더 데스 레이스

THE DEATH RACE

게임엔진활용 (Unity 3D)

[텀 프로젝트 최종 보고서]

게임소프트웨어학과 91716218 김규리

Contents

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 데스 레이스

THE DEATH RACE

게임 소개 - 게임 내용

모두가 좀비로 감염된 학교에서 홀로 감염되지 않은 한 학생이 탈출하는 게임

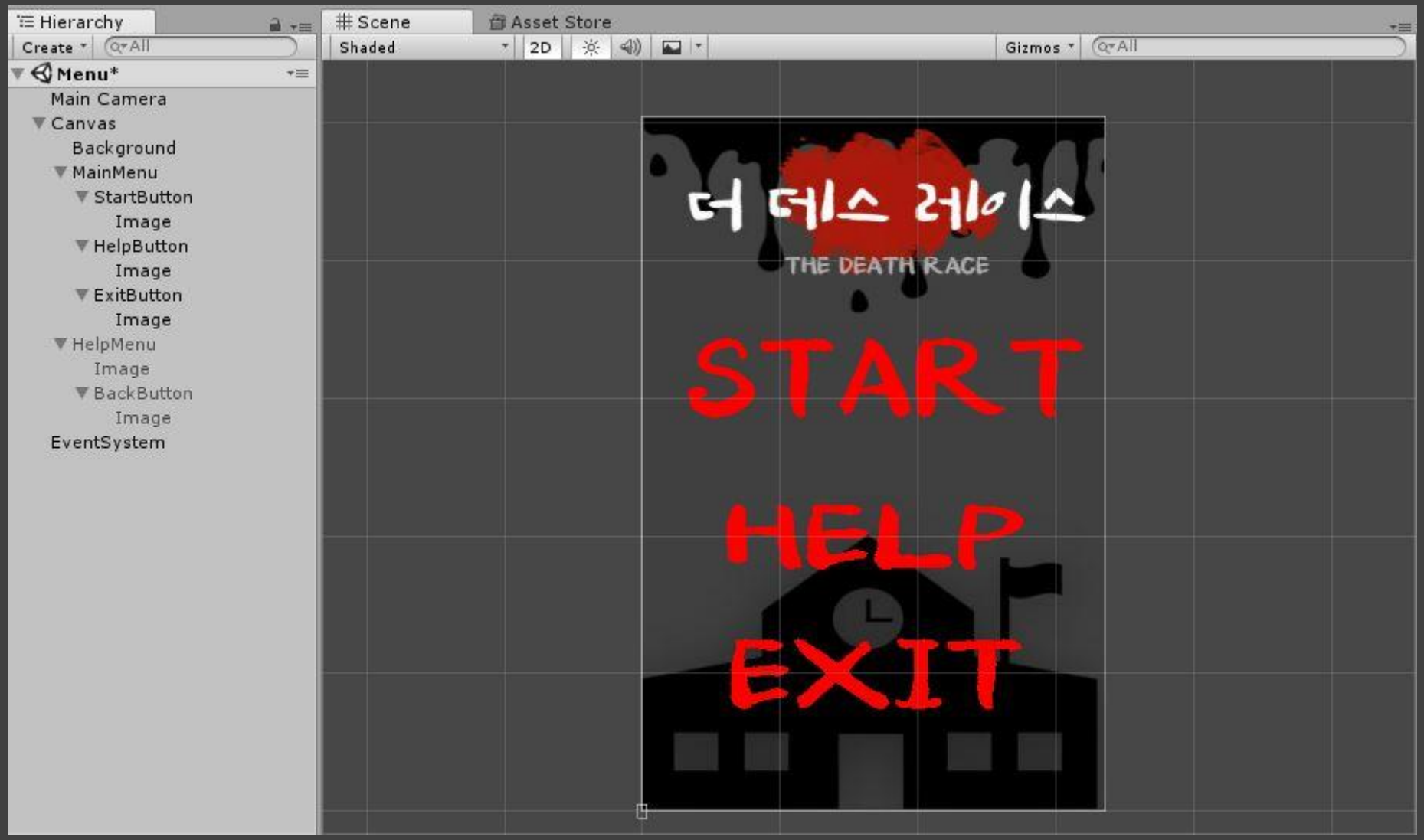
게임 소개 - 게임 조작법

주인공을 좌우방향키로 조종하여
아래에서 올라오는 좀비의 손길로부터 도망쳐
