

더 데스 레이스

THE DEATH RACE

게임엔진 활용 (Unity 3D)
[팀 프로젝트 최종 보고서]

김규리

Contents

X

1. 게임 소개

- 게임 이름
- 게임 내용
- 게임 조작법

2. Menu Scene

- Main Menu
- Button

3. Game Scene

- Player
- Platform
- Level Generator
- Kill Zone
- Finish Line
- Game Over
- Score
- Message Output
- Audio Manager
- Sound

1

게임 소개

게임 소개 - 게임 제목

더 더스 레이스

THE DEATH RACE

게임 소개 - 게임 내용

모두가 좀비로 감염된 학교에서 홀로 감염되지 않은 한 학생이 탈출하는 게임



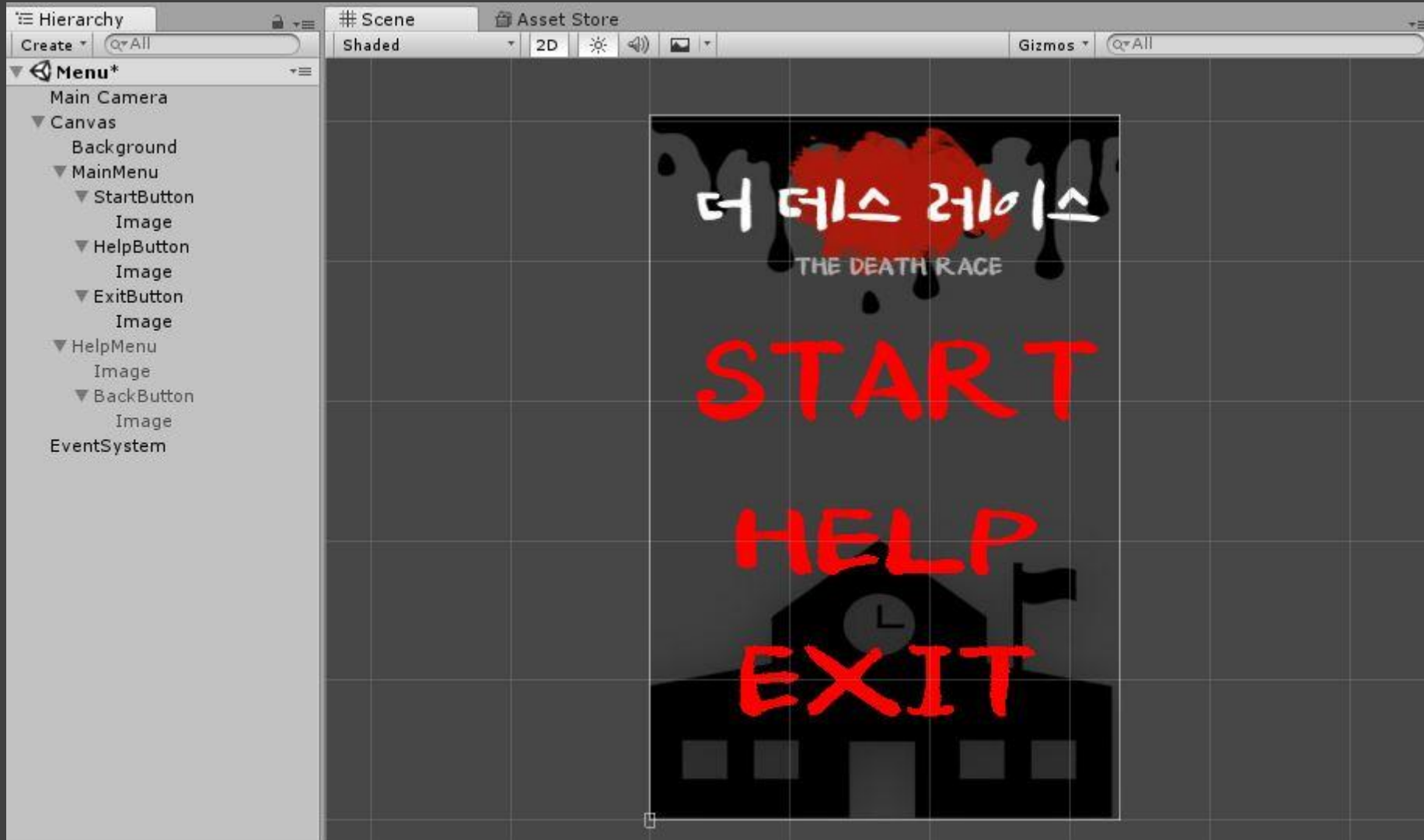
게임 소개 - 게임 조작법

주인공을 좌우방향키로 조종하여
아래에서 올라오는 좀비의 손길로부터 도망쳐
위로 내려가는 게임

2

Menu Scene

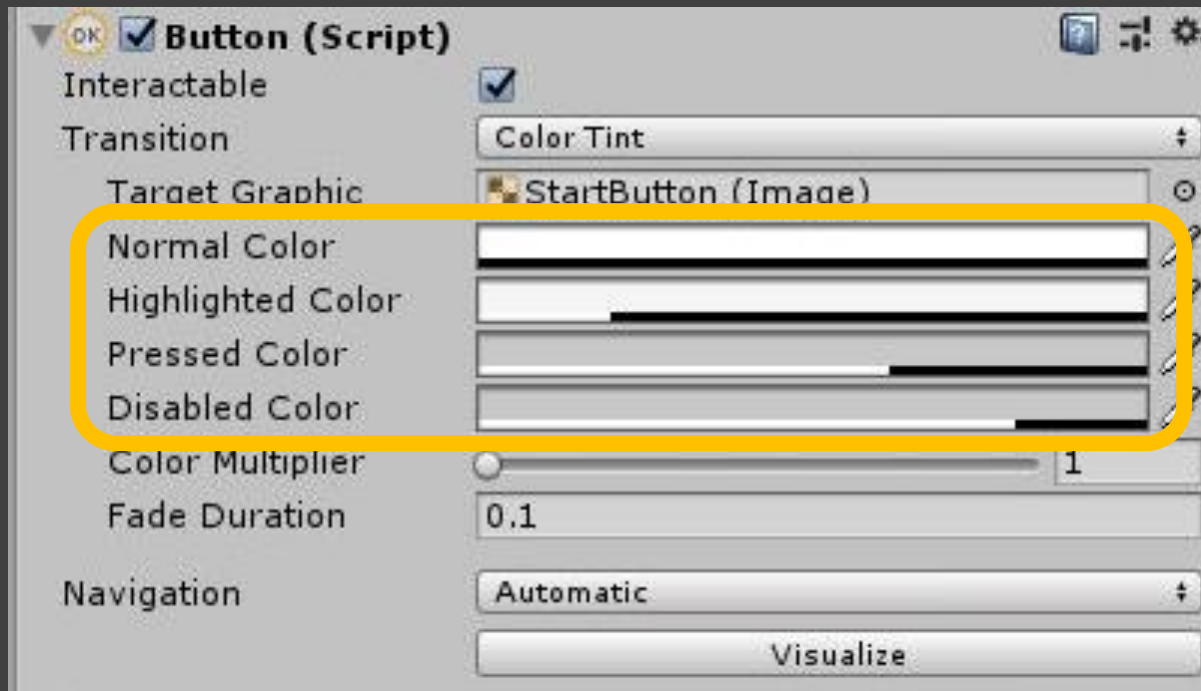
Menu Scene



Menu Scene – Main Menu

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5
