talk with other avatars in the game and use F key to open the door and drive the car. The player use R key to reload the bullet and use mouse left click to fire.

Audio:

For this game, I use AudioSource to do the audio module. In this game when you shoot it will has a sound, when you walk it will have sound. When you trigger the even a alert sound will apper. When you drive the car it will also has a sound. All the sounds were player through the AudioSource.

Animation:

Character Animation: In this part, it include the avatar walk, fire(shoot) and jump. Figure 4(b) shown how the character animation works in the Unity3D. From the figure, we can know that, 'idle' is the waiting animation, 'fire' is the shoot animation, 'fightrun' is the running animation, and 'reload' is reloading bullet animation. In this module, 'fire' and 'reload' animation was player by the function "Animator.Play(animation name)". For running, I was use a bool variable called 'isMove', it is shown in figure 4(a). If the variable is true, it will run, else it will stay and waiting.

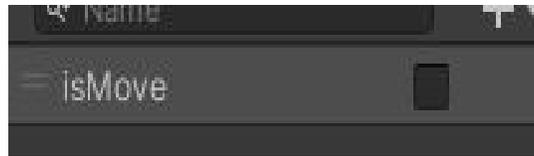


Figure 4 (a)

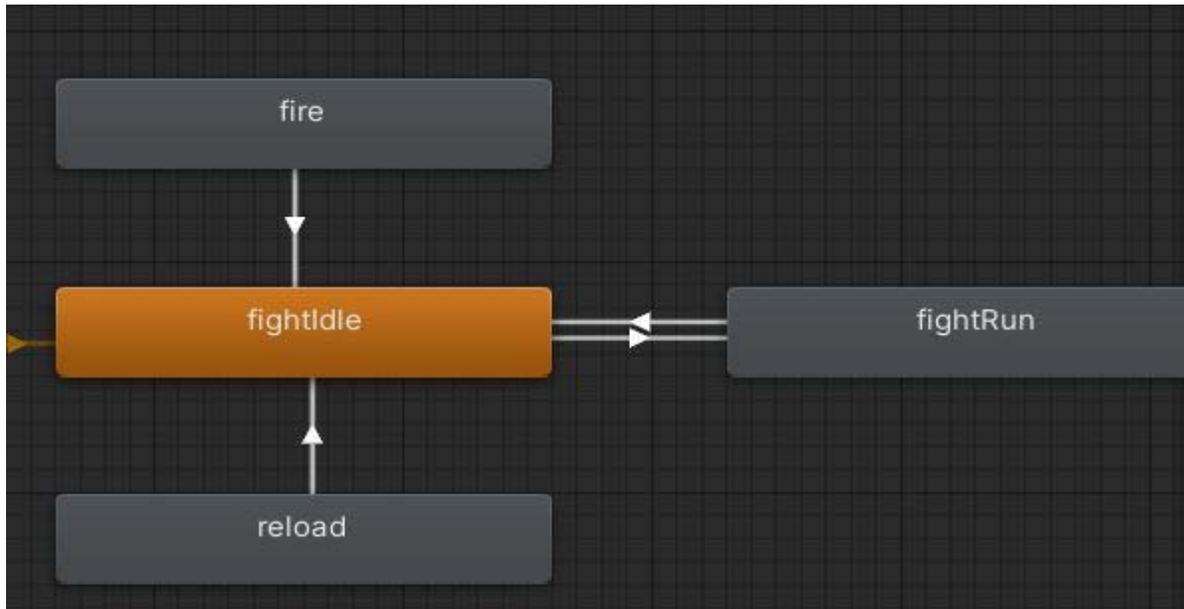


Figure 4(b)

Enemy Animation:

For the enemy animation, it is also use animator to control. The animation include standby animation 'idle', the death animation 'die', and fire animation 'shoot'. It is shown in figure 5(a). For the enemy animation I use two bool variables: 'isEnemyDie' and 'isShoot'. If 'isEnemyDie' is true it means the enemy is die, it will play the death animation. If 'isShoot' is true, it will fire else it will standby.