위로 내려가는 게임

2

Menu Scene

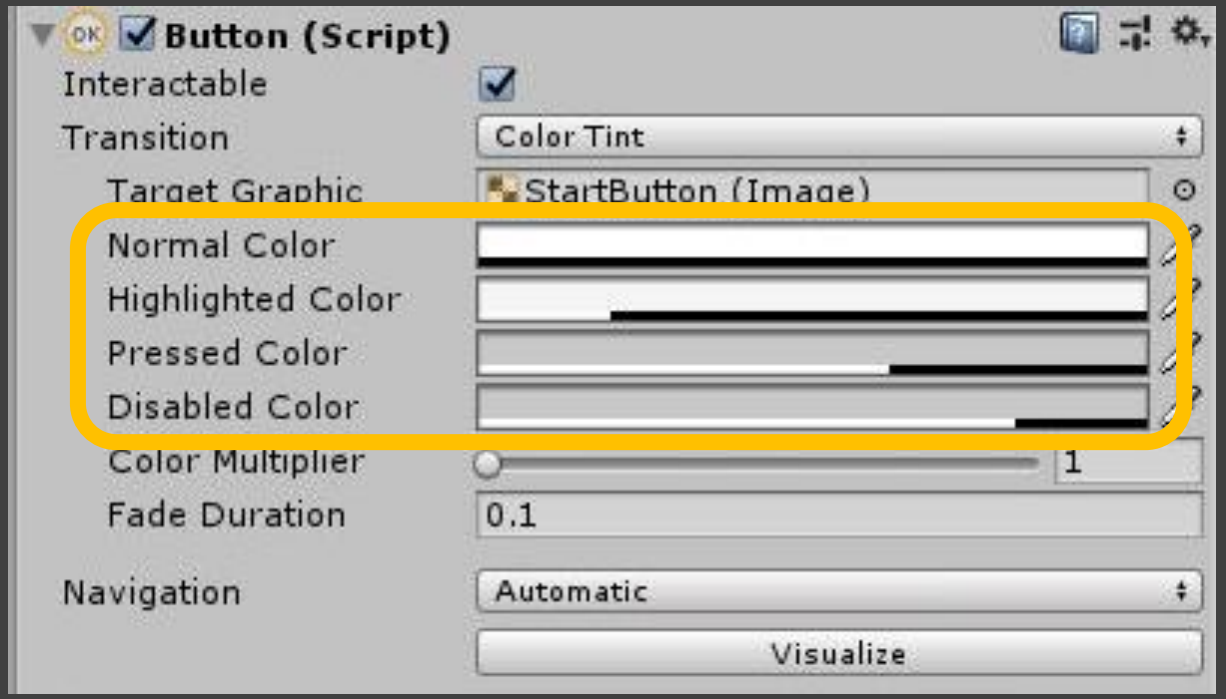
Menu Scene


Hierarchy
Create
All
Menu*
Main Camera
Canvas
Background
MainMenu
StartButton
Image
HelpButton
Image
ExitButton
Image
HelpMenu
Image
BackButton
Image
EventSystem
Scene
Asset Store
Shaded
2D
Gizmos
All
더 데스 레이스
THE DEATH RACE
START
HELP
EXIT

