Unity를 이용한 2D 게임프로그래밍

Lecture 9
간단한 아케이드 게임
적의 동작과 각종 효과

강영민

동명대학교 게임공학과

학습목표

- 적의 슈팅 제어
- 각종 이펙트 추가
  - 입자시스템
  - 음향 효과

점수 관리 방법

- 적이 파괴되면 ➔ 점수가 증가

고칠 것: Health.cs

새로 만들 것: ScoreKeeper.cs

Health.cs

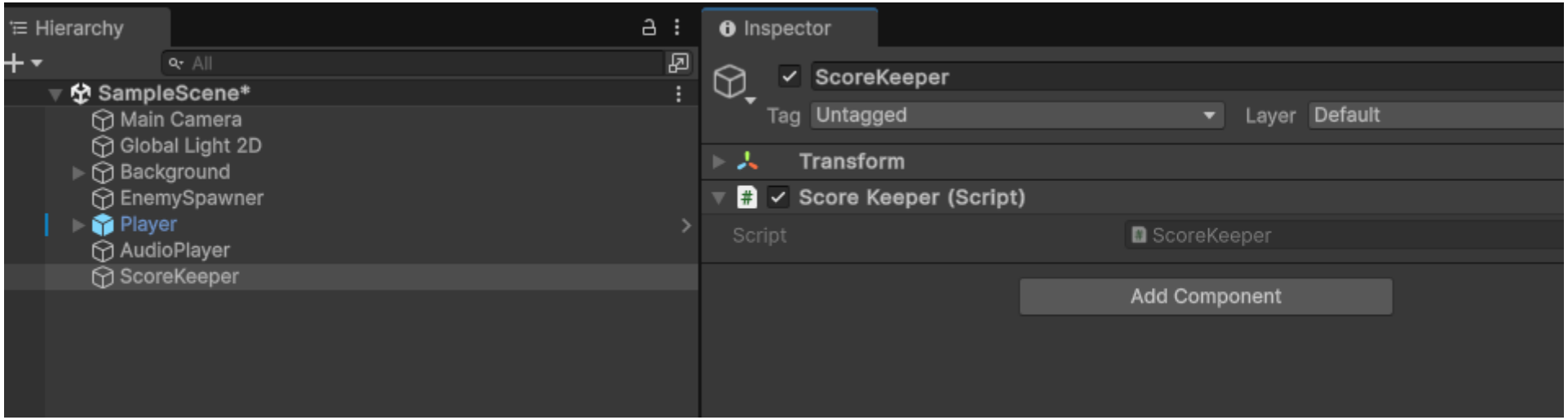
- 적인지 플레이언지 구분
- 적인데 파괴되면 스코어 증가

ScoreKeeper.cs

- 스코어 관리
- 스코어 변경 메소드 구현

ScoreKeeper

- 객체 만들고 스크립트 연결



점수 관리

- Getter/Modifier

```
public class ScoreKeeper : MonoBehaviour
{
    4 references
    int Score = 50;

    0 references
    public int GetScore()
    {
        return Score;
    }

    0 references
    public void ChangeScore(int value)
    {
        Score += value;
        if (Score < 0)
        {
            Score = 0; // Prevent negative scores
        }
    }
}
```

Health.cs에서 점수 다루기

```
1 reference
[SerializeField] bool isPlayer = false;
3 references
ShakeCam shakeCam;

3 references
AudioPlayer audioPlayer;
1 reference
ScoreKeeper scoreKeeper;

0 references
void Awake()
{
    shakeCam = Camera.main.GetComponent<ShakeCam>();
    audioPlayer = FindAnyObjectByType<AudioPlayer>();
    scoreKeeper = FindAnyObjectByType<ScoreKeeper>();
}
```