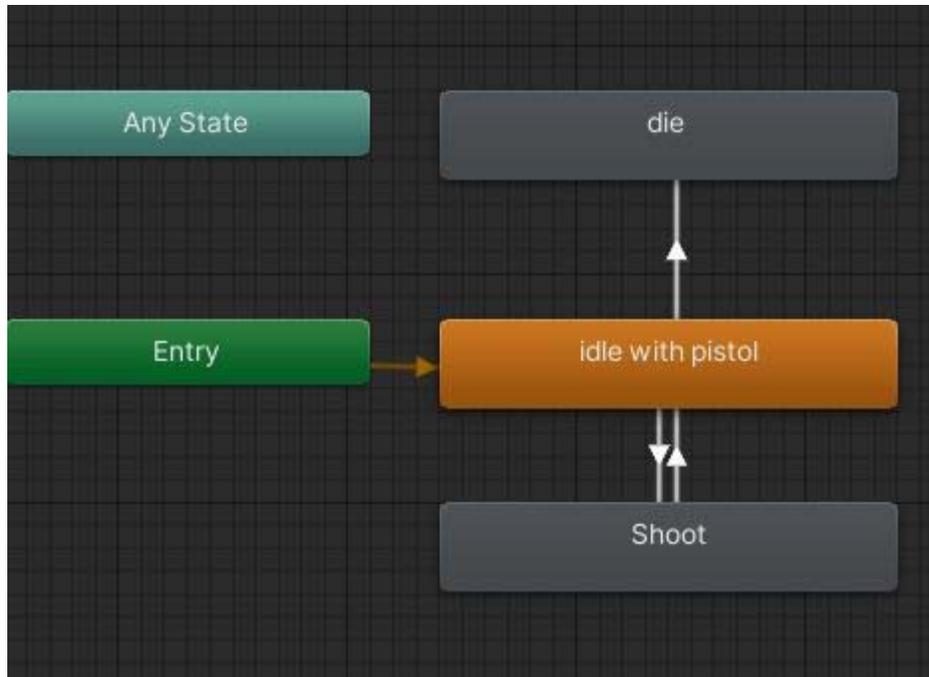


Figure 5(a)

NPC animation:

For the NPC animation, I use 'NavMeshAgent' to navigate the scene using the NavMesh. The NPC will walk from position 1 to position 2 then back to position 1 again.

Door Animation:

From figure 6 shown below, we can know that the default status for the door is closed. I use a bool variable 'isOpen', if it is true it will open the door, else it will close the door. I also use a sensor for the door open/close. I will talk it later in the sensor part.

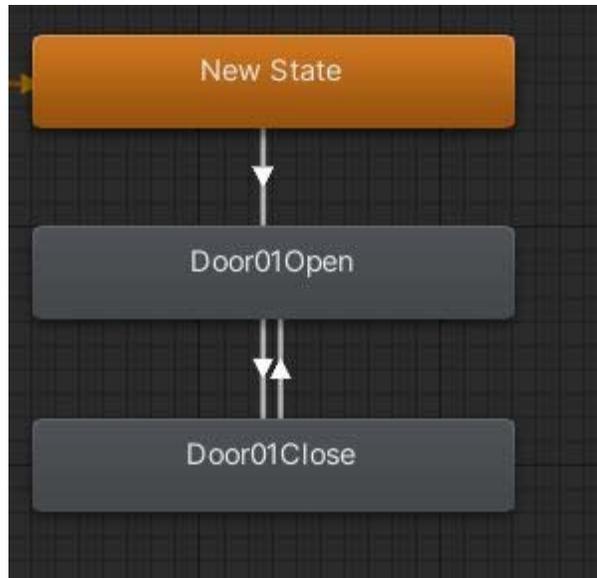


Figure 6

Sensor:

Timer:

The first sensor is the sensor. When the user press 'K' on the keyboard, the event will be triggered. The time will cut down. I use a 'timer', let 'timer -= Time.deltaTime'. We can see figure 7 is a part of code how to do the cutdown timer.

```
if (isEventStart && timer>0 && isFnihed == false)
{
    timer -= Time.deltaTime;
    currentTimerText.text = "CountDown:" + ((int)timer).ToString()+"S";
}
```

Figure 7

Enemy Collider:

I put a collider on the enemy object, it is used to detect the users. From figure 8 we can see that , if the user is in the detect range, the enemy will shoot the user.



Figure 8

Door sensor:

I add a trigger for the doors. It also a box collider. When the user is in the range, it will open the door and close the door when they are out of the range. It is shown in figure 9.

