

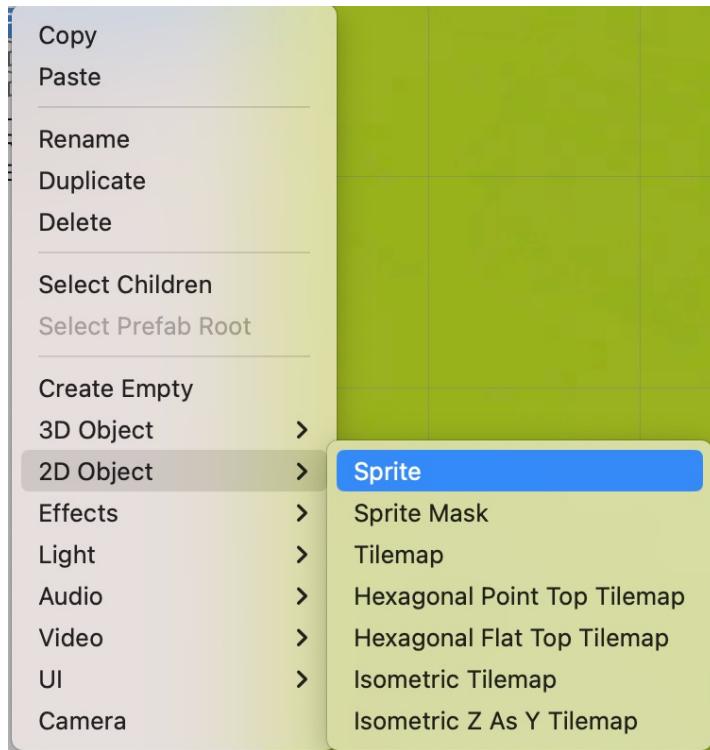
chapter06

충돌처리

1. 충돌 처리
2. 코인 만들기
3. 점수 출력

01 충돌 처리

■ Player 에 그림자 처리



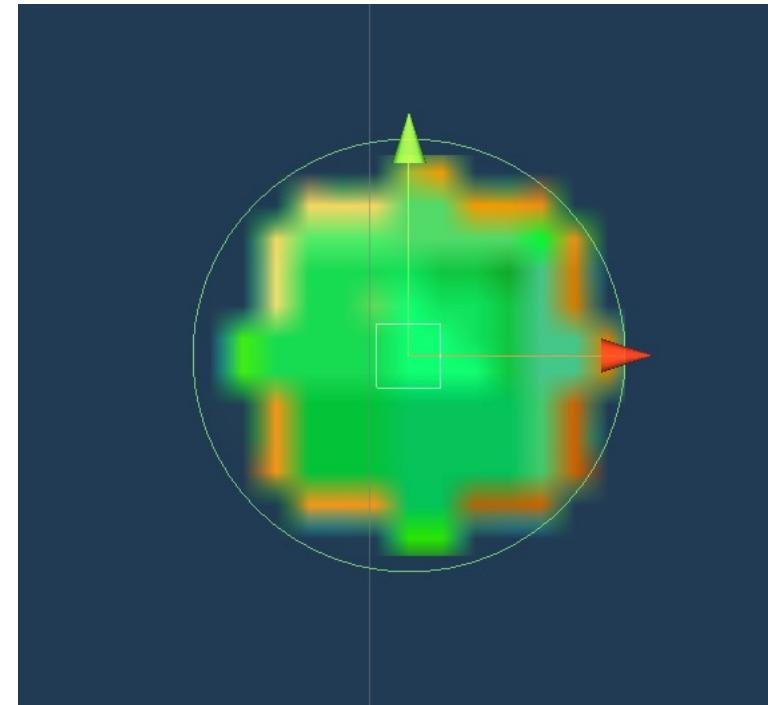
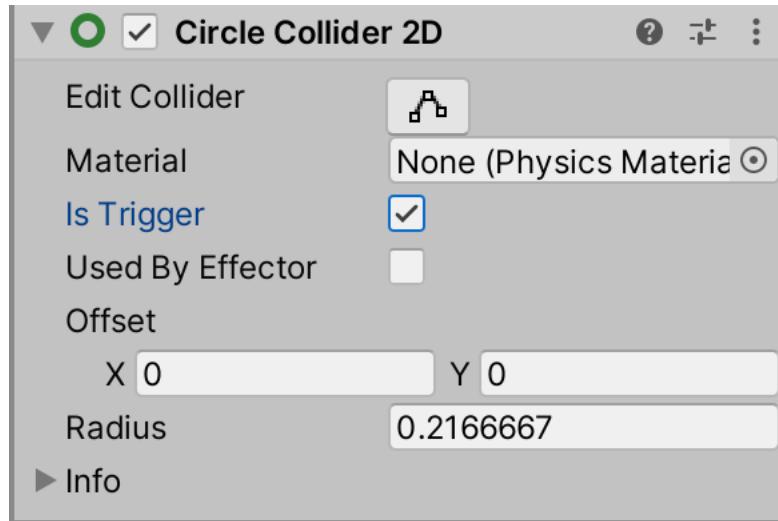
The Unity Hierarchy panel shows a scene named 'SampleScene*' containing objects: Main Camera, Background1, Background2, Player, ShootTransform, Shadow, Left Wall, Right Wall, and Enemy Spawner. The 'Shadow' object is currently selected, indicated by a blue highlight. The Inspector panel is open for the 'Shadow' object, showing the 'Sprite Renderer' component. The 'Sprite' dropdown is set to 'Knob', and the 'Material' dropdown is set to 'Sprites-Default'. Other settings include Color (dark gray), Draw Mode (Simple), Mask Interaction (None), Sprite Sort Point (Center), and Sorting Layer (Default).



01 충돌 처리



■ Weapon 에 Circle Collider2D 추가, Is Trigger 체크



01 충돌 처리

- Enemy에 미사일이 닿았을때? -> 체력 저하 -> 체력 고갈(사망)
- Enemy에 Player가 닿았을때?

탐색기 ... C# Enemy.cs C# Wepon.cs X

Assets > Scripts > C# Wepon.cs

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class Wepon : MonoBehaviour
6  {
7      [SerializeField]
8      private float moveSpeed = 10f;
9      public float damage = 1f;
10
11     void Start()
12     {
13         Destroy(gameObject, 1f); // 1초 후 소멸 예약
14     }
15     // Update is called once per frame
16     void Update()
17     {
18         transform.position += Vector3.up * moveSpeed * Time.deltaTime;
19     }
20 }
21

```

01 충돌 처리

■ Enemy에 미사일 ■ Enemy에 Player

탐색기 ...

- 열려 있는 편집기
 - C# Wepon.cs Assets/Scripts
 - X C# Enemy.cs Assets/Scripts
- STARSHOOTER
 - .vscode
 - Assets
 - Prefabs
 - Scenes
 - Scripts
 - C# Background.cs
 - C# Enemy.cs**
 - C# EnemySpawner.cs
 - C# Player.cs
 - C# Wepon.cs
 - Sprites
 - ridgidbody.cs
 - Logs
 - Packages
 - Assembly-CSharp.csproj
 - StarShoot.sln
 - StarShooter.sln

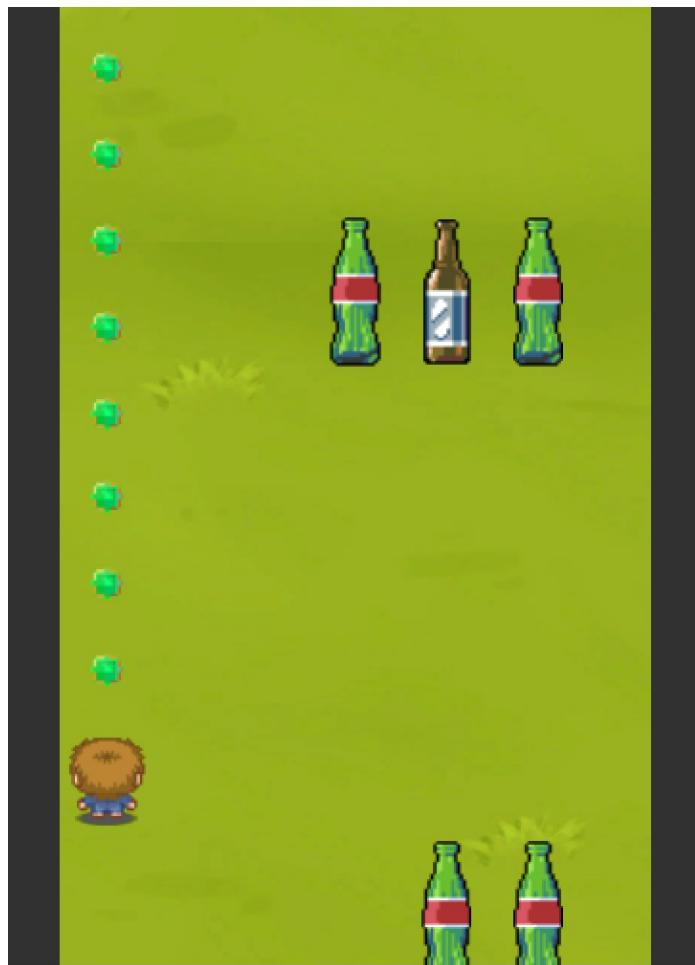
C# Wepon.cs C# Enemy.cs X

```

Assets > Scripts > C# Enemy.cs
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class Enemy : MonoBehaviour
6  {
7      [SerializeField]
8      private float moveSpeed = 10f;
9      private float minY = -7f;
10     private float hp = 1f;
11
12     public void SetMoveSpeed(float moveSpeed){
13         this.moveSpeed = moveSpeed;
14     }
15     void Update()
16     {
17         transform.position += Vector3.down * moveSpeed * Time.deltaTime;
18         //게임 플레이 화면 아래쪽을 벗어나면 없어지도록..
19         if(transform.position.y < minY){
20             Destroy(gameObject);
21         }
22     }
23
24     //충돌 감지 되면 자동으로 호출되는 메소드
25     private void OnTriggerEnter2D(Collider2D other){
26         if(other.gameObject.tag == "Wepon"){
27             Wepon wepon = other.gameObject.GetComponent<Wepon>();
28             hp -= wepon.damage;
29             if(hp <= 0){
30                 Destroy(gameObject);
31             }
32             Destroy(other.gameObject);
33         }
34     }
35 }
```

01 충돌 처리

- Play 해보기
- Player가 Enemy와 충돌해도 살아있다!



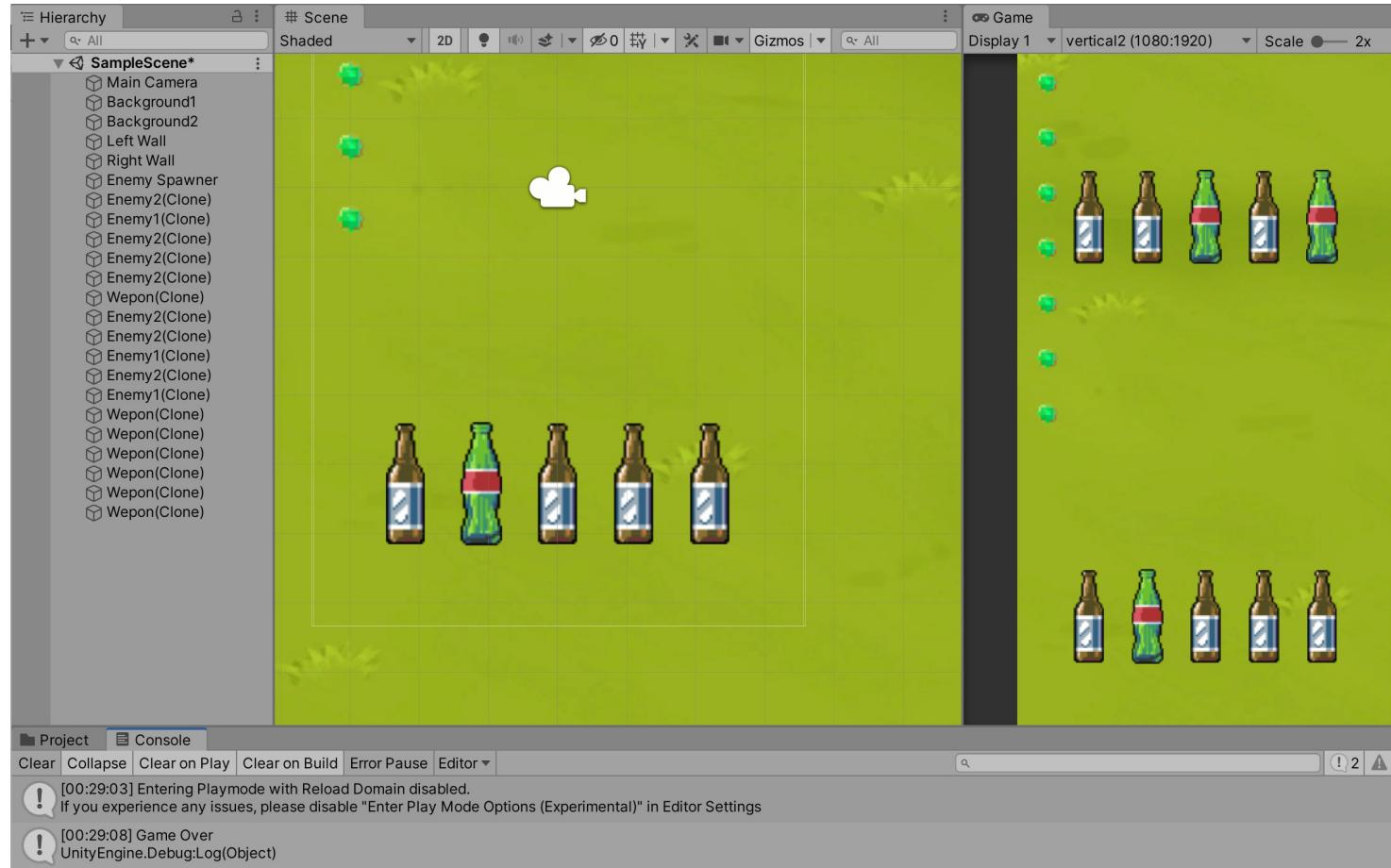
01 충돌 처리

■ Player.cs 에 OnTriggerEnter2D() 추가

