

Practical No. 8

Aim : Develop a 3D-Space-Shooter Game in Unity Game Engine

<https://unity3d.com/learn/tutorials/s/space-shooter-tutorial>

code

1. Player-Controller.cs file

```
1  using UnityEngine;
2  using System.Collections;
3
4  [System.Serializable]
5  public class Boundary
6  {
7      public float xMin, xMax, zMin, zMax;
8  }
9  public class PlayerController : MonoBehaviour
10 {
11
12     public float speed;
13     public float tilt;
14     public float firerate = 0.5f;
15     public GameObject shot;
16     public Transform shotspawn;
17     private float mytime = 0.0f;
18     private float nextfire = 0.5f;
19     private Rigidbody rb;
20     public Boundary boundary;
21     private Touch theTouch;
22     private Vector2 touchStartPosition, touchEndPosition;
23     private string direction;
24     private AudioSource audio;
25     void Start()
26     {
27         rb = GetComponent<Rigidbody>();
28         audio = GetComponent<AudioSource>();
29     }
30     void Update()
31     {
```

```

32     mytime = mytime + Time.deltaTime;
33     //For Mobile Based Touch controls
34     Gametouchfire();
35     // For Desktop/Web Keyboard based controls.
36     //if ((Input.GetButton ("Fire1")) && mytime > nextfire) {
37     //    nextfire = mytime + firerate;
38     //    Instantiate (shot, shotspawn.position,
39     //    shotspawn.rotation);
40     //    audio.Play ();
41     //    nextfire = nextfire - mytime;
42     //    mytime = 0.0f;
43     //}
44 }
45 void Gametouchfire()
46 {
47     if (Input.touchCount > 0)
48     {
49         theTouch = Input.GetTouch(0);
50
51         if (theTouch.phase == TouchPhase.Began)
52         {
53             touchStartPosition = theTouch.position;
54         }
55
56         else if (theTouch.phase == TouchPhase.Moved || theTouch.phase
57         == TouchPhase.Ended)
58         {
59             touchEndPosition = theTouch.position;
60
61             float x = touchEndPosition.x - touchStartPosition.x;
62             float y = touchEndPosition.y - touchStartPosition.y;
63
64             if ((Mathf.Abs(x) == 0 && Mathf.Abs(y) == 0) && mytime >
65             nextfire)
66             {
67                 nextfire = mytime + firerate;
68                 Instantiate(shot, shotspawn.position,
69                 shotspawn.rotation);
70                 audio.Play();
71             }
72         }
73     }
74 }
```

```

68             nextfire = nextfire - mytime;
69             mytime = 0.0f;
70         }
71     }
72 }
73 }
74 void FixedUpdate()
75 {
76     //float moveHorizontal = Input.GetAxis ("Horizontal");
77     //float moveVertical = Input.GetAxis ("Vertical");
78     float moveHorizontal = Input.acceleration.x;
79     float moveVertical = 0.0f;
80     Vector3 movement = new Vector3(moveHorizontal, 0.0f,
81     ↘moveVertical);
82     rb.velocity = movement * speed;
83     rb.position = new Vector3
84     (
85         Mathf.Clamp(rb.position.x, boundary.xMin, boundary.xMax),
86         0.0f,
87         Mathf.Clamp(rb.position.z, boundary.zMin, boundary.zMax)
88     );
89
90     rb.rotation = Quaternion.Euler(0.0f, 0.0f, rb.velocity.x *
91     ↘-tilt);
92 }
```

2. Destroy-By-Boundary.cs file

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;

4
5 public class DBBoundary : MonoBehaviour {

6
7     void OnTriggerExit(Collider other)
8     {
9         Destroy(other.gameObject);
10    }
```

11 }

3. Destroy-By-Contact.cs file

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;

4

5  public class DestroyByContact : MonoBehaviour {

6

7      public GameObject Explosion;
8      public GameObject PExplosion;
9      private GameController gamecontroller;
10     public int scoreval;
11     void Start(){
12         GameObject gc = GameObject.FindGameObjectWithTag("GameController");
13         if (gc != null) {
14             gamecontroller = gc.GetComponent<GameController> ();
15         }
16         if (gamecontroller == null) {
17             Debug.Log ("Game Controller not found");
18         }
19
20     }
21     void OnTriggerEnter(Collider other)
22     {
23         if (other.tag == "Boundary")
24         {
25             return;
26         }
27         Instantiate (Explosion,transform.position,transform.rotation);
28         if (other.tag == "Player") {
29             Instantiate (PExplosion,other.transform.position,other.transf ↴
29             orm.rotation);
30             gamecontroller.GameOvers ();
31         }
32         gamecontroller.addScore (scoreval);
33         Destroy(other.gameObject);
34         Destroy(gameObject);
35     }
}
```