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  }
```

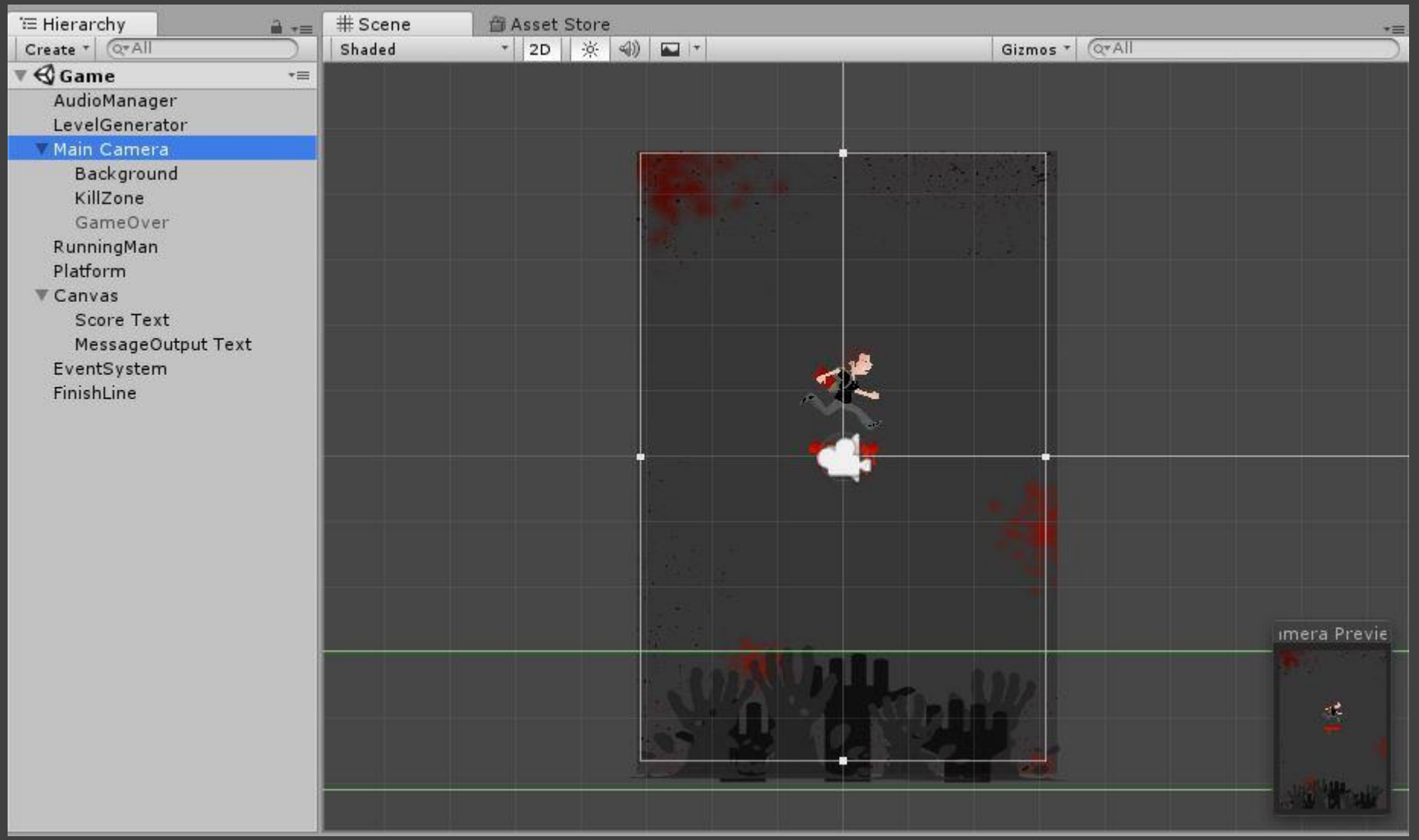
Menu Scene - Button 메뉴의 효과



3

Game Scene

Game Scene


Hierarchy
Create
All
Game
AudioManager
LevelGenerator
Main Camera
Background
KillZone
GameOver
RunningMan
Platform
Canvas
Score Text
MessageOutput Text
EventSystem
FinishLine
Scene
Asset Store
Shaded
2D
Gizmos
All
imera Previe

Game Scene - Player (Running Man)

Inspector | Services

RunningMan — Static

Tag: Untagged — Layer: Default

Transform

	X	Y	Z
Position	-0.01	1.07	0
Rotation	0	0	0
Scale	0.2567217	0.2565144	1.4149

Sprite Renderer

Sprite	Runningman
Color	
Flip	X Y
Material	Sprites-Default
Draw Mode	Simple
Sorting Layer	Default
Order in Layer	4
Mask Interaction	None
Sprite Sort Point	Center

Rigidbody 2D

Body Type	Dynamic
Material	None (Physics Material 2D)
Simulated	☑
Use Auto Mass	☐
Mass	1
Linear Drag	0
Angular Drag	0.05
Gravity Scale	1
Collision Detection	Continuous
Sleeping Mode	Start Awake
Interpolate	None

Box Collider 2D

Edit Collider

Material	None (Physics Material 2D)
Is Trigger	☐
Used By Effector	☐
Used By Composite	☐
Auto Tiling	☐
Offset	X 0.3163635 Y -1.815137
Size	X 4.32728 Y 1.679725
Edge Radius	0

Info

Player (Script)

Script	Player
Movement Speed	10

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                  SetGaravityScale(0);
94                  break;
95              case "FinalPlatform":
96                  GameManager.currentState = GameState.Clear;
97                  print("Clear Level.");
98                  break;
99          }
100     }
101 }
```