```
private void OnTriggerEnter2D(Collider2D other){  
    if(other.gameObject.tag == "Enemy"){  
        Debug.Log("Game Over");  
        Destroy(gameObject);  
    }  
}
```

01 충돌 처리

■ Play 해보기



01 충돌 처리

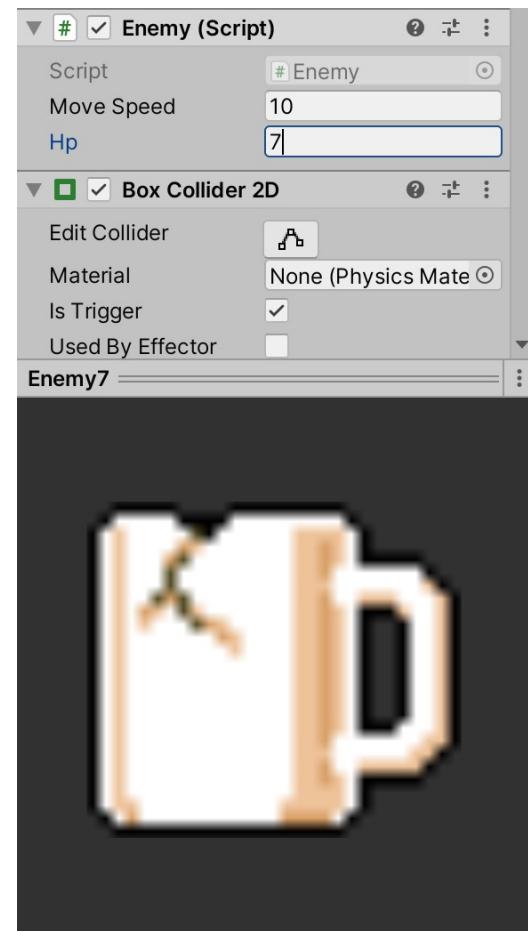
■ Enemy.cs 의 hp -> SerializeField 로

```
public class Enemy : MonoBehaviour
{
    [SerializeField]
    private float moveSpeed = 10f;
    private float minY = -7f;
    [SerializeField]
    private float hp = 1f;

