



9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
24
25
26

```
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;  
  
void start  
{  
}  
  
void update  
{  
}
```

게임엔진기초 기말 프로젝트

ICT융합공학부
202104320 임민석



9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
24
25
26



목차

1. 개요

3. 주요 코드

2. 게임 방법 및 소개

4. 게임 플레이

9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
24
25
26



1. 개요



2D 플랫폼머

플랫폼이란 발판을 의미하므로 이는 발판
이 등장하는 게임을 뜻한다. 구체적으로
는 플레이어가 캐릭터를 조종할 때 발판
위를 뛰어다니는 점프 컨트롤이 중요한
게임 장르이다.



2D 플랫폼머 장르의 대표적인 게임



9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
24
25
26



2. 게임 방법 및 소개

조작법

플레이어 이동 :

A, S키 or ←, →

플레이어 점프 :

스페이스바



게임 재시작 : R



9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
24
25
26

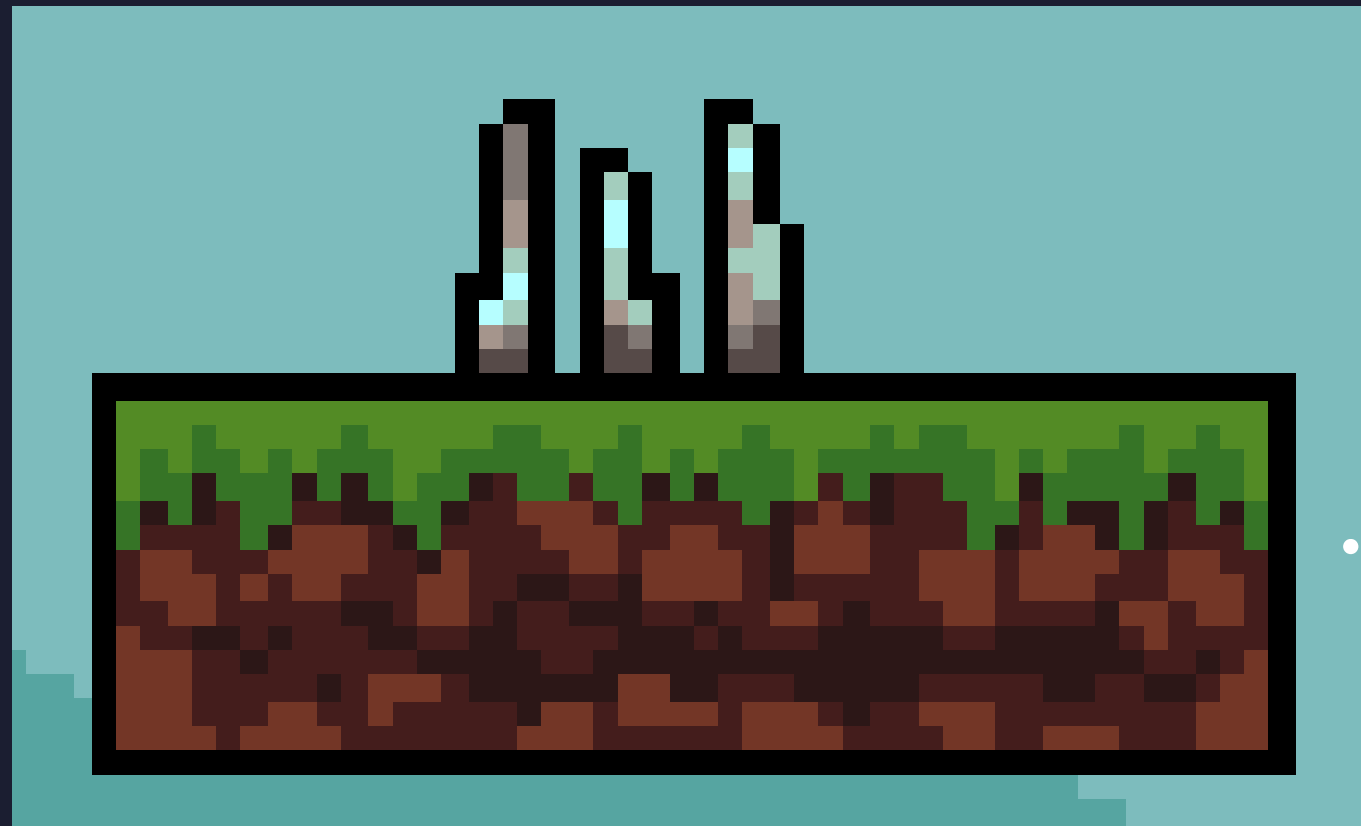


2. 게임 방법 및 소개

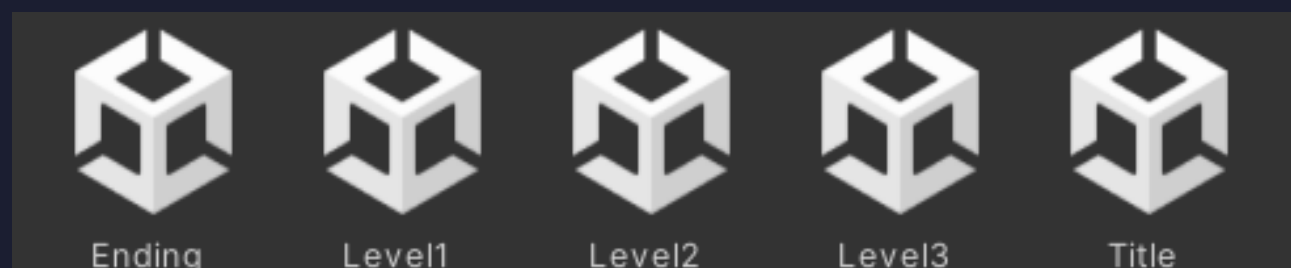
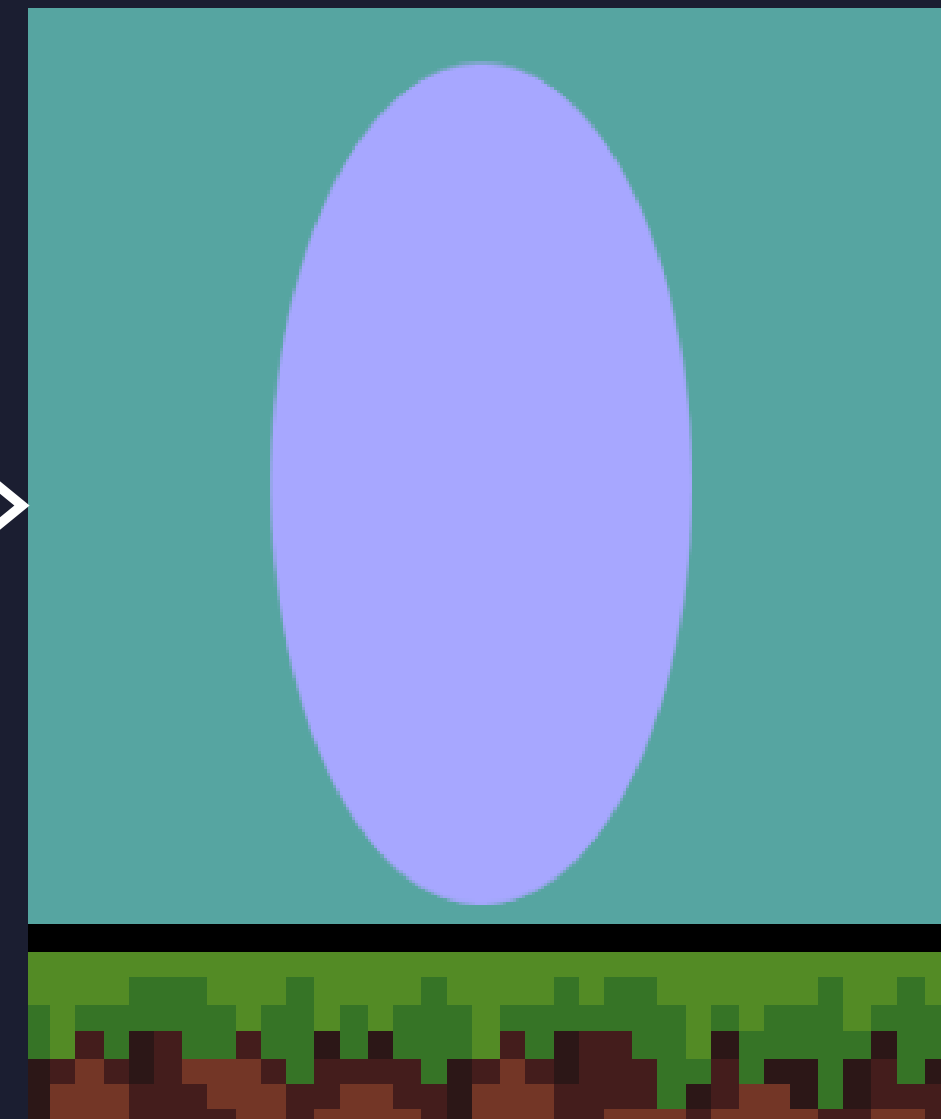
게임 목표