Game Scene - Platform

Inspector | Services

Platform — Static

Tag: Untagged — Layer: Default

Transform

	X	Y	Z
Position	0	0	0
Rotation	0	0	0
Scale	0.4721169	0.6143647	1

Sprite Renderer

Sprite	Platform
Color	
Flip	X Y
Material	Sprites-Default
Draw Mode	Simple
Sorting Layer	Default
Order in Layer	0
Mask Interaction	None
Sprite Sort Point	Center

Edge Collider 2D

Edit Collider

Material	None (Physics Material 2D)
Is Trigger	
Used By Effector	
Offset	X 0 Y 0
Edge Radius	0

Points

Size	2
Element 0	X -1.050326 Y 0.3635605
Element 1	X 1.073687 Y 0.3635605

Info

Platform (Script)

Script	Platform
Jump Force	10

Platform Effector 2D

Use Collider Mask	☑
Collider Mask	Everything
Rotational Offset	0

One Way

Use One Way	☑
Use One Way Grouping	☐
Surface Arc	180

Sides

Game Scene - Platform

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

public class Platform : MonoBehaviour
{
    public float jumpForce = 5f;

    public void SetActivation(bool activation)
    {
        gameObject.SetActive(activation);
    }

    public void SetPosition(Vector3 position)
    {
        transform.position = position;
    }

    void OnCollisionEnter2D(Collision2D collision)
    {
        if (GameManager.currentState.Equals(GameState.Gameover)) return;

        if (collision.relativeVelocity.y <= 0f )
        {
            Rigidbody2D rb = collision.collider.GetComponent<Rigidbody2D>();

            if (rb != null)
            {
                Vector2 velocity = rb.velocity;
                velocity.y = jumpForce;
                rb.velocity = velocity;
            }
        }
    }