    public void SetMoveSpeed(float moveSpeed){
        this.moveSpeed = moveSpeed;
    }
}
```

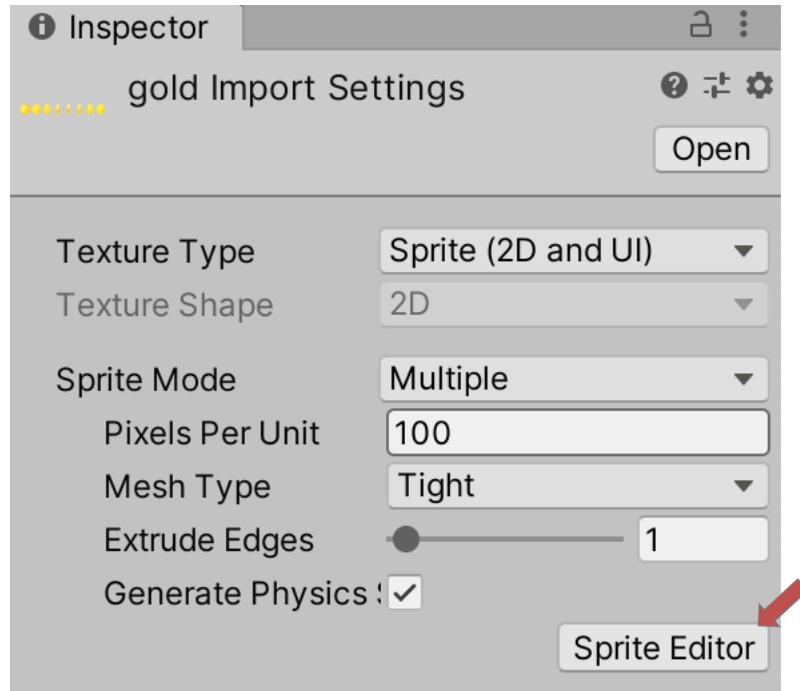
01 충돌 처리

- Enemy1 의 hp -> 1
- Enemy2 의 hp -> 2
- ...
- Enemy7 의 hp -> 7



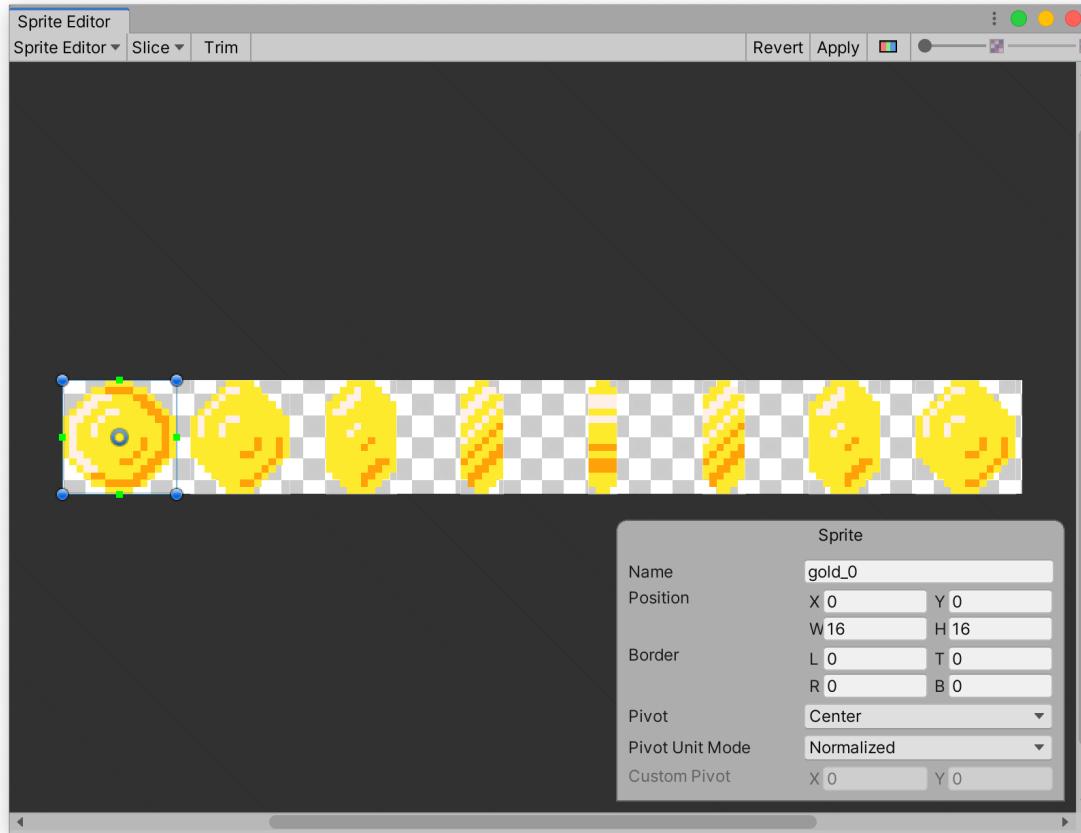
02 코인 만들기

■ 코인으로 사용할 이미지 선택 – Inspector에서 Sprite Mode -> Multiple



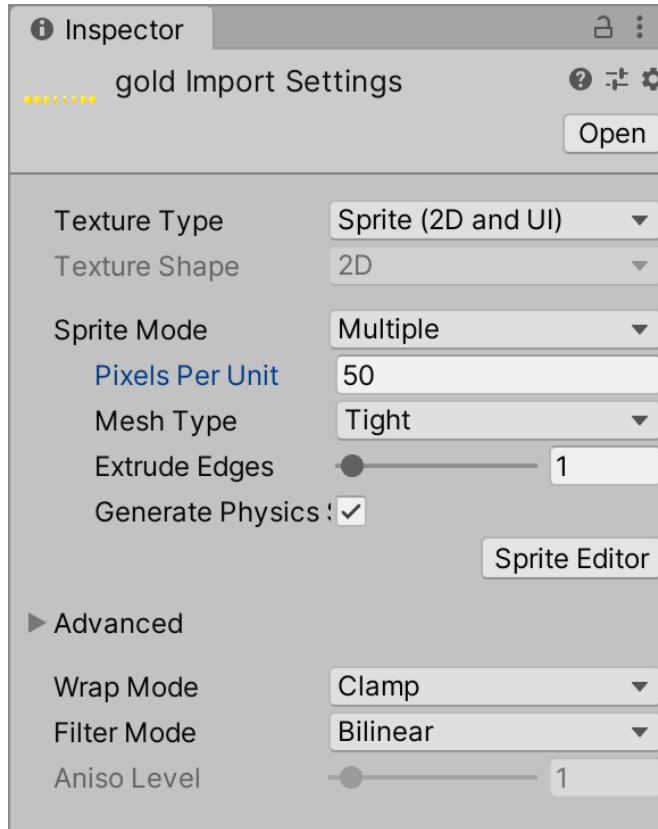
02 코인 만들기

■ 코인으로 사용할 이미지 선택 – Inspector에서 Sprite Mode -> Multiple



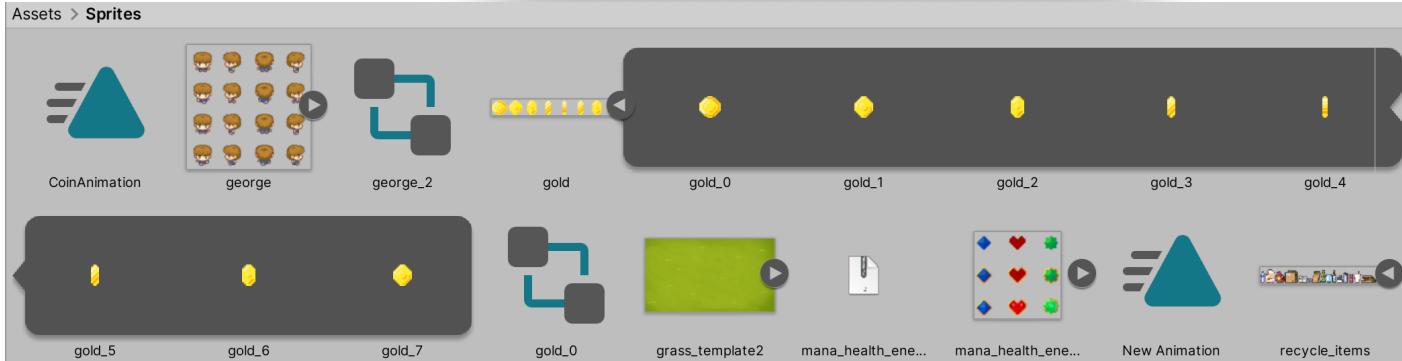
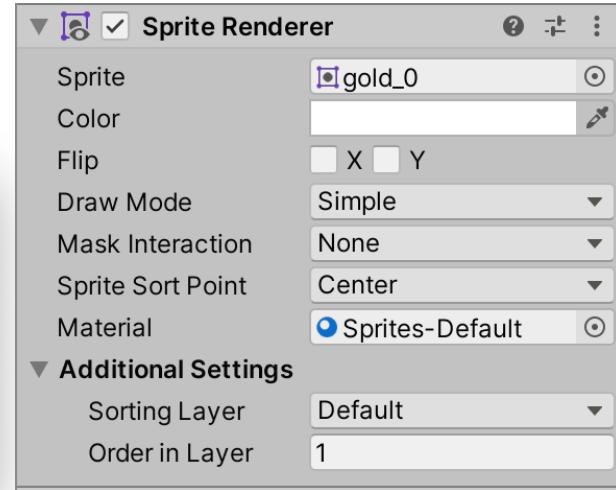
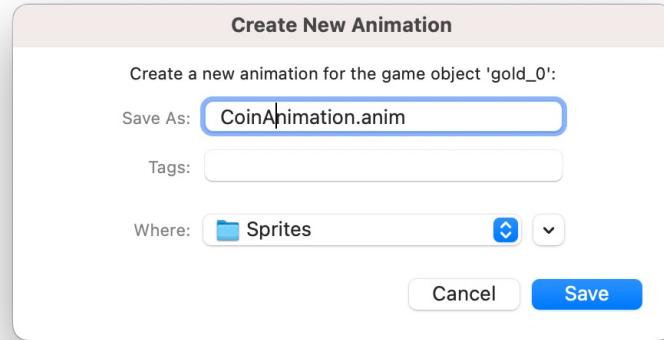
02 코인 만들기

■ Pixels Per Unit -> 50



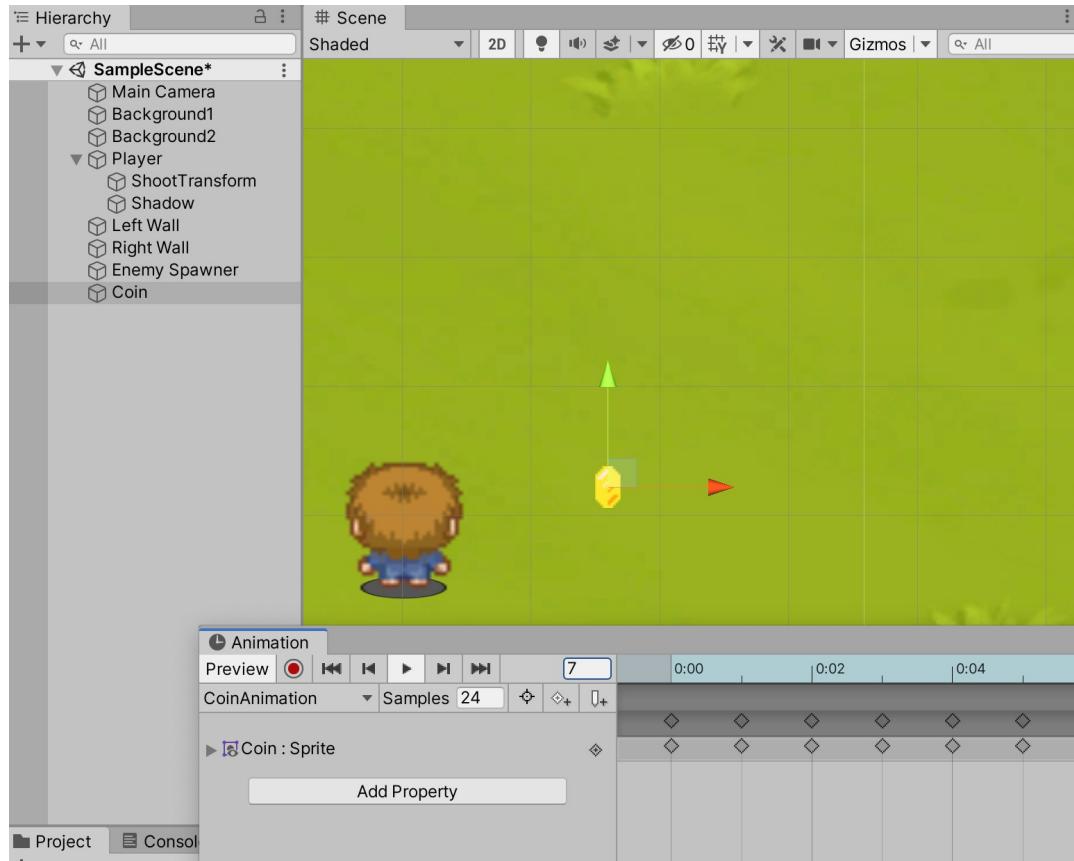
02 코인 만들기

- 씬에 코인 추가
- Coin 으로 이름 변경
- Order in Layer -> 1
- animation 적용



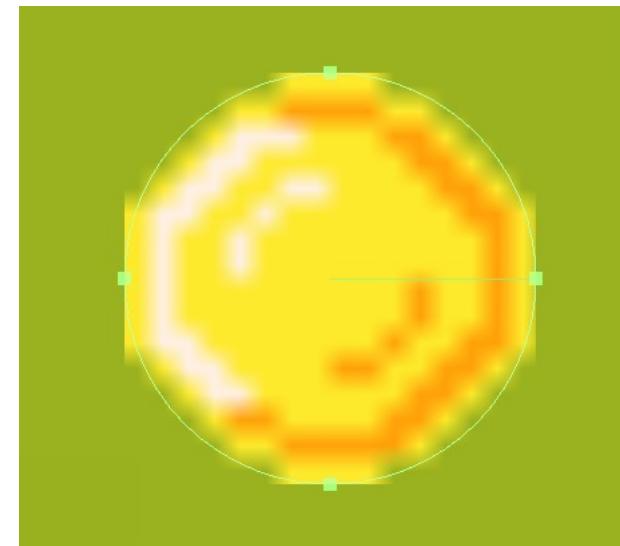
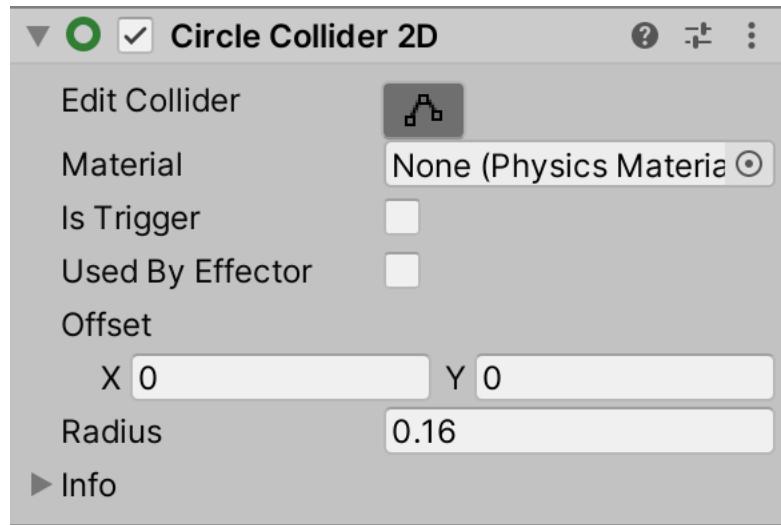
02 코인 만들기

■ animation 확인



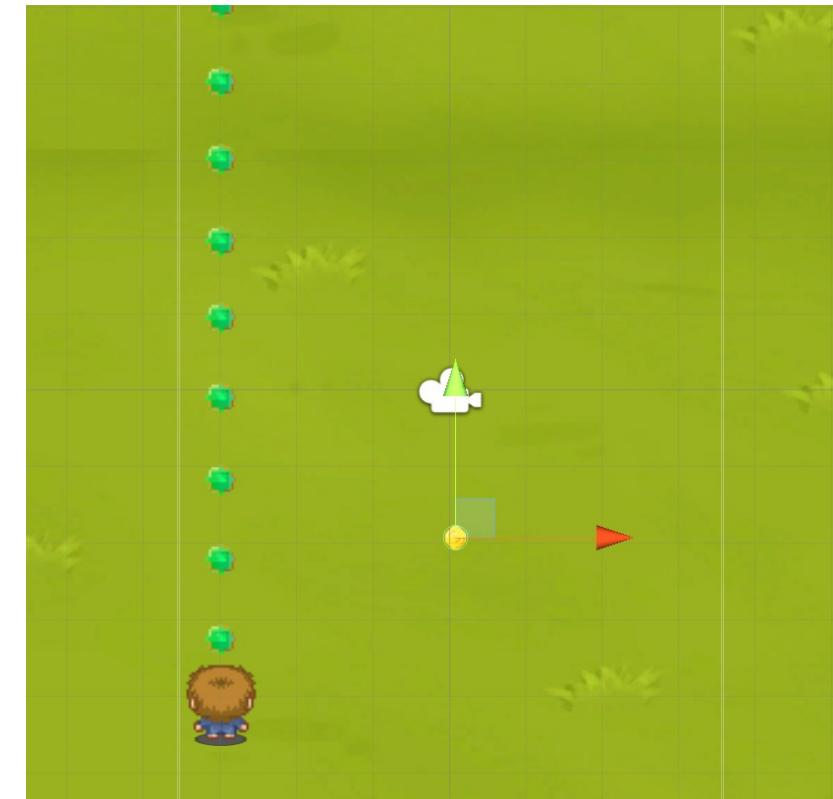
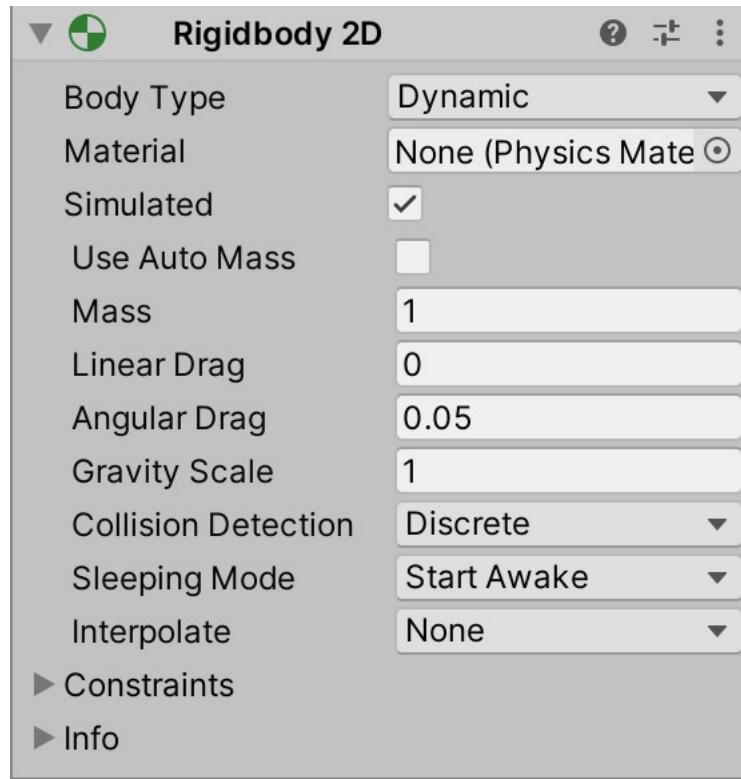
02 코인 만들기

■ Coin 에 Circle Collider2D 추가



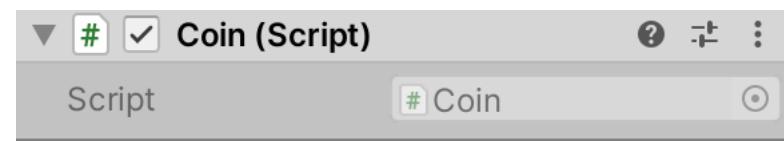
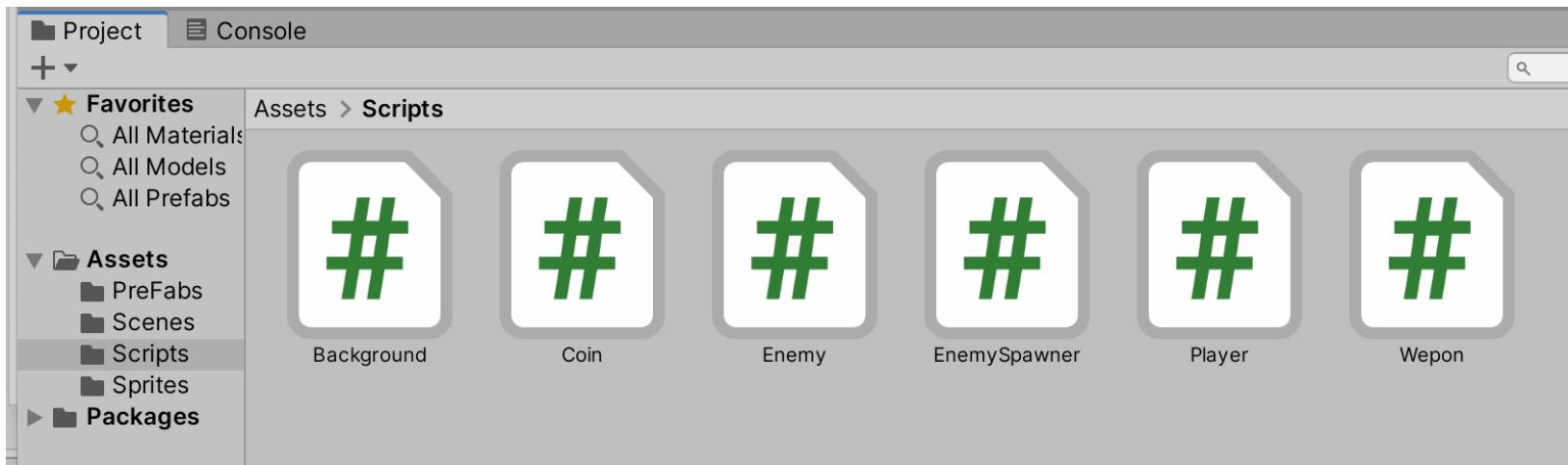
02 코인 만들기

■ Coin 에 RigidBody2D 추가 -> 이상태로 Play 시 Coin은 그냥 떨어짐



02 코인 만들기

- Coin 생성후 자유 낙하 포뮬션 적용
- Coin 에 Script 추가



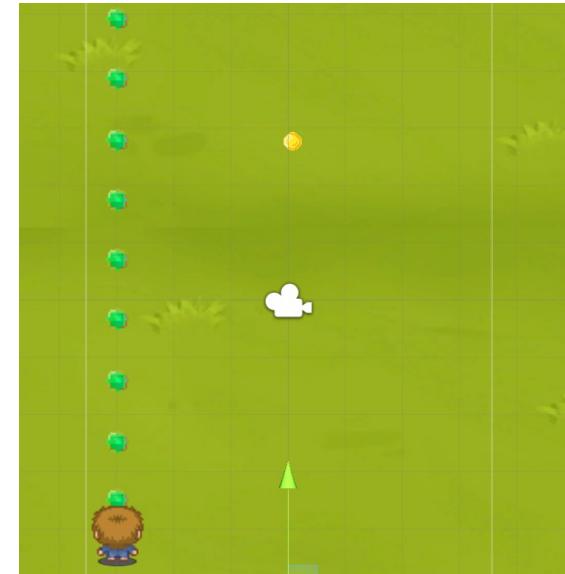
02 코인 만들기

Coin.cs 수정 및 확인