Health.cs에서 점수 다루기

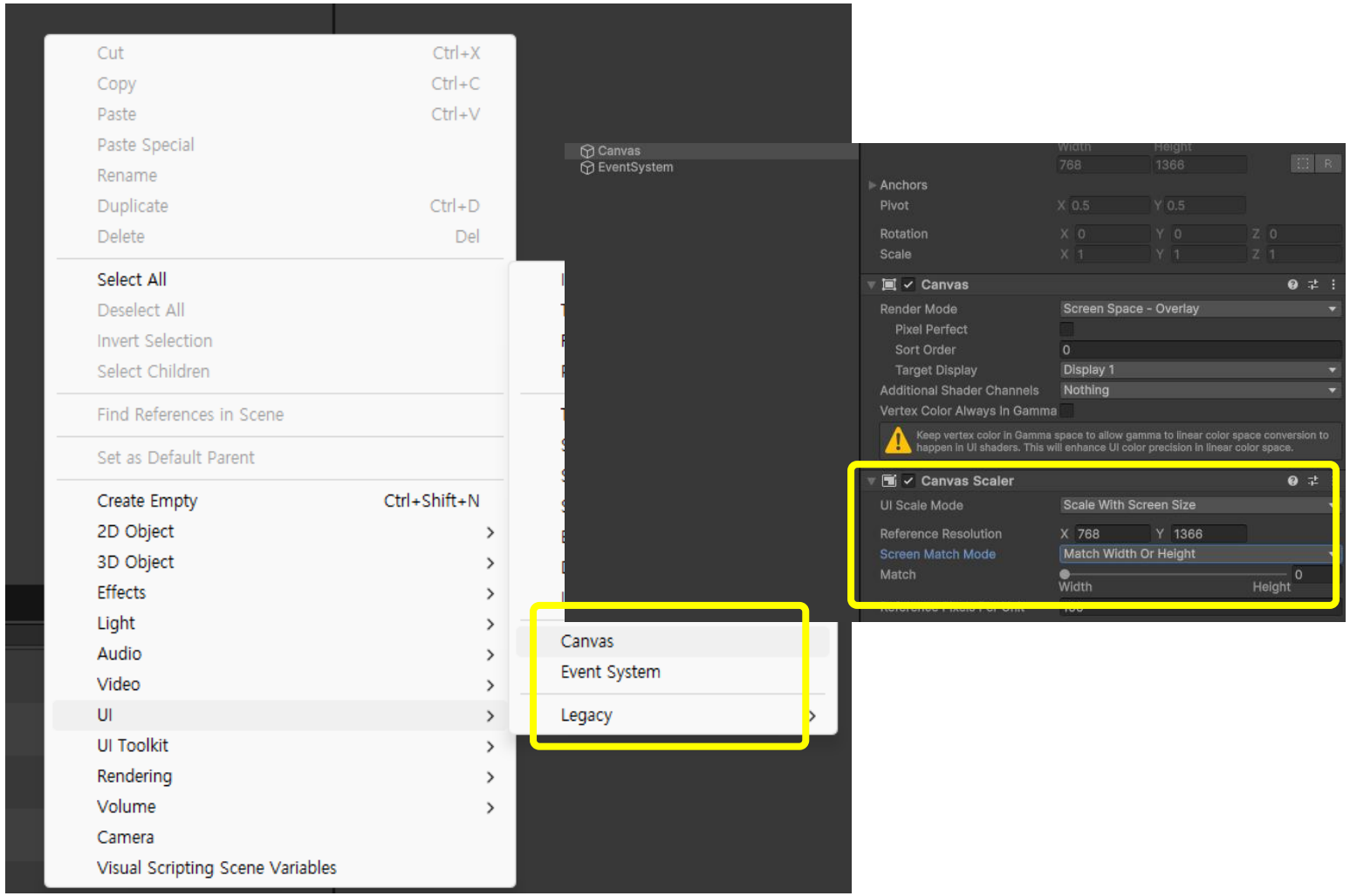
```
void TakeDamage(int damage)
{
    health -= damage;

    ShakeCamera();
    if (health <= 0)
    {
        if (explosion != null)
        {
            ExplosionEffect();
        }
        Die();

    }
    else
    {
        DamageEffect();
    }
}

1 reference
void Die()
{
    if (scoreKeeper != null && !isPlayer)
    {
        scoreKeeper.ChangeScore(30);
    }
    Destroy(gameObject);
    Debug.Log($"Enemy Destroyed {scoreKeeper.GetScore()}");
}
```