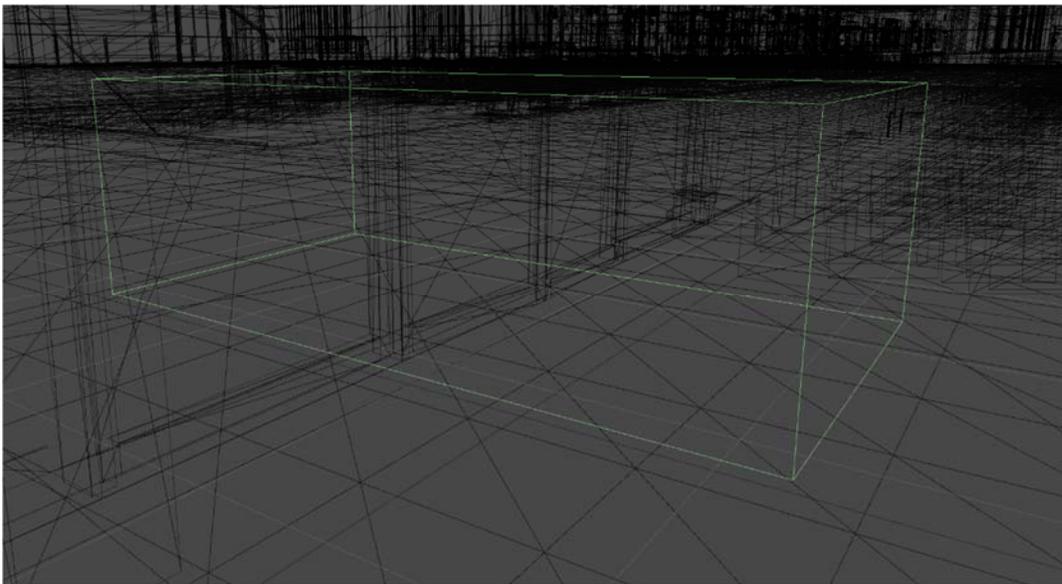


Figure 9

I also add trigger for other avatars in the game, it is used to detect if the user is in the detect range and then they can have a conversation. I use capsule collider and sphere for the teacher and student. I use Sphere Collider for the police. It is shown in figure 10.

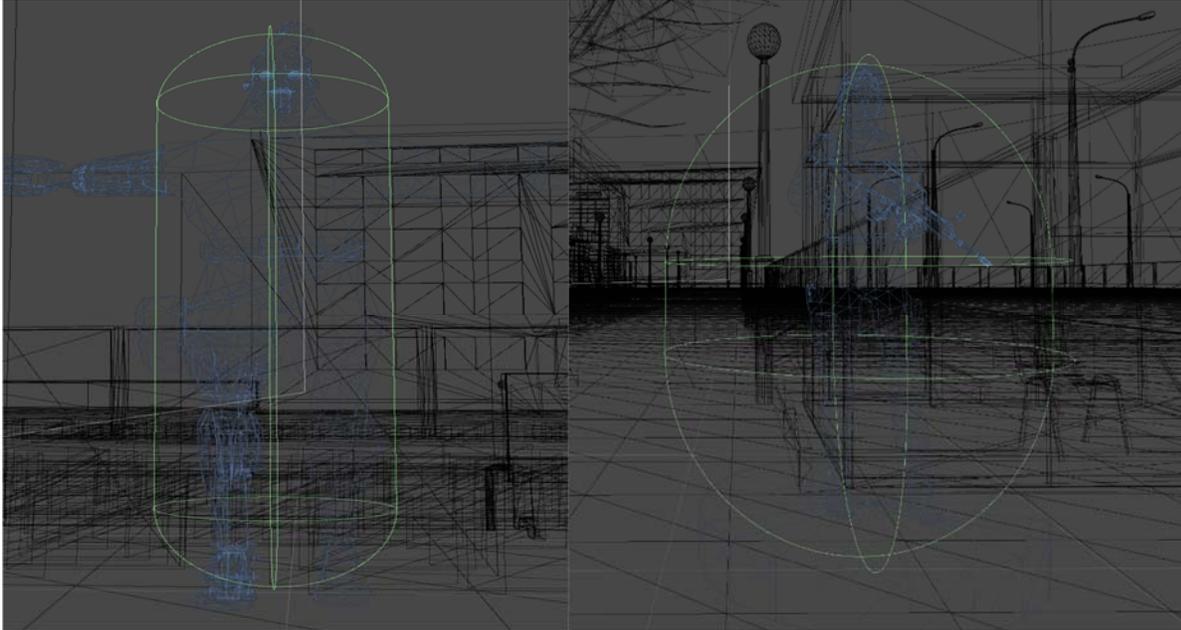


Figure 10 (a) Capsule Collider

Figure 10(b) Sphere Collider

For the car, I also use collider to detect if any users are near the car, it detects the user, the user can use 'K' to drive the car. I add a box collider for the car. It is shown in the figure 11.