6  public class MainMenu : MonoBehaviour {
7
8      public void PlayGame()
9      {
10         SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
11     }
12
13     public void QuitGame()
14     {
15
16         Debug.Log("QUIT!");
17         Application.Quit();
18     }
19
20
21
22 }
```

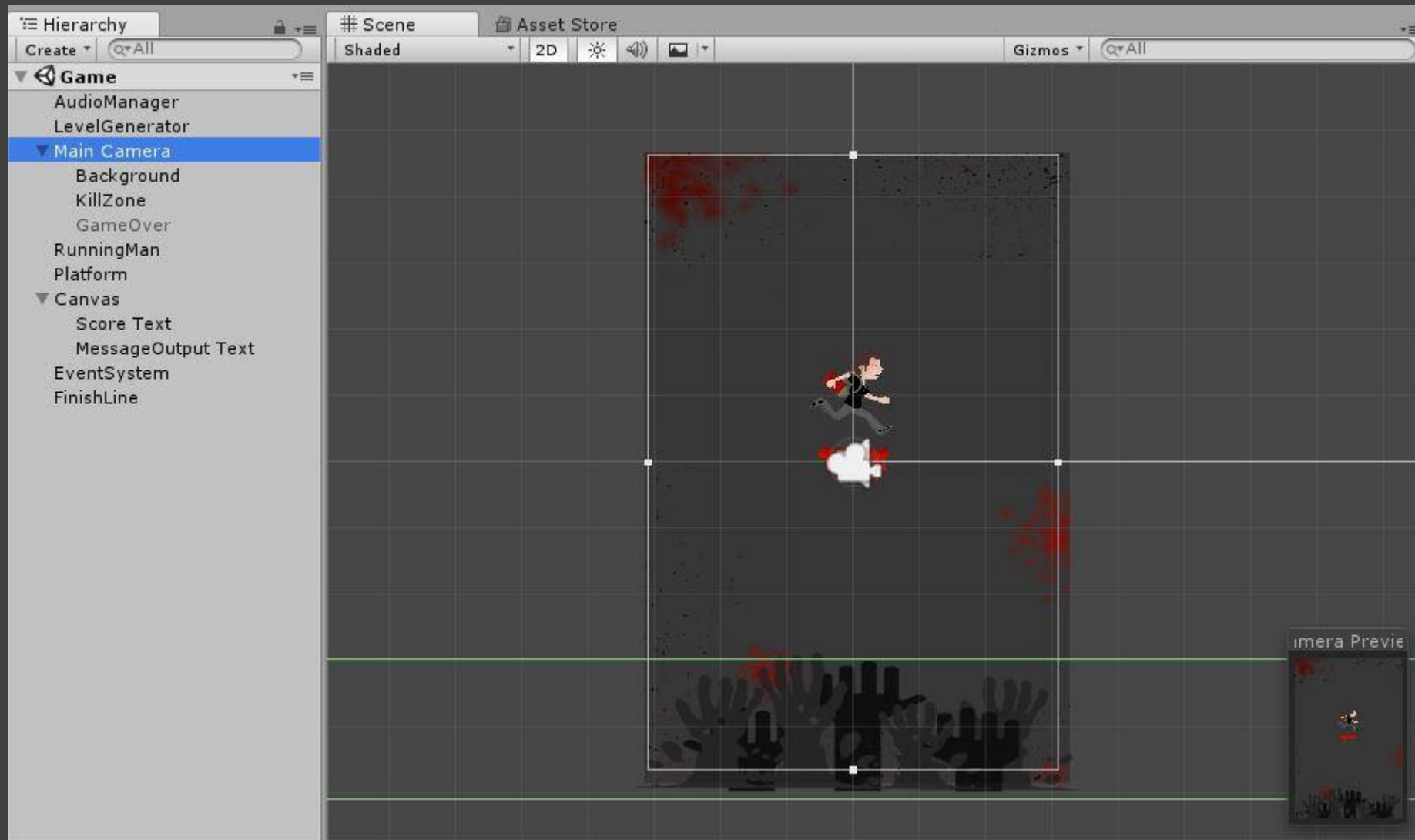
Menu Scene - Button 메뉴의 효과



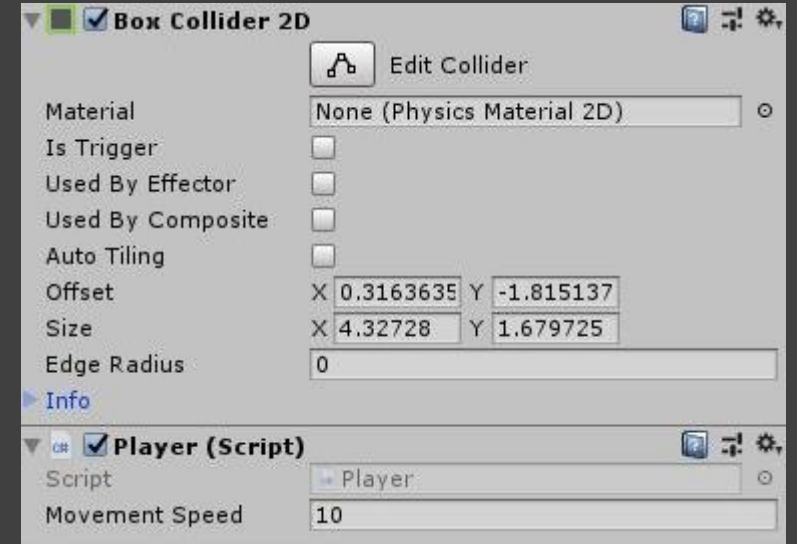
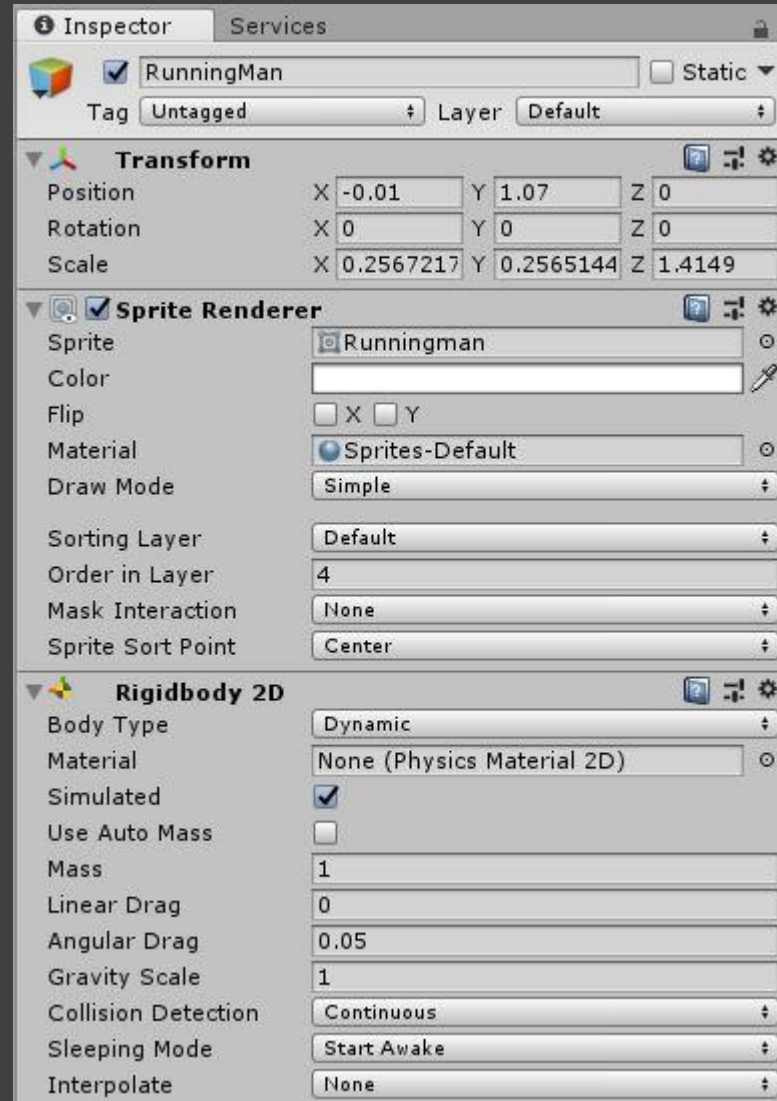
3

Game Scene

Game Scene



Game Scene – Player (Running Man)

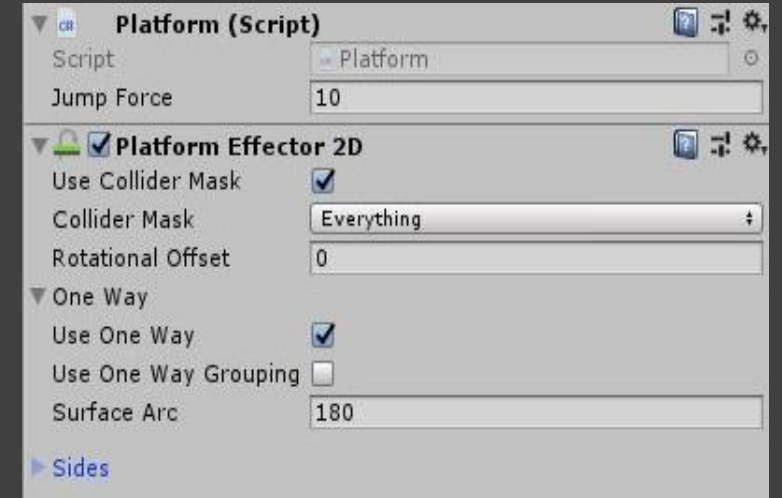
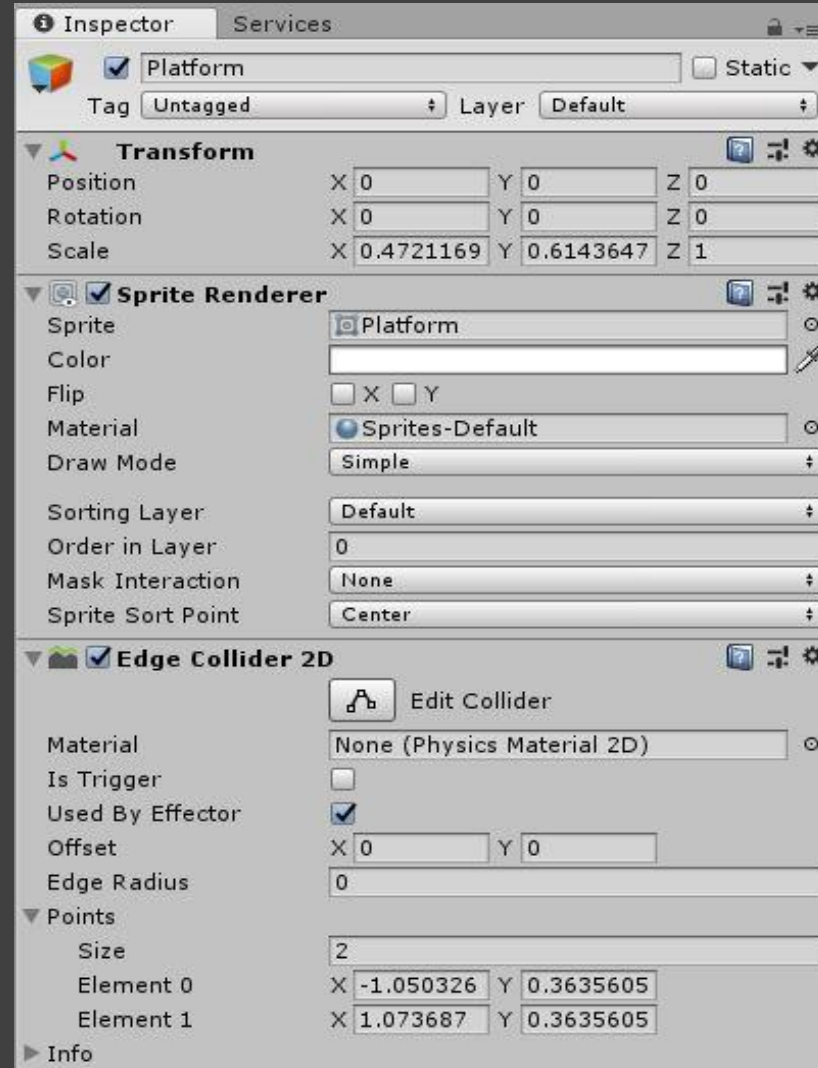


Game Scene - Player (Running Man)