```
// Start is called before the first frame update
void Start()
{
    Jump();
}

void Jump(){
    Rigidbody2D rigidBody = GetComponent<Rigidbody2D>();
    float randomJumpForce = Random.Range(4f, 8f);
    Vector2 jumpVelocity = Vector2.up * randomJumpForce;
    rigidBody.AddForce(jumpVelocity, ForceMode2D.Impulse);
}

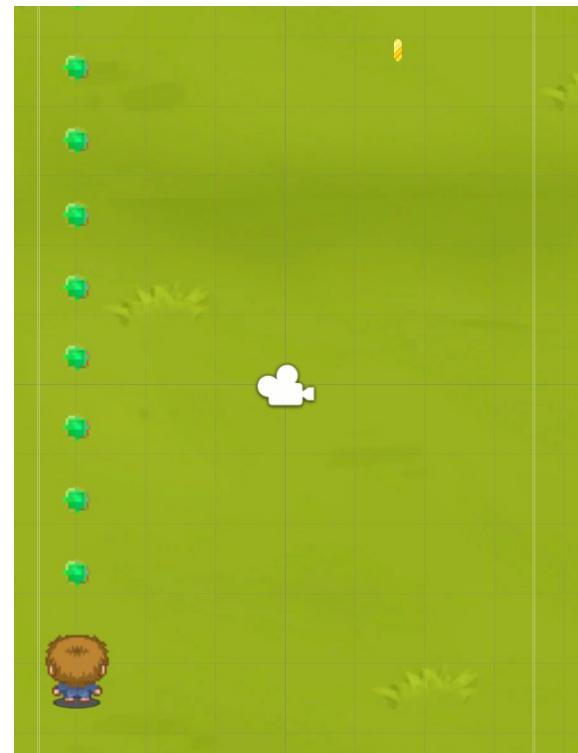
// Update is called once per frame
void Update()
{
}
```



02 코인 만들기

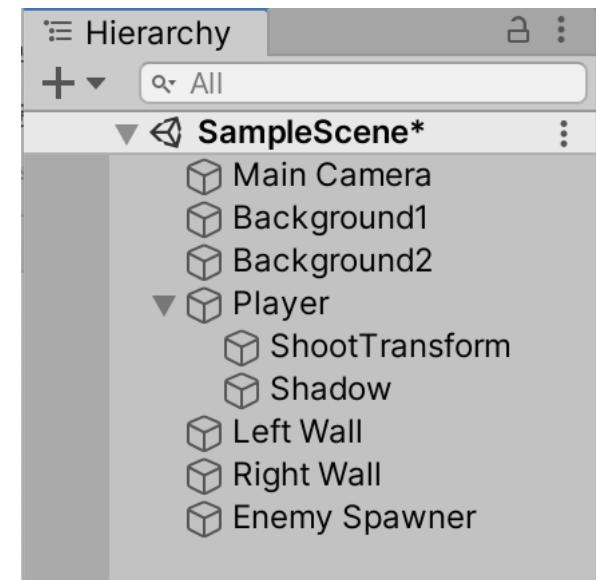
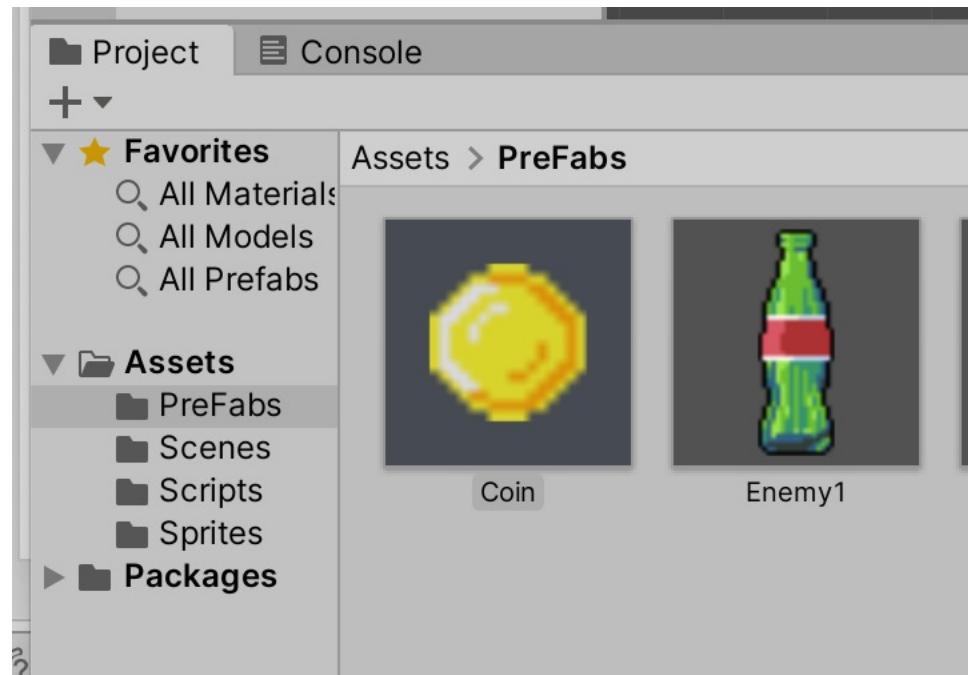
■ 좌우로 랜덤하게 낙하시키기

```
void Jump(){  
    Rigidbody2D rigidBody = GetComponent<Rigidbody2D>();  
    float randomJumpForce = Random.Range(4f, 8f);  
    Vector2 jumpVelocity = Vector2.up * randomJumpForce;  
    jumpVelocity.x = Random.Range(-2f, 2f);  
    rigidBody.AddForce(jumpVelocity, ForceMode2D.Impulse);  
}
```



02 코인 만들기

- Coin -> PreFabs
- Hierarchy 뷰에서 Coin은 삭제



02 코인 만들기

■ Enemy.cs 설정

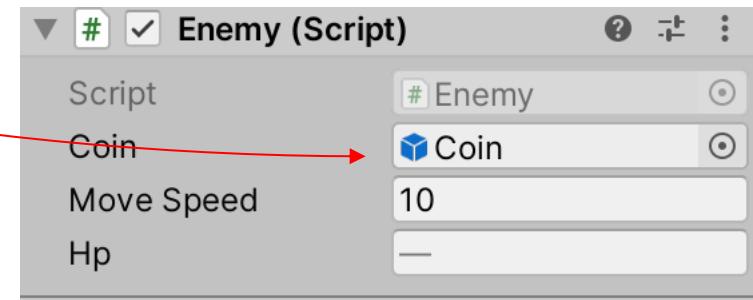
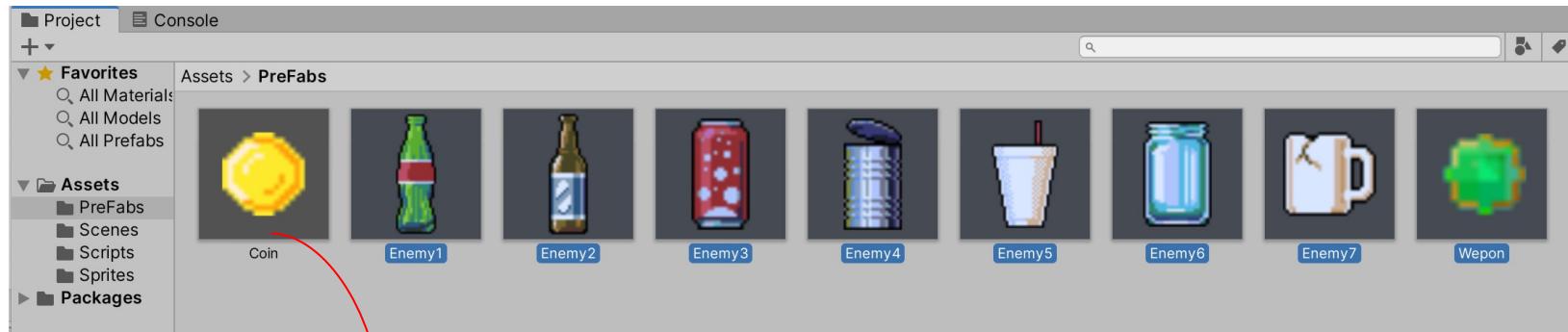
```
[SerializeField]
private GameObject coin;
[SerializeField]
private float moveSpeed = 10f;
private float minY = -7f;
[SerializeField]
private float hp = 1f;

public void SetMoveSpeed(float moveSpeed){
    this.moveSpeed = moveSpeed;
}
void Update()
{
    transform.position += Vector3.down * moveSpeed * Time.deltaTime;
    //게임 플레이 화면 아래쪽을 벗어나면 없어지도록..
    if(transform.position.y < minY){
        Destroy(gameObject);
    }
}

