

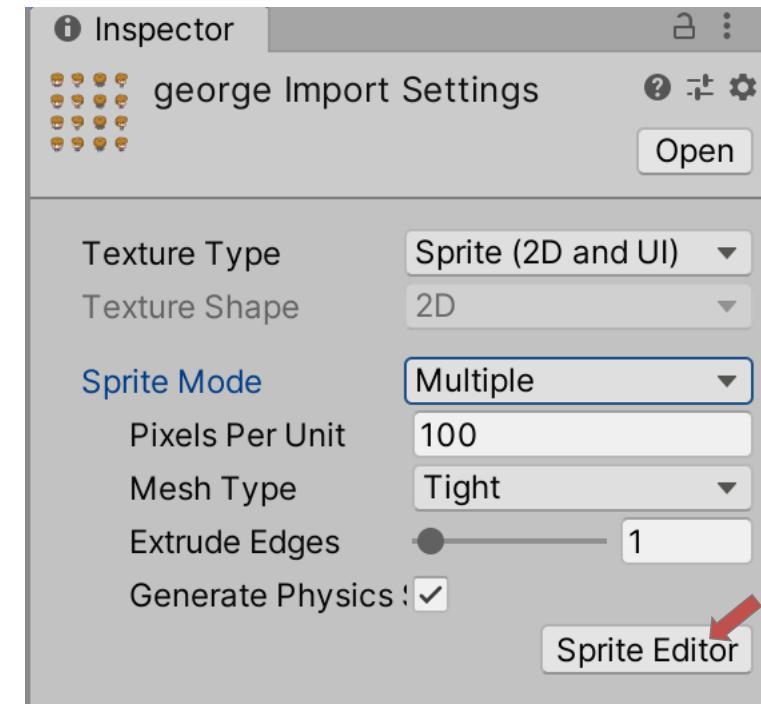
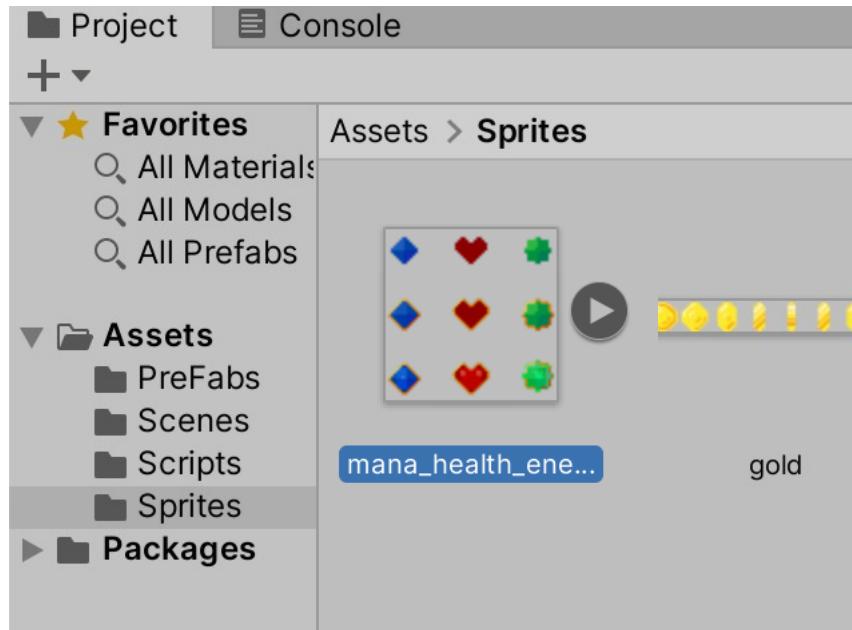
# chapter05