```
71 private void MoveClamp()  
72 {  
73  
74     Vector2 left = Camera.main.ViewportToWorldPoint(new Vector2(0, 0));  
75  
76     Vector2 right = Camera.main.ViewportToWorldPoint(new Vector2(1, 1));  
77  
78     Vector2 player = transform.position;  
79  
80     player.x = Mathf.Clamp(player.x, left.x + 0.8F, right.x - 0.8F);  
81  
82     player.y = Mathf.Clamp(player.y, left.y + 1, right.y - 1);  
83  
84     transform.position = player;  
85 }
```

```
86 private void OnTriggerEnter2D(Collider2D collision)  
87 {  
88     if (GameManager.currentState.Equals(GameState.Gameover)) return;  
89     switch(collision.tag)  
90     {  
91         case "KillZone":  
92             GameManager.currentState = GameState.Gameover;  
93             SetGravityScale(0);  
94             break;  
95         case "FinalPlatform":  
96             GameManager.currentState = GameState.Clear;  
97             print("Clear Level.");  
98             break;  
99     }  
100 }  
101 }
```


Game Scene - Platform



Game Scene - Platform

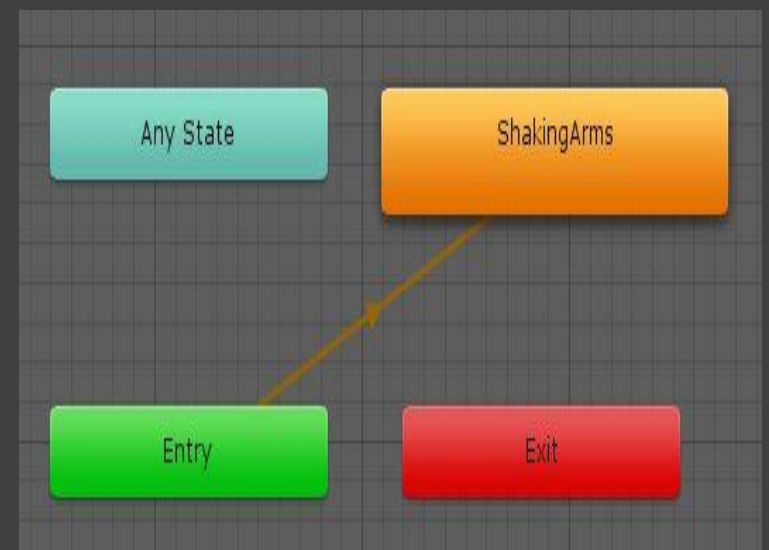
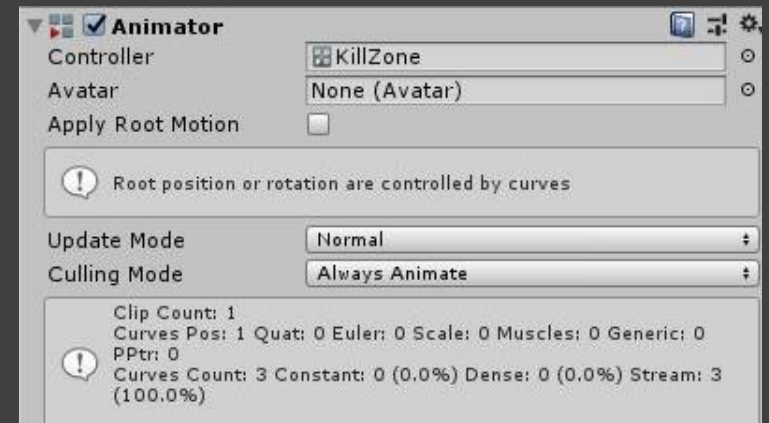
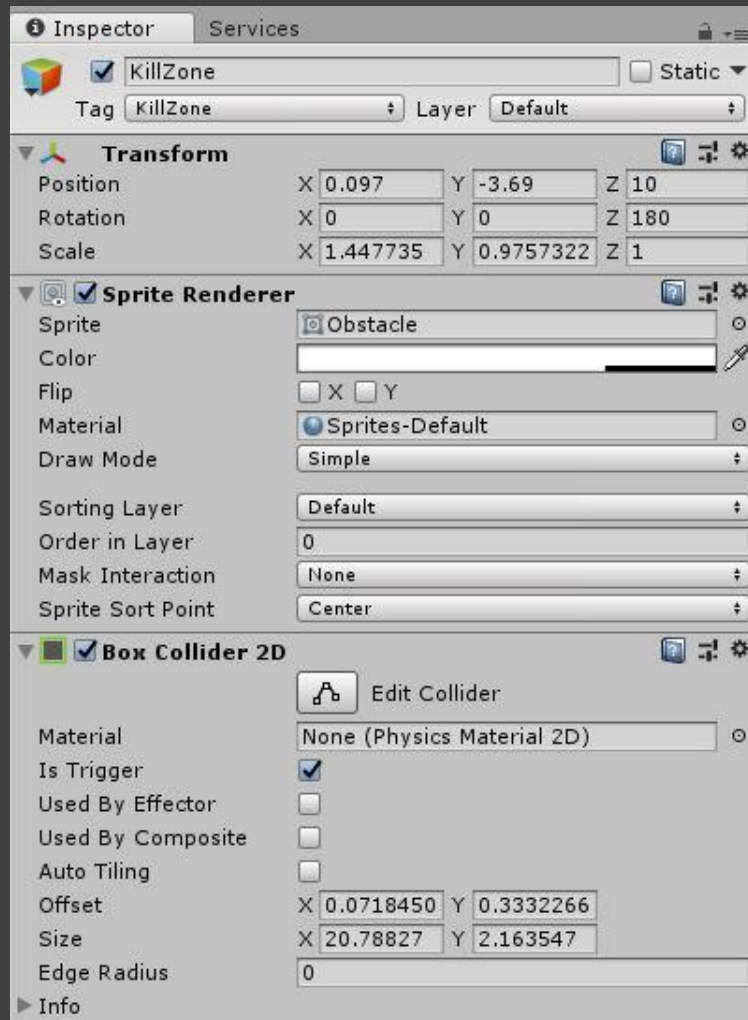
```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class Platform : MonoBehaviour
6  {
7      public float jumpForce = 5f;
8
9      public void SetActivation(bool activation)
10     {
11         gameObject.SetActive(activation);
12     }
13
14     public void SetPosition(Vector3 position)
15     {
16         transform.position = position;
17     }
18