//총돌 감지 되면 자동으로 호출되는 메소드
private void OnTriggerEnter2D(Collider2D other){
    if(other.gameObject.tag == "Wepon"){
        Wepon wepon = other.gameObject.GetComponent<Wepon>();
        hp -= wepon.damage;
        if(hp <= 0){
            Destroy(gameObject);
            Instantiate(coin, transform.position, Quaternion.identity);
        }
        Destroy(other.gameObject);
    }
}
```

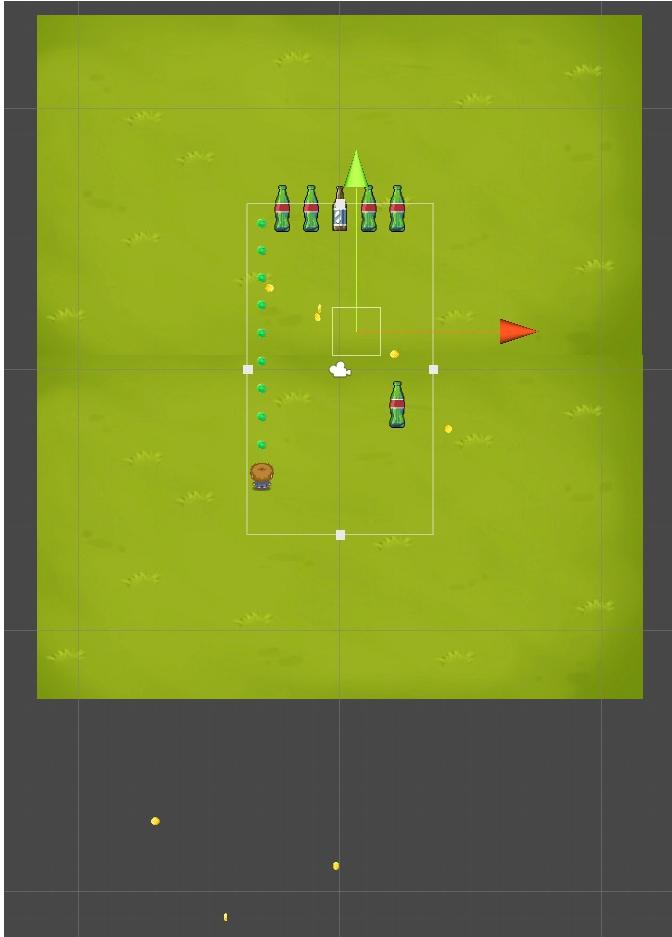
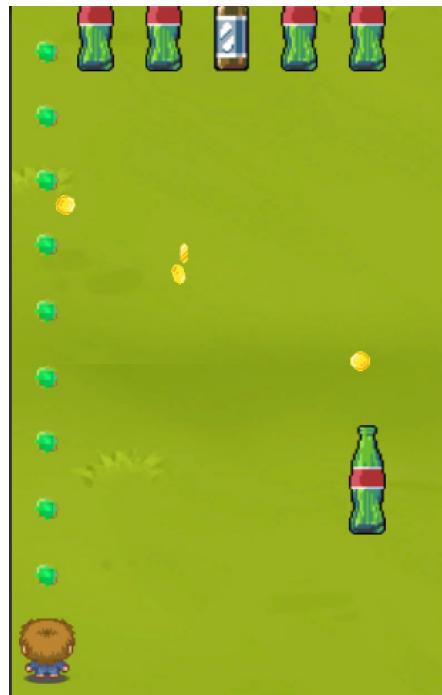
02 코인 만들기

■ 모든 Enemy 에 Coin Prefab 적용



02 코인 만들기

- Play 해보기
- Coin 이 계속 발생하는 문제점



02 코인 만들기

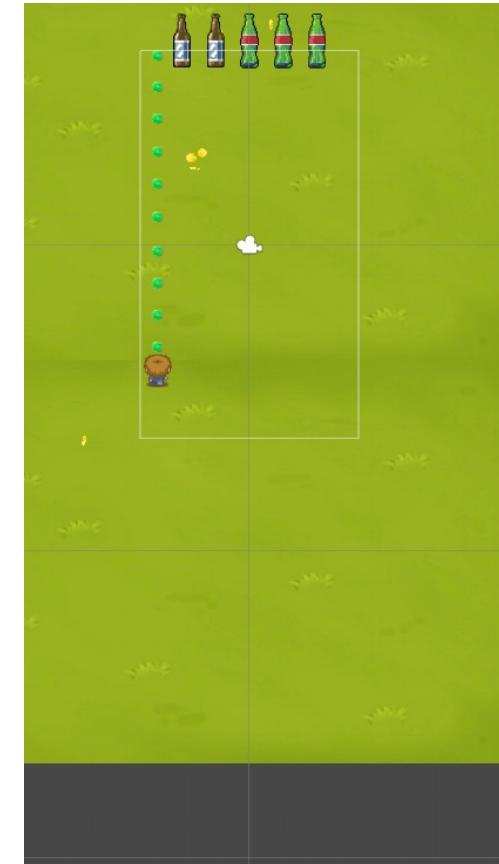
Coin.cs 수정

```
private float minY = -7f;

// Start is called before the first frame update
void Start()
{
    Jump();
}

void Jump(){
    Rigidbody2D rigidBody = GetComponent<Rigidbody2D>();
    float randomJumpForce = Random.Range(4f, 8f);
    Vector2 jumpVelocity = Vector2.up * randomJumpForce;
    jumpVelocity.x = Random.Range(-2f, 2f);
    rigidBody.AddForce(jumpVelocity, ForceMode2D.Impulse);
}

// Update is called once per frame
void Update()
{
    if(transform.position.y < minY){
        Destroy(gameObject);
    }
}
```



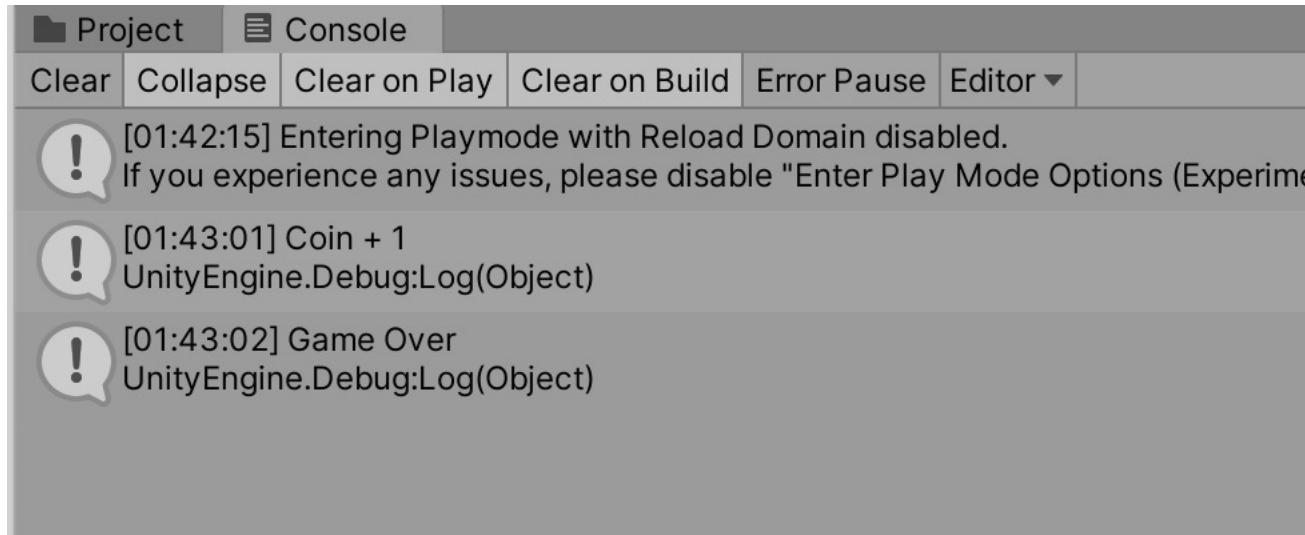
02 코인 만들기

- Player 가 Coin 과 충돌 하면? Coin +1 & 사라짐
- Player.cs 수정