장애물 오브젝트를 피하면서 포탈 오브젝트에 도달하면 해당 Level 클리어

장애물 오브젝트



포탈 오브젝트



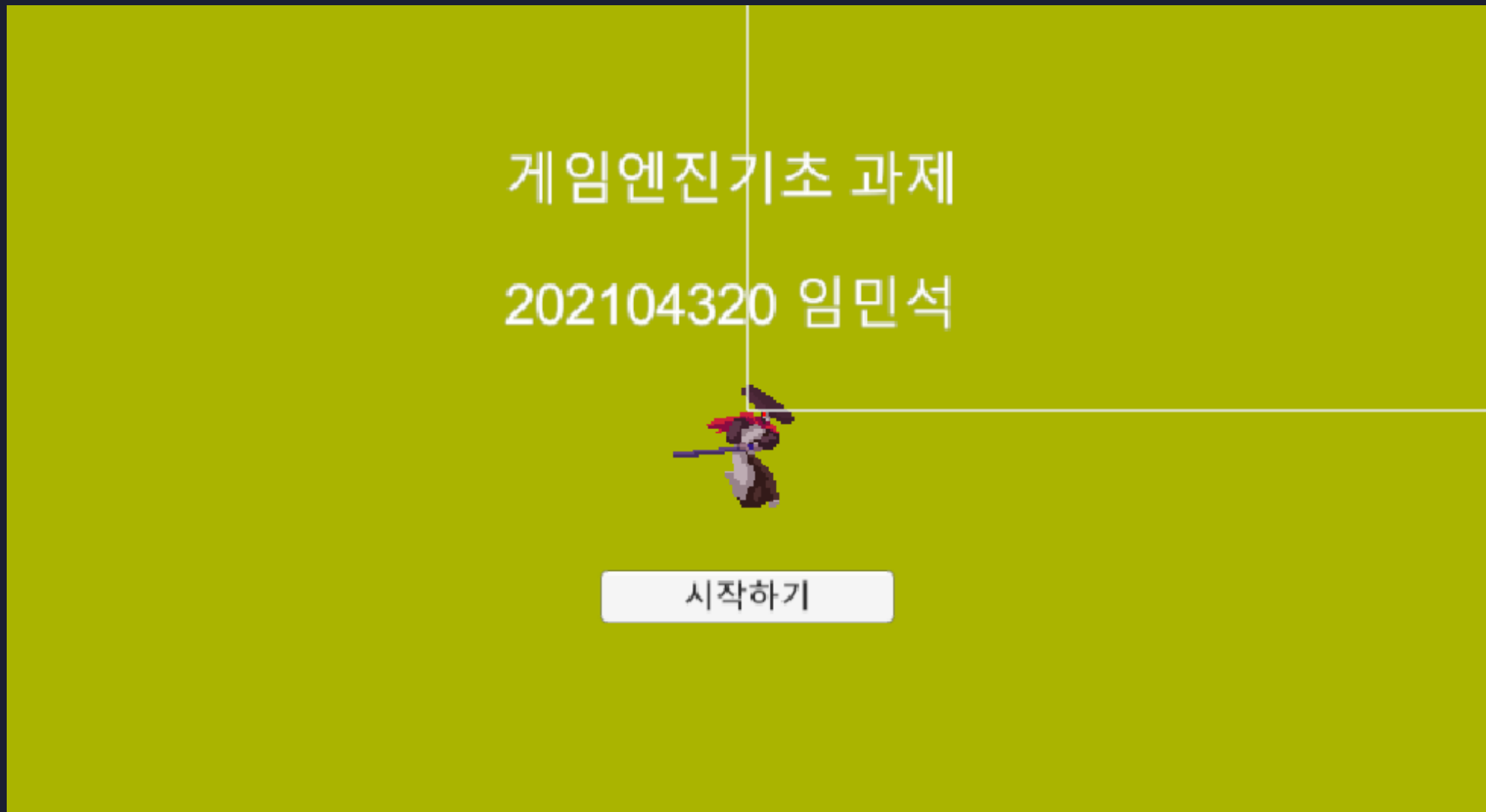
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
24
25
26



