

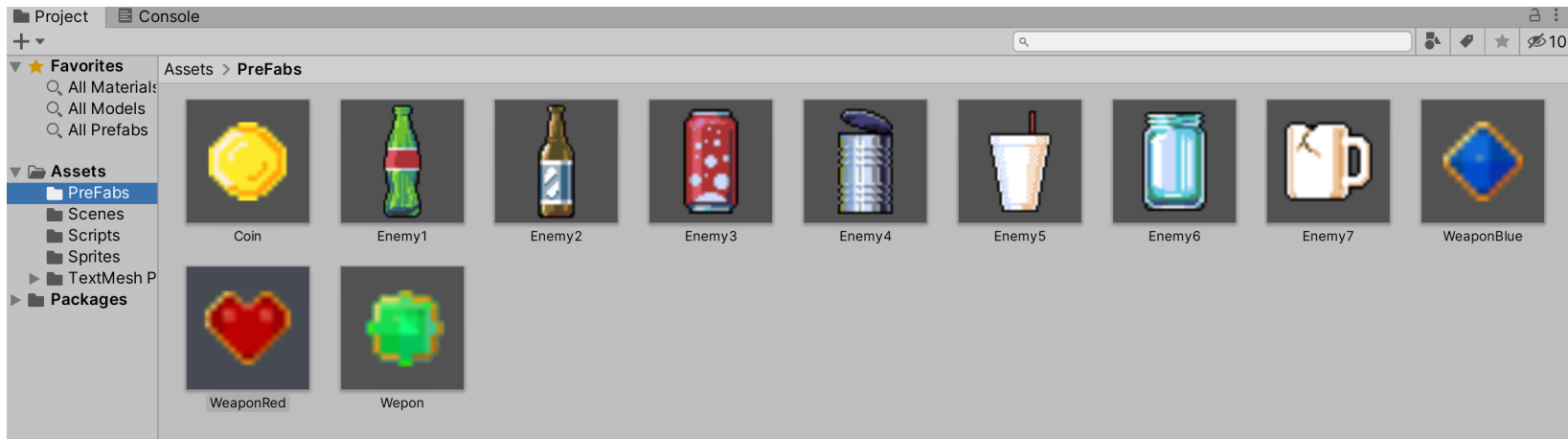
chapter07

게임 진행 제어

1. 무기 업그레이드
2. 보스 만들기
3. 게임 오버 처리
4. 결과 화면

01 무기 업그레이드

- 획득한 Coin 수에 따른 무기 업그레이드
- Prefabs 의 Weapon 복사
- Sprite 이미지에서 무기 이름에 맞는 이미지 선택후 적용



01 무기 업그레이드

■ Player.cs 수정

```
[SerializeField]
```

```
private GameObject[] weapons;
```

```
private int weaponIndex = 0;
```

```
public void Upgrade(){
```

```
    weaponIndex ++;
```

```
    if(weaponIndex >= weapons.Length){
```

```
        weaponIndex = weapons.Length - 1;
```

```
    }
```

```
}
```

01 무기 업그레이드

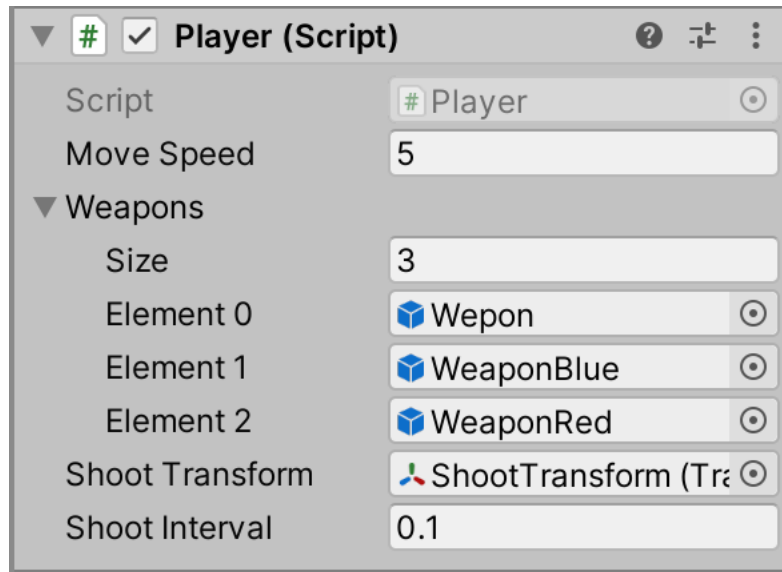
■ GameManager.cs 수정

```
public void IncreaseCoin(){
    coin++;
    text.SetText(coin.ToString());

    if(coin % 30 == 0){
        Player player = FindObjectOfType<Player>();
        if(player != null){
            player.Upgrade();
        }
    }
}
```