19     void OnCollisionEnter2D(Collision2D collision)
20     {
21         if (GameManager.currentState.Equals(GameState.Gameover)) return;
22
23         if (collision.relativeVelocity.y <= 0f )
24         {
25             Rigidbody2D rb = collision.collider.GetComponent<Rigidbody2D>();
26
27             if (rb != null)
28             {
29                 Vector2 velocity = rb.velocity;
30                 velocity.y = jumpForce;
31                 rb.velocity = velocity;
32             }
33         }
34     }
35
36 }
```

Game Scene - Level Generator

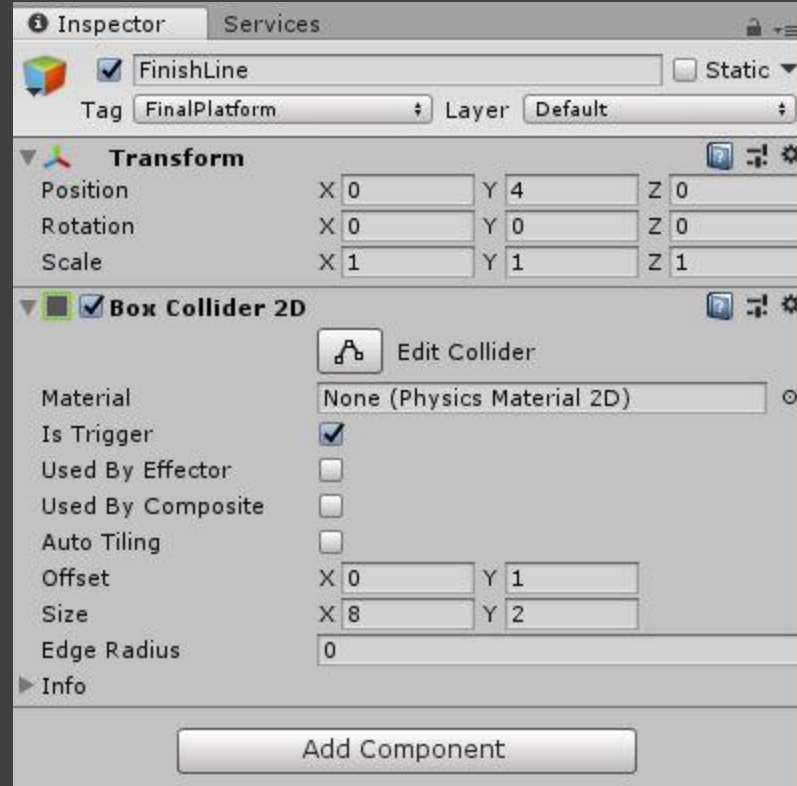
```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class LevelGenerator : MonoBehaviour
6 {
7
8     public GameObject platformPrefab;
9     public GameObject finishLine;
10    public int numberOfBlocks = 50;
11    public float blockHeight = 1f;
12    public int numberOfPlatformsInBlock = 3;
13
14    public float levelWidth = 2.5f;
15
16    public float offset = 0.25f;
17
18    private List<Platform> platformList = new List<Platform>();
19
20    public void Start()
21    {
22
23        int maxCount = numberOfBlocks * numberOfPlatformsInBlock;
24        Transform platformsParent = new GameObject("Platforms").transform;
25        for(int i=0; i < maxCount; i++)
26        {
27            Platform platformTemp = Instantiate(platformPrefab, platformsParent).GetComponent<Platform>();
28            platformList.Add(platformTemp);
29            platformTemp.SetActivation(false);
30        }
31        print(platformsParent.childCount.ToString());
32    }
33 }
```