UI

- Health
  - Slider
  - Text
  - Sprite Change
  - ...
- Score
  - text

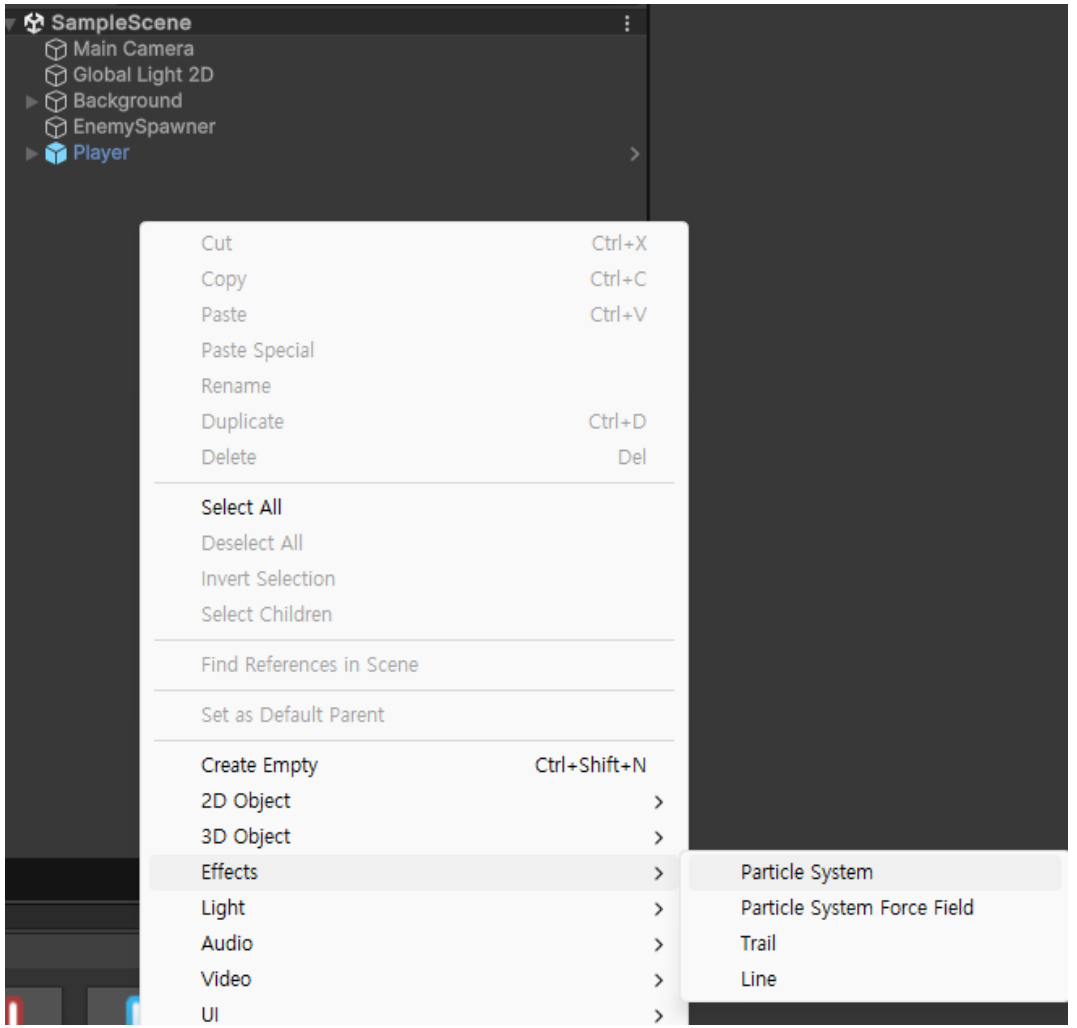
UI는 캔버스를 필요로 한다

- Canvas
  - Canvas Scaler 조정



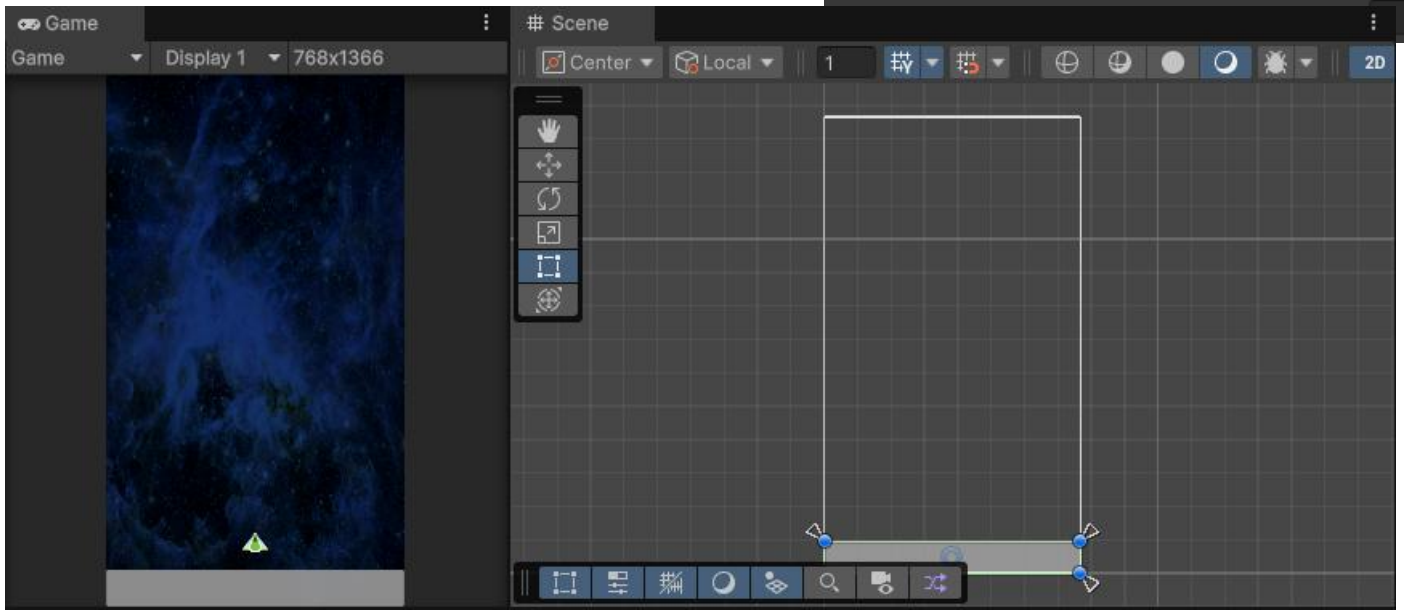
폭발 효과 – Particle System