## 적 구현

1. 미사일 만들기
2. 적 만들기
3. 랜덤으로 적 만들기
4. 적 무한 생성하기

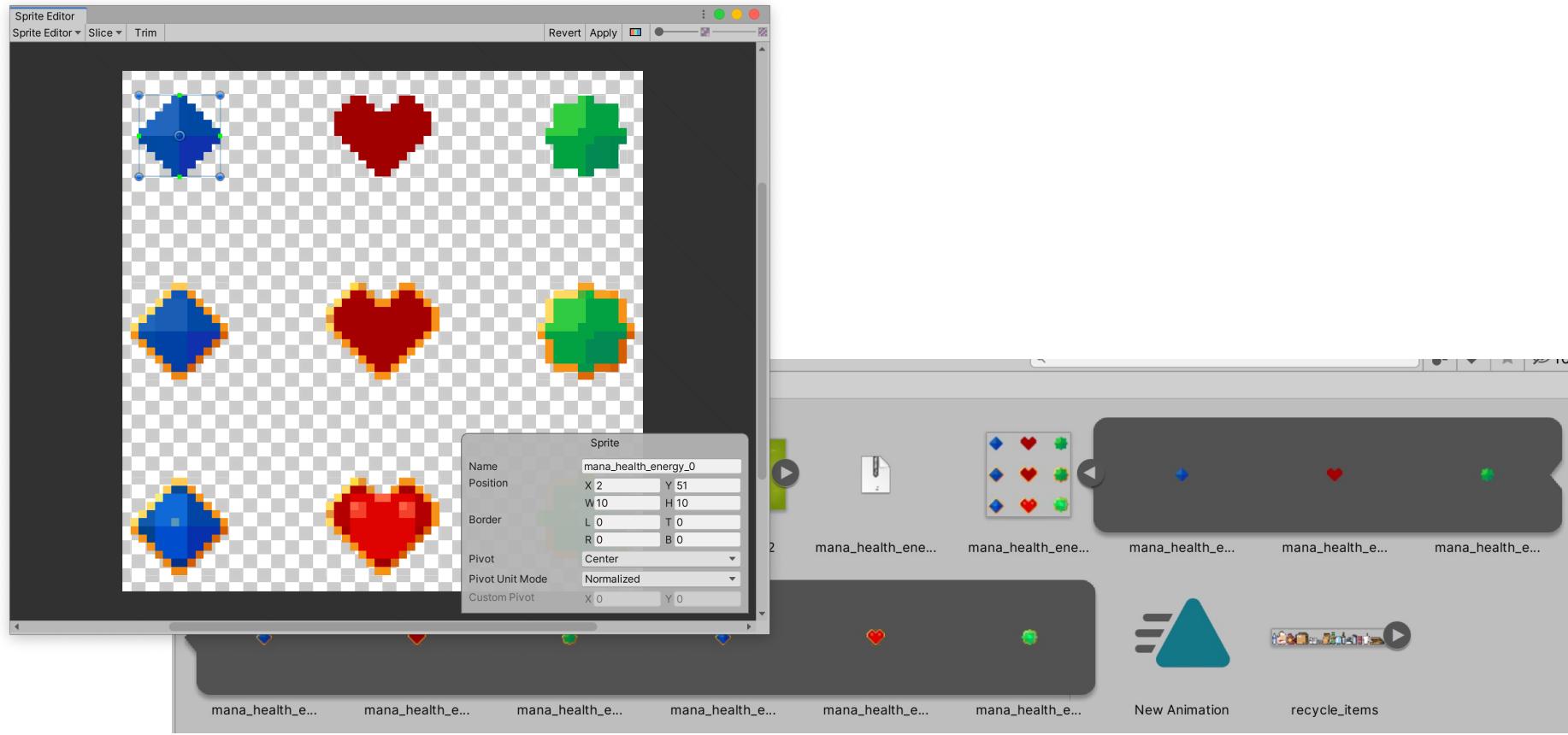
# 01 미사일 만들기

## ■ 미사일로 사용할 이미지 선택 – Inspector에서 Sprite Mode -> Multiple



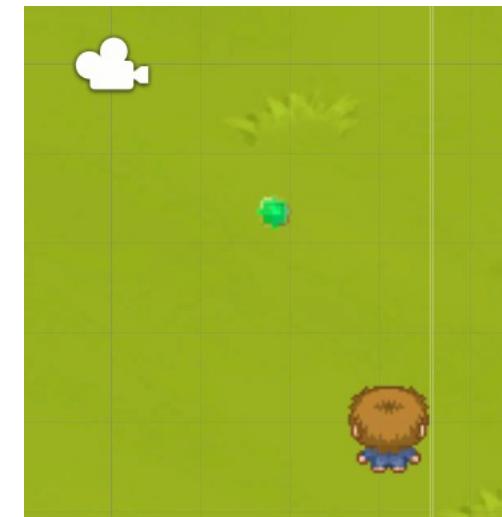
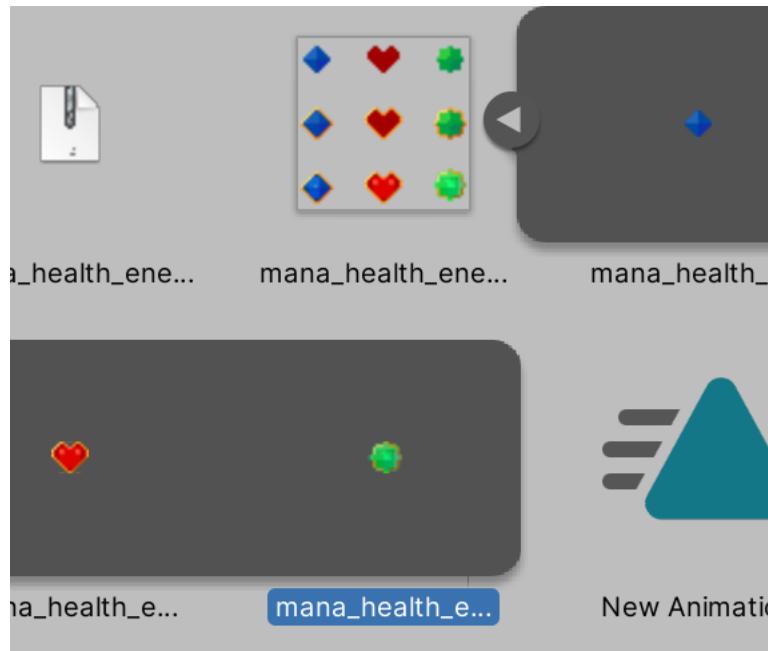
# 01 미사일 만들기

## ■ Sprite Editor – automatic slice 후 apply



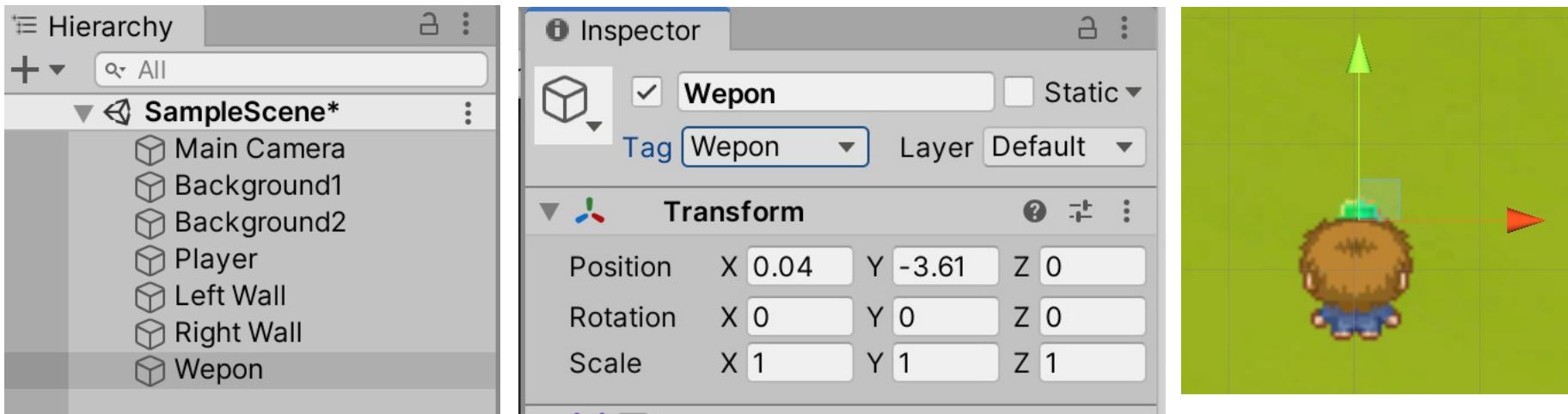
## 01 미사일 만들기

- Pixels Per Unit -> 30
- Order in layer -> 1



# 01 미사일 만들기

- Game Object 명 -> Weapon
- Tag -> Wepon



## 01 미사일 만들기

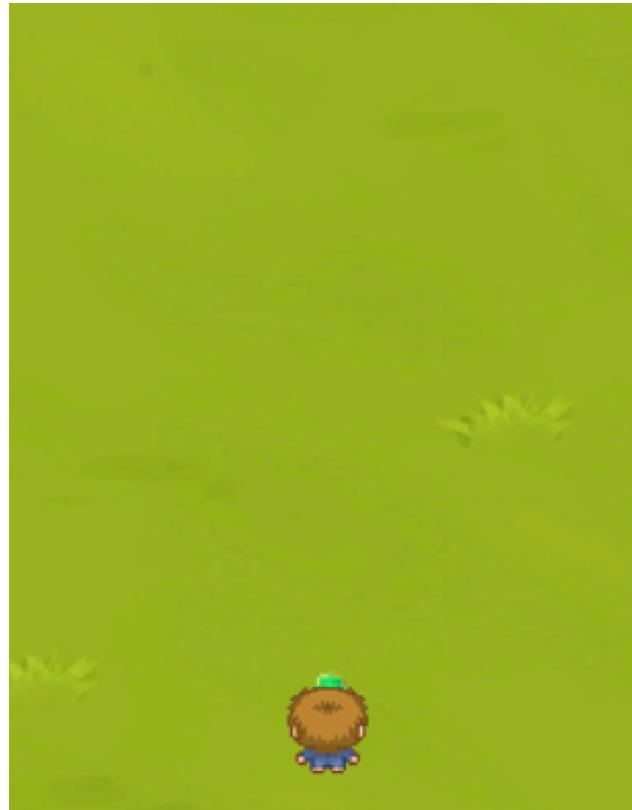
### ■ Script -> Weapon

[SerializeField]

```
private float moveSpeed = 10;  
// Update is called once per frame  
void Update()  
{  
    transform.position += Vector3.up * moveSpeed * Time.deltaTime;  
}
```