```
34 public void LevelStart(int level)
35 {
36     float blockHeightTemp = blockHeight + (level - 1) * 0.25f;
37
38
39     for (int i = 0; i < numberOfBlocks; i++)
40     {
41         float minY = i * blockHeightTemp;
42
43         float maxY = (i + 1) * blockHeightTemp;
44
45         for (int j = 0; j < numberOfPlatformsInBlock; j++)
46         {
47             Vector3 spawnPosition = new Vector3();
48             spawnPosition.y = Random.Range(minY, maxY);
49             spawnPosition.x = Random.Range(-levelWidth, levelWidth);
50             int index = i * j;
51             platformList[index].SetPosition(spawnPosition);
52             platformList[index].SetActivation(true);
53         }
54     }
55     float finishLineY = blockHeightTemp * numberOfBlocks;
56     finishLine.transform.position = new Vector2(0, finishLineY);
57 }
58 }
```

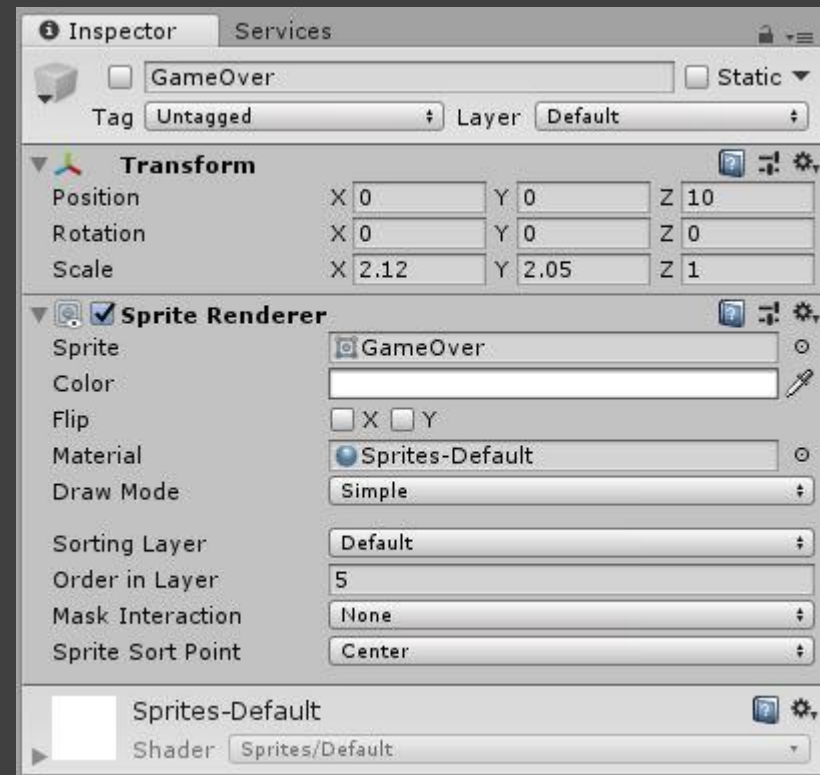
Game Scene – Kill Zone



Game Scene – Finish Line



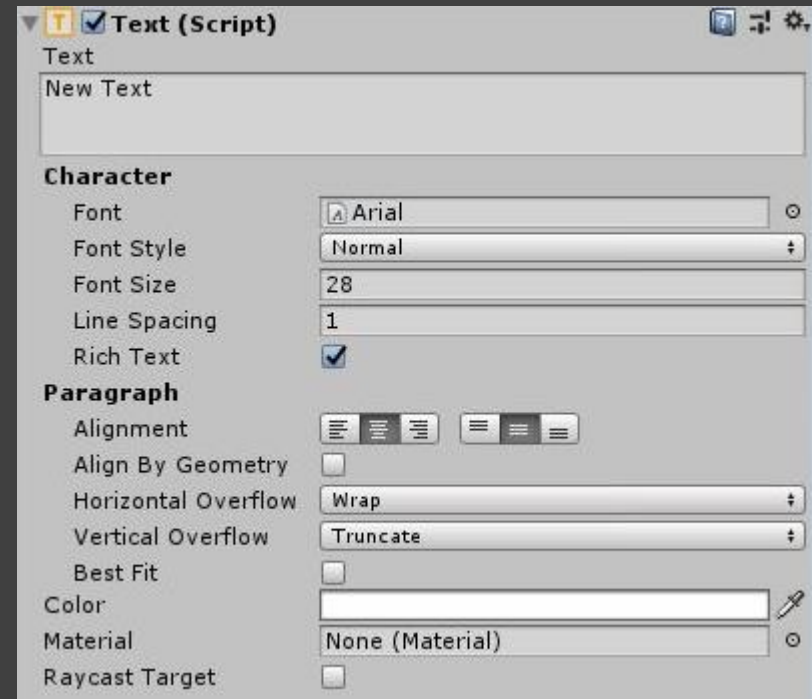
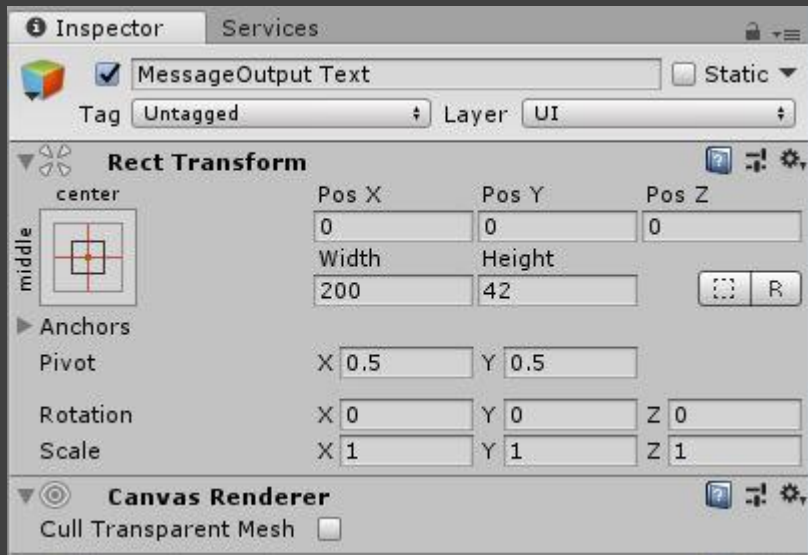
Game Scene - Game Over



Game Scene - Score