- Particle System 생성



Anchor 설정

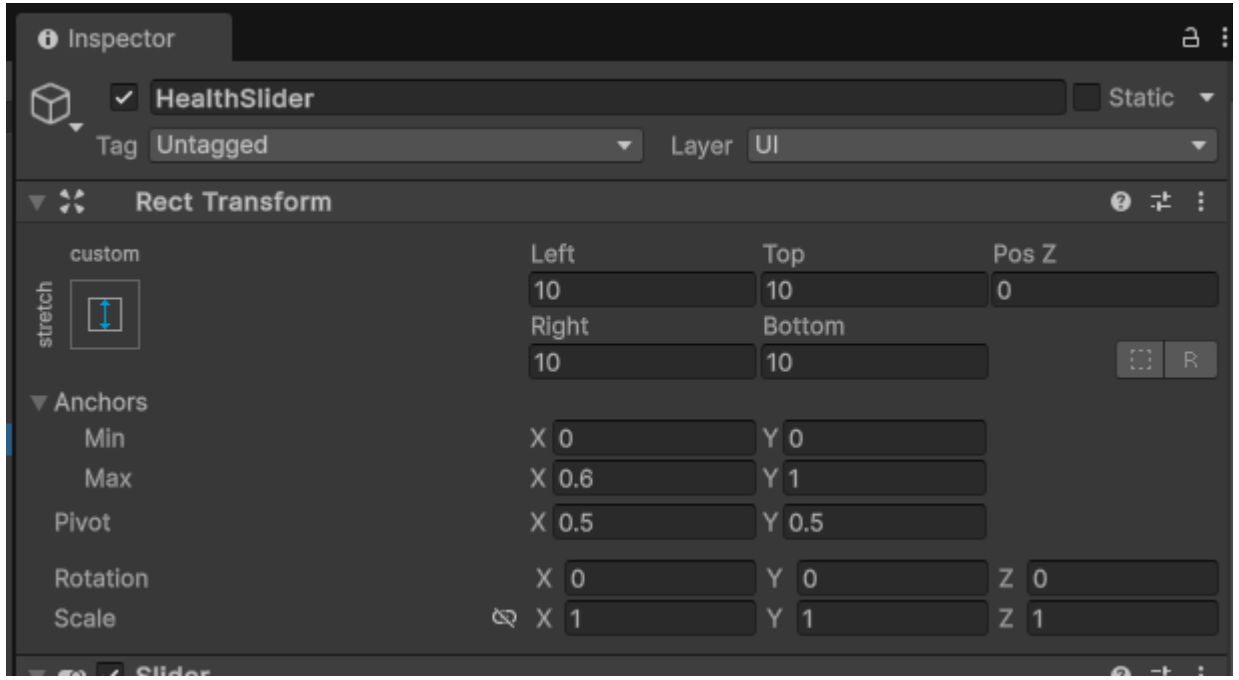
- Panel
  - Anchor
- Top ➔ 0

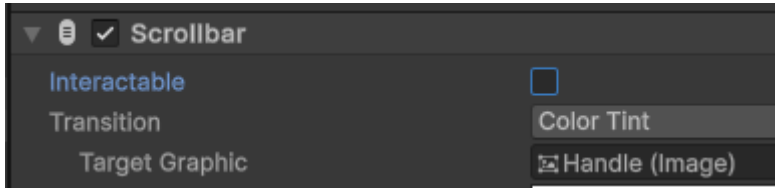


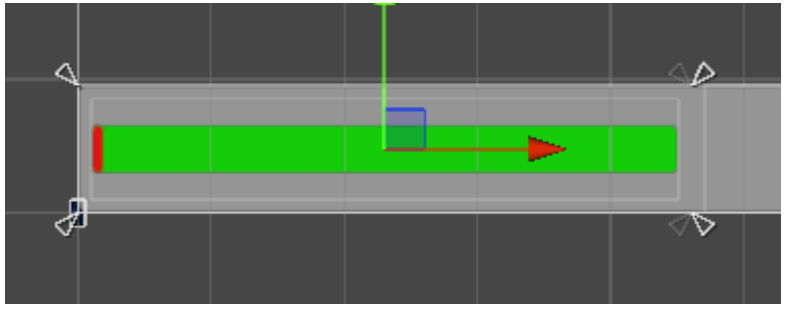
패널에 슬라이더 장착


Canvas
Panel
HealthSlider
Background
Fill Area