}
```

Game Scene - Level Generator

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

public class LevelGenerator : MonoBehaviour
{

    public GameObject platformPrefab;
    public GameObject finishLine;
    public int numberOfBlocks = 50;
    public float blockHeight = 1f;
    public int numberOfPlatformsInBlock = 3;

    public float levelWidth = 2.5f;

    public float offset = 0.25f;

    private List<Platform> platformList = new List<Platform>();

    public void Start()
    {

        int maxCount = numberOfBlocks * numberOfPlatformsInBlock;
        Transform plaformsParent = new GameObject("Plaforms").transform;
        for(int i=0; i < maxCount;i++)
        {
            Platform platformTemp = Instantiate(platformPrefab, plaformsParent).GetComponent<Platform>();
            platformList.Add(platformTemp);
            platformTemp.SetActivation(false);
        }
        print(plaformsParent.childCount.ToString());
    }
```

```
    public void LevelStart(int level)
    {
        float blockHeightTemp = blockHeight + (level - 1) * 0.25f;


        for (int i = 0; i < numberOfBlocks; i++)
        {
            float minY = i * blockHeightTemp;

            float maxY = (i + 1) * blockHeightTemp;

            for (int j = 0; j < numberOfPlatformsInBlock; j++)
            {
                Vector3 spawnPosition = new Vector3();
                spawnPosition.y = Random.Range(minY, maxY);
                spawnPosition.x = Random.Range(-levelWidth, levelWidth);
                int index = i * j;
                platformList[index].SetPosition(spawnPosition);
                platformList[index].SetActivation(true);
            }
        }
        float finishLineY = blockHeightTemp * numberOfBlocks;
        finishLine.transform.position = new Vector2(0, finishLineY);
    }
}
```

Game Scene - Kill Zone

Inspector | Services

KillZone — Static

Tag: KillZone — Layer: Default

Transform

	X	Y	Z
Position	0.097	-3.69	10
Rotation	0	0	180
Scale	1.447735	0.9757322	1

Sprite Renderer

Sprite: Obstacle
Color:
Flip: X Y
Material: Sprites-Default
Draw Mode: Simple
Sorting Layer: Default
Order in Layer: 0
Mask Interaction: None
Sprite Sort Point: Center

Box Collider 2D

Edit Collider
Material: None (Physics Material 2D)
Is Trigger: ☑
Used By Effector: ☐
Used By Composite: ☐
Auto Tiling: ☐