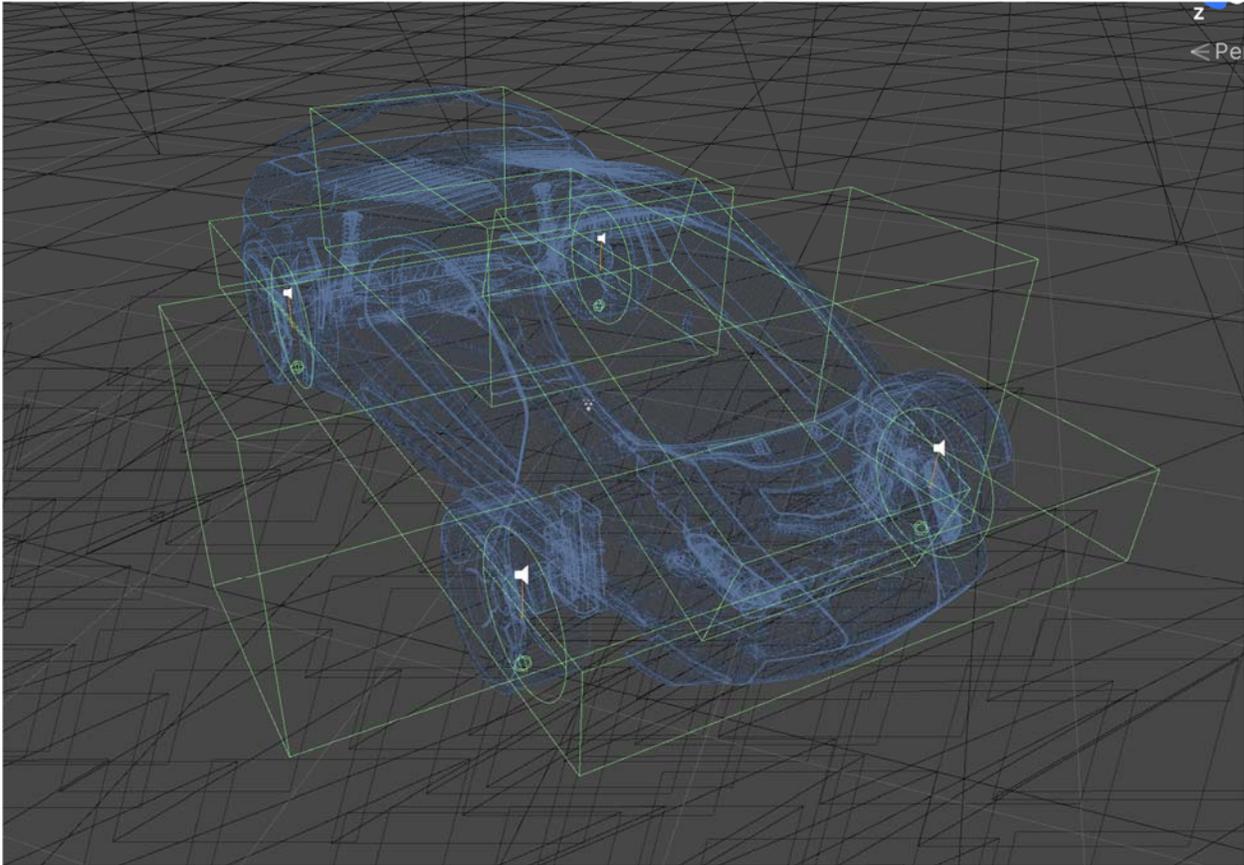


Figure 11

User Interactive:

- 1) Talk with NPC in the game. The user can talk with the student, police and teacher in the game. In this game I use collider to detect if the user is near the NPC, if detect the user, and it will notice the user 'Use K to talk with the NPC'.
- 2) Open the door. The second interactive is to open the door. I set up a trigger to detect the user. If detect the user it will notice the user" Press F to open the door".
- 3) Drive the car is the third interactive. I also use a trigger for the car module. If detect the user, it will notice "press k to drive the car". When the user is in the car, it will disable the camera and use the carcamera to follow the car.
- 4) Trigger the event. In this game when you press K the event will be triggered.

AI:

- 1) When the event is triggered, all the NPC will go to the gym automatically. The Police will stand at the door to protect the people.
- 2) The enemy can detect the user and attack the user automatic.

Multi User:

For this game I use photon unity network (PUN) to realize multiuser environment. The reason I choose Photon was that I do not need to build my own server. I can use the PUN server. I only need to register an account and got the ID from PUN then I can use it. One problem for PUN is that it does not support Unity 2020 version.

From figure 12, we can see that, we first need to connect the server. You should have the AppId you apply from the PUN website, if you do not have that you can not connect the server.