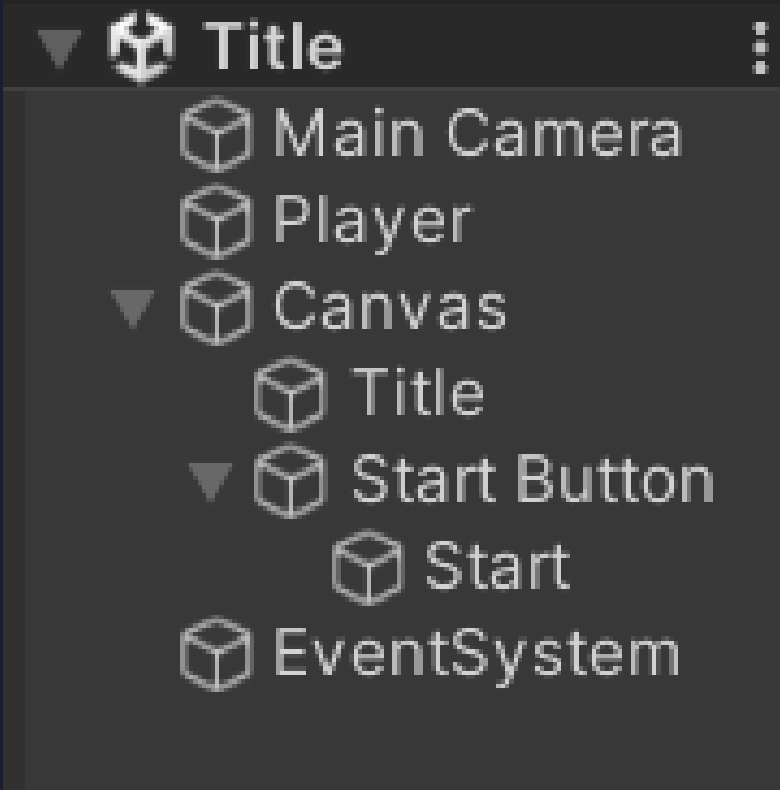
2. 게임 방법 및 소개

Title Scene

Game Scene



Hierarchy



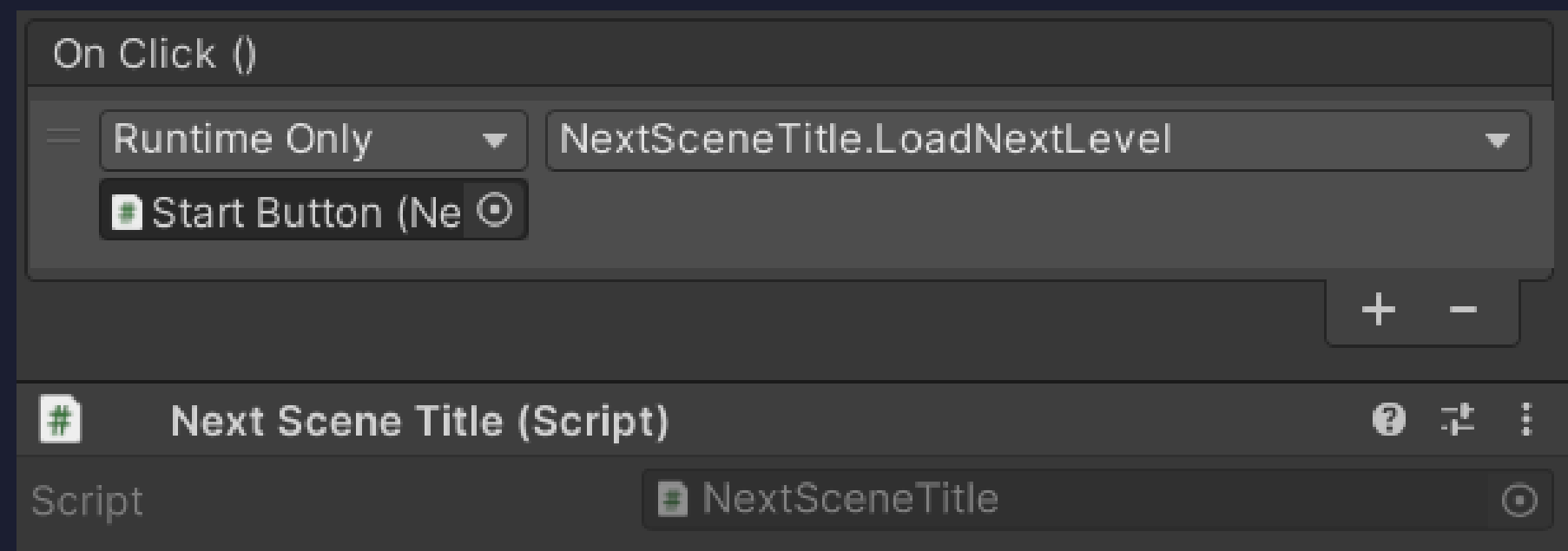
2. 게임 방법 및 소개

NextSceneTitle

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

Unity 스크립트(자산 참조 1개) | 참조 0개
public class NextSceneTitle : MonoBehaviour
{
    참조 0개
    public void LoadNextLevel()
    {
        SceneManager.LoadScene("Level1");
    }
}
```

Start Button Inspector

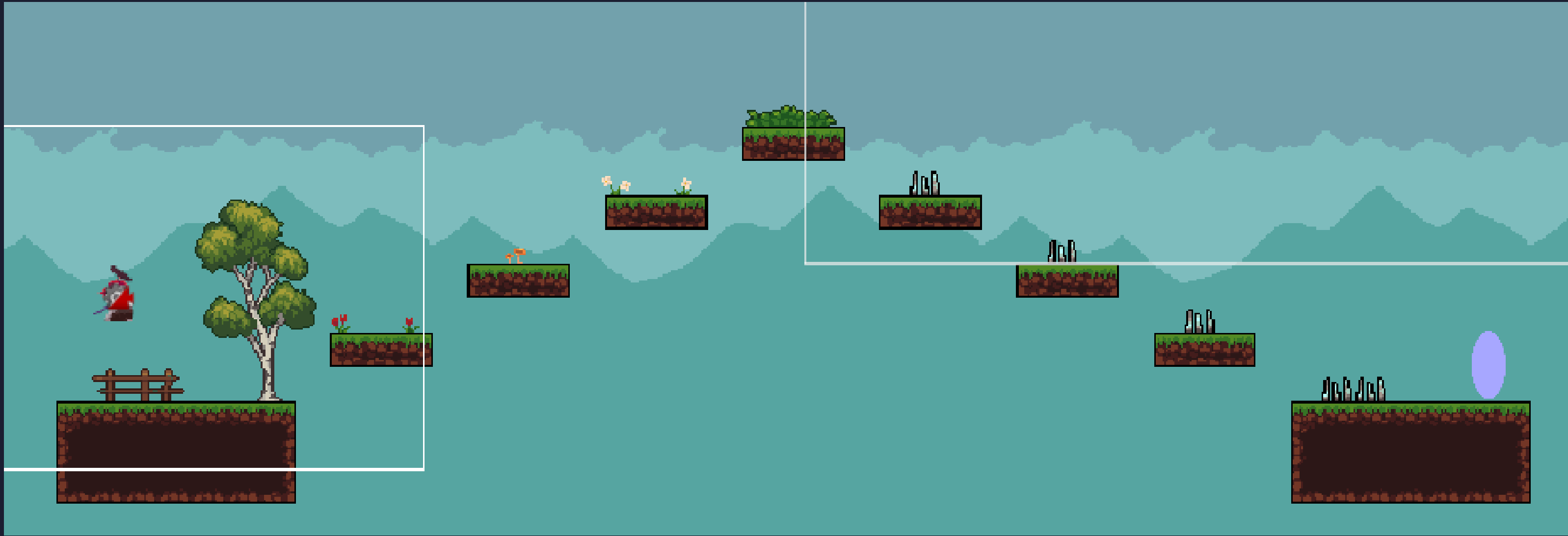


9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
24
25
26



2. 게임 방법 및 소개

Level1 Scene

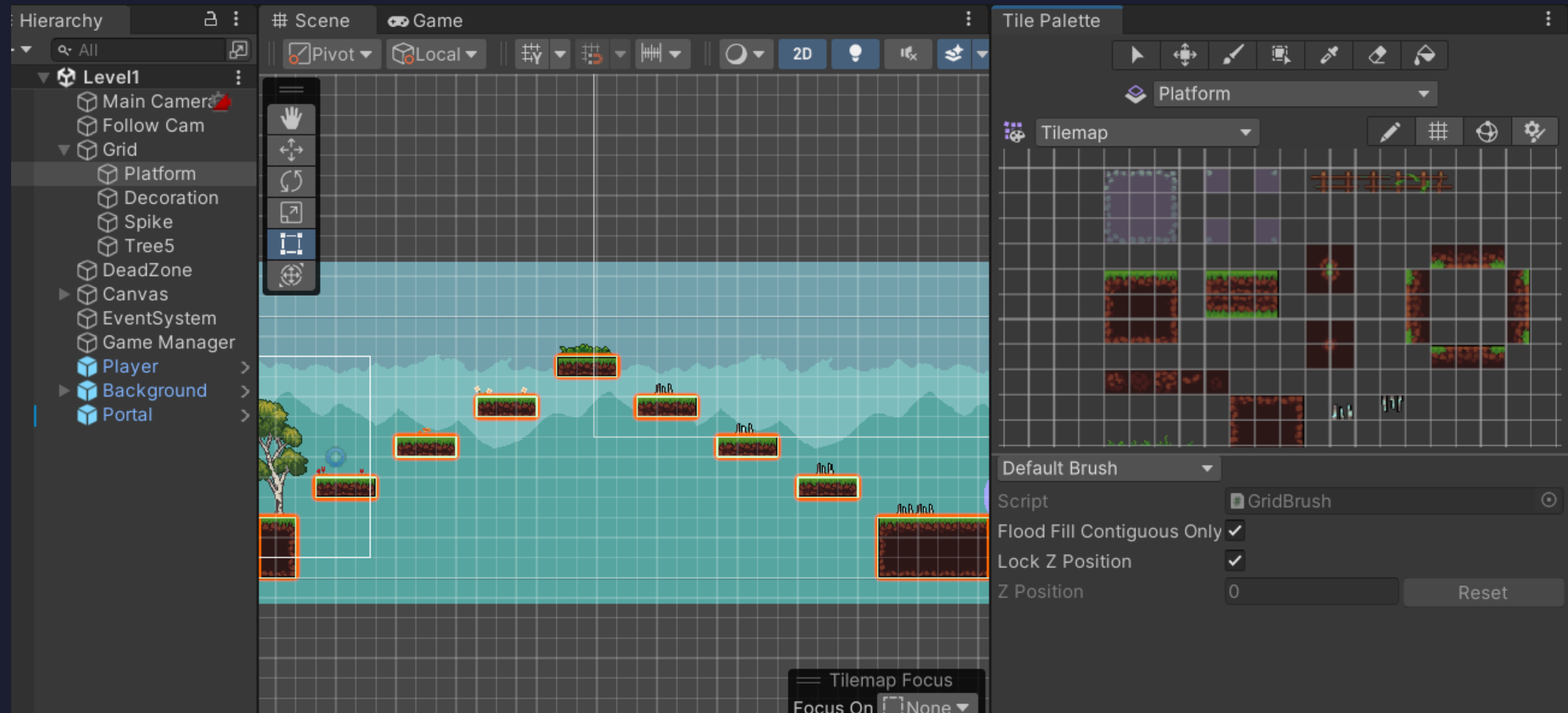




2. 게임 방법 및 소개

TileMap

타일맵 (Tilemap) 이란, 2D 레벨을 구현할때, 타일 에셋들을 가져와 관리하고 타일들의 배치와 속성을 관리하는 컴포넌트입니다.

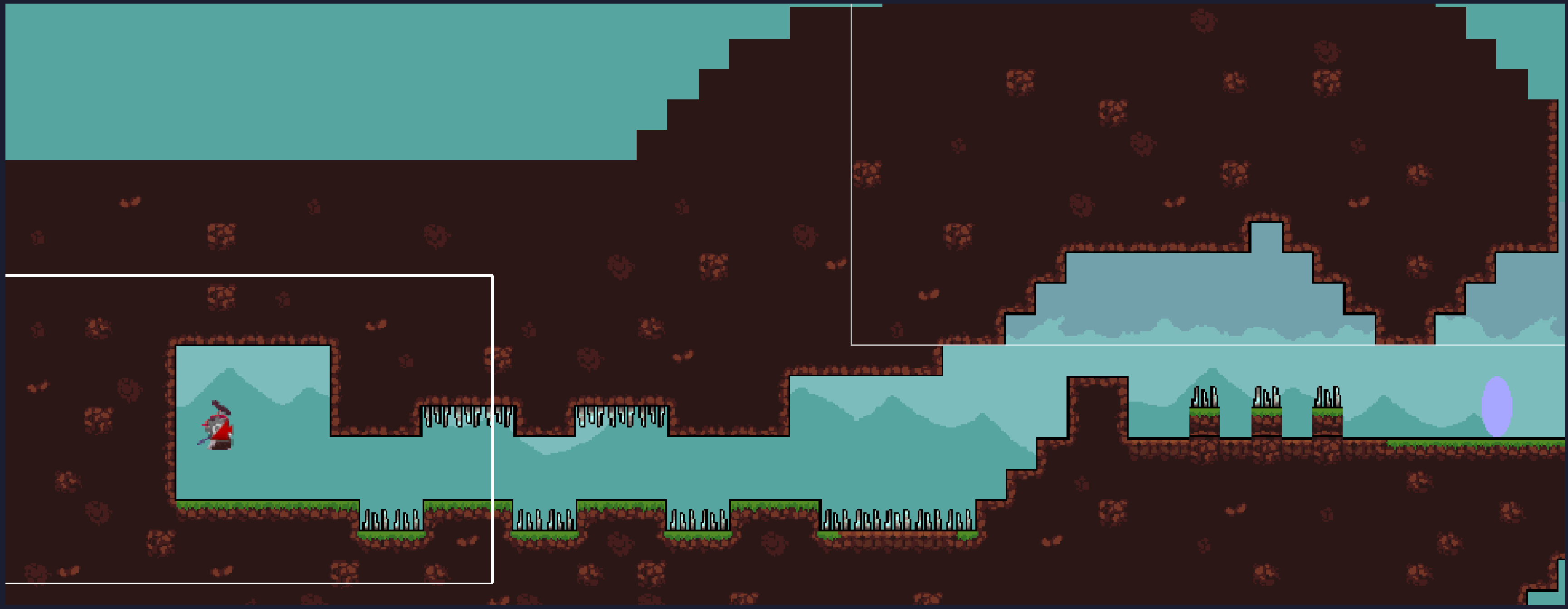