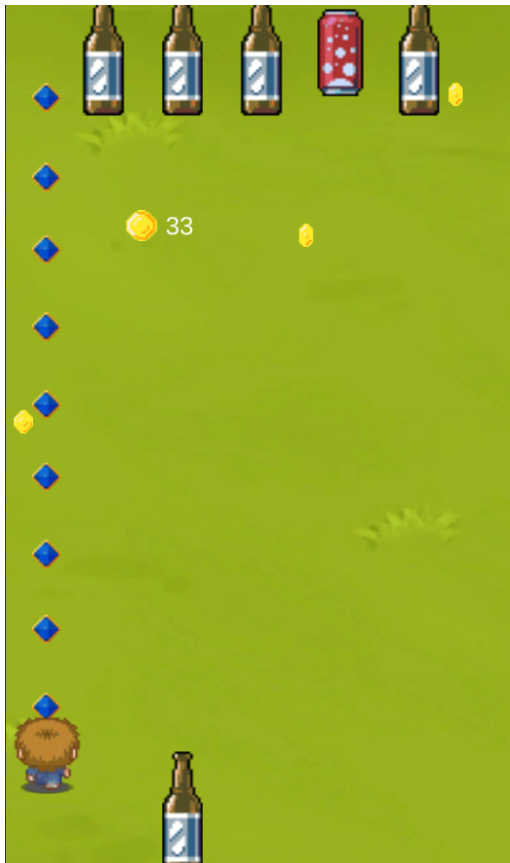
01 무기 업그레이드

■ Prefabs 의 무기들을 Player.Weapons 배열로 추가



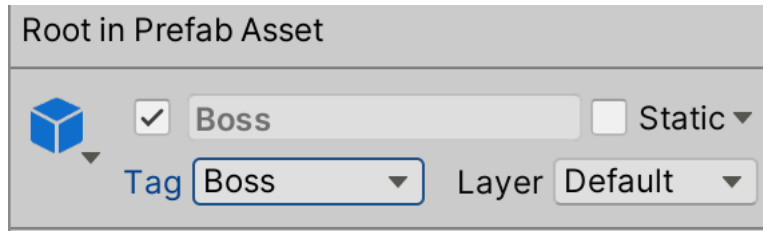
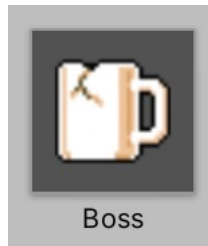
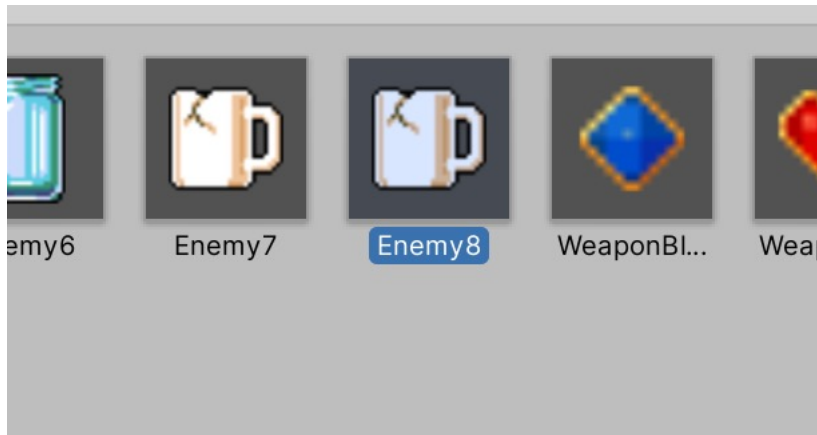
01 무기 업그레이드

■ Play 해보기, 무기 damage변경 및 테스트



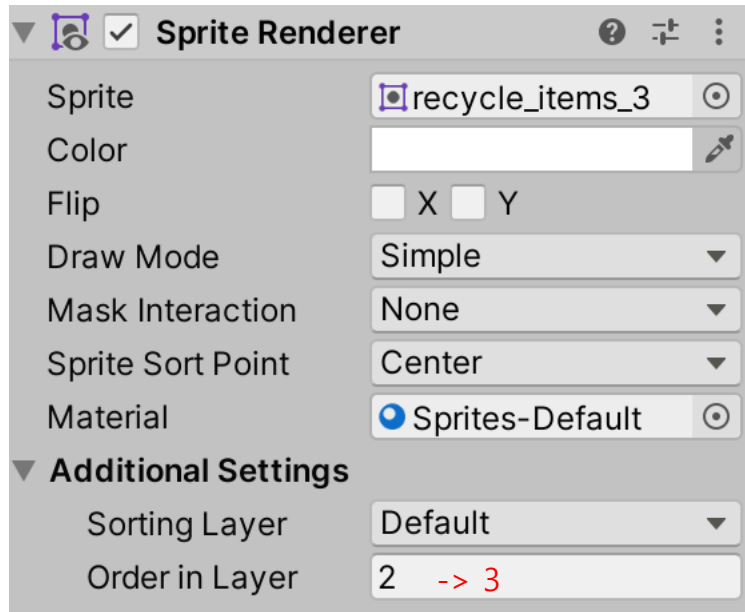
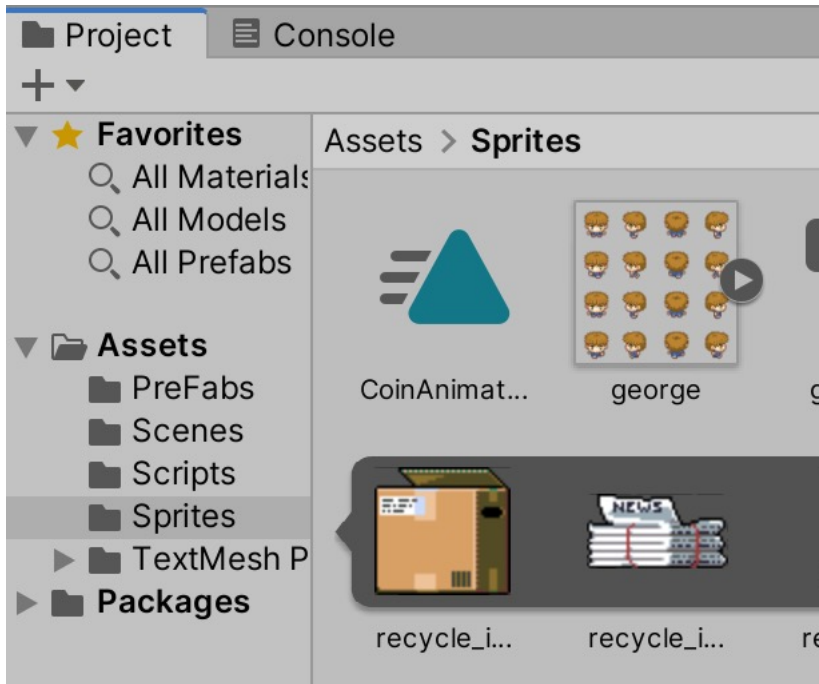
02 보스 만들기