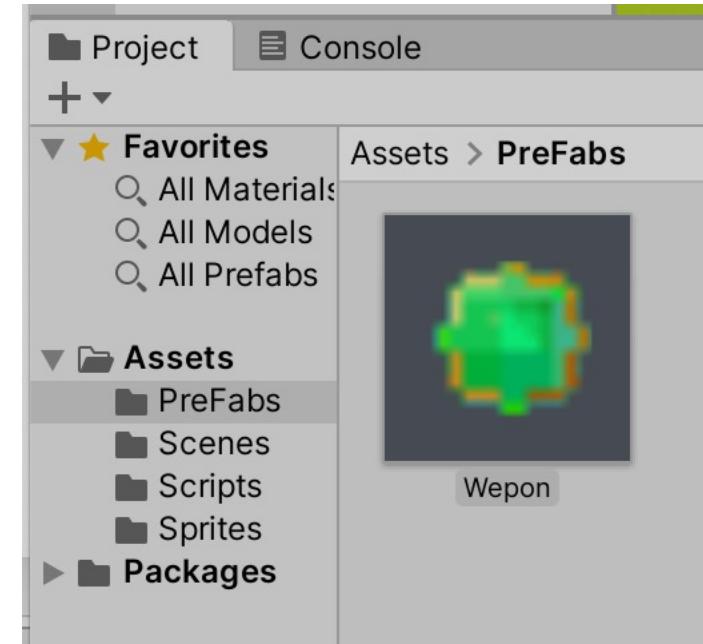
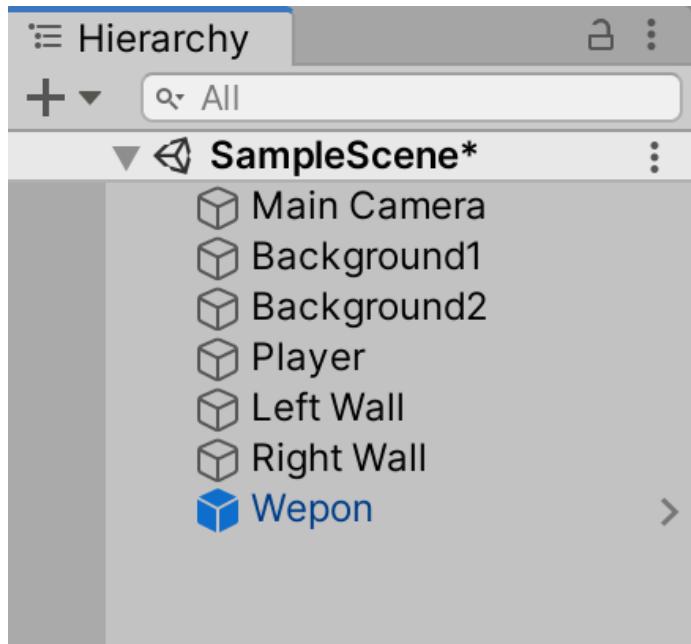
## 01 미사일 만들기

## ■ Play 해보기



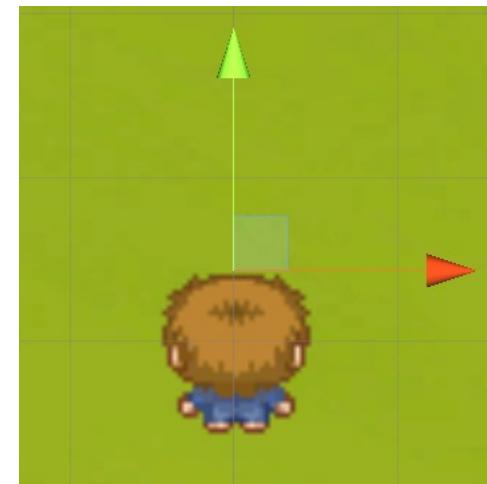
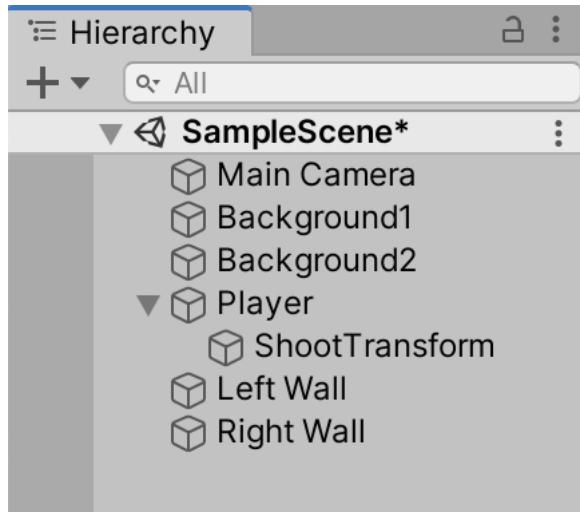
# 01 미사일 만들기

■ Game Object Wepon -> PreFabs로 이동 -> Game Object에서 Wepon 지움



## 01 미사일 만들기

- Game Object 의 Player 하위에 Create Empty 만들고 이름 ShootTransform
- 초기 위치를 Player 머리 위쪽으로 이동