	X	Y
Offset	0.0718450	0.3332266
Size	20.78827	2.163547

Edge Radius: 0
Info

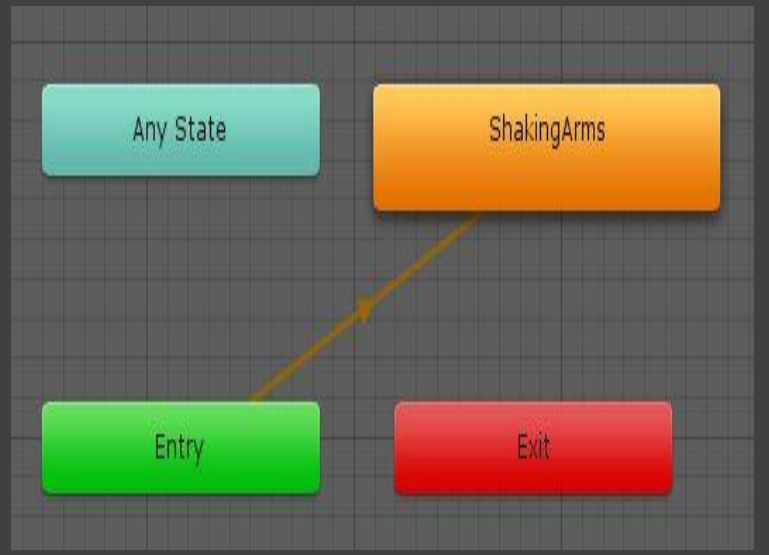
Animator

Controller: KillZone
Avatar: None (Avatar)
Apply Root Motion: ☐

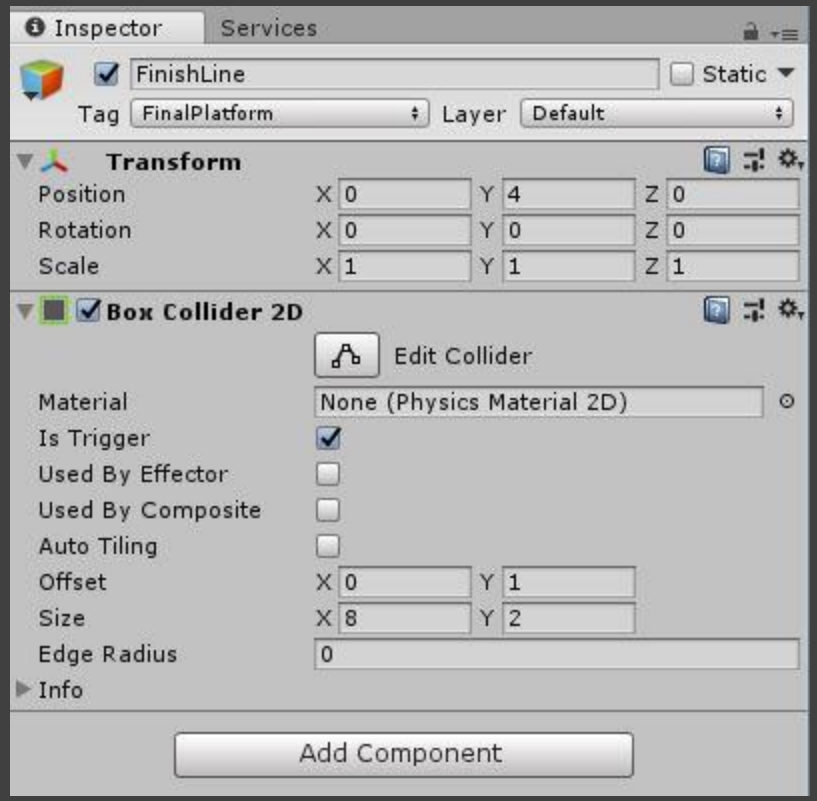
Root position or rotation are controlled by curves

Update Mode: Normal
Culling Mode: Always Animate

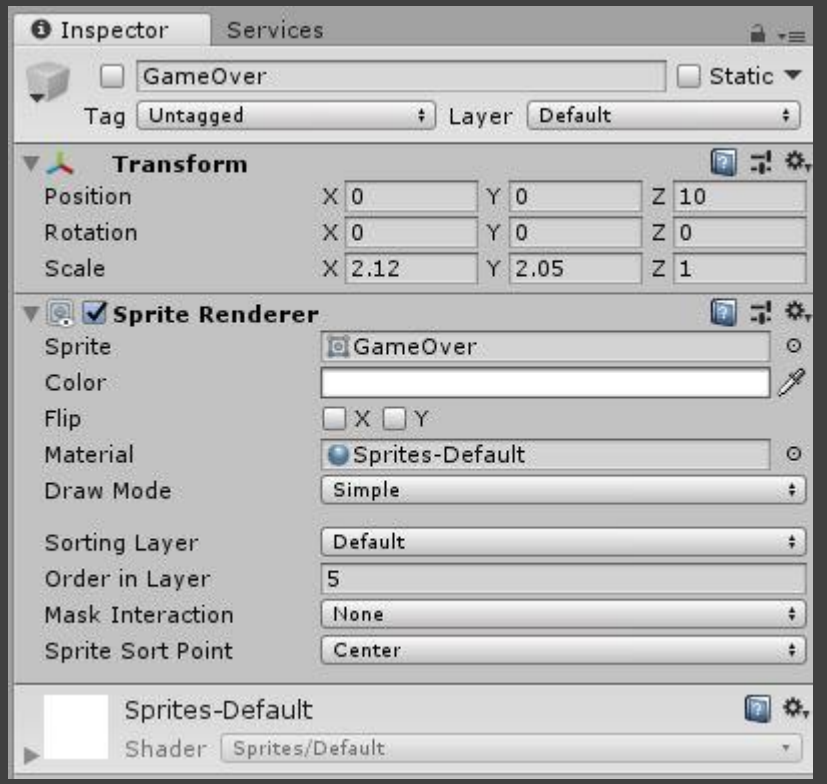
Clip Count: 1
Curves Pos: 1 Quat: 0 Euler: 0 Scale: 0 Muscles: 0 Generic: 0 PPtr: 0
Curves Count: 3 Constant: 0 (0.0%) Dense: 0 (0.0%) Stream: 3 (100.0%)