- Prefabs 에 있는 Enemy중 하나를 Duplicate 한후 Boss 로 변경
- Tag 를 Boss 로 변경



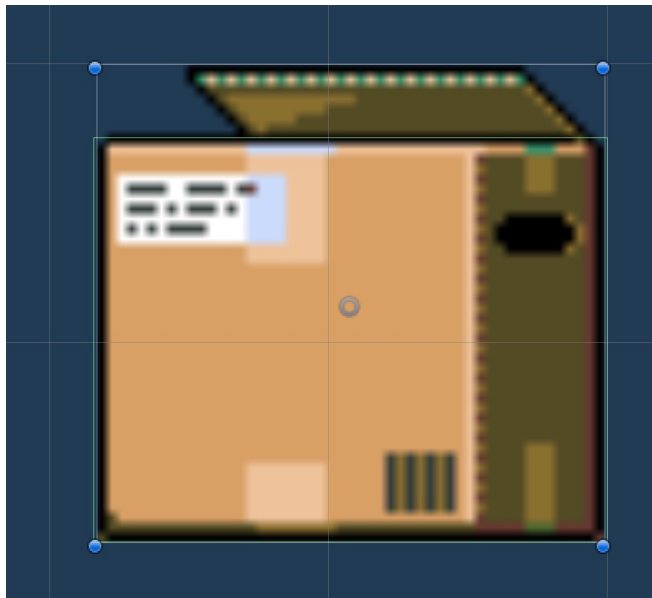
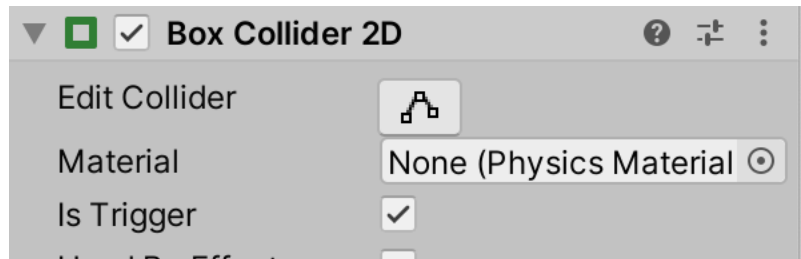
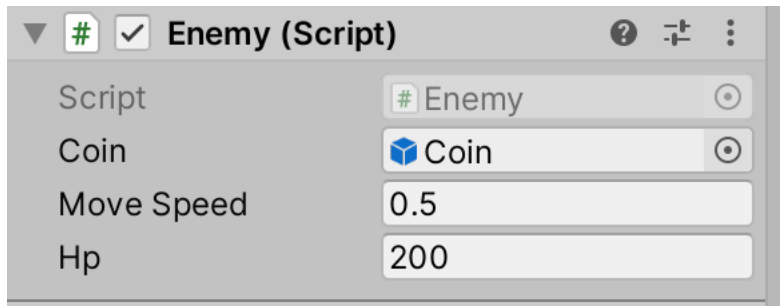
02 보스 만들기

■ Boss 이미지 변경



02 보스 만들기

- Boss Move Speed -> 0.5
- Boss Hp -> 200
- 충돌영역 수정

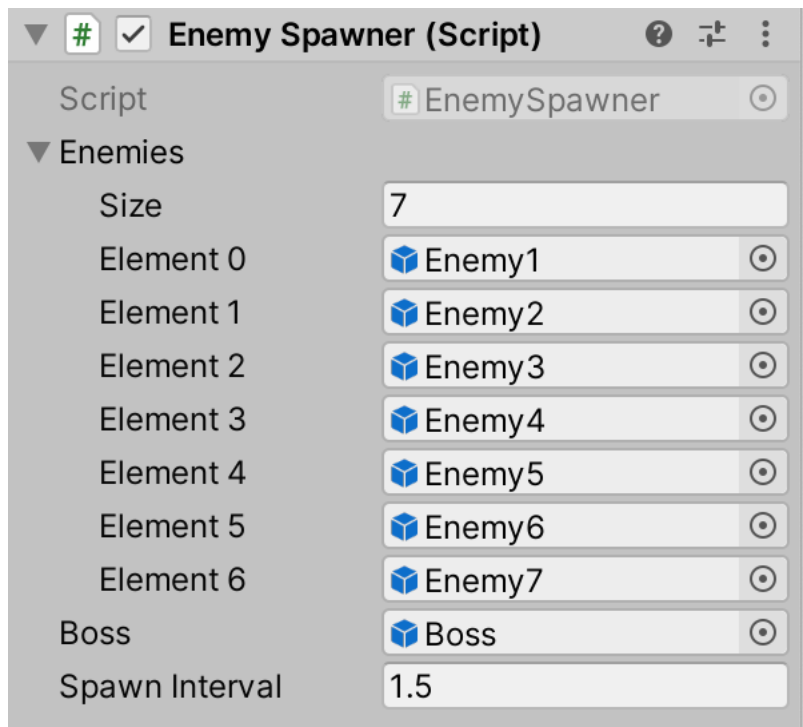


02 보스 만들기

- EnemySpawner.cs 수정 (Boss 변수 추가)
- Unity Editor 에서 Boss 변수에 Prefabs 의 Boss 추가

[SerializeField]

```
private GameObject boss;
```



■ EnemySpawner.cs 수정

```
IEnumerator EnemyRoutine(){
    yield return new WaitForSeconds(3f);
    float moveSpeed = 5f;
    int spawnCount = 0;
    int enemyIndex = 0;
    while(true){
        foreach(float posX in arrPosX){
            SpawnEnemy(posX, enemyIndex, moveSpeed);
        }
        spawnCount++;
        if(spawnCount % 10 == 0){
            enemyIndex ++;
            moveSpeed += 2;
        }

        if(enemyIndex >= enemies.Length){
            SpawnBoss();
            enemyIndex = 0;
            moveSpeed = 5f;
        }
        yield return new WaitForSeconds(spawnInterval);
    }
}

void SpawnBoss(){
    Instantiate(boss, transform.position, Quaternion.identity);
}
```

02 보스 만들기

■ Player.cs 수정

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

02 보스 만들기

■ Play 해보기



03 게임 오버 처리

- Coroutine() 동작 멈추기
- 게임 오버 알림 및 재시작 여부 묻기
- EnemySpawner.cs 수정 (StopEnemyRoutine() 추가)

```
public void StopEnemyRoutine(){  
    StopCoroutine("EnemyRoutine");  
}
```

03 게임 오버 처리

■ GameManager.cs 수정 (SetGameOver() 추가)

```
public void SetGameOver(){  
    EnemySpawner enemySpawner = FindObjectOfType<EnemySpawner>();  
    if(enemySpawner != null){  
        enemySpawner.StopEnemyRoutine();  
    }  
}
```

03 게임 오버 처리

■ Player.cs 수정 (플레이어가 적이나 보스와 충돌하면 사망)