Fill
Handle Slide Area
Handle

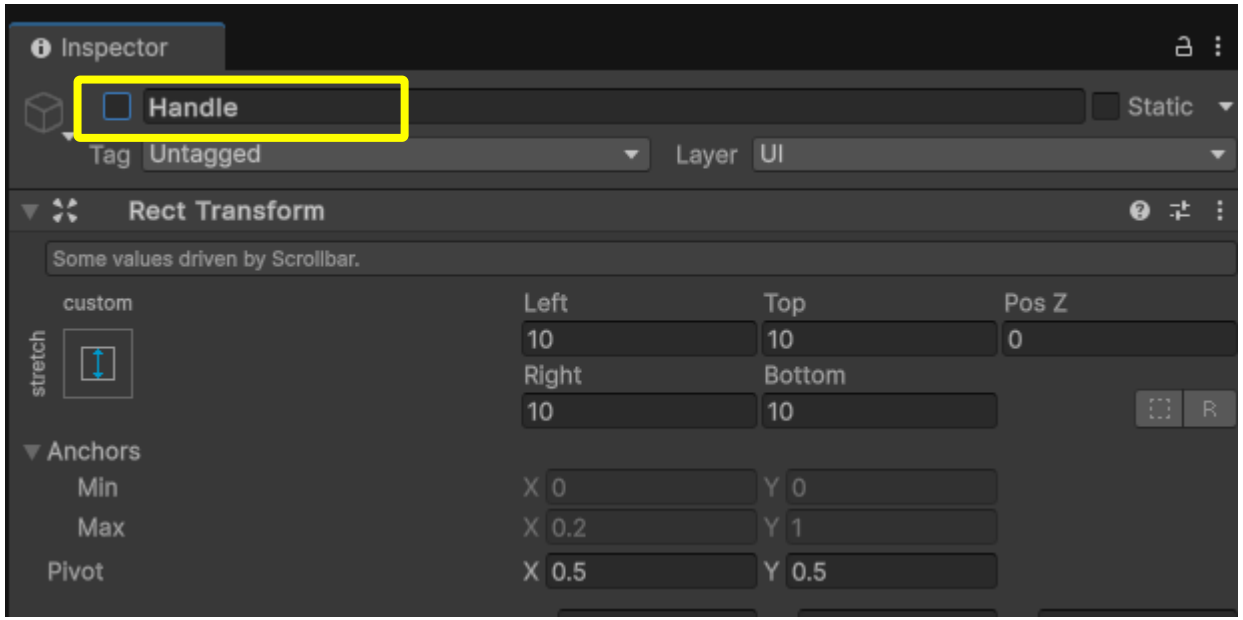
슬라이더 크기 조정 / interactable: false



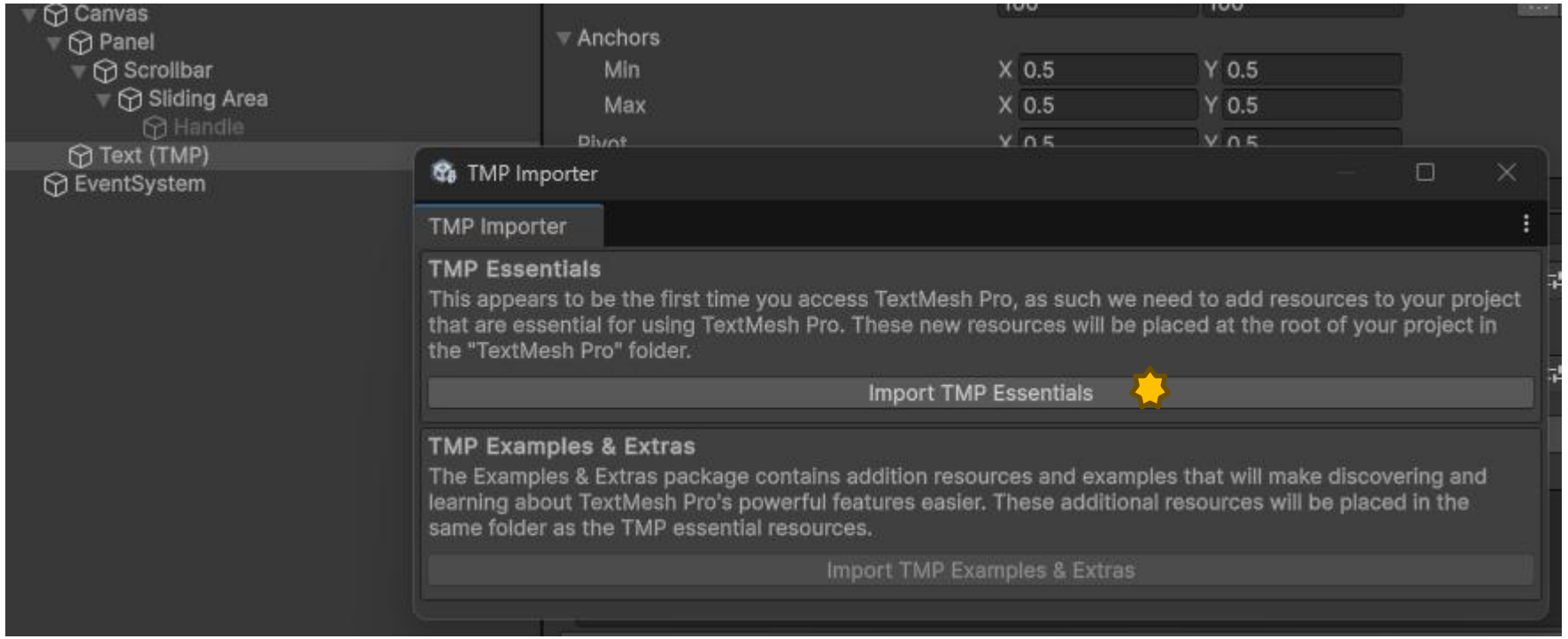




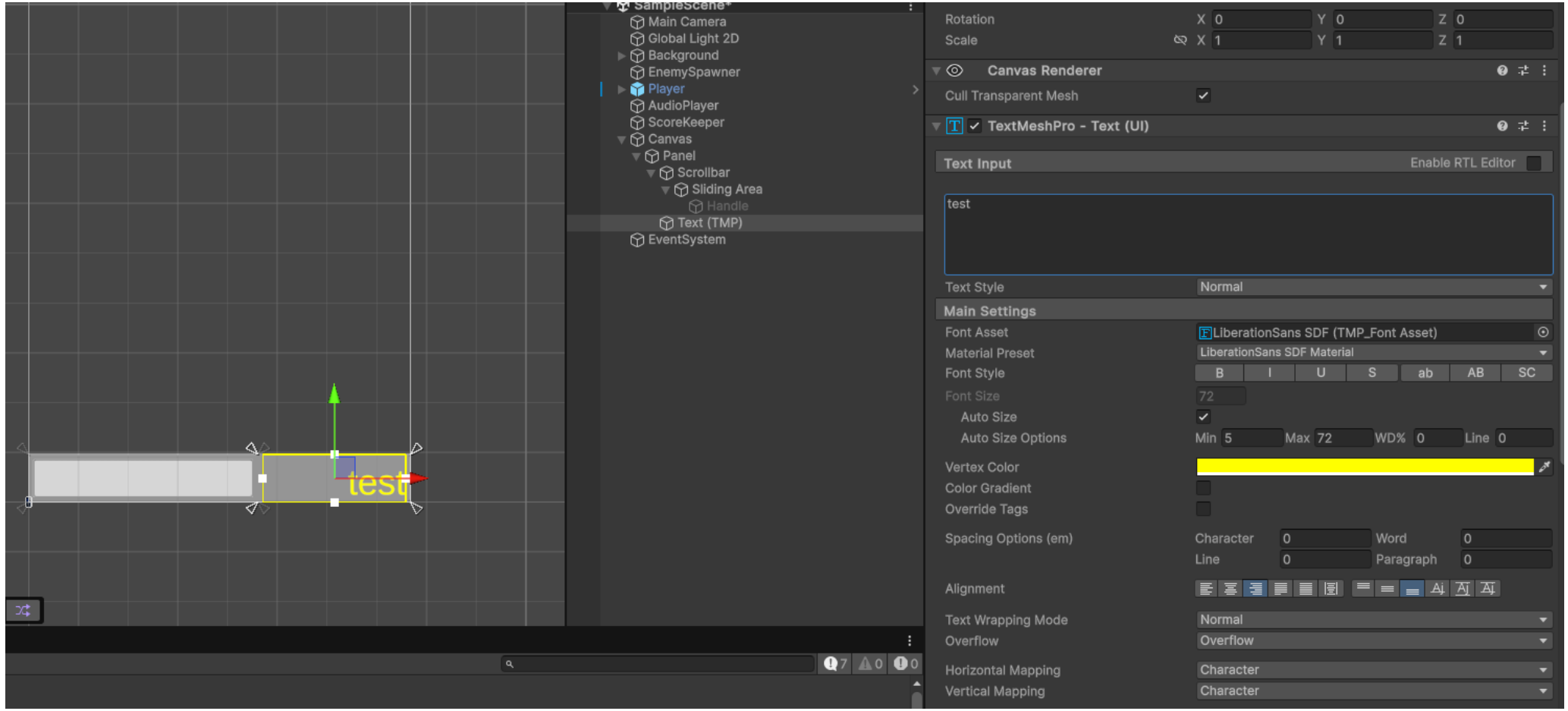
Handle은 Disable



Text Mesh Pro도 추가 (Score 표시)