9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
24
25
26



2. 게임 방법 및 소개

Level2 Scene



9



2. 게임 방법 및 소개

10

11

Level3 Scene

12

13



24

24

25

26

2. 게임 방법 및 소개

MovingPlatform Script

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

// Unity 스크립트(자산 참조 1개) | 참조 0개
public class MovingPlatform : MonoBehaviour
{
    public Transform point;
    public float speed = 2.8f;

    private Vector3 targetPosition;

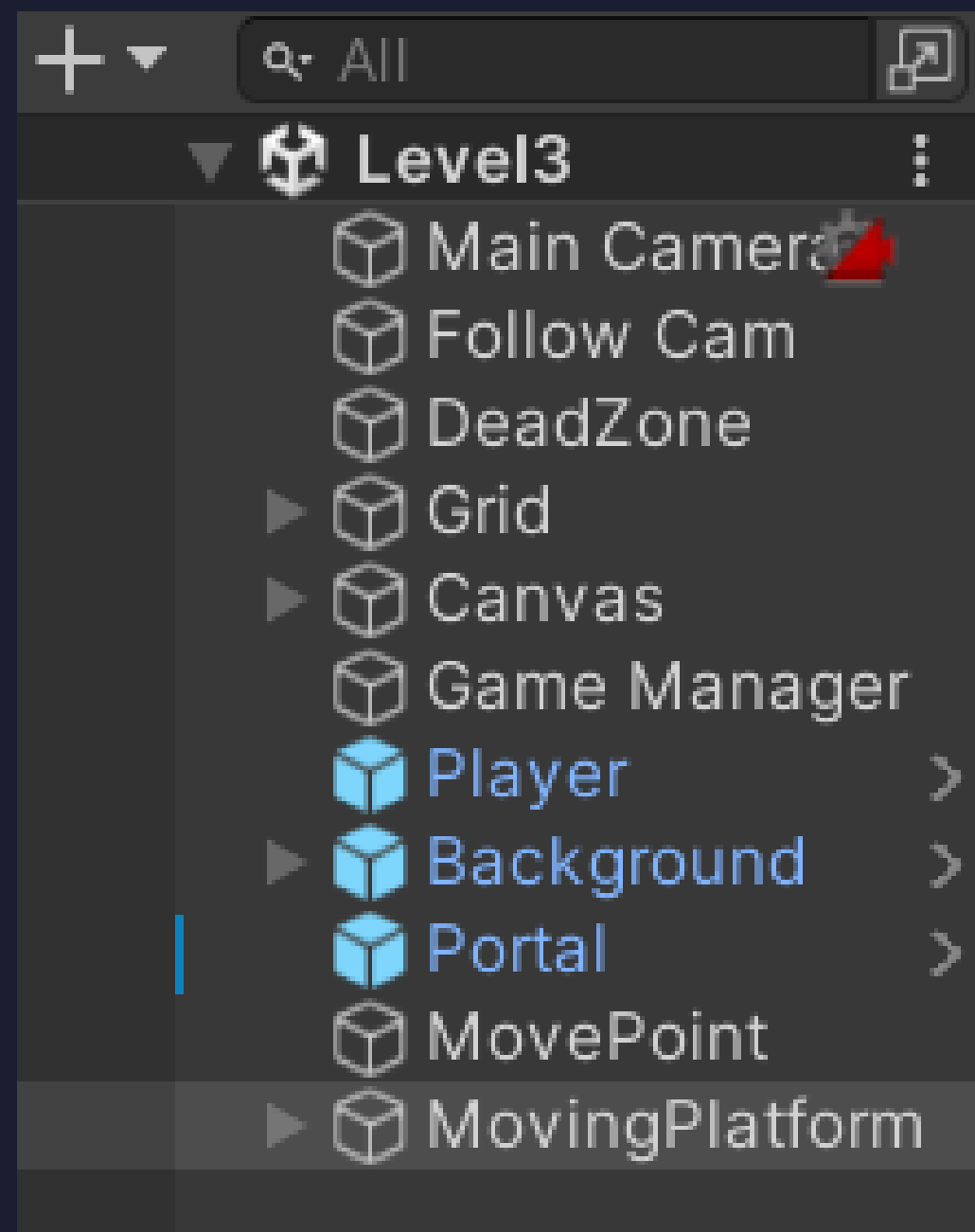
    // Unity 메시지 | 참조 0개
    void Start()
    {
        targetPosition = point.position;
    }

    // Unity 메시지 | 참조 0개
    void Update()
    {
        transform.position = Vector3.MoveTowards(transform.position, targetPosition, speed * Time.deltaTime);
    }
}
```