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


03 게임 오버 처리

- Play -> 플레이어가 적이나 보스와 충돌하면 사망 확인

03 게임 오버 처리

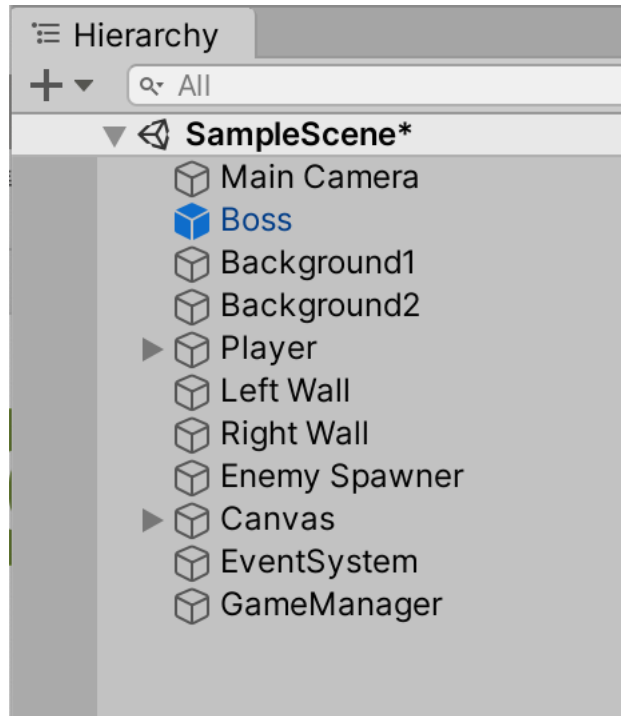
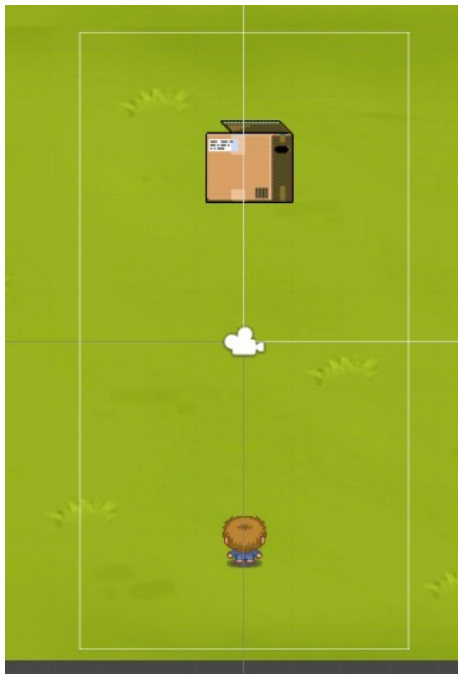
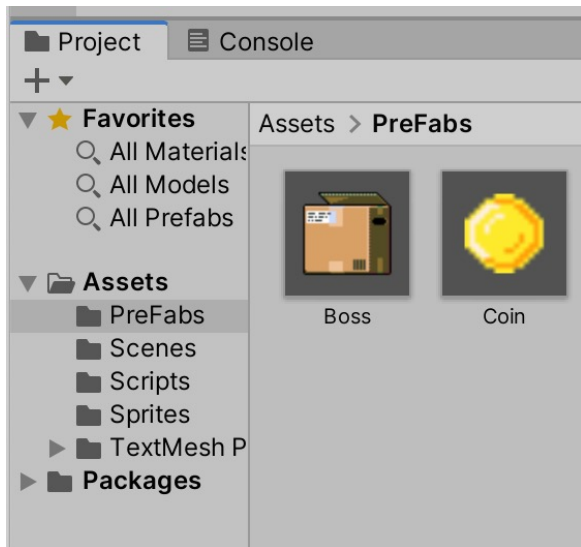
- 플레이어가 보스 처치시에도 게임 오버!
- Enemy.cs 수정

//충돌 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){  
            if(gameObject.tag == "Boss" ){  
                GameManager.instance.SetGameOver();  
            }  
            Destroy(gameObject);  
            Instantiate(coin, transform.position, Quaternion.identity);  
        }  
        Destroy(other.gameObject);  
    }  
}
```

03 게임 오버 처리

- Play 해보기
- 임시로 Prefabs 의 Boss 를 Scene 에 등장시켜서 확인



03 게임 오버 처리

- Game Over 시 미사일 발사 중지
- GameManager.cs 에 게임종료 여부 변수 추가 및 SetGameOver() 수정

[HideInInspector]

```
public bool isGameOver = false;
```

```
public void SetGameOver(){
```

```
    isGameOver = true;
```

```
    EnemySpawner enemySpawner = FindObjectOfType<EnemySpawner>();
```

```
    if(enemySpawner != null){
```

```
        enemySpawner.StopEnemyRoutine();
```

```
    }
```

```
}
```

03 게임 오버 처리

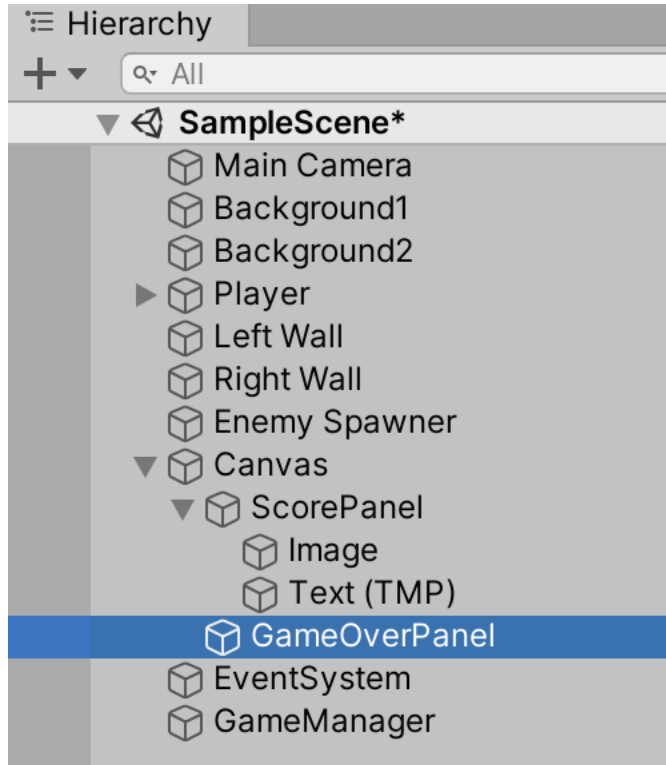
■ Player.cs 수정

```
// Update is called once per frame
void Update()
{
    Vector3 mousePos = Camera.main.ScreenToWorldPoint(Input.mousePosition);
    float toX = Mathf.Clamp(mousePos.x, -3.0f, 3.0f);
    transform.position = new Vector3(toX, transform.position.y, transform.position.z);

