Project Team – 03

ShootAR Game

Code Documentation
Milestone - 2

Object:

Document Title
Code Documentation (Milestone-2)

Submitted To:

Project Director
Ms. Simranpreet Kaur

Project Customer
Ms. Shivani Kapadia

Submitted By:

Team Members
Kameswara Saidatta Srinivas Peddada - 110081329 Kartik Peddinti – 110086484 Keneel chirag shah – 110073464 Krishna Sravanthi Telapudi - 1100727725 Mrinal Walia - 110066886 Varnita Sharma - 110074272 V V R Sricharan Apparayacheruvu – 110086372 Venkata Sai Vardhan Seepala - 110074717

Date : **20-March-2022**

Table of Contents

Code Documentation

Website link: https://blog.srpeddada.com/shootar

GitHub link: https://github.com/abhiwalia15/ShootAR

Demo Video link: https://www.youtube.com/watch?v=N9PmWcF_jSE

(This code documentation is for milestone -2. All the new features added will be represented with Asterisk on the heading)*

1. ShootAR Explained

In this article, we will be explaining the code base and features of the ShootAR game.

Our shootAR game is built with:

1. Unity

2. C#

The above are the two technologies used. However, we will display C# code snippets and explain them below.

2. Rendering Camera for AR feature

We will be using the camera, as our game is augmented reality, and we will be using the below C# script to render the 3d objects into the real world.
In the below code, we have three functions:

start()
This function executes for the first time when the class is loaded. So, we will add the core functionality for rending the camera in this start method.
GameObject cameraParent = new GameObject ("camParent");

Fire()
In this function, we have written the code for bullet firing. When the shoot button is clicked, we will call this function.

update()
This function is called once per frame. So, we will write the code which requires

continuous updating—for instance, changing the level or scene when the requirement is fulfilled.

Asset/webCamScript.cs

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

        public class CameraScript : MonoBehaviour
        { // Start is called before the first frame update
            public GameObject webCameraPlane;
           public Button fireButton;

            void Start()
            {

         if (Application.isMobilePlatform) {
              GameObject cameraParent = new GameObject ("camParent");
              cameraParent.transform.position = this.transform.position;
              this.transform.parent = cameraParent.transform;
              cameraParent.transform.Rotate (Vector3.right, 90);
            }

        //Input.gyro.enabled =
            Input.gyro.enabled = true;

          fireButton.onClick.AddListener (fire);


                WebCamTexture webCameraTexture = new WebCamTexture();
                webCameraPlane.GetComponent().material.mainTexture =
webCameraTexture;
                webCameraTexture.Play();
            }


          void fire(){

            GameObject bullet = Instantiate(Resources.Load("bullet",
typeof(GameObject))) as GameObject;
            Rigidbody rb = bullet.GetComponent();
            bullet.transform.rotation = Camera.main.transform.rotation;
            bullet.transform.position = Camera.main.transform.position;
            rb.AddForce(Camera.main.transform.forward * 500f);
            Destroy (bullet, 3);

            GetComponent().Play ();
          }
```

```
            // Update is called once per frame
            void Update()
            {
                 Quaternion cameraRotation = new Quaternion
(Input.gyro.attitude.x, Input.gyro.attitude.y, -Input.gyro.attitude.z, -
Input.gyro.attitude.w);
            this.transform.localRotation = cameraRotation;

        // this is the code activates when the play button is hit
            if (GameObject.FindGameObjectsWithTag("GameController").Length ==
0){
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);

            }

            }
        }
```

3. Collision Script

We have to blast the spaceship when the bullet hits the enemy models. So, we gave implemented one function in the below code.

OnTriggerEnter()
In this function, we have implemented the collision feature where the 3d enemy models will be busted when in contact with the bullet.

Assets/collisionScript.cs

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

        public class collisionScript : MonoBehaviour {

          // Use this for initialization
          void Start () {

          }

          // Update is called once per frame
          void Update () {


          }

          // This is the code for collision
          void OnTriggerEnter (Collider col)
          {
            GameObject explosion = Instantiate(Resources.Load("FlareMobile",
typeof(GameObject))) as GameObject;
            explosion.transform.position = transform.position;
            Destroy(col.gameObject);
            Destroy (explosion, 2);


            // destroying the game object
            Destroy (gameObject);


          }

        }
```

4. Movement of Enemies

We also have to move the enemies to move in the 3d space around the player. So we have implemented the **move()** function in the below code. In this function, we will be giving vector values with which the enemy models will be moving in the 3d space.

Assets/enemyScript.cs

```
using UnityEngine;
       using System.Collections;

       public class enemyScript : MonoBehaviour {



         // Use this for initialization
         void Start () {

           StartCoroutine ("Move");
         }

         // Update is called once per frame
         void Update () {

           transform.Translate(Vector3.forward * 3f * Time.deltaTime);
         }

         IEnumerator Move() {
           while (true) {
             yield return new WaitForSeconds (3.5f);
             transform.eulerAngles += new Vector3 (0, 180f, 0);
           }
         }
       }
```