```
1
2  using UnityEngine;
3  using UnityEngine.UI;
4
5  public class Score : MonoBehaviour {
6
7      public Transform player;
8      public Text scoreText;
9      public int lastScore = 0;
10     public int currentScore = 0;
11     public void SetLastScore()
12     {
13         lastScore = currentScore;
14         currentScore = 0;
15     }
16     // Update is called once per frame
17     void Update () {
18         currentScore = lastScore + (int)player.position.y * 10;
19         scoreText.text = currentScore.ToString();
20     }
21 }
```

Game Scene - Message Output



Game Scene – Audio Manager

```
1 using UnityEngine.Audio;
2 using UnityEngine;
3
4 public class AudioManager : MonoBehaviour {
5
6     public Sound[] sounds;
7
8     // Use this for initialization
9     void Awake () {
10         foreach (Sound s in sounds)
11         {
12             s.source = gameObject.AddComponent<AudioSource>();
13             s.source.clip = s.clip;
14
15             s.source.volume = s.volume;
16             s.source.pitch = s.pitch;
17         }
18     }
19
20     // Update is called once per frame
21     void Update () {
22
23     }
24 }
```

Game Scene - Sound

```
1  using UnityEngine.Audio;
2  using UnityEngine;
3
4  [System.Serializable]
5  public class Sound {
6
7      public string name;
8
9      public AudioClip clip;
10
11     [Range(0f, 1f)]
12     public float vloume;
13     [Range(.1f, 3f)]
14     public float pitch;
15
16     [HideInInspector]
17     public AudioSource source;
18
19 }
```


감사합니다.