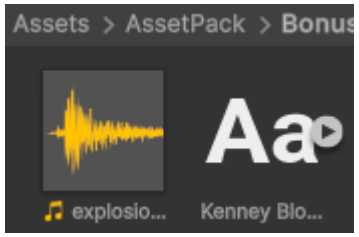


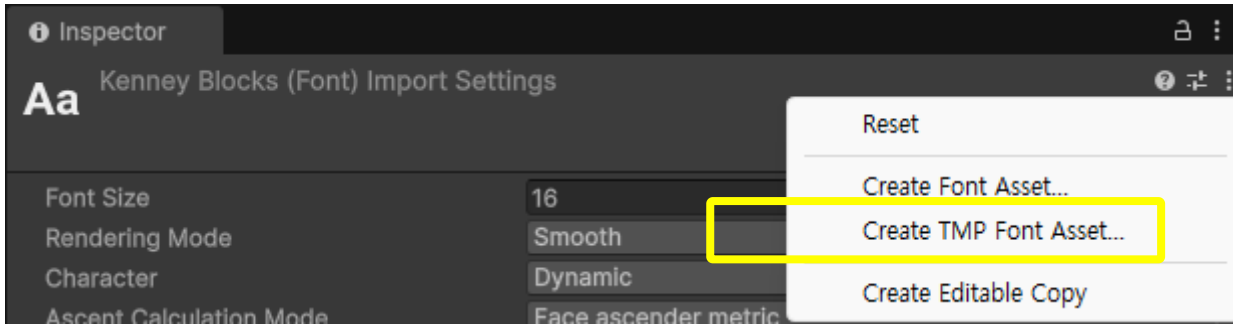
슬라이더와 텍스트 영역의 Anchor 설정

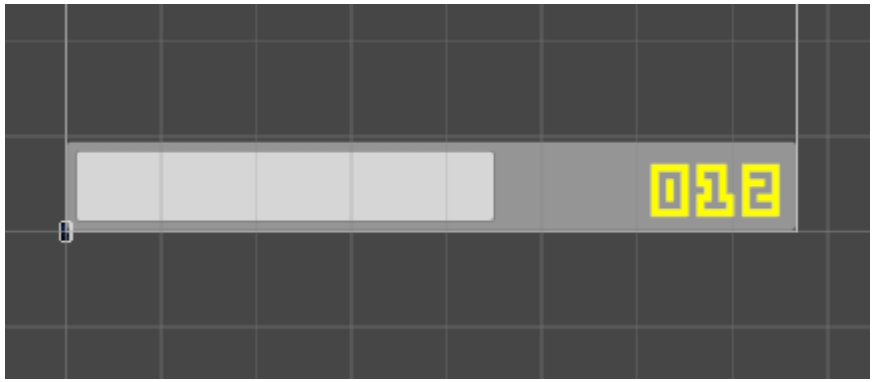


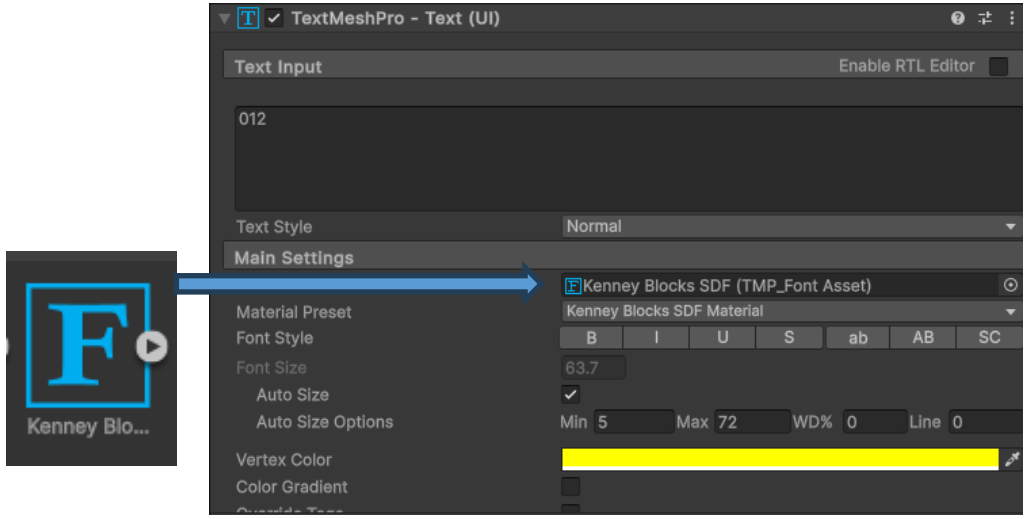
Font를 바꾸어 보자

Kenney.nl에서 폰트 다운로드









정보를 출력하는 스크립트 작성

- 주요 기능
  - 정보 출력을 위한 스크립트 작성 UIDisplay.cs
  - 스크립트를 캔버스에 연결

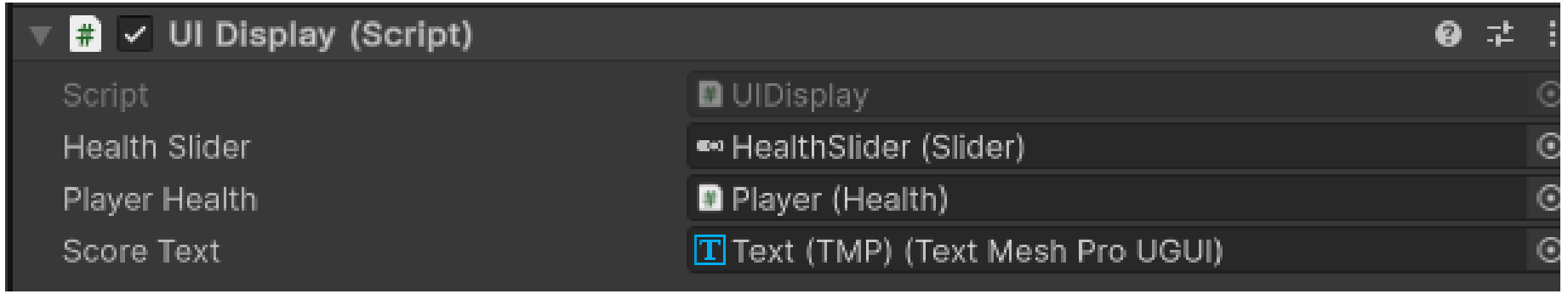
```
using UnityEngine.UI;
using TMPro;
```

  - 스크립트를 이용하여 정보에 맞춰 UI 객체 업데이트
    - Score from ScoreKeeper.cs
    - Health from Health.cs