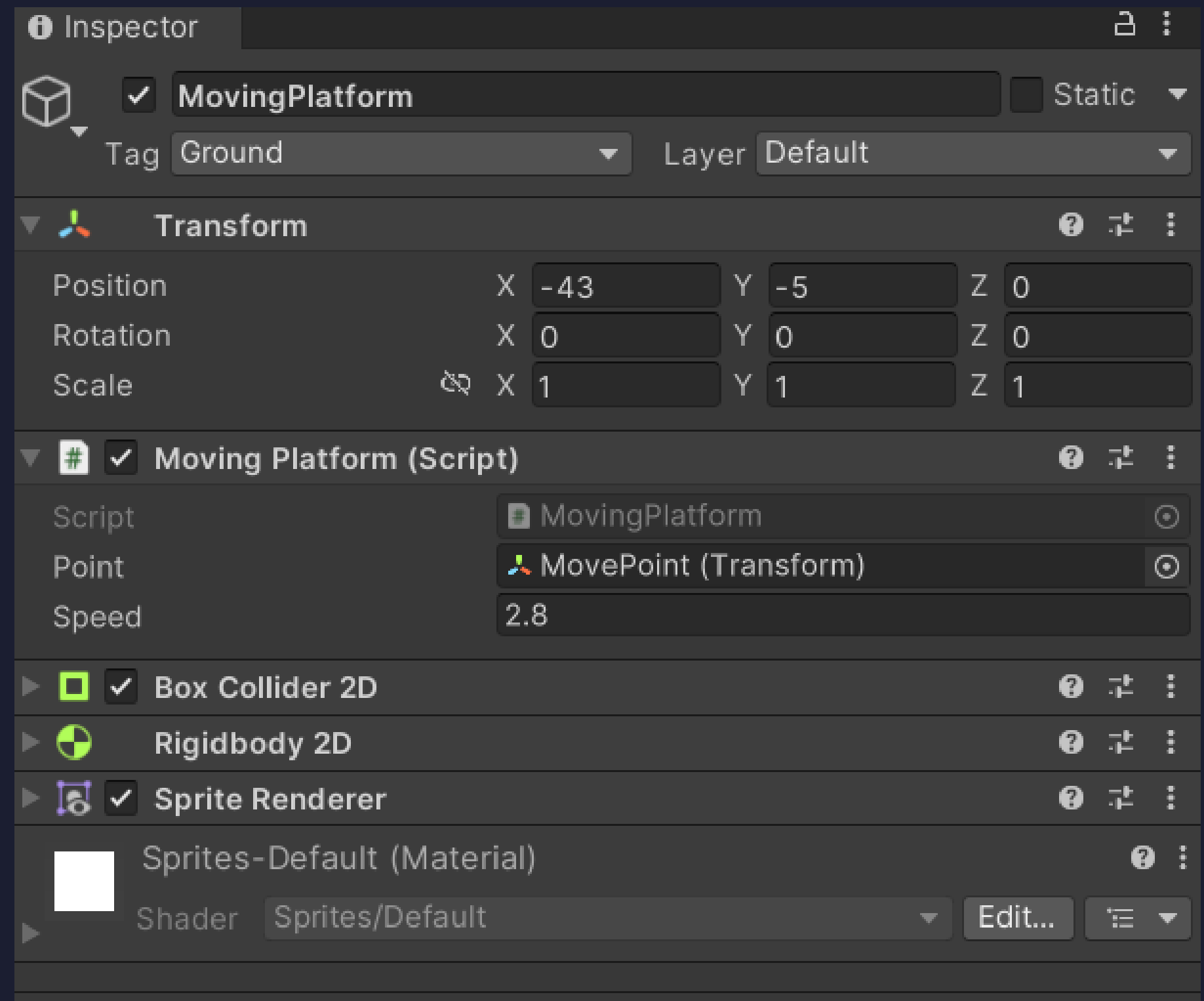
2. 게임 방법 및 소개

MovingPlatform

Hierarchy



Inspector



9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
24
25
26