5. Countdown Timer

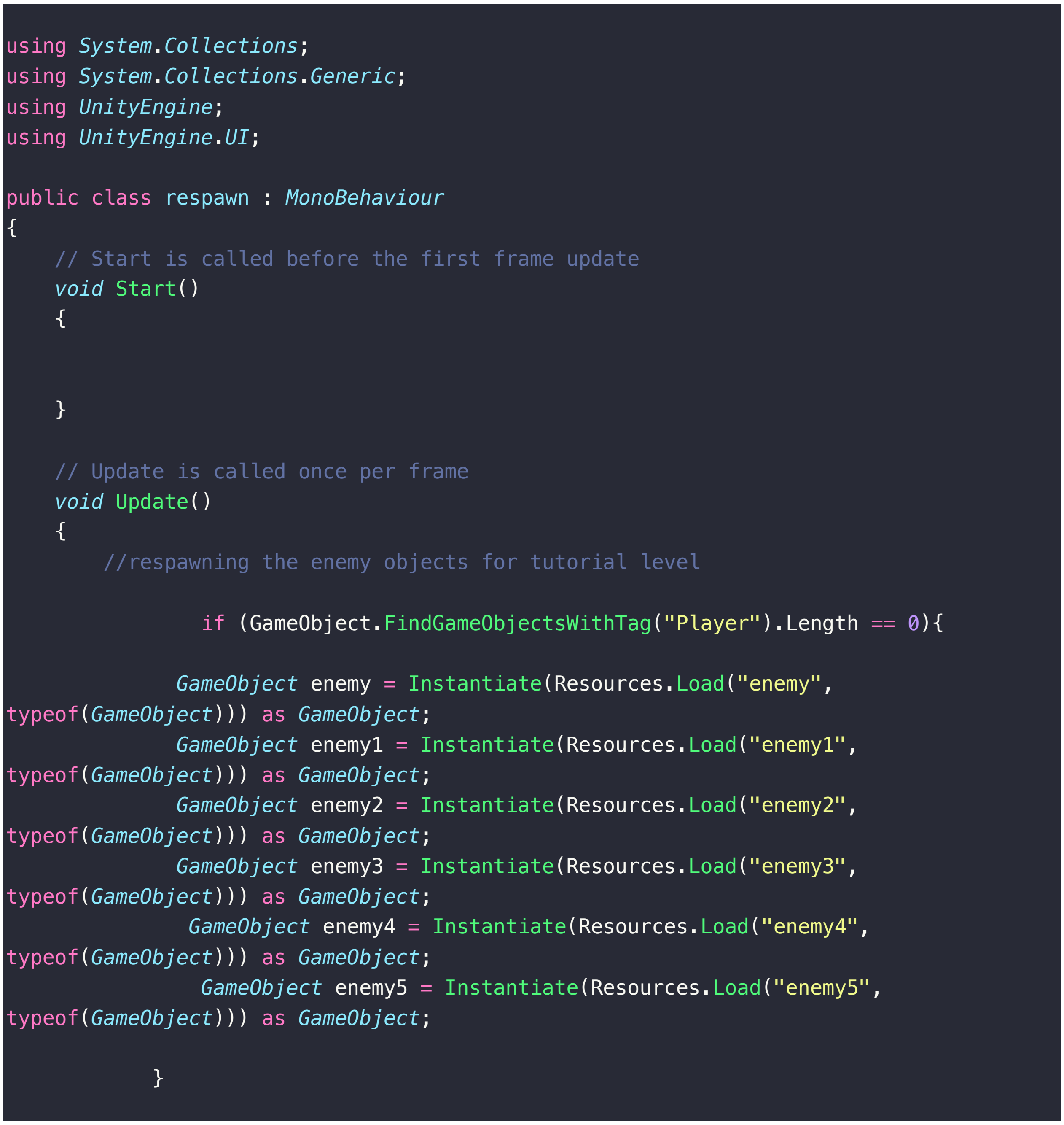
We have also partially implemented the count down timer for the warm-up scene (i.e. level 1 in the game) we will be updating the code further once we finish the implementation.

Assets/countdownTimer.cs

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

        public class countdownTimer : MonoBehaviour
        {
            public float Timer = 10f;
            public Text timerSeconds;
            bool timerActive = true;
            // Start is called before the first frame update
            void Start()
            {
                timerSeconds = GetComponent();

            }

            // Update is called once per frame
            void Update()
            {
                if(timerActive == true){
        Timer -= Time.deltaTime;

        if(Timer <= 0){
                    timerActive = false;
                    Timer = 0f;
                    Debug.Log("Timer finished");
                }
                }

                timerSeconds.text = Timer.ToString("f2");
```

```
            }
        }
```

6. **Respawn* :**
   We have implemented respawning function where all the enemy models re generate in the tutorial level.



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

public class respawn : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {


    }

    // Update is called once per frame
    void Update()
    {
        //respawning the enemy objects for tutorial level

                if (GameObject.FindGameObjectsWithTag("Player").Length == 0){

              GameObject enemy = Instantiate(Resources.Load("enemy",
typeof(GameObject))) as GameObject;
              GameObject enemy1 = Instantiate(Resources.Load("enemy1",
typeof(GameObject))) as GameObject;
              GameObject enemy2 = Instantiate(Resources.Load("enemy2",
typeof(GameObject))) as GameObject;
              GameObject enemy3 = Instantiate(Resources.Load("enemy3",
typeof(GameObject))) as GameObject;
               GameObject enemy4 = Instantiate(Resources.Load("enemy4",
typeof(GameObject))) as GameObject;
                GameObject enemy5 = Instantiate(Resources.Load("enemy5",
typeof(GameObject))) as GameObject;

            }
    }
}
```

7. **CameraScript.cs* :**

   This contains the code for changing one scene to another. This can be used to promote user from one level to another.

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

public class CameraScript : MonoBehaviour
{ // Start is called before the first frame update
    void Start()
    {

    }
/ Update is called once per frame
    void Update()
    {
// This code is to move from one scene to another
    if (GameObject.FindGameObjectsWithTag("GameController").Length == 0){
SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);

    }

    }
}
```