UIDisplay.cs

```
using UnityEngine;
using UnityEngine.UI;
using TMPro;


0 references
public class UIDisplay : MonoBehaviour
{
    2 references
    [SerializeField] Slider healthSlider;
    3 references
    [SerializeField] Health playerHealth; // Reference to the player's health script
    3 references
    [SerializeField] TextMeshProUGUI scoreText;
```

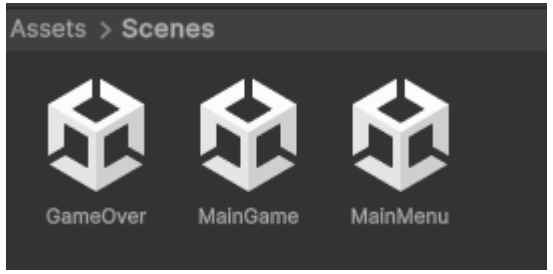
SerializeField 연결

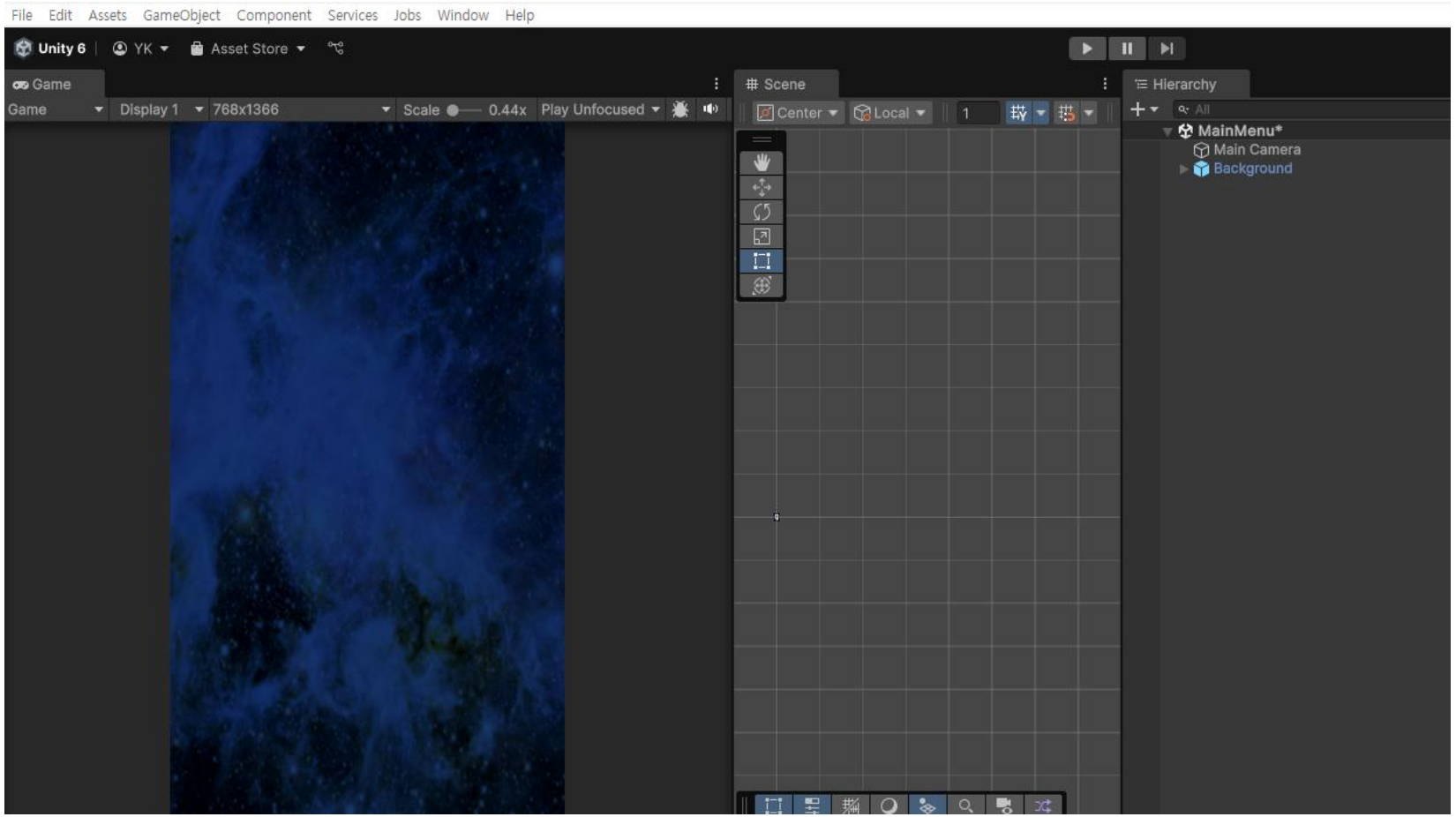


```
ScoreKeeper scoreKeeper;

// Start is called once before the first execution of Update after the MonoBehaviour is created
0 references
void Start()
{
    scoreKeeper = FindAnyObjectByType<ScoreKeeper>();

    if (scoreText != null)
    {
        scoreText.text = "Score: " + scoreKeeper.GetScore();
    }

    healthSlider.maxValue = playerHealth.GetHealth(); // Assuming GetMaxHealth() is a method in Health script
}

// Update is called once per frame
0 references
void Update()
{
    healthSlider.value = playerHealth.GetHealth(); // Update the health slider value
    scoreText.text = scoreKeeper.GetScore().ToString(); // Update the score text
    if (playerHealth.GetHealth() <= 0)
    {
        // Optionally, you can handle the game over state here
        // For example, you might want to show a game over screen or stop the game
        Debug.Log("Game Over");
    }
}
```

뭐가 남았지?