```
private void OnTriggerEnter2D(Collider2D other){  
    if(other.gameObject.tag == "Enemy"){  
        Debug.Log("Game Over");  
        Destroy(gameObject);  
    }else if(other.gameObject.tag == "Coin"){  
        Debug.Log("Coin + 1");  
        Destroy(other.gameObject);  
    }  
}
```

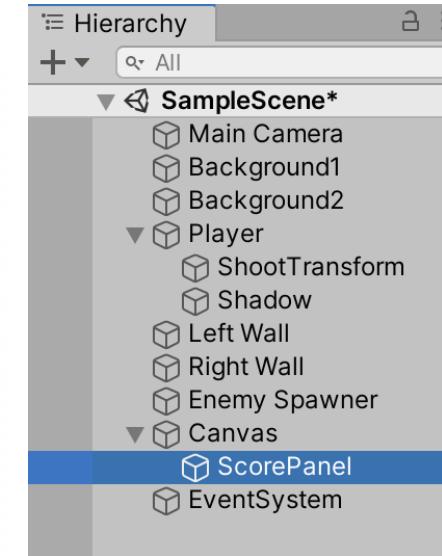
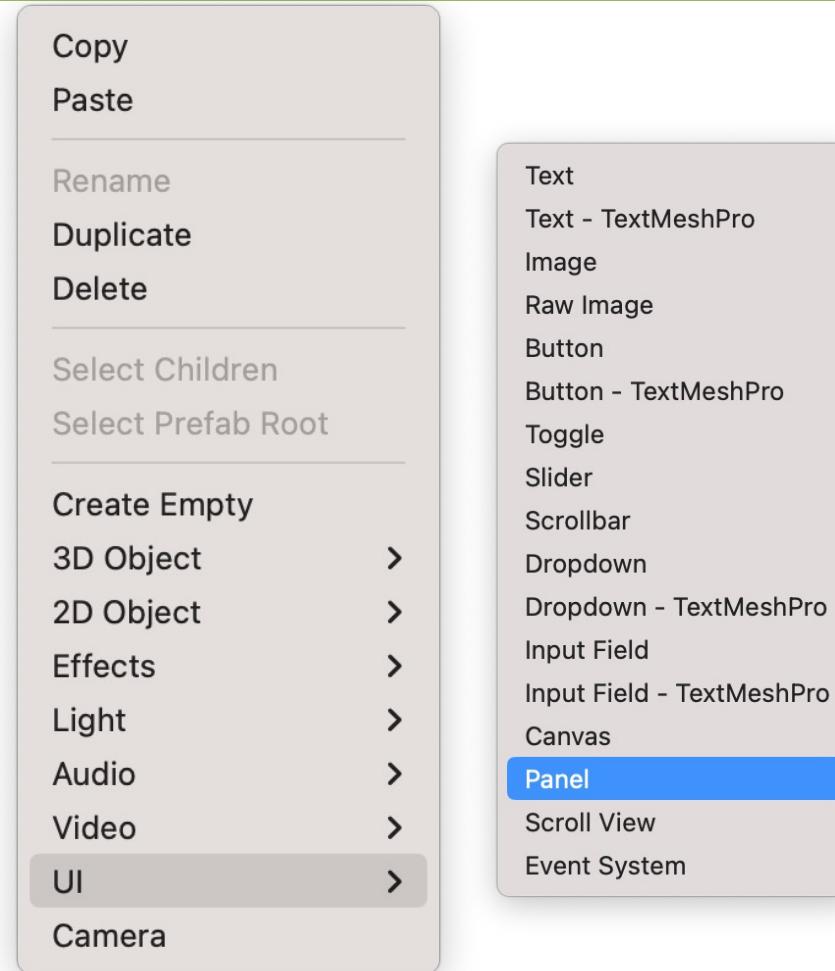
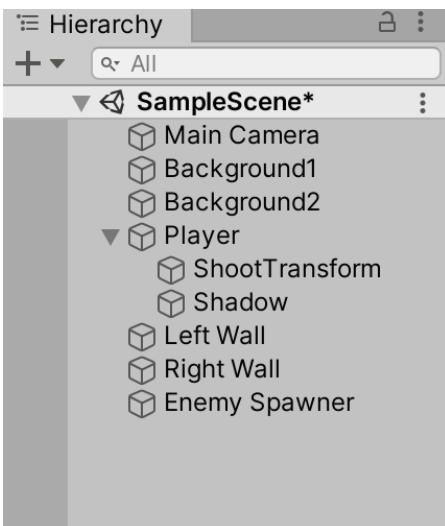
02 코인 만들기

■ Play 해보고 문제점 있으면 수정 하기



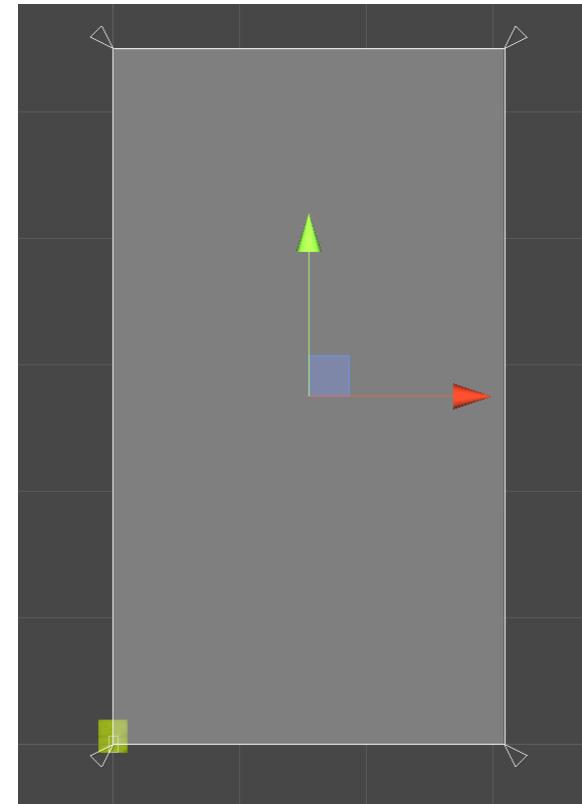
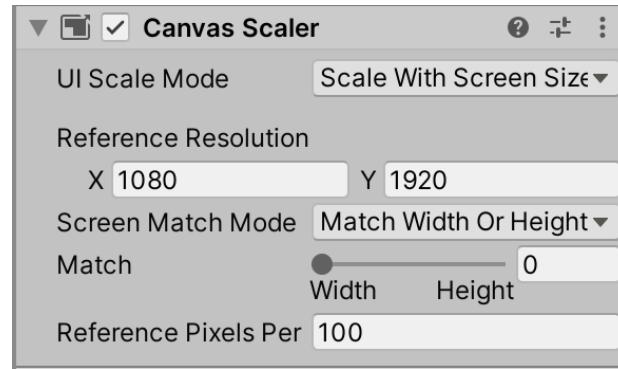
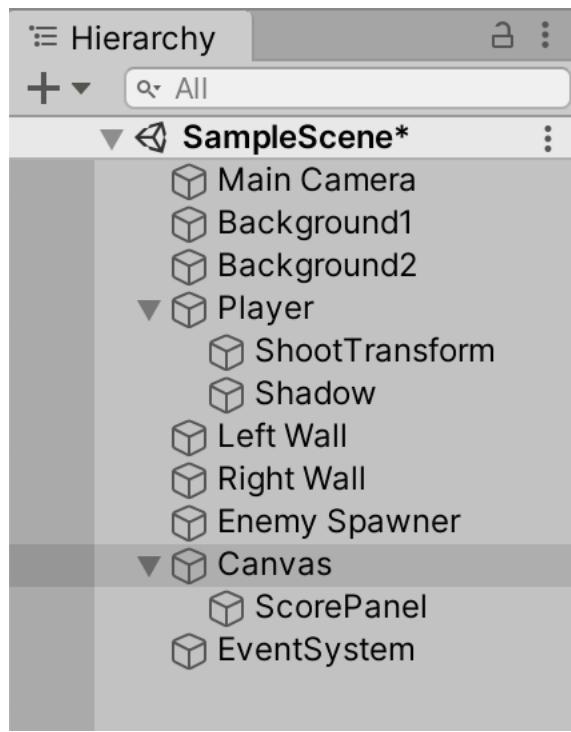
03 점수 출력

- 점수 UI 추가
- ScorePanel 변경



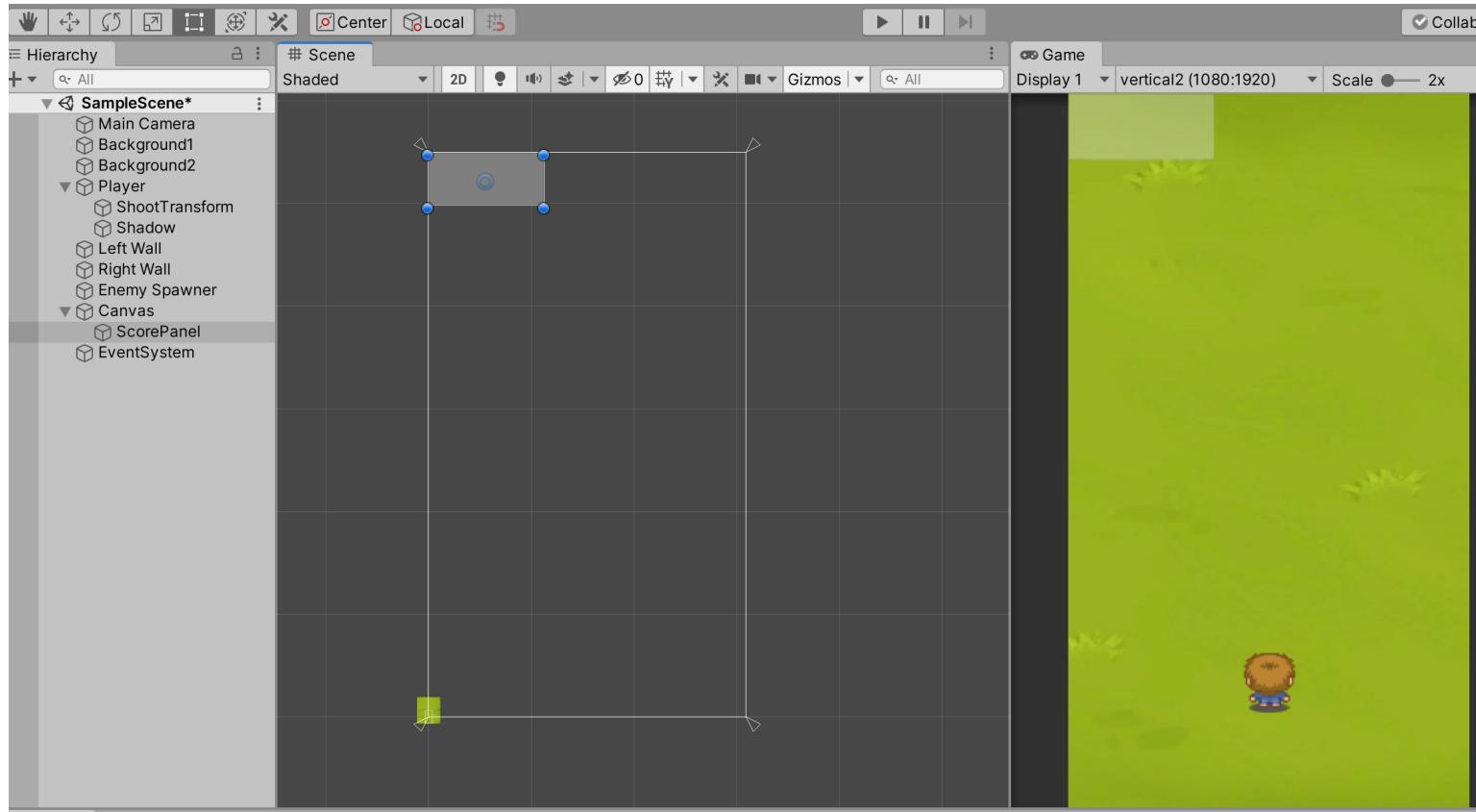
03 점수 출력

- Canvas Scaler에서 UI Scale Mode를 Scale With Screen Size로 변경
- Resolution도 화면 크기와 같이 맞춤



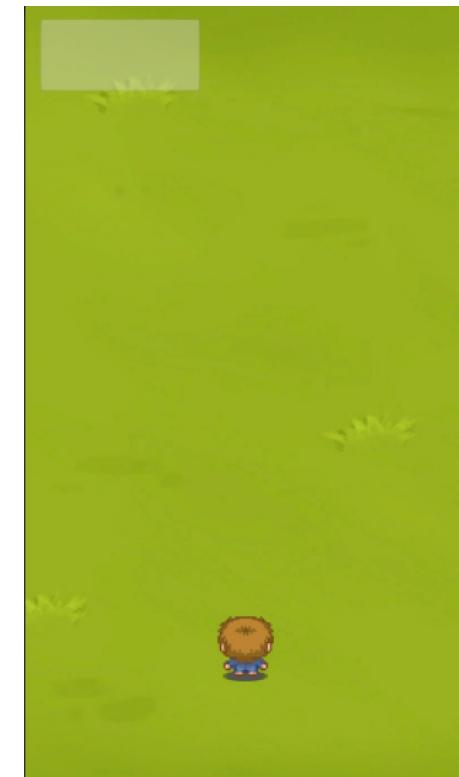
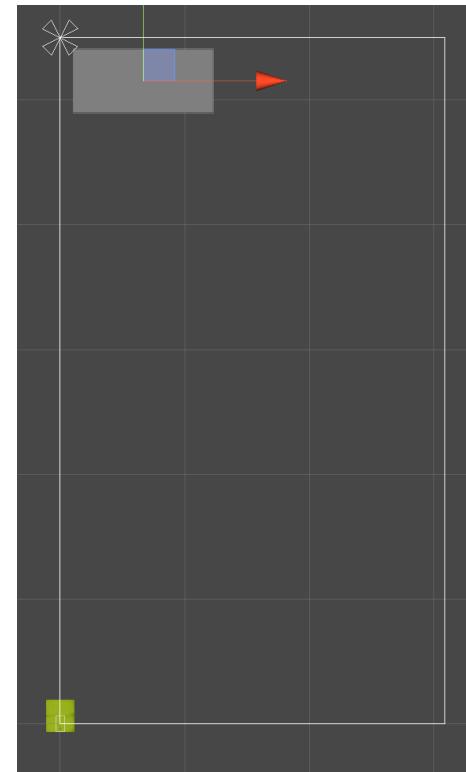
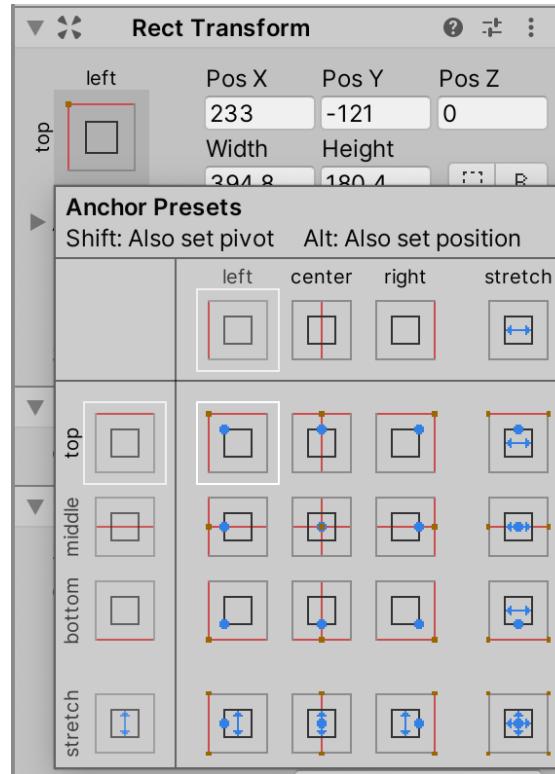
03 점수 출력

Score Panel 사이즈, 위치 변경



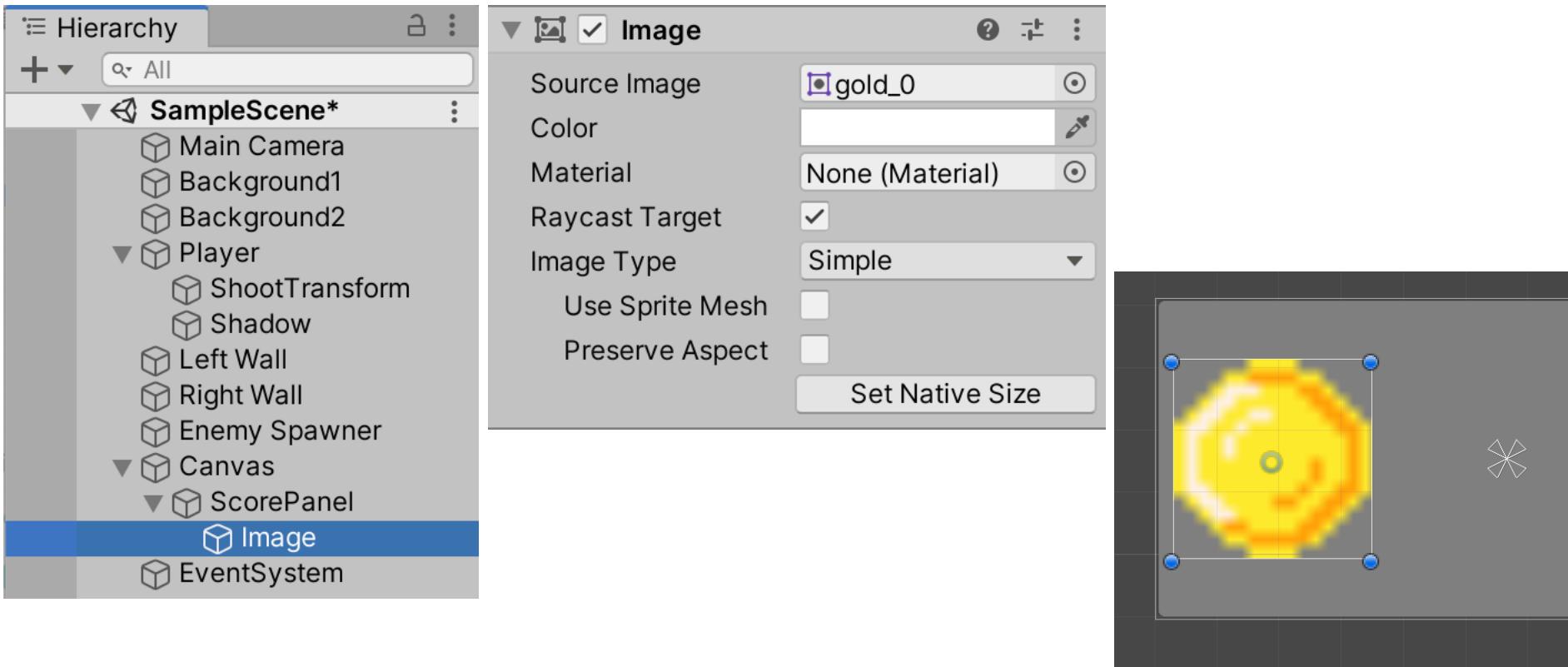
03 점수 출력

- Score Panel 사이즈, 위치 변경
- Anchor Preset 이용하면 사전에 정의된 위치로 쉽게 이동 가능



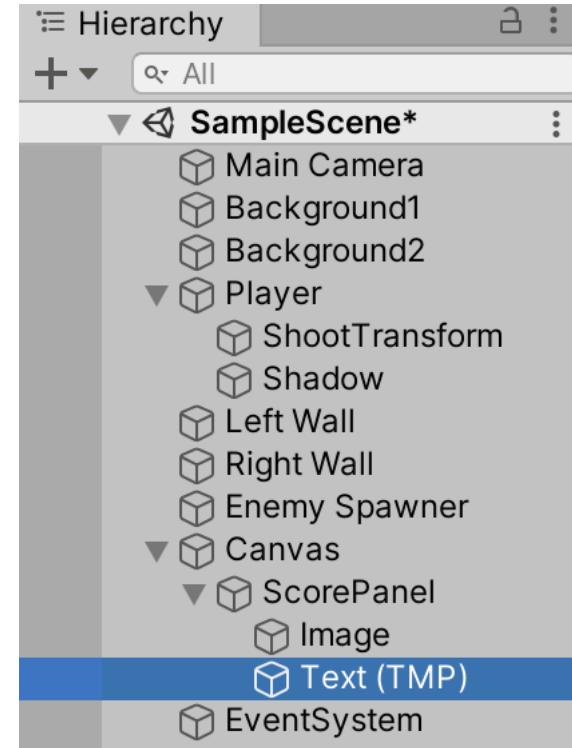
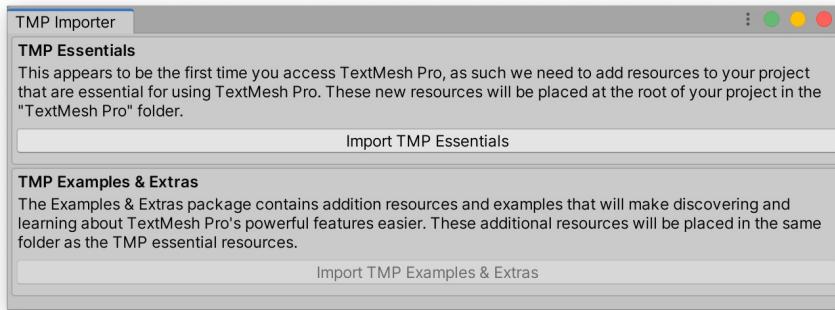
03 점수 출력

- Score Panel 안쪽에 UI Image 추가, Sprite Coin 이미지 추가
- Set Native Size 버튼 클릭 후 작아진 이미지 적당히 키우기