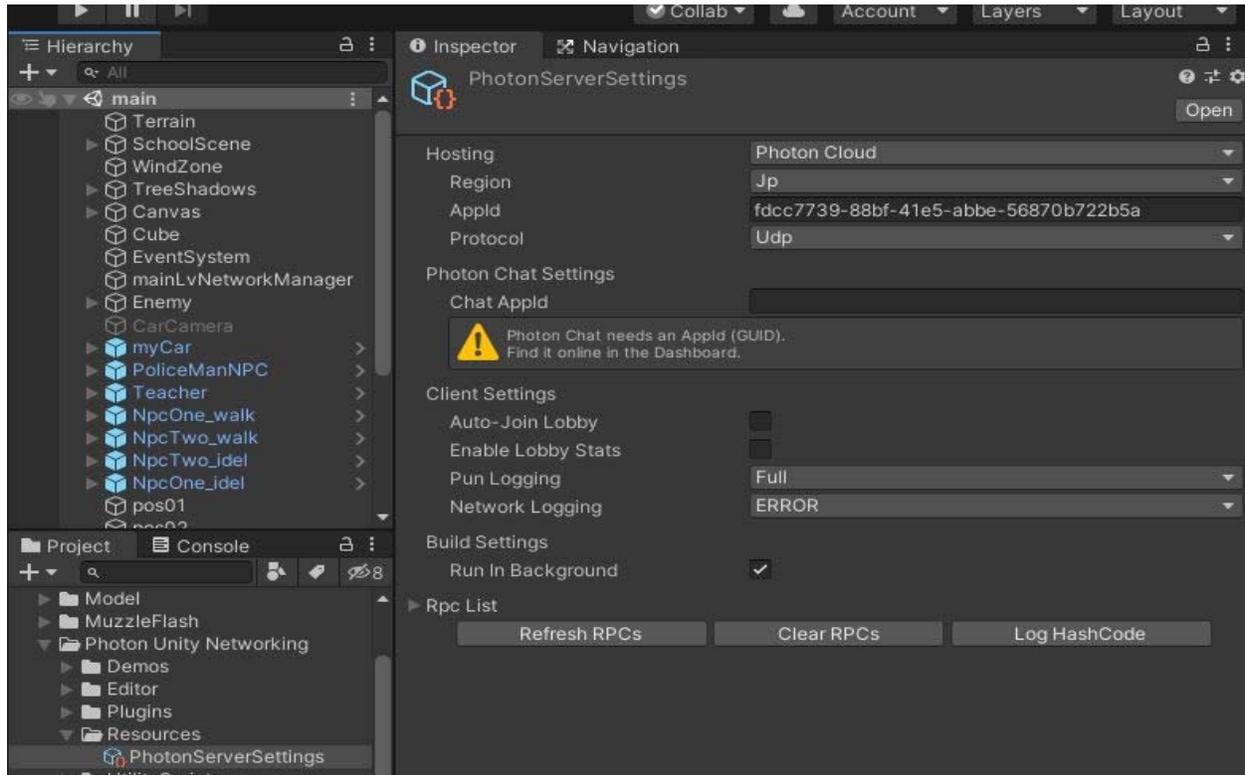


Figure 12

After you connect the server, you can import the Photon library. You can use 'using Photon' to import the library in your code. (See figure 13)

```
using System.Collections.Generic;
using UnityEngine;
using Photon;
using UnityEngine.UI;
```

Figure 13

Then you need to test if you connect the server successfully. You can use the code in figure 14 to connect the server. If you succeed you will see “connected to masterserver” (shown in figure 15).

```
void Start()
{
    PhotonNetwork.ConnectUsingSettings("0.0.1");
}
```

Figure 14

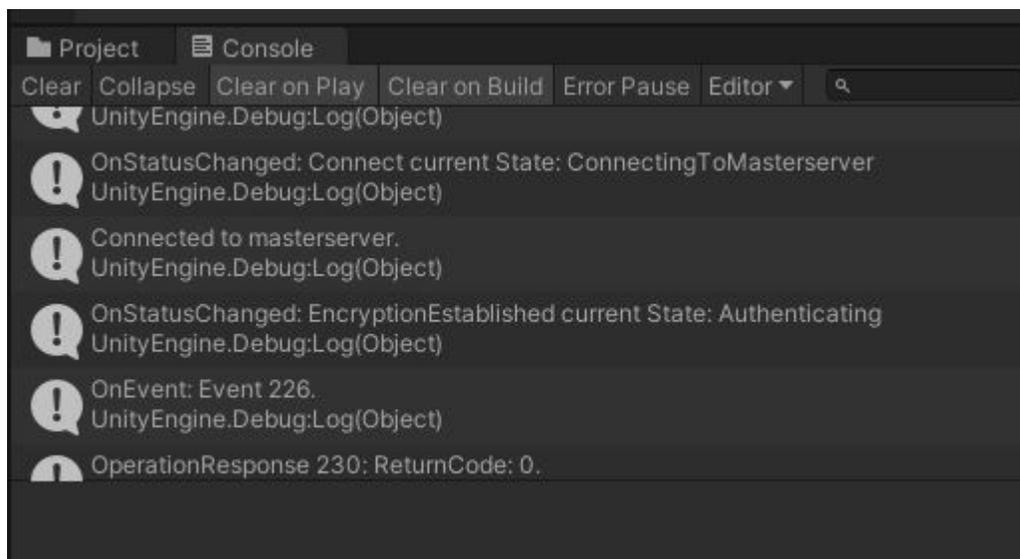


Figure 15

Synchronize in the game:

After you configure the server. You can start to add more details in your game. The most important things for multi user game is the synchronize. For PUN it use component 'PhotonView' to do this function. It is shown in figure 16. From figure 16 we can know that each object in the game has different ViewID. The ID is not enough we should add component 'Photon TransformView'(shown in figure 17). We need to synchronize the animation so we also need add the component 'PhotonAnimatorView'(show in figure 18). We put all these components to the Photon View. The position and animation can synchronize(figure 16).