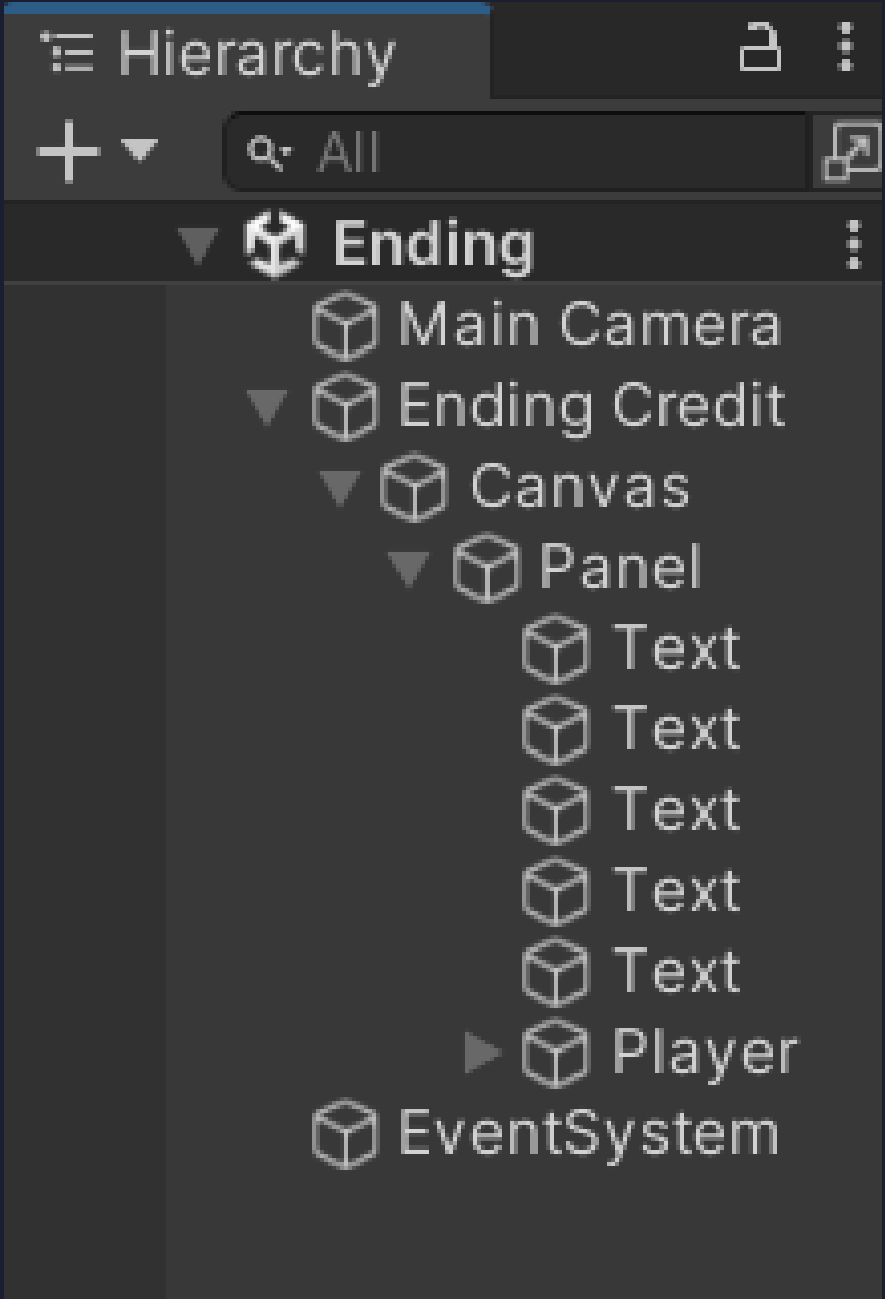


2. 게임 방법 및 소개

Ending Scene



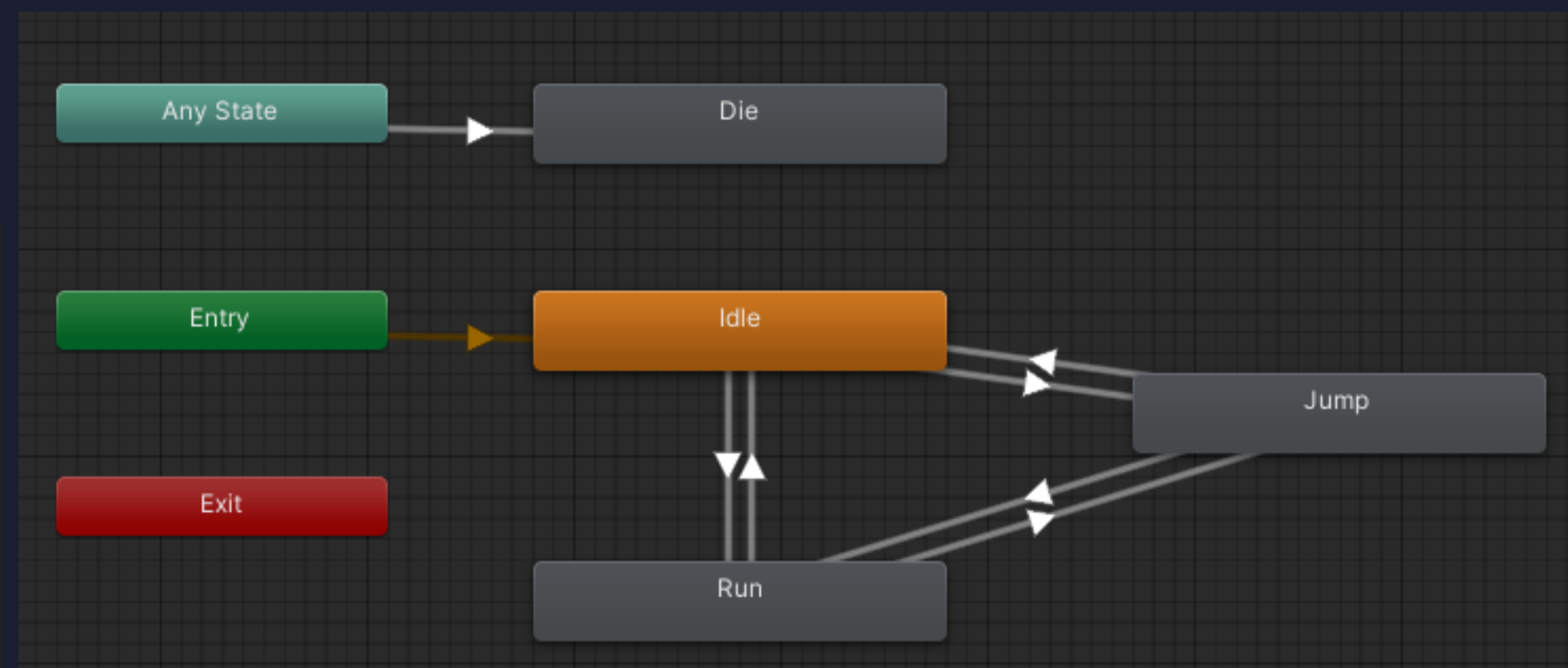
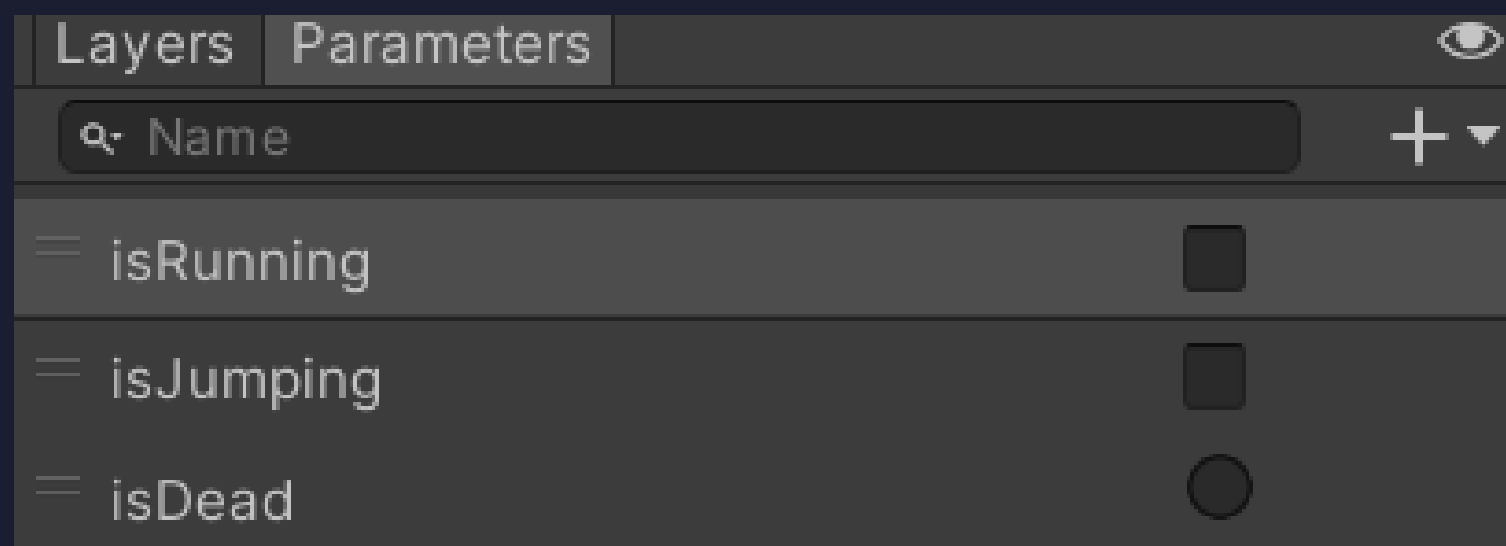
Hierarchy