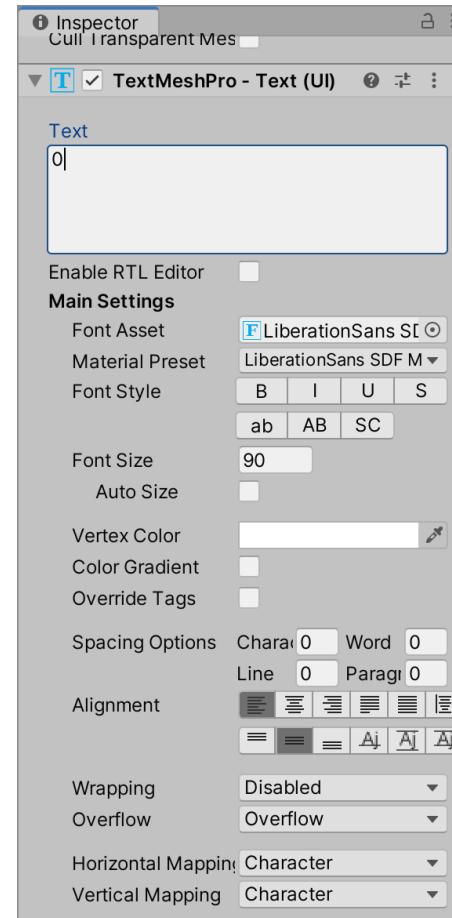
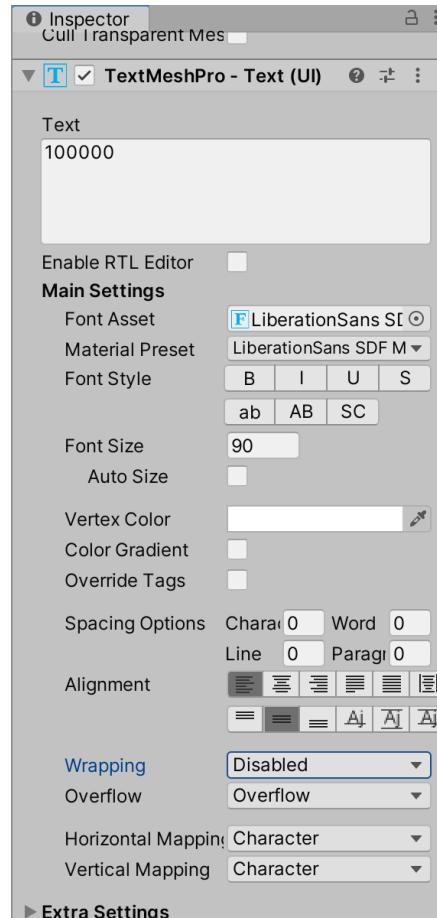
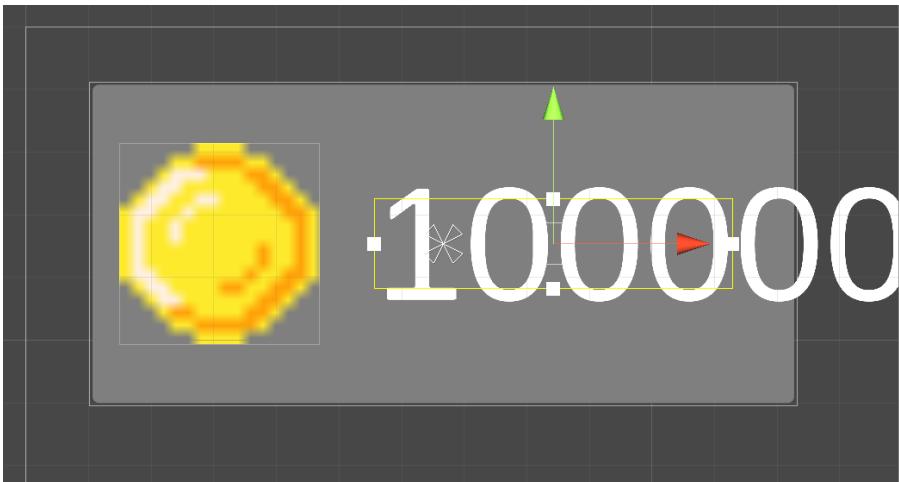
03 점수 출력

- Score Panel 안쪽에 UI TextMeshPro 추가
- Set Native Size 버튼 클릭 후 작아진 이미지 적당히 키우기



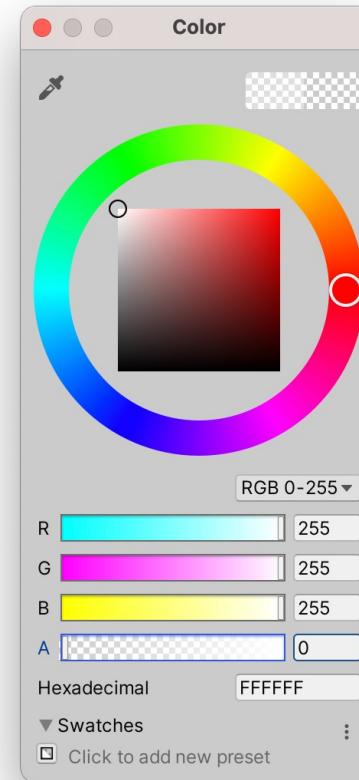
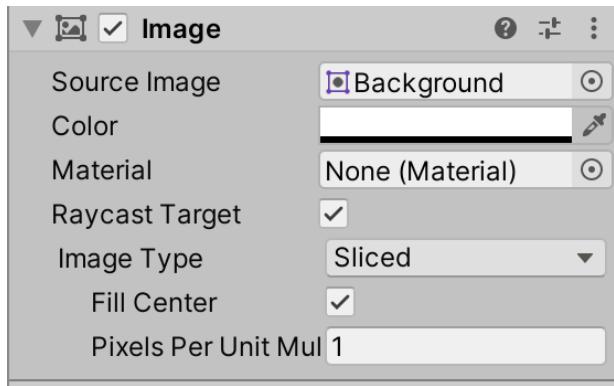
03 점수 출력

- UI TextMeshPro 위치 변경
- 폰트크기 90
- 정렬 위치, Wrapping disable



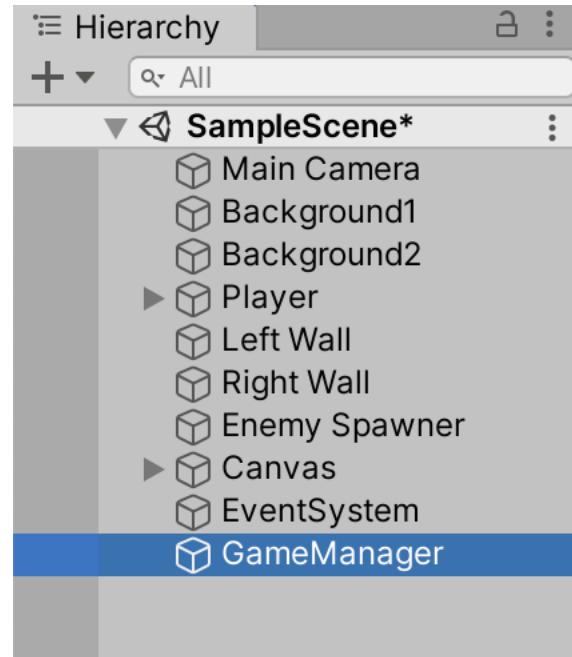
03 점수 출력

Score Panel 투명도 0



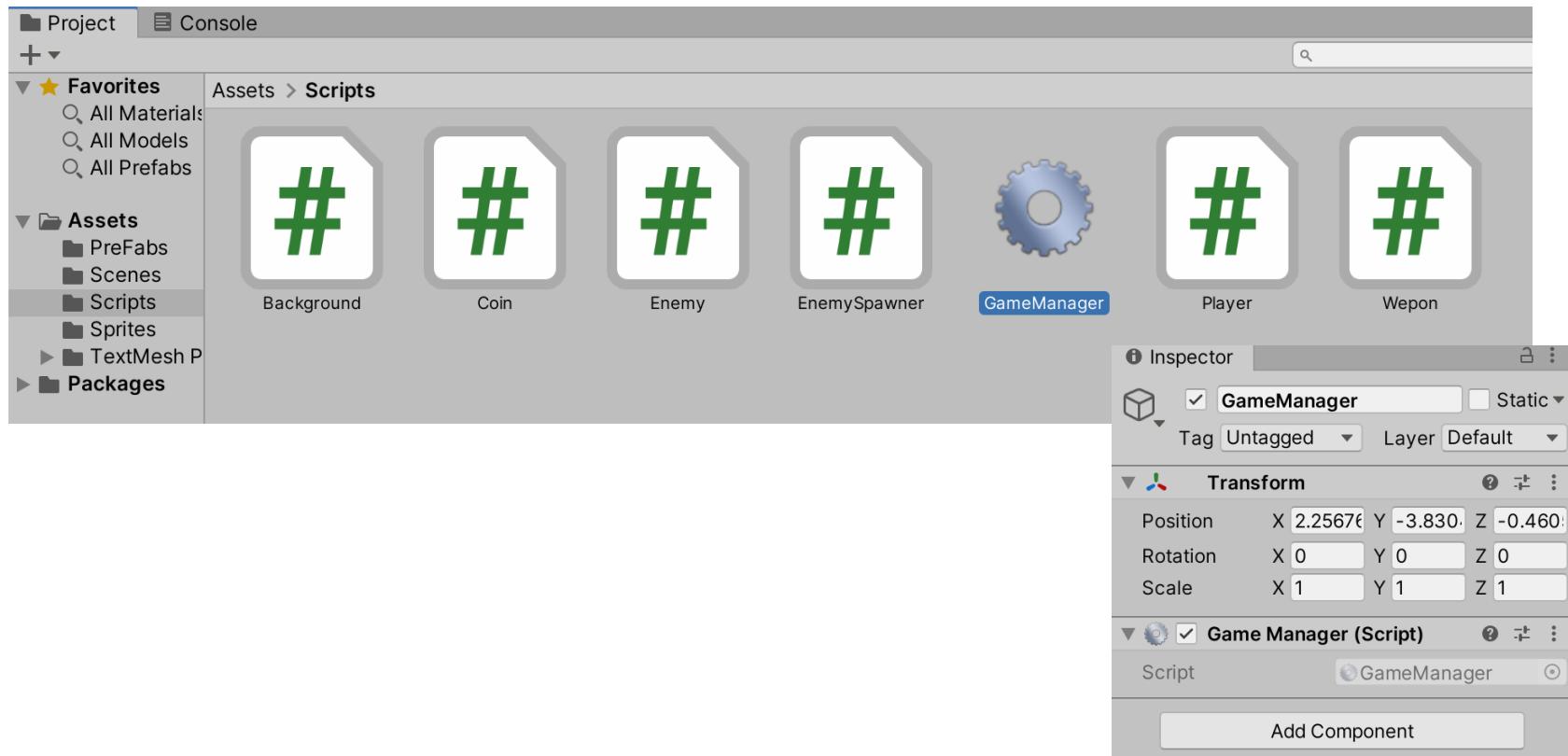
03 점수 출력

■ Create Empty 로 GameManager 추가



03 점수 출력

■ GameManager.cs 추가 및 GameManager Object에 적용



■ GameManager.cs 수정

C# GameManager.cs X C# Player.cs

Assets > Scripts > C# GameManager.cs

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using TMPro;
5
6  public class GameManager : MonoBehaviour
7  {
8      // singleton
9      public static GameManager instance = null;
10
11     [SerializeField]
12     private TextMeshProUGUI text;
13
14     private int coin = 0;
15
16     // Start() 보다 먼저 호출되는 메소드
17     void Awake()
18     {
19         if(instance == null){
20             instance = this;
21         }
22     }
23
24     public void IncreaseCoin()
25     {
26         coin++;
27         text.SetText(coin.ToString());
28     }
}
```