Figure 16

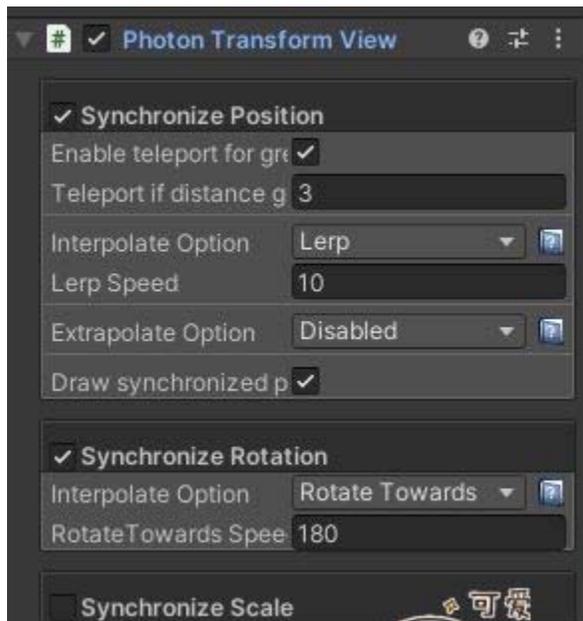


Figure 17

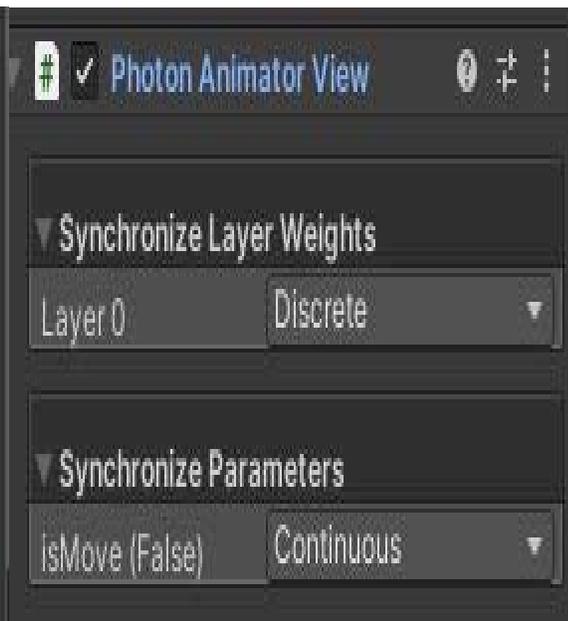


Figure 18

How the server sent information to the client? We can add [PunRPC] before the functions. It is shown in figure 19. We let the function become an RPC functions. For the code in figure 19, when we press k in the game the event will be triggered. All the enemy will appear, and the alert will rise, the cut down timer will appear. We should take care of the calling function method in this module. We should use the code in figure 20 to call the function.

```
[PunRPC]
public void OnEventStart ()
{
    isEventStart = true;
    tipText.text = "The enemy appeared! ";
    dangerAudio.Play ();

    enemmyGo01.SetActive (true);
    enemmyGo02.SetActive (true);
    enemmyGo03.SetActive (true);
    enemmyGo04.SetActive (true);
}
```

Figure 19

```
PhotonView pv = GetComponent<PhotonView> ();
pv.RPC ("OnEventStart", PhotonTargets.AllBuffered);
```

Figure 20

Game Play:

When you start the game, you will see the home page first. The home page is shown in figure 21 In the middle you can select the avatars you want to choose. On the top right you will see how many players are in the game. After you select the avatars you can start the game by click the match button on the bottom of the avatars.

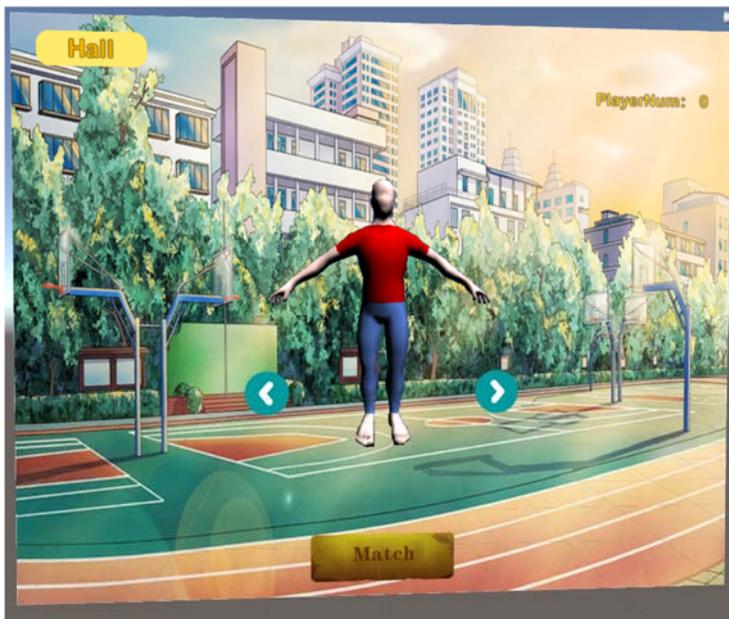


Figure 21

In this game you have three choice, teacher(figure 22(c)), student(figure 22(b)) and police(figure 22(a)). You can use left or right button to change the avatars. After you select the avatars you can start the game.