    if(GameManager.instance.isGameOver == false){
        Shoot();
    }
}
```

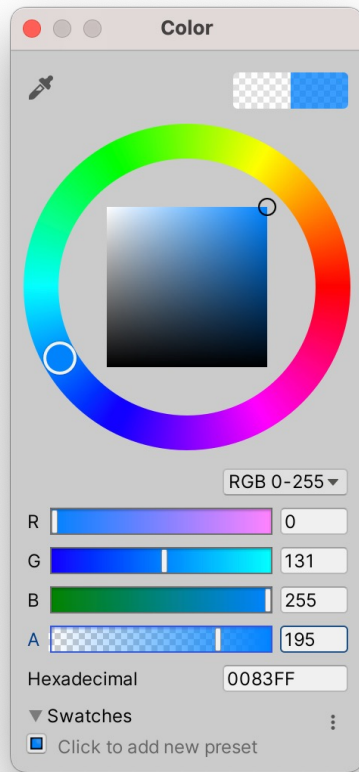
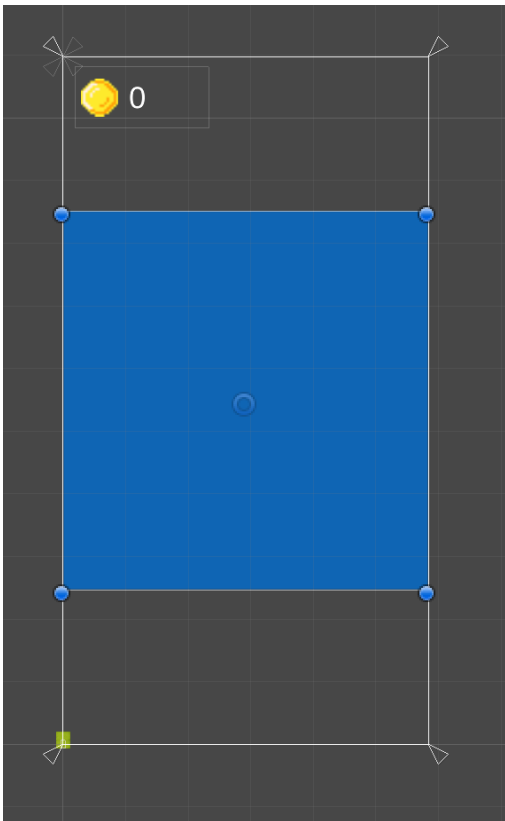
04 결과 화면

- 게임 종료 처리를 위한 Panel 추가(Canvas 아래쪽에 UI – Panel 추가)
- 이름을 GameOverPanel 로 변경



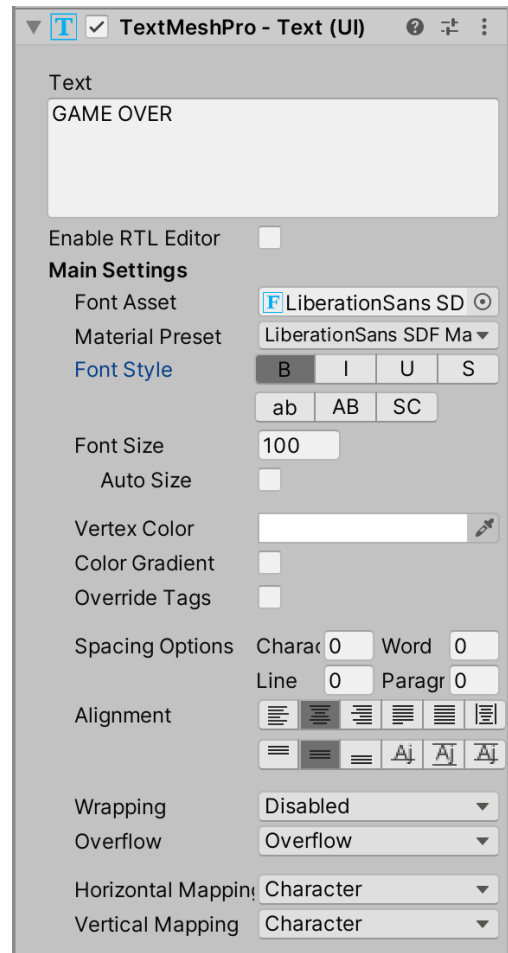
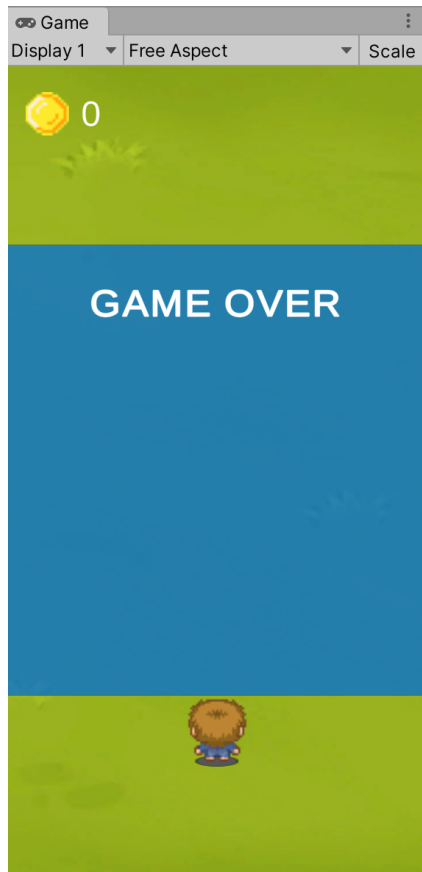
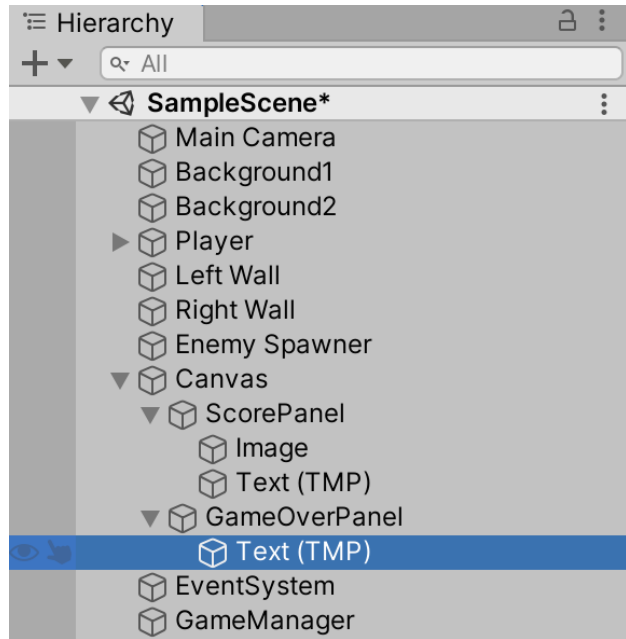
04 결과 화면

■ GameOverPanel 크기 변경 및 컬러 지정



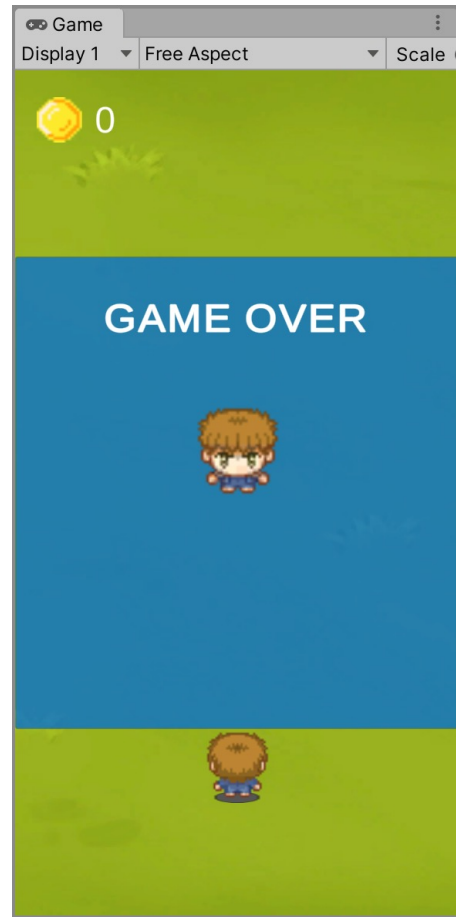