## 01 미사일 만들기

### ■ Player 의 Script 설정

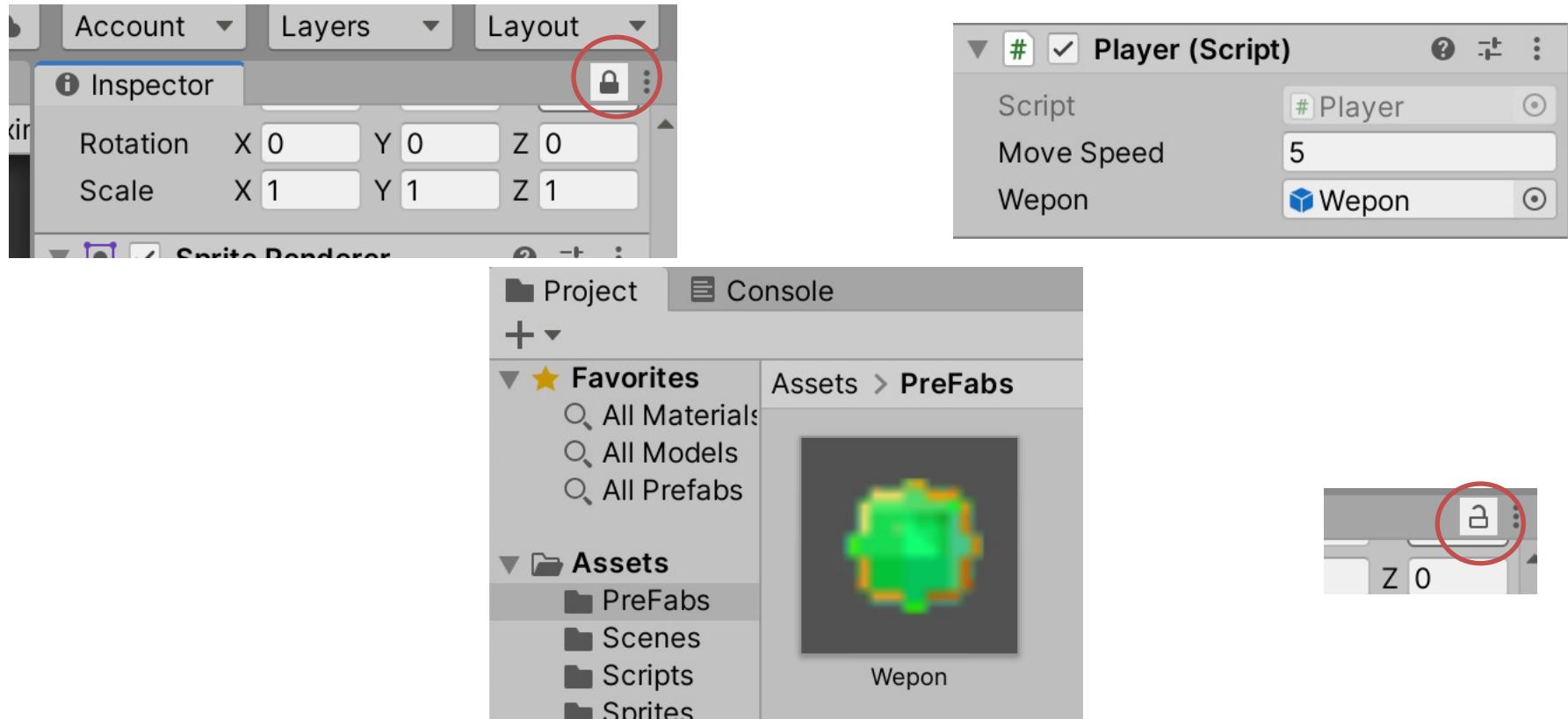
```
public class Player : MonoBehaviour
{
    [SerializeField]
    private float moveSpeed;


    [SerializeField]
    private GameObject wepon;

    // Update is called once per frame
    void Update()
    {
        // float horizontalInput = Input.GetAxisRaw("Horizontal");
        // float verticalInput = Input.GetAxisRaw("Vertical");
        // Vector3 moveTo = new Vector3(horizontalInput, verticalInput, 0f);
    }
}
```

# 01 미사일 만들기

## ■ Prefabs Wepon 을 Player Script 의 속성값으로 사용 가능하도록 등록하기



# 01 미사일 만들기

## ■ Player 의 Script 수정

```
[SerializeField]
private GameObject wepon;

[SerializeField]
private Transform shootTransform;

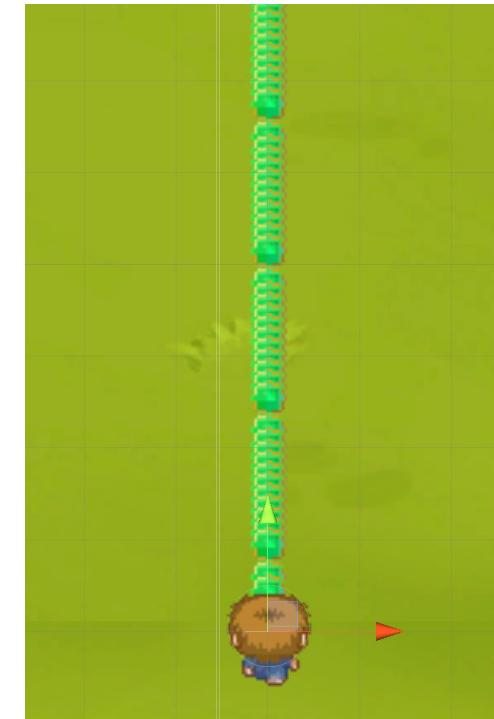
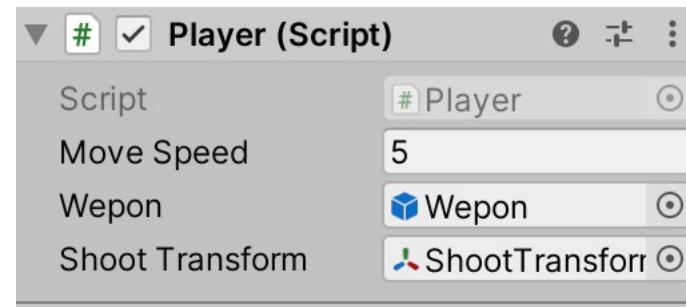
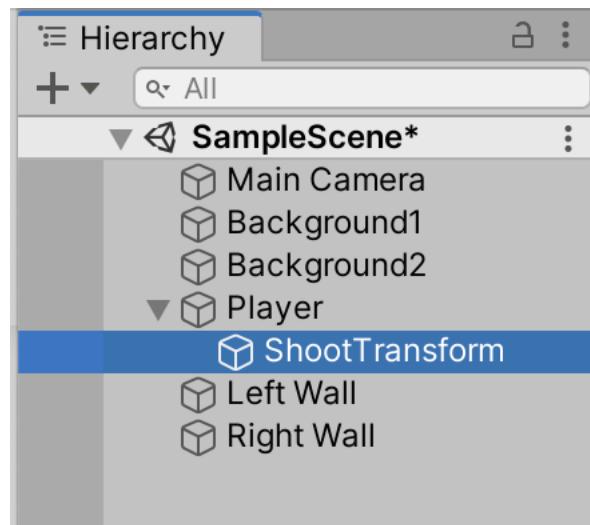
// Update is called once per frame
void Update()
{
    Vector3 mousePos = Camera.main.ScreenToWorldPoint(Input.mousePosition);
    float toX = Mathf.Clamp(mousePos.x, -3.0f, 3.0f);
    transform.position = new Vector3(toX, transform.position.y, transform.position.z);

    Shoot();
}

void Shoot(){
    Instantiate(wepon, shootTransform.position, Quaternion.identity);
}
```

# 01 미사일 만들기

## ■ ShootTransform Object 를 Player의 Script 하위의 Shoot Transform 로 복사



# 01 미사일 만들기

## ■ Player 의 Script 수정

[SerializeField]

```
private GameObject wepon;
```

[SerializeField]

```
private Transform shootTransform;
```

[SerializeField]

```
private float shootInterval = 0.05f;
```

```
private float lastShootTime = 0.0f;
```

```
void Shoot(){
```

```
    if(Time.time - lastShootTime > shootInterval){
```

```
        Instantiate(wepon, shootTransform.position, Quaternion.identity);
```

```
        lastShootTime = Time.time;
```