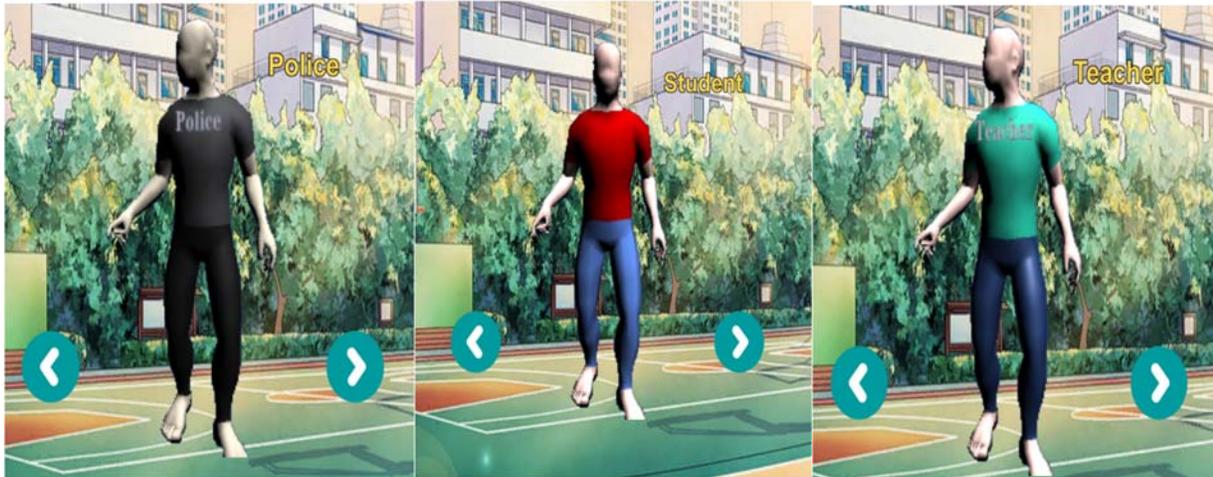


Figure 22(a)

Figure 22(b)

Figure 22(c)

In the game you can talk with the NPCs by press 'F' on you keyboard(see figure 23).



Figure 23

You can also drive the car in the game. From figure 24, you can see that when you near the car, press F to use the car.