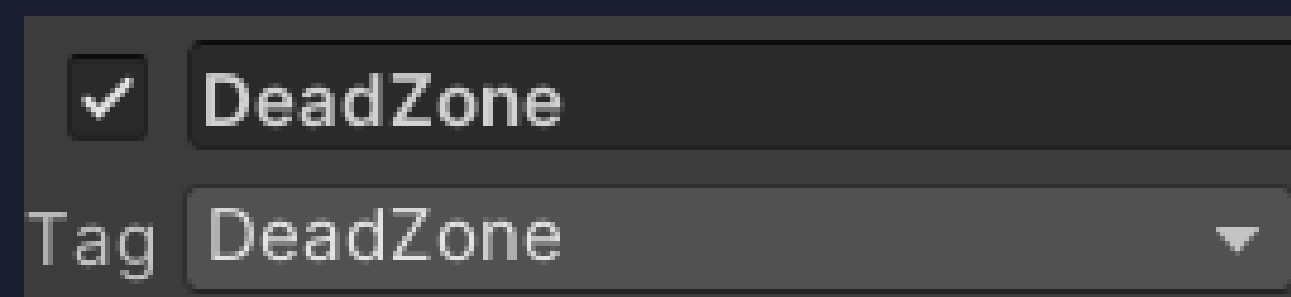
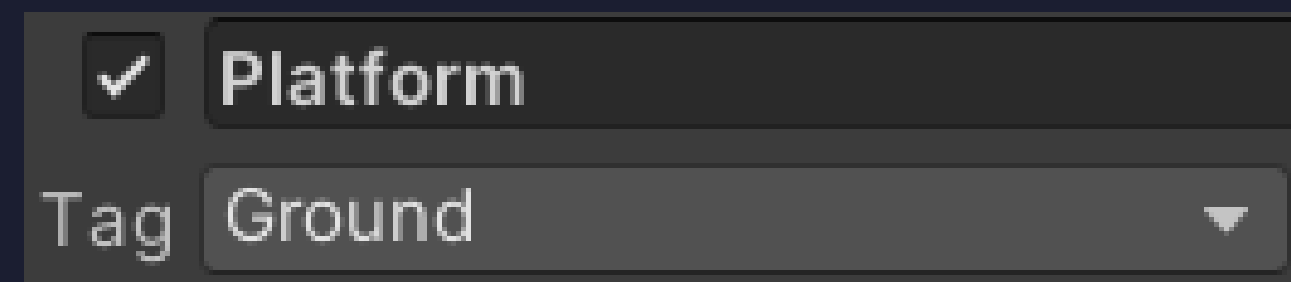
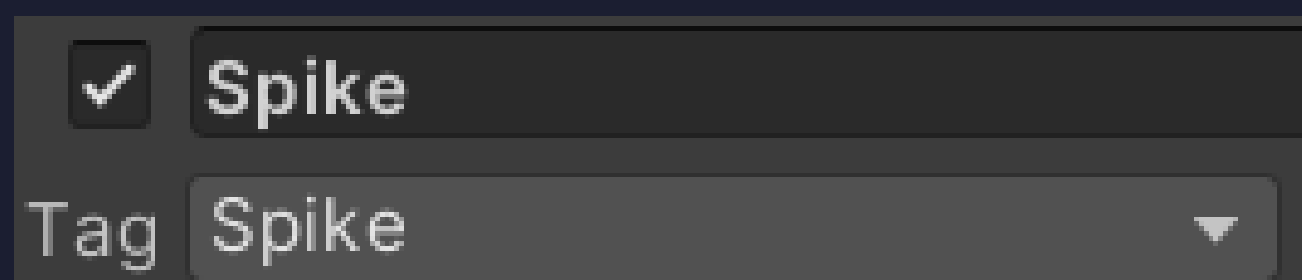
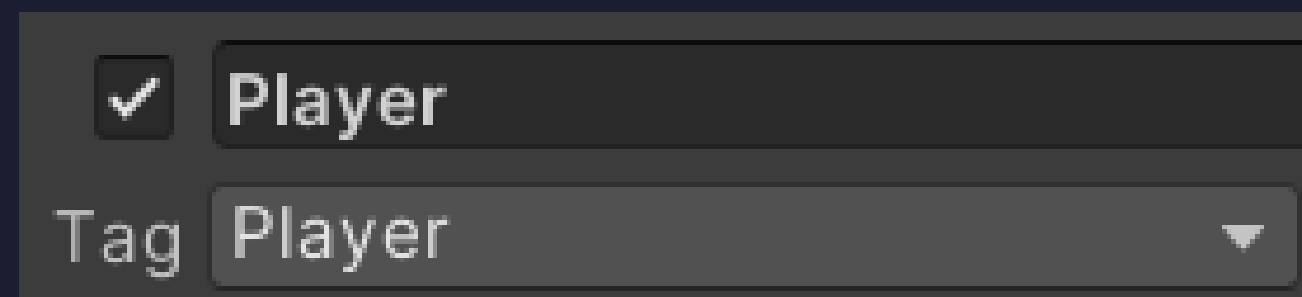


3. 주요 코드

Player Animator



오브젝트 태그 설정



3. 주요 코드

PlayerController Script

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

Unity 스크립트(자산 참조 1개) | 참조 0개
public class PlayerController : MonoBehaviour
{
    public float moveSpeed = 5f;
    public float jumpForce = 5f;
    private bool isGrounded;
    private bool isGameOver = false;

    private Rigidbody2D playerRigidbody;
    private Animator animator;

    Unity 메시지 | 참조 0개
    void Start()
    {
        playerRigidbody = GetComponent<Rigidbody2D>();
        animator = GetComponent<Animator>();
    }
}
```

```
void Update()
{
    if (isGameOver)
    {
        return;
    }
    float moveInput = Input.GetAxis("Horizontal");
    playerRigidbody.velocity = new Vector2(moveInput * moveSpeed, playerRigidbody.velocity.y);
    if (moveInput != 0)
    {
        animator.SetBool("isRunning", true);
    }
    else
    {
        animator.SetBool("isRunning", false);
    }

    if (Input.GetButtonDown("Jump") && isGrounded)
    {
        playerRigidbody.velocity = new Vector2(playerRigidbody.velocity.x, jumpForce);
        animator.SetTrigger("isJumping");
    }
    FlipSprite();
}