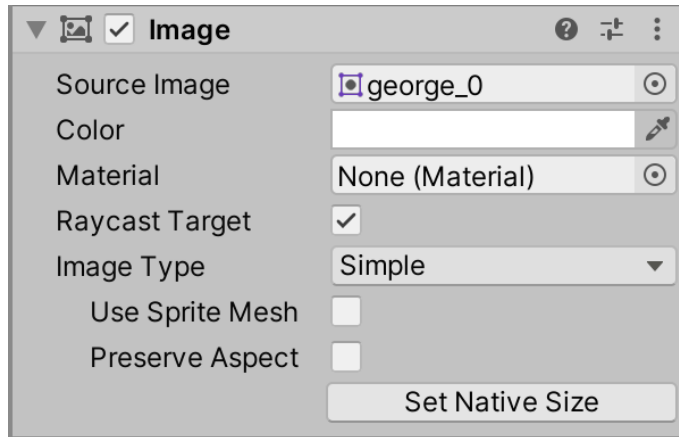
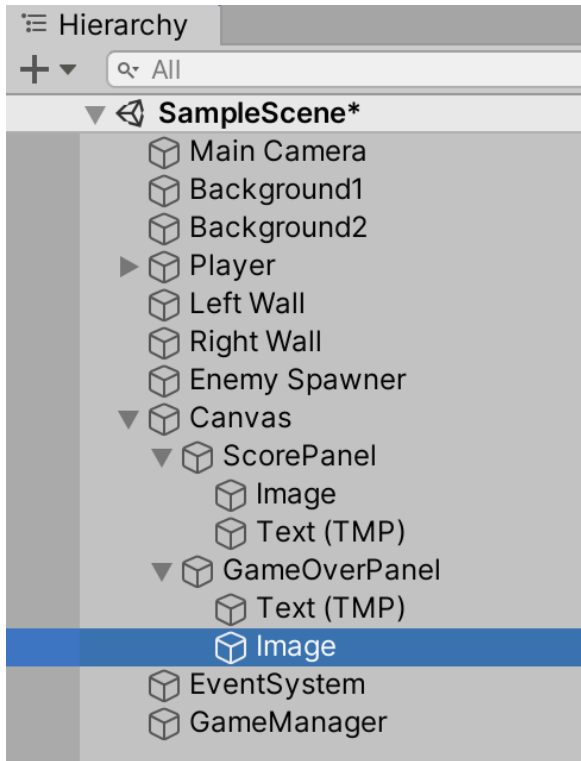
04 결과 화면

GameOverPanel 에 Text Mesh Pro 추가



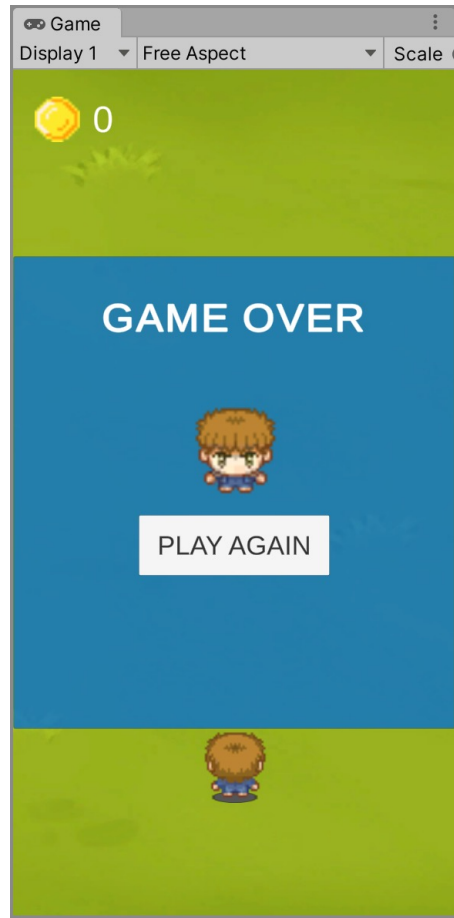
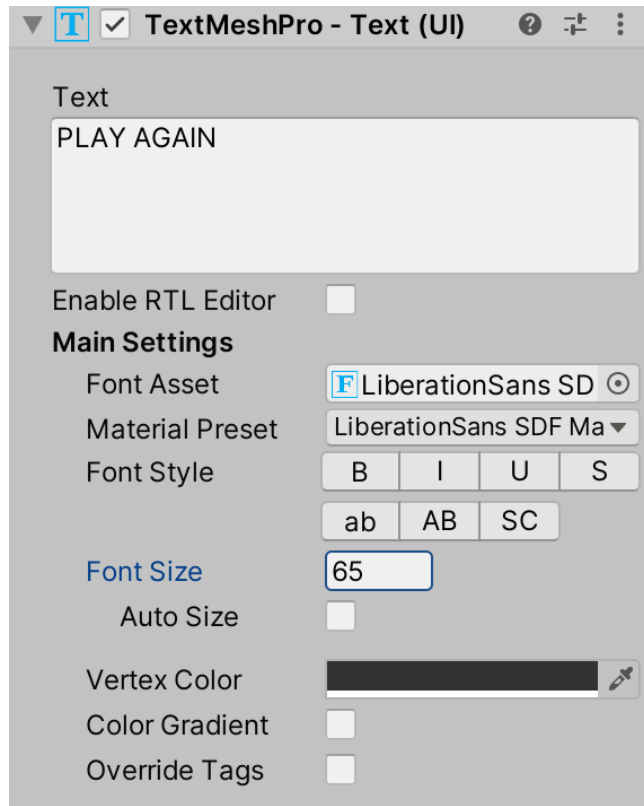
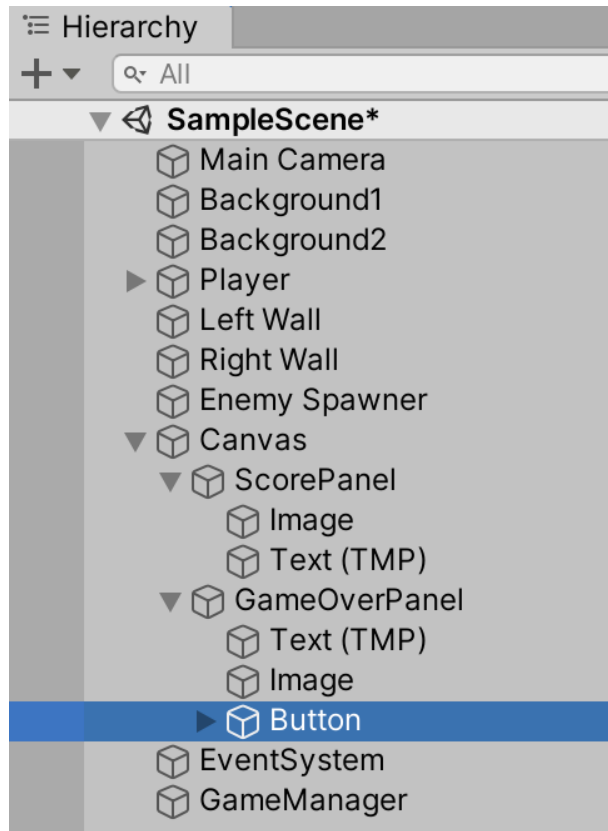
04 결과 화면

GameOverPanel 에 Image 추가



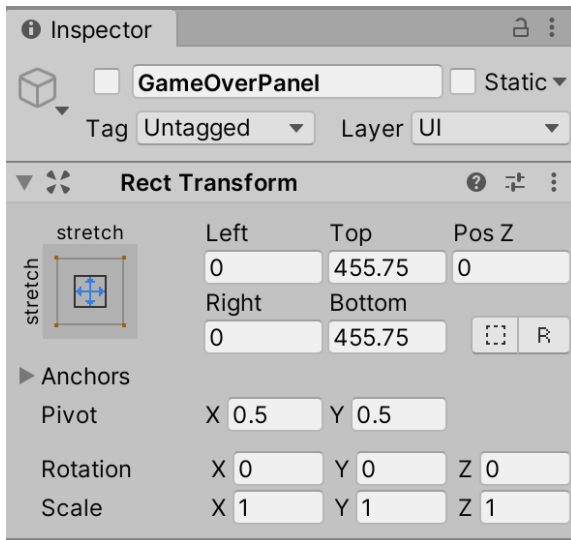
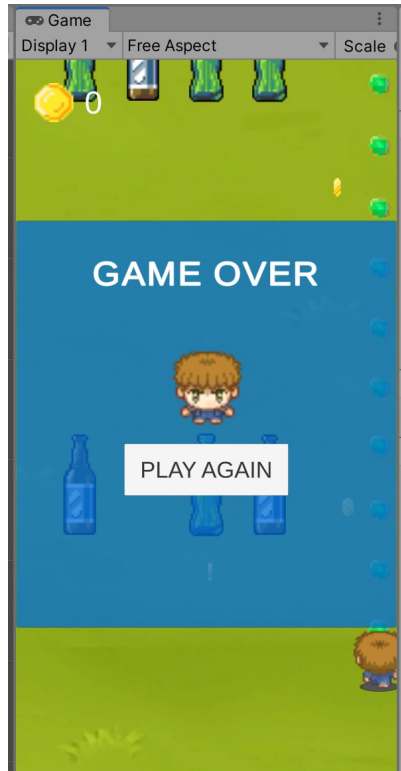
04 결과 화면

GameOverPanel 에 UI Button-TextMeshPro 추가



04 결과 화면

- Play 해보기