Game Scene - Finish Line



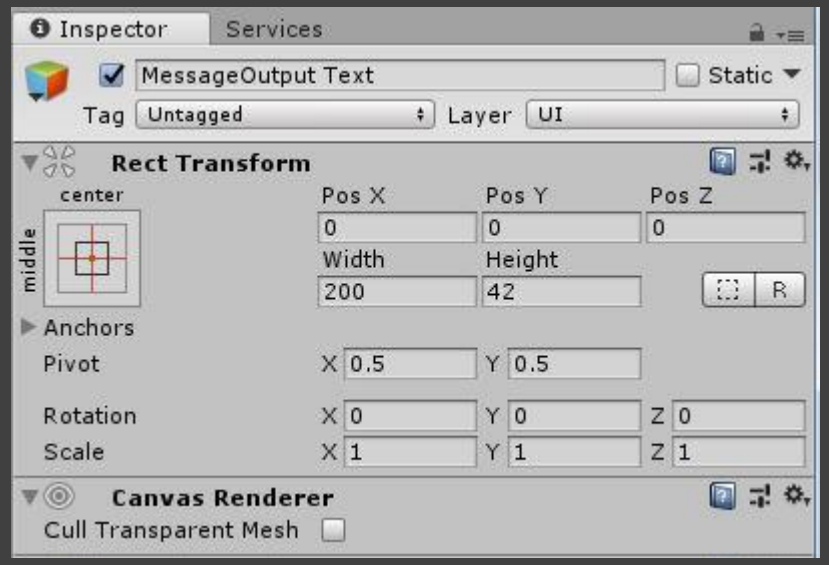
Game Scene - Game Over

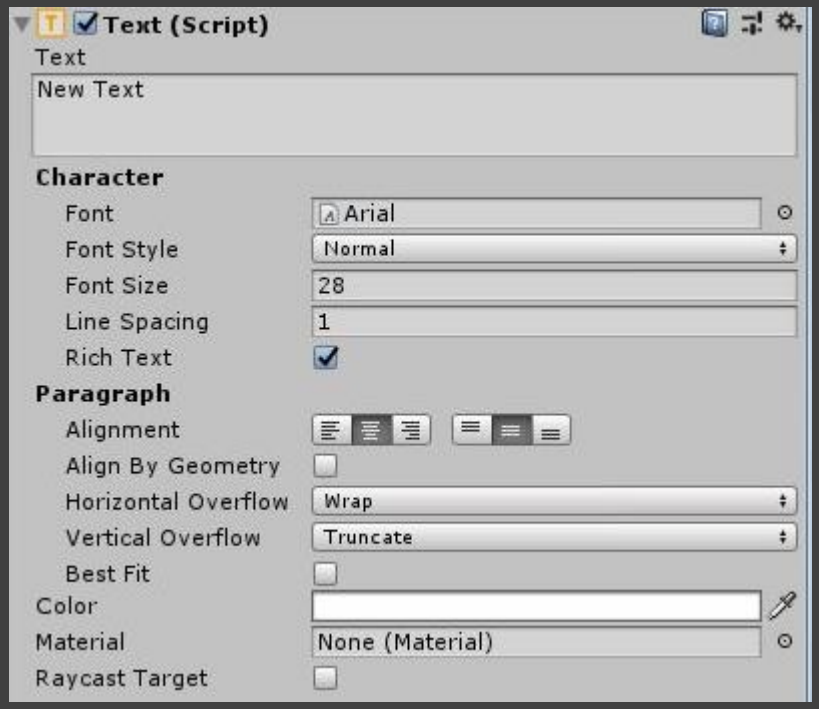


Game Scene - Score

```
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 currentSocre = 0;
11      public void SetLastScore()
12      {
13          lastScore = currentSocre;
14          currentSocre = 0;
15      }
16      // Update is called once per frame
17      void Update () {
18          currentSocre = lastScore + (int)player.position.y * 10;
19          scoreText.text = currentSocre.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.vloume;
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.