More Scenes

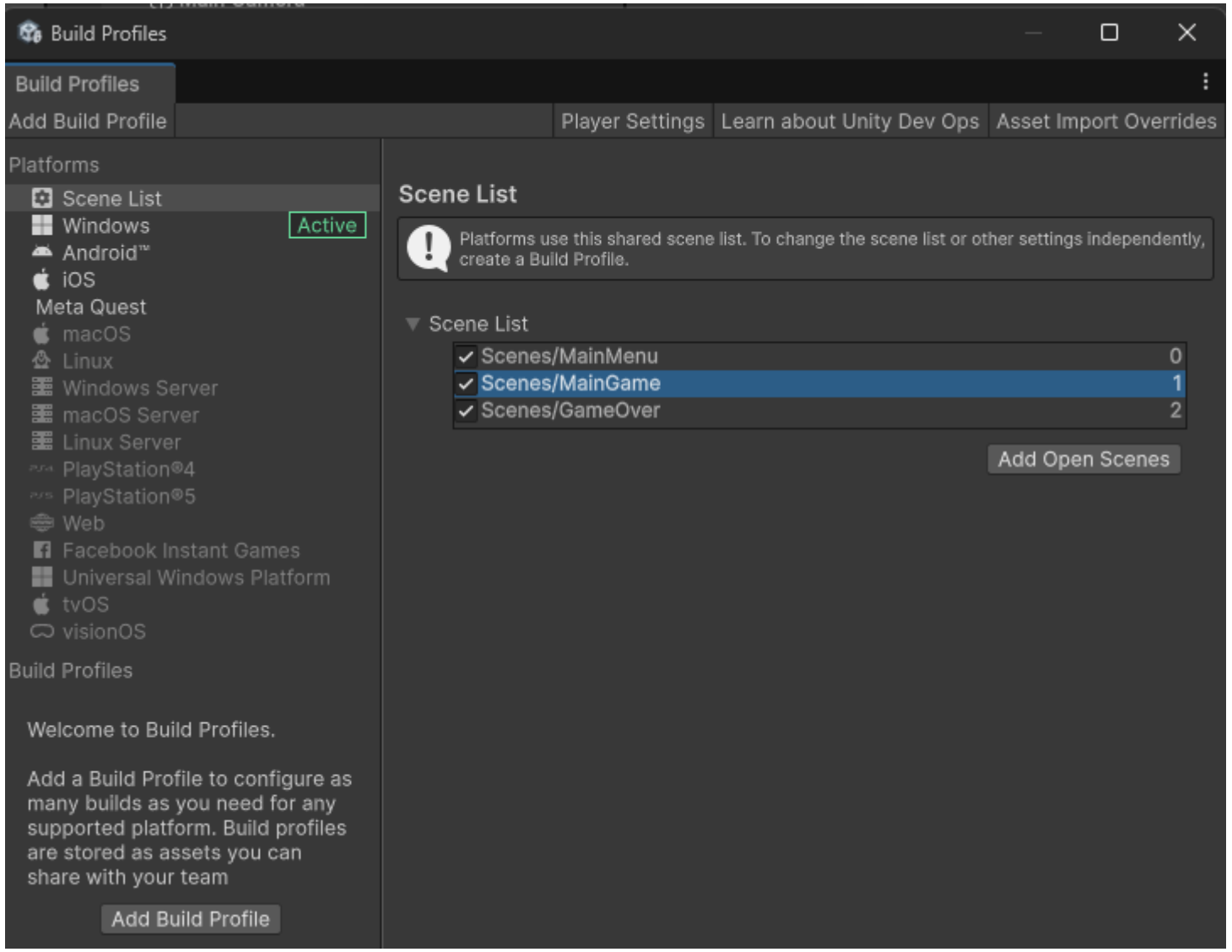




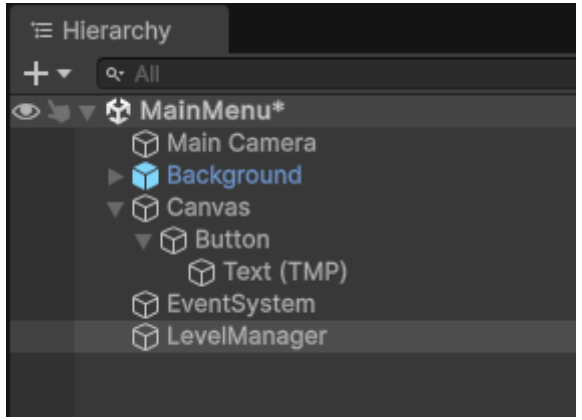
Background를 prefab해서 활용

Build Profiles

Scene 추가

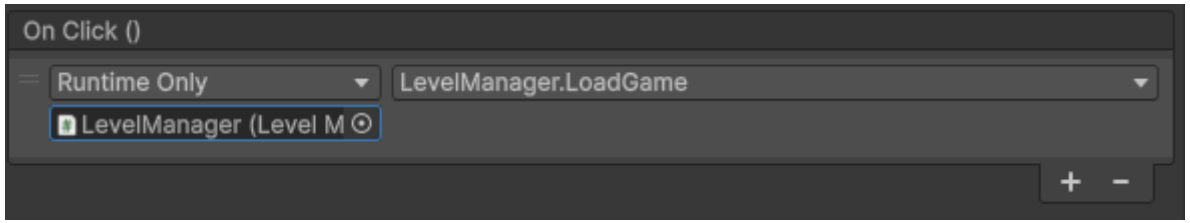


Level Manager




```
using UnityEngine;
using UnityEngine.SceneManagement;
0 references
public class LevelManager : MonoBehaviour
{
    0 references
    public void LoadGame()
    {
        SceneManager.LoadScene("MainGame");
    }
    0 references
    public void LoadMainMenu()
    {
        SceneManager.LoadScene("MainMenu");
    }
    0 references
    public void LoadGameOver()
    {
        SceneManager.LoadScene("GameOver");
    }
    0 references
    public void QuitGame()
    {
        Application.Quit();
        Debug.Log("Game has been quit.");
    }
}
```

Button Action



전체를 완성해 볼까요?
– GameOver까지