- GameOverPanel 비활성 해두기



04 결과 화면

■ GameManager.cs 수정 및 객체 연결

[SerializeField]

private GameObject gameOverPanel;

public void SetGameOver(){

isGameOver = true;

EnemySpawner enemySpawner = FindObjectOfType<EnemySpawner>();

if(enemySpawner != null){

 enemySpawner.StopEnemyRoutine();

}

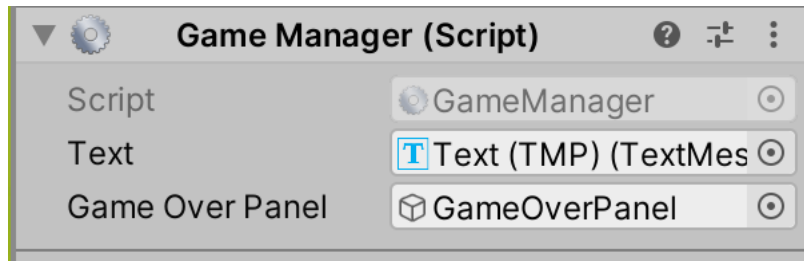
Invoke("ShowGameOverPanel", 1f); // 1초 뒤 ShowGameOverPanel() 호출하라

}

void ShowGameOverPanel(){

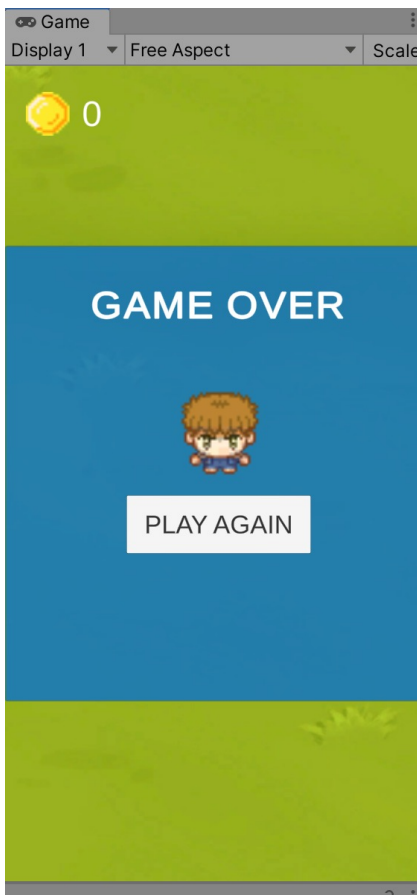