36 }

4. Destroy-By-Time.cs file

```
1 using UnityEngine;
2 using System.Collections;
3
4 public class DestroyByTime : MonoBehaviour
5 {
6     public float lifetime;
7     void Start(){
8         Destroy (gameObject, lifetime);
9     }
10 }
```

5. Mover.cs file

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Mover : MonoBehaviour {
6
7     public float speed;
8     private Rigidbody rb;
9
10    void Start () {
11        rb = GetComponent<Rigidbody> ();
12        rb.velocity = transform.forward * speed;
13    }
14
15 }
16 }
```

6. Game-Controller.cs file

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
```

```
5
6 public class GameController : MonoBehaviour {
7     public GameObject hazard;
8     public Vector3 spawnvalue;
9     public int hazardcount;
10    public float spawnwait;
11    public float startwait;
12    public float wavewait;
13    public Text score;
14    public Text restart;
15    public Text GameOver;
16    private bool go;
17    private bool rs;
18    private int scr;
19    public Button rst;
20    private Touch theTouch;
21    private Vector2 touchStartPosition, touchEndPosition;
22    private string direction;
23
24    void Start(){
25        go = false;
26        rs = false;
27        restart.text = "";
28        GameOver.text = "";
29        scr = 0;
30        UpdateScore ();
31        StartCoroutine(SpawnWaves ());
32
33    }
34
35    IEnumerator SpawnWaves(){
36        yield return new WaitForSeconds (startwait);
37        while(!go){
38            for (int i=0;i<hazardcount;i++) {
39                Vector3 spawnPosition = new Vector3 (Random.Range
40                    (-spawnvalue.x, spawnvalue.x), spawnvalue.y,
41                    spawnvalue.z);
42                Quaternion srotation = Quaternion.identity;
43                Instantiate (hazard, spawnPosition, srotation);
44                yield return new WaitForSeconds (spawnwait);
45            }
46        }
47    }
48}
```

```
43     }
44     yield return new WaitForSeconds (wavewait);
45     if (go) {
46         restart.text = "Restart";
47         rs = true;
48         break;
49     }
50 }
51 }

52 void Update(){
53     if (rs) {
54         if (Input.touchCount > 0){
55             theTouch = Input.GetTouch(0);
56             if (theTouch.phase == TouchPhase.Began){
57                 touchStartPosition = theTouch.position;
58             }
59             else if (theTouch.phase == TouchPhase.Moved || theTouch.phase
59             == TouchPhase.Ended){
60                 touchEndPosition = theTouch.position;
61                 float x = touchEndPosition.x - touchStartPosition.x;
62                 float y = touchEndPosition.y - touchStartPosition.y;
63                 if (Mathf.Abs(x) == 0 && Mathf.Abs(y) == 0){
64                     Application.LoadLevel(Application.loadedLevel);
65                 }
66             }
67         }
68     }
69 }
70 }

71 public void addScore(int newscr){
72     scr += newscr;
73     UpdateScore ();
74 }
75 void UpdateScore(){
76     score.text = "Score : " + scr.ToString ();
77 }
78 public void GameOvers(){
79     GameOver.text = "Game Over";
80     go = true;
81 }
```

82 }

7. Random-Rotator.cs file

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;