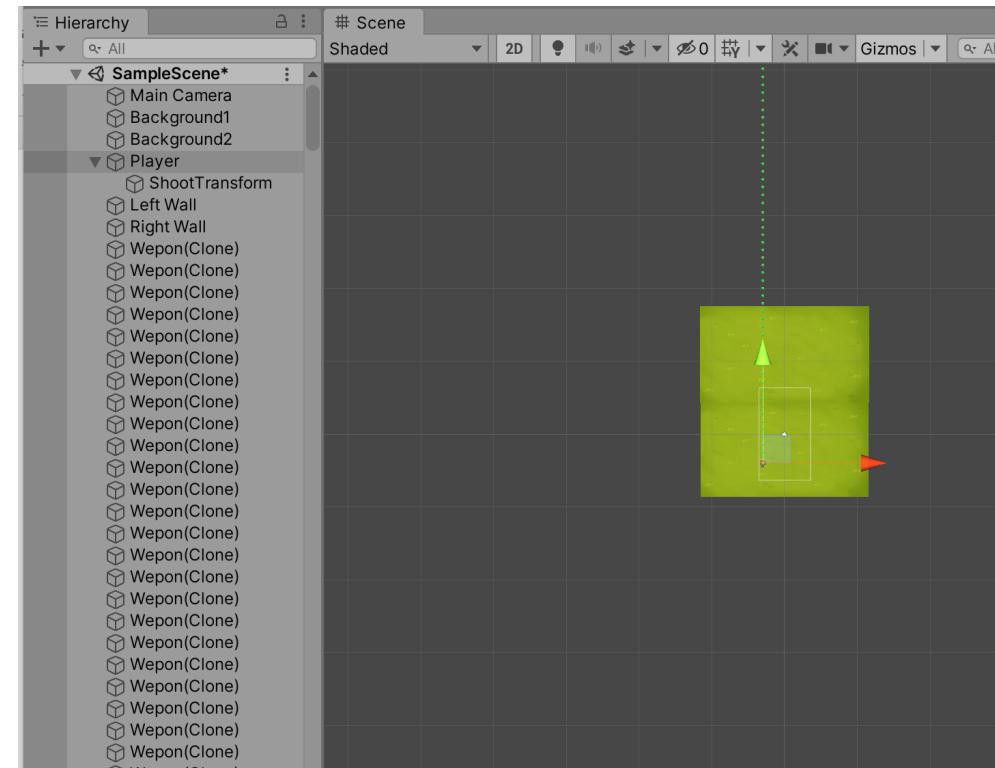
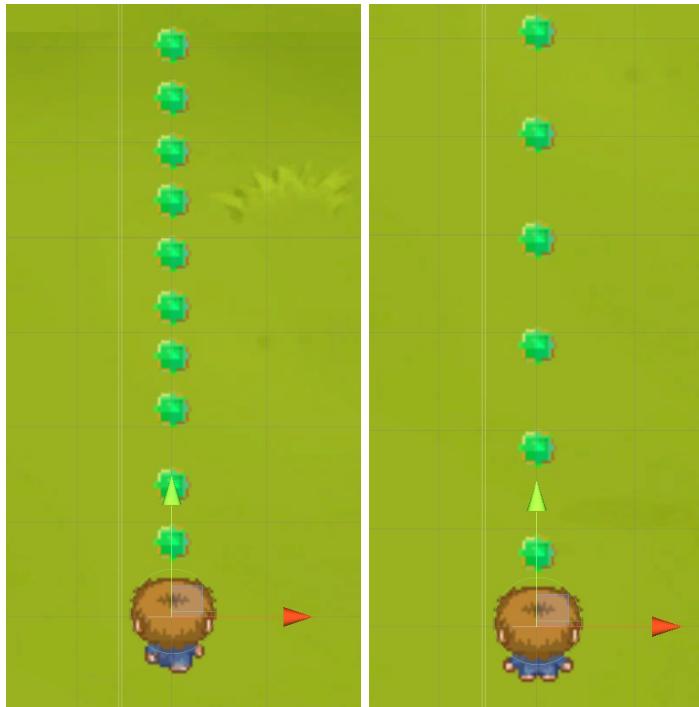
```
}
```

- Instantiate() => Unity가 제공하는 오브젝트 생성용 메소드(오브젝트, 위치, 회전값)
- Quaternion.identity = 회전없음

```
}
```

# 01 미사일 만들기

- Play 확인
- Shoot Interval -> 0.1
- Wepon Object 소멸시키기



# 01 미사일 만들기

## ■ Wepon Script 수정

```
public class Wepon : MonoBehaviour
{
    [SerializeField]
    private float moveSpeed = 10;

    void Start()
    {
        Destroy(gameObject, 1f); // 1초 후 소멸 예약
    }

    // Update is called once per frame
    void Update()
    {
        transform.position += Vector3.up * moveSpeed * Time.deltaTime;
    }
}
```