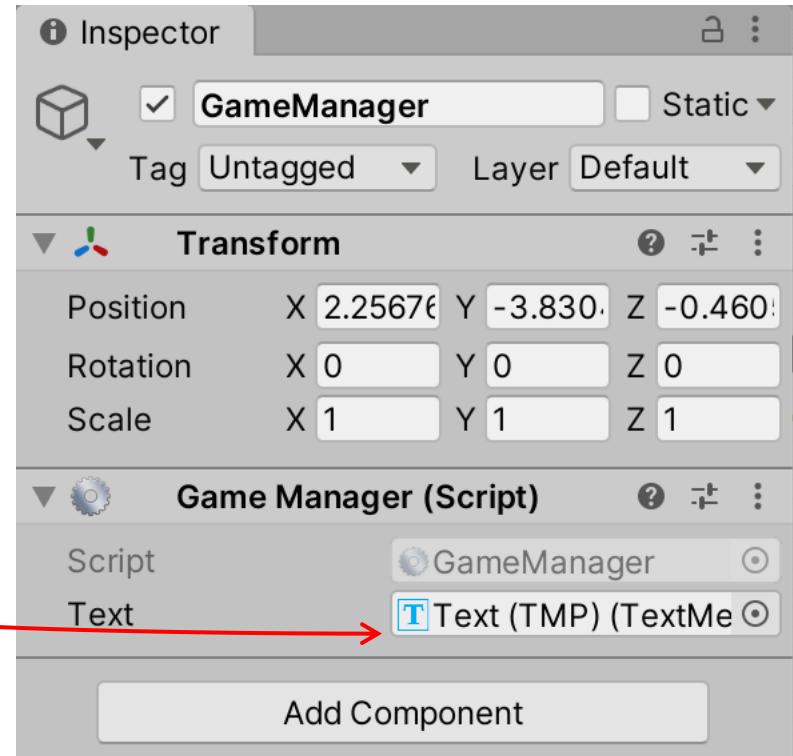
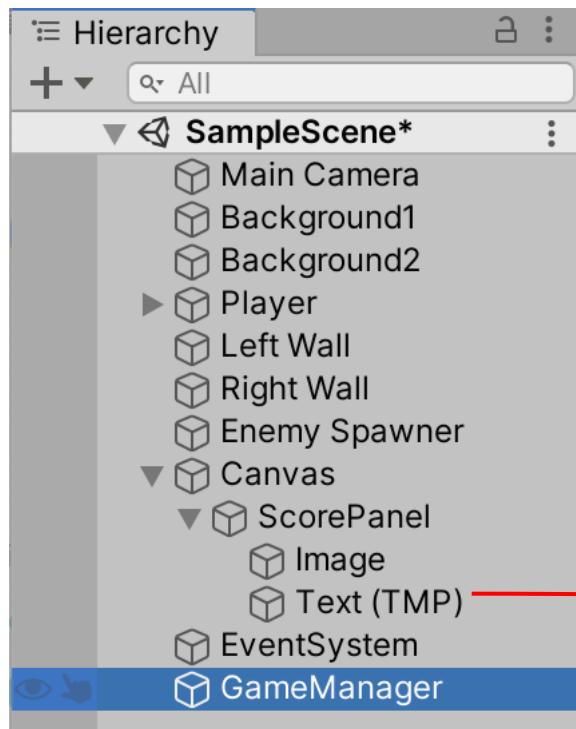
03 점수 출력

■ Player.cs 설정

```
private void OnTriggerEnter2D(Collider2D other){
    if(other.gameObject.tag == "Enemy"){
        Debug.Log("Game Over");
        Destroy(gameObject);
    }else if(other.gameObject.tag == "Coin"){
        // Debug.Log("Coin + 1");
        GameManager.instance.IncreaseCoin();
        Destroy(other.gameObject);
    }
}
```

03 점수 출력

■ GameManager 에 Script 아래 Text 영역에 Score Panel 의 Text(TMP) 연결



03 점수 출력

■ Play 해보기