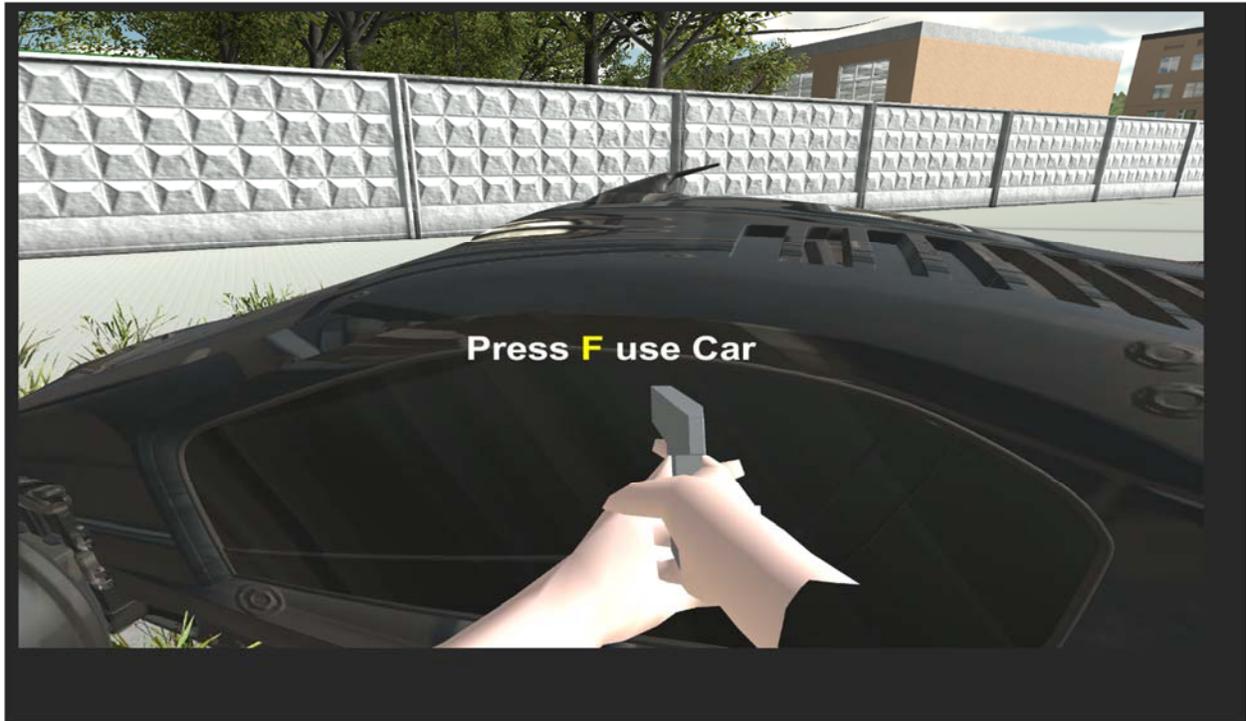


Figure 24

When you want to go into the building, you can use F to open the door. (see figure 25)

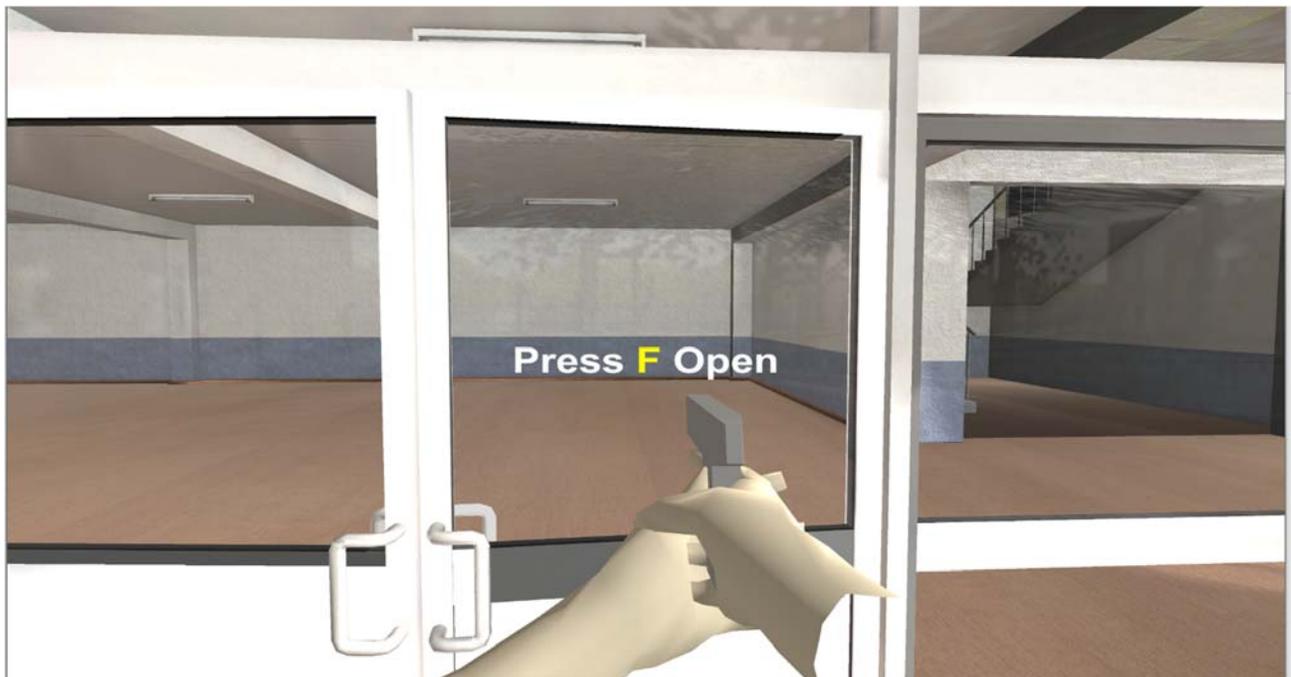


Figure 25

In the game, you can use 'K' on the keyboard to trigger the event. When the event is triggered, the enemy will appear and a timer will appear also(see figure 26). You have 60 seconds to finish the task. When you destroy all four enemies or the timer become 0, the task will end(figure 27).



Figure 26



Figure 27

Conclusion and future Works:

For this game, I was to do a campus active shoot response. The users are in the campus and when the event was triggered, all the avatars in the game will go to a safe space(the gym) to take refuge. The police will stand by the door to protect the people in the gym. The user will against with the enemy in the campus to find them and destroy them. I use Photon component to realize the multi-user environment.

In the future, I will add more avatars in the game and find more safe place for people to take refuge when the event is triggered, Currently the game can maxim has 6 players at same time. I will make it have more players in the future.