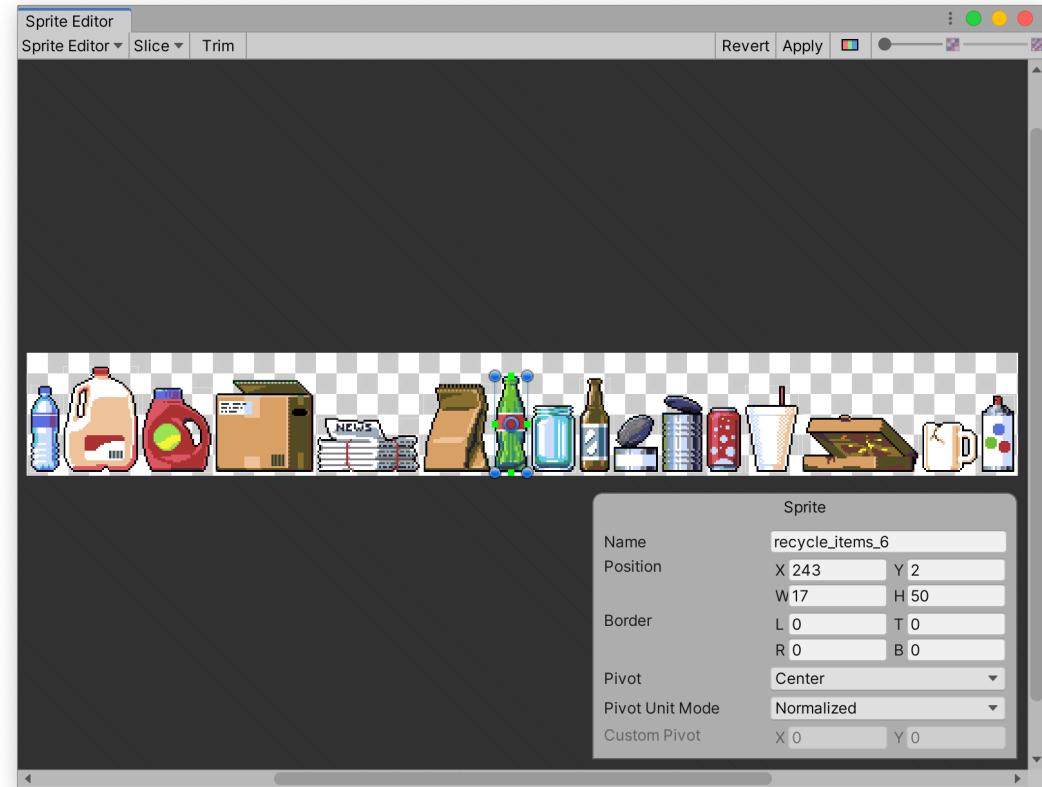
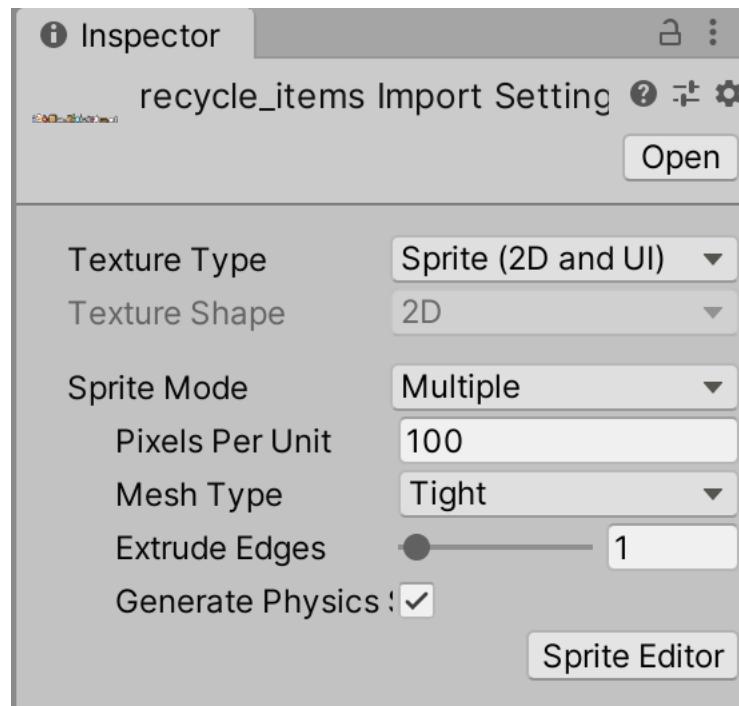
# 01 미사일 만들기

## ■ 소멸 확인



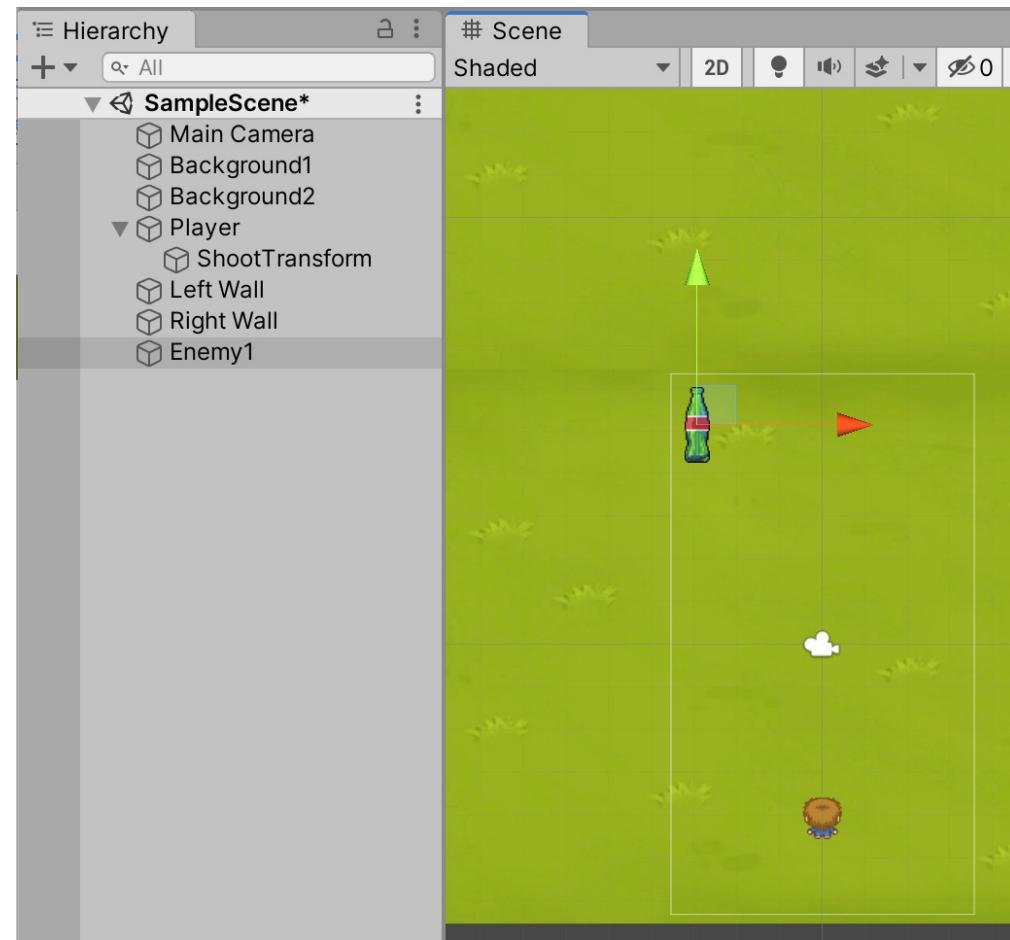
## 02 적 만들기

### ■ 적으로 사용할 이미지 선택 – Inspector에서 Sprite Mode -> Multiple



## 02 적 만들기

- Pixels Per Unit -> 28
- Order in layer -> 1
- Object 명 -> Enemy1



## 02 적 만들기

### ■ Enemy1 Object 에 Scrip Enemy 적용 후 수정

[SerializeField]

```
private float moveSpeed = 10f;  
private float minY = -7f;  
// Update is called once per frame  
void Update()  
{  
    transform.position += Vector3.down * moveSpeed * Time.deltaTime;  
    //게임 플레이 화면 아래쪽을 벗어나면 없어지도록..  
    if(transform.position.y < minY){  
        Destroy(gameObject);  
    }  
}
```