4

5  public class RandomRotator : MonoBehaviour {

6

7      public float tumble;
8      private Rigidbody rb;
9      void Start ()
10     {
11         rb = GetComponent<Rigidbody> ();
12         rb.angularVelocity = Random.insideUnitSphere * tumble;
13     }
14 }
```

Output

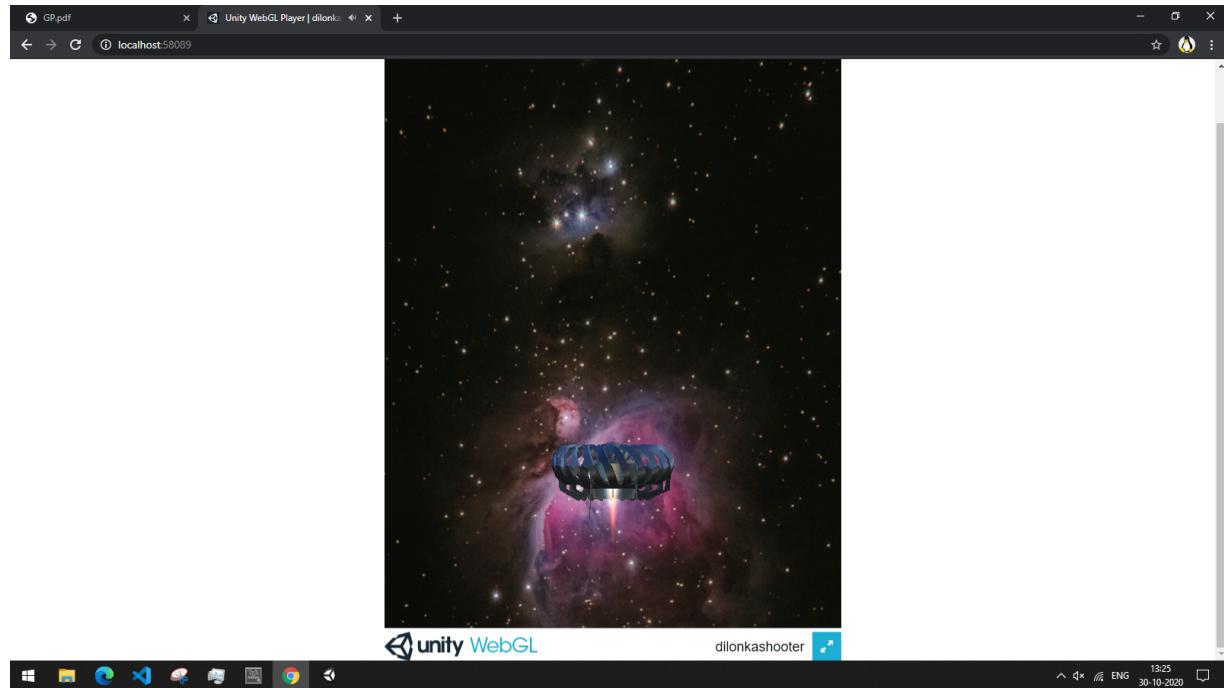


Figure 1: Space-Shooter On WebGL

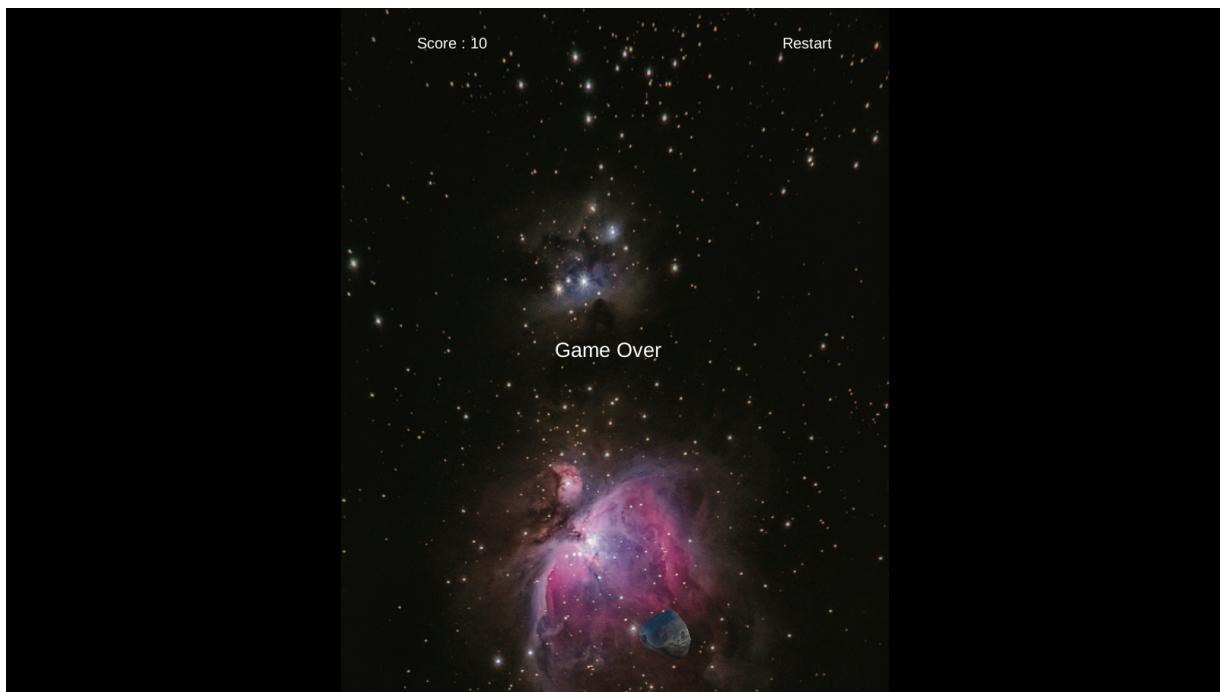


Figure 2: Space-Shooter On WebGL Full Screen



Figure 3: Space-Shooter On Android