참조 1개
void FlipSprite()
{
    transform.localScale = new Vector2(Mathf.Sign(playerRigidbody.velocity.x), 1f);
}
```


3. 주요 코드

PlayerController Script

Unity 메시지 | 참조 0개

```
void OnCollisionEnter2D(Collision2D collision)
{
    if (collision.gameObject.CompareTag("Ground"))
    {
        isGrounded = true;
        animator.SetBool("isJumping", false);
    }
}
```

Unity 메시지 | 참조 0개

```
void OnCollisionExit2D(Collision2D collision)
{
    if (collision.gameObject.CompareTag("Ground"))
    {
        isGrounded = false;
        animator.SetBool("isJumping", true);
    }
}
```

```
void OnTriggerEnter2D(Collider2D other)
{
    if (other.gameObject.name == "DeadZone")
    {
        Die();
    }
    else if (other.gameObject.name == ("Spike"))
    {
        Die();
    }
}
참조 2개
void Die()
{
    animator.SetTrigger("isDead");

    isGameOver = true;

    GameManager.Instance.OnPlayerDead();
}
```

3. 주요 코드

GameManager

```
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;
// Unity 스크립트(자산 참조 3개) | 참조 2개
public class GameManager : MonoBehaviour
{
    public static GameManager Instance;
    public bool isGameOver = false;
    public GameObject gameOverUI;
    // Start is called before the first frame update
    // Unity 메시지 | 참조 0개
    void Awake()
    {
        if (Instance == null)
        {
            Instance = this;
        }
        else
        {
            Debug.LogWarning("썬에 두 개 이상의 게임 매니저가 존재합니다!");
            Destroy(gameObject);
        }
    }
}
```

```
void Update()
{
    if (Input.GetKeyDown(KeyCode.R))
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().name);
    }
}
```

참조 1개

```
public void OnPlayerDead()
{
    isGameOver = true;
    gameOverUI.SetActive(true);
}
```

3. 주요 코드

NextScene Script

Imported Object



Next Scene Level 1 (Mono Script)

Assembly Information

Filename	Assembly-CSharp.dll
----------	---------------------

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

```
public class NextSceneLevel1 : MonoBehaviour
{
    void OnTriggerEnter2D(Collider2D other)
    {
        if (other.gameObject.name == ("Player"))
        {
            LoadNextLevel();
        }
    }
    void LoadNextLevel()
    {
        SceneManager.LoadScene("Level2");
    }
}
```



Next Scene Level 2 (Mono Script)

Assembly Information

Filename	Assembly-CSharp.dll
----------	---------------------

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

```
public class NextSceneLevel2 : MonoBehaviour
{
    void OnTriggerEnter2D(Collider2D other)
    {
        if (other.gameObject.name == ("Player"))
        {
            LoadNextLevel();
        }
    }
    void LoadNextLevel()
    {
        SceneManager.LoadScene("Level3");
    }
}
```

Imported Object



Next Scene Level 3 (Mono Script)

Assembly Information

Filename	Assembly-CSharp.dll
----------	---------------------

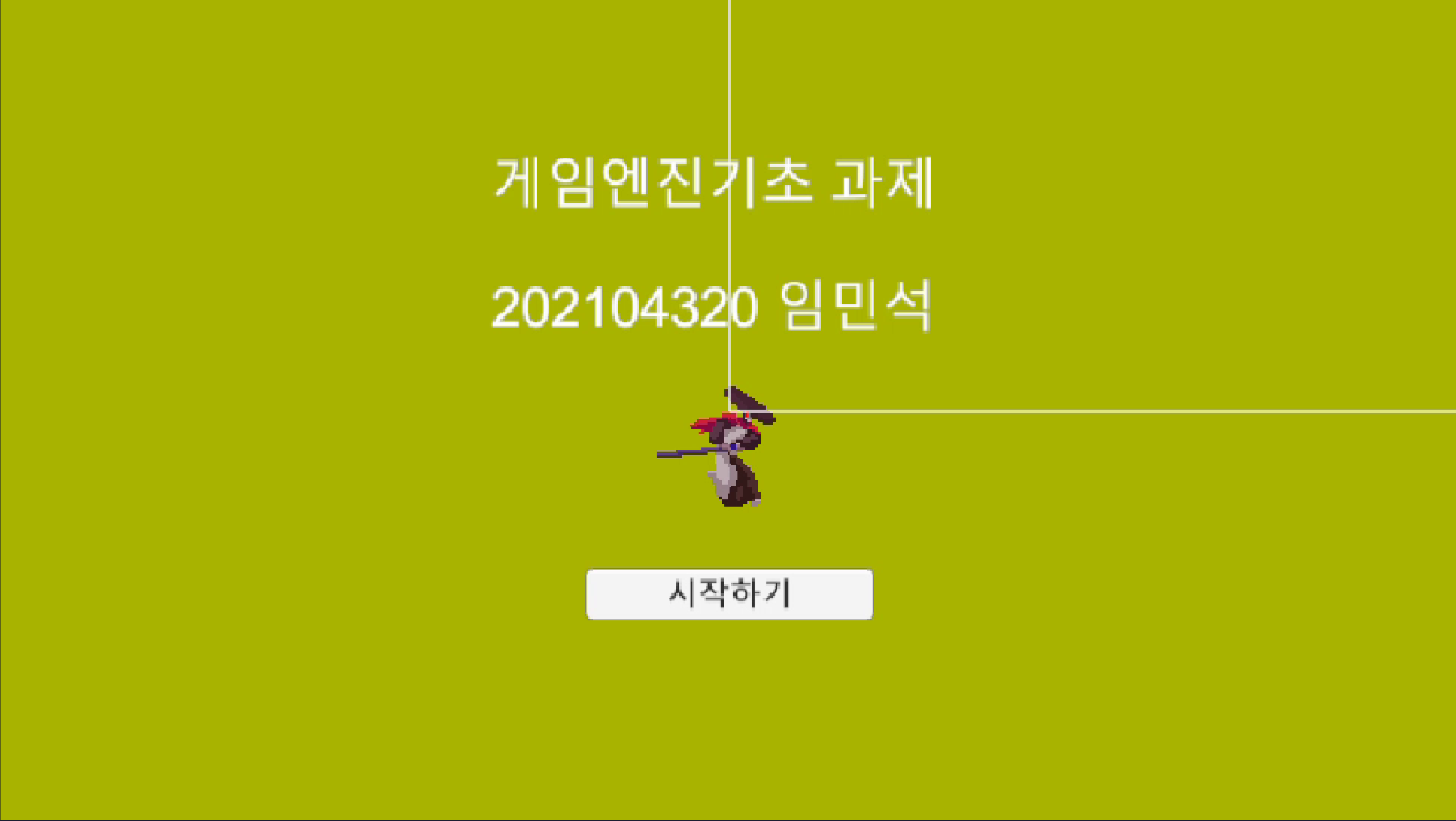
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
```

```
public class NextSceneLevel3 : MonoBehaviour
{
    void OnTriggerEnter2D(Collider2D other)
    {
        if (other.gameObject.name == ("Player"))
        {
            LoadNextLevel();
        }
    }
    void LoadNextLevel()
    {
        SceneManager.LoadScene("Ending");
    }
}
```

9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
24
25
26



4. 게임 플레이



9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
24
25
26



Q&A ____

감사합니다!!