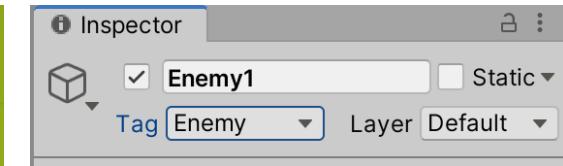
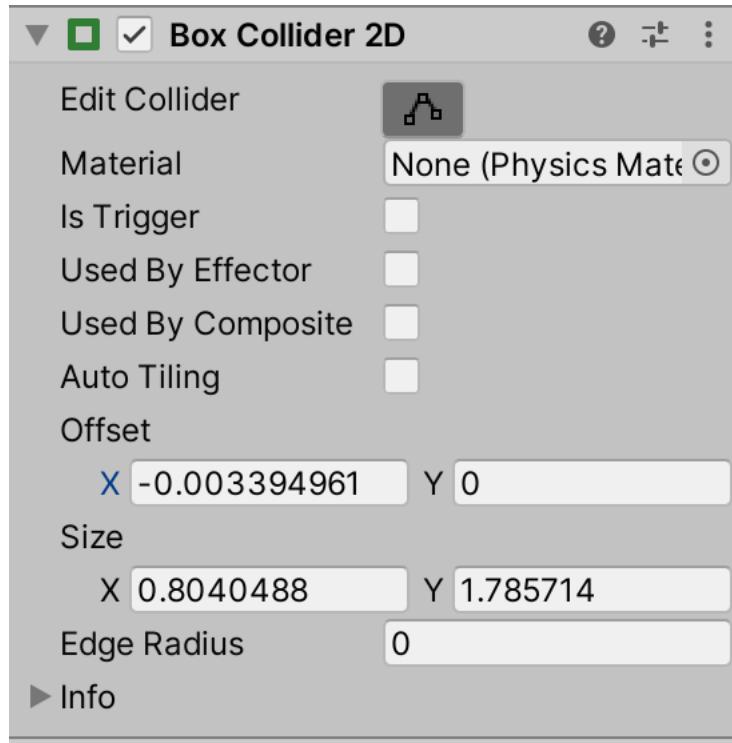
## 02 적 만들기

- Play 해보기
- Move Speed -> 5



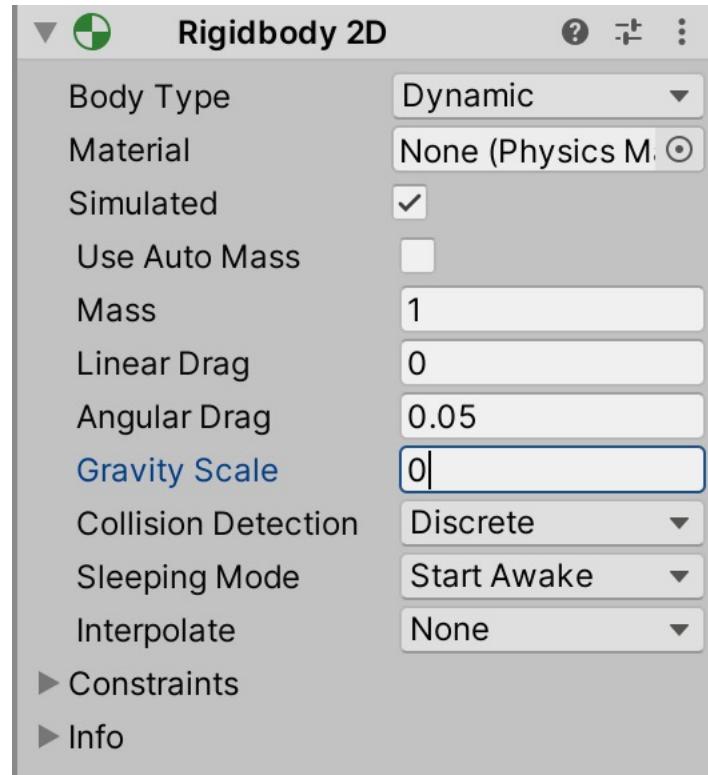
## 02 적 만들기

- 충돌영역 설정 – Enemy1 에 Box Collider 2D 컴포넌트 추가후 영역 수정
- Enemy Tag 지정



## 02 적 만들기

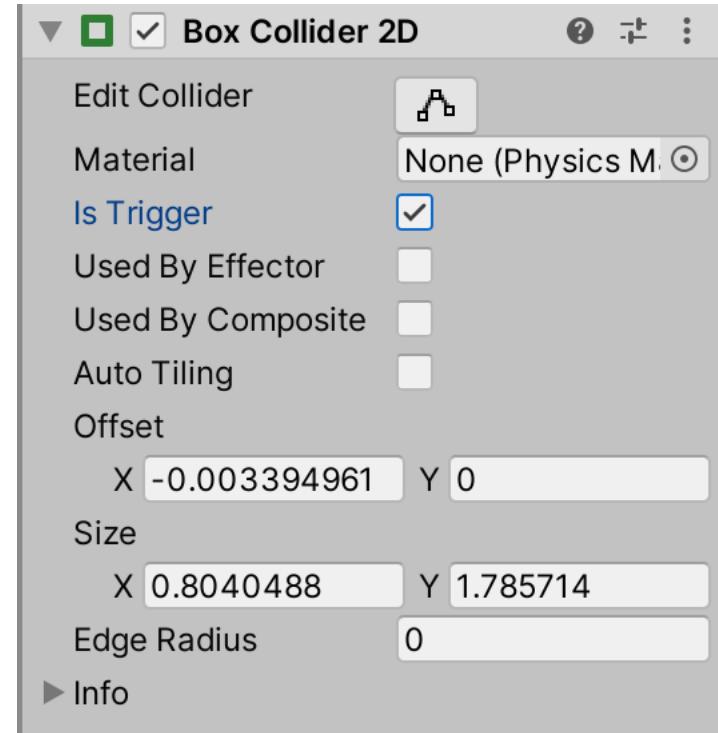
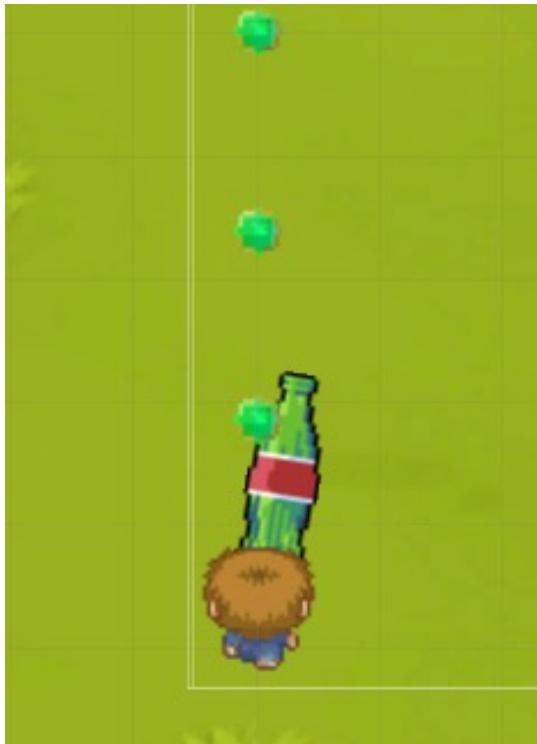
### ■ Enemy1 에 Rigidbody 2D 컴포넌트 추가후 Gravity Scale -> 0