 gameOverPanel.SetActive(true);

}



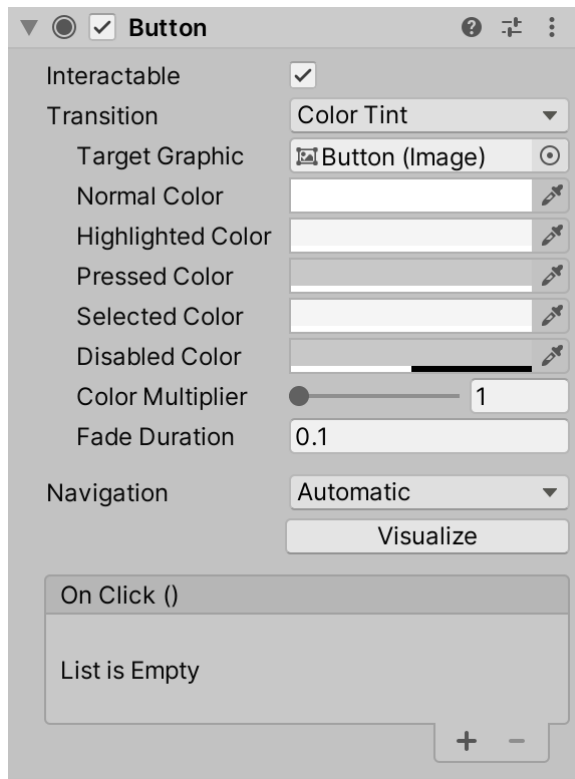
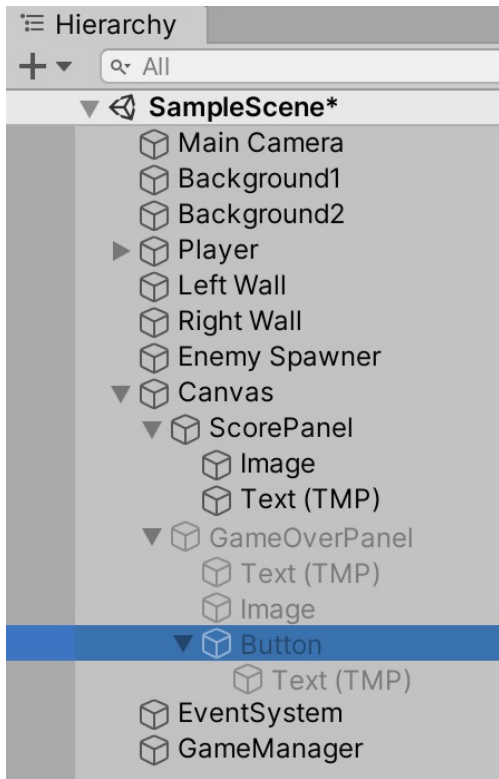
04 결과 화면

■ Play 해보기



04 결과 화면

■ PLAY AGAIN 버튼 동작 설정



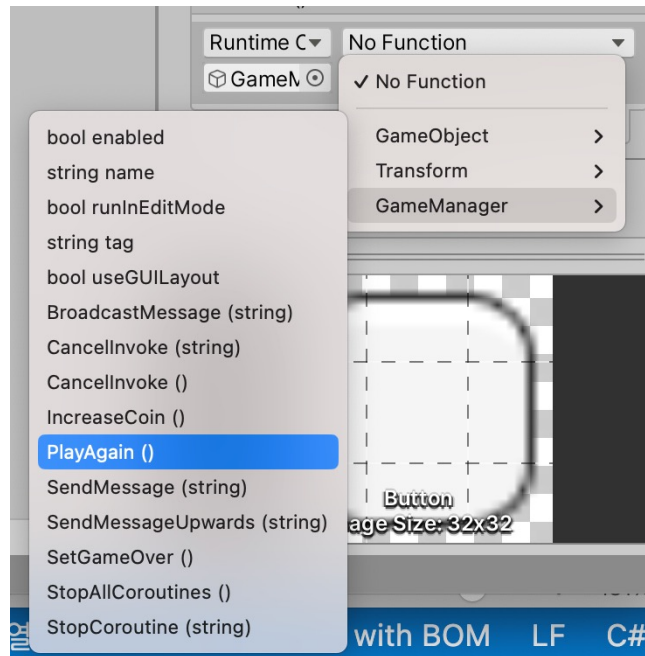
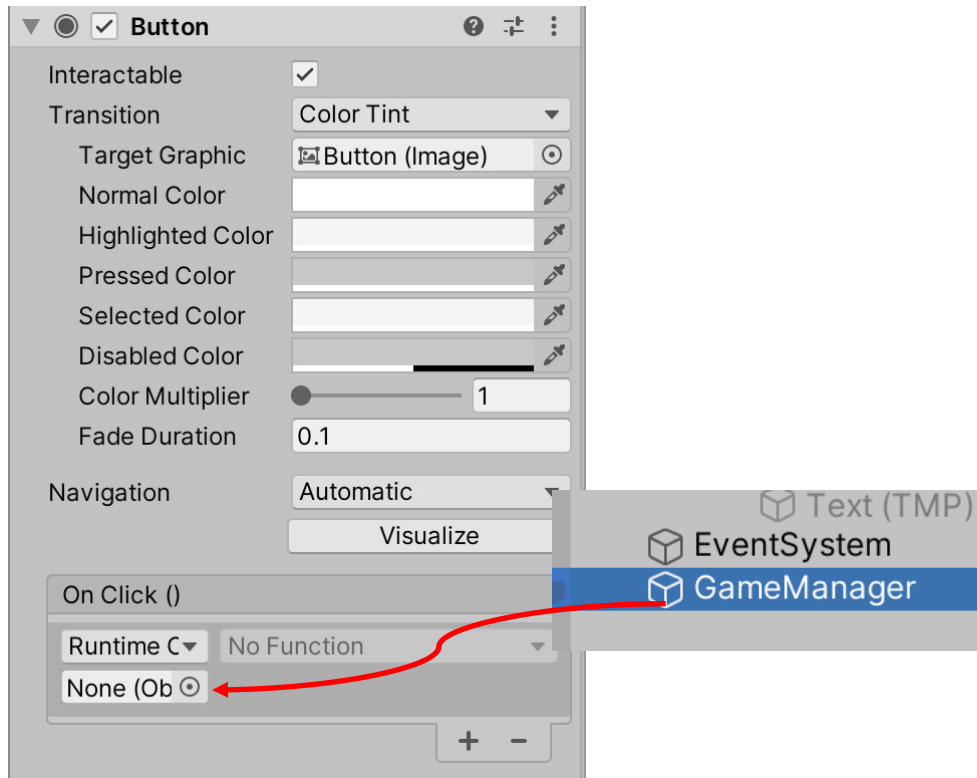
GameManager.cs

```
using UnityEngine.SceneManagement;

public void PlayAgain(){
    SceneManager.LoadScene("SampleScene");
}
```

04 결과 화면

■ PLAY AGAIN 버튼 동작 설정



04 결과 화면

■ Play !!!