## 02 적 만들기

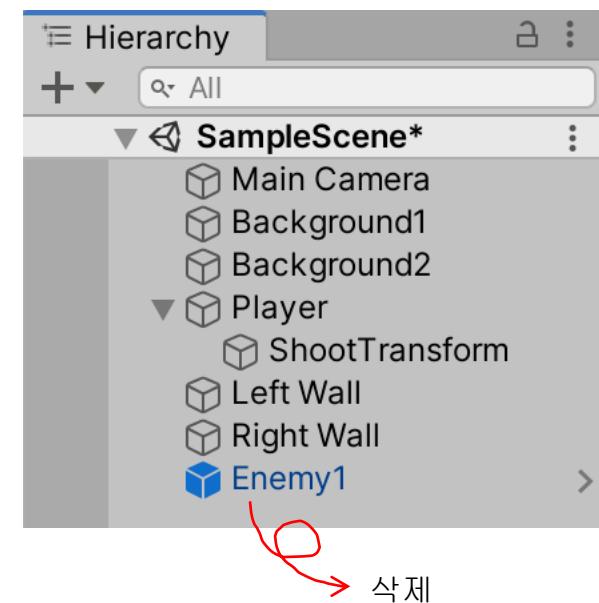
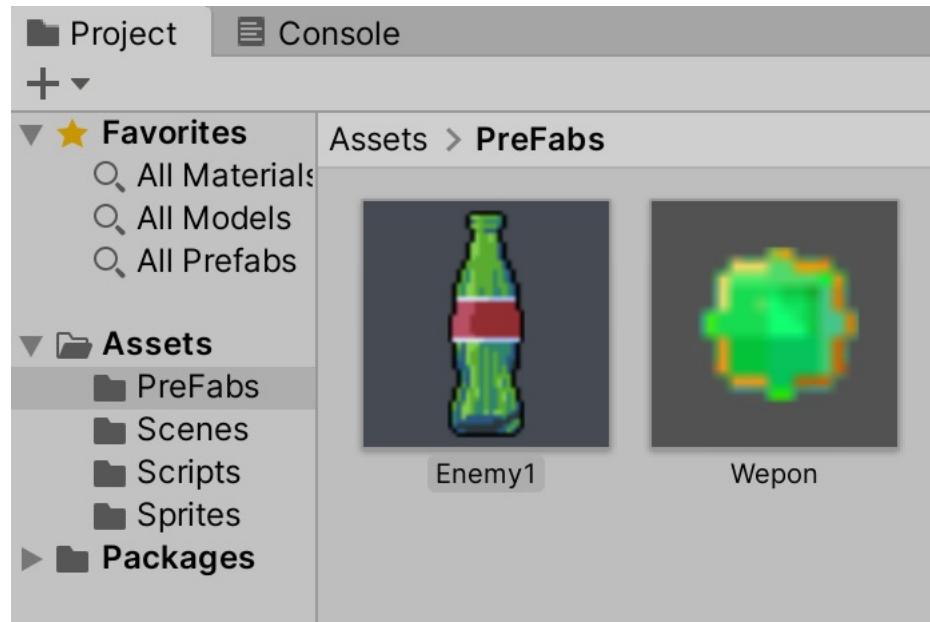
■ Play 해보기

■ Enemy1 의 Box Collider 2D 의 is trigger 체크해두기



## 02 적 만들기

### ■ Enemy1 을 Prefabs 로 준비



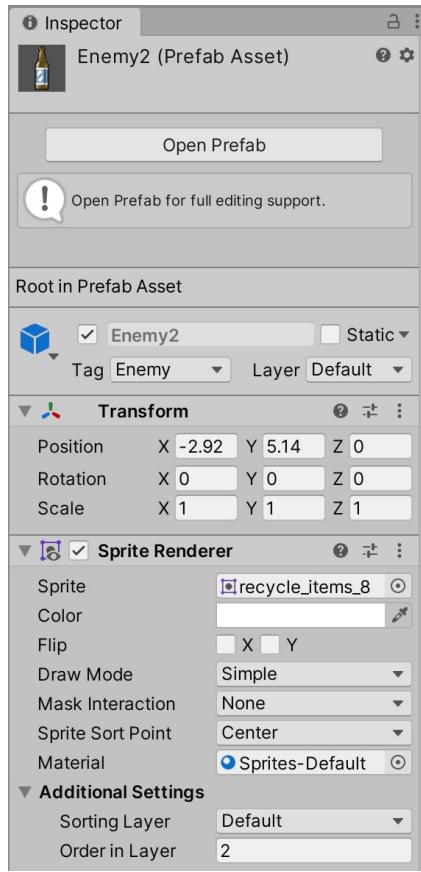
## 02 적 만들기

### ■ Prefabs 의 Enemy1 을 Duplicate ... 7개 준비



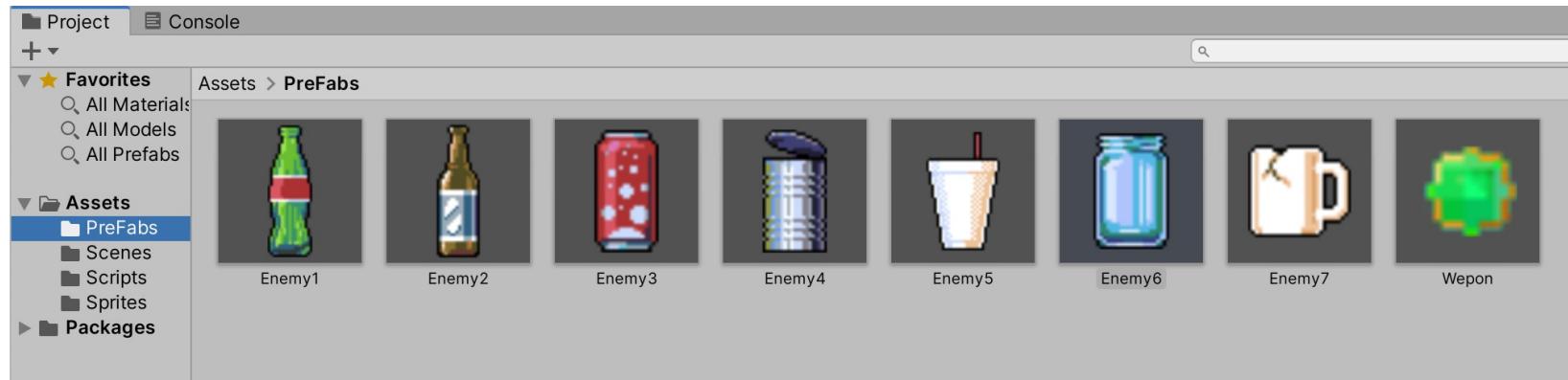
## 02 적 만들기

### ■ Enemy2 선택 후 Sprite Renderer의 Sprite 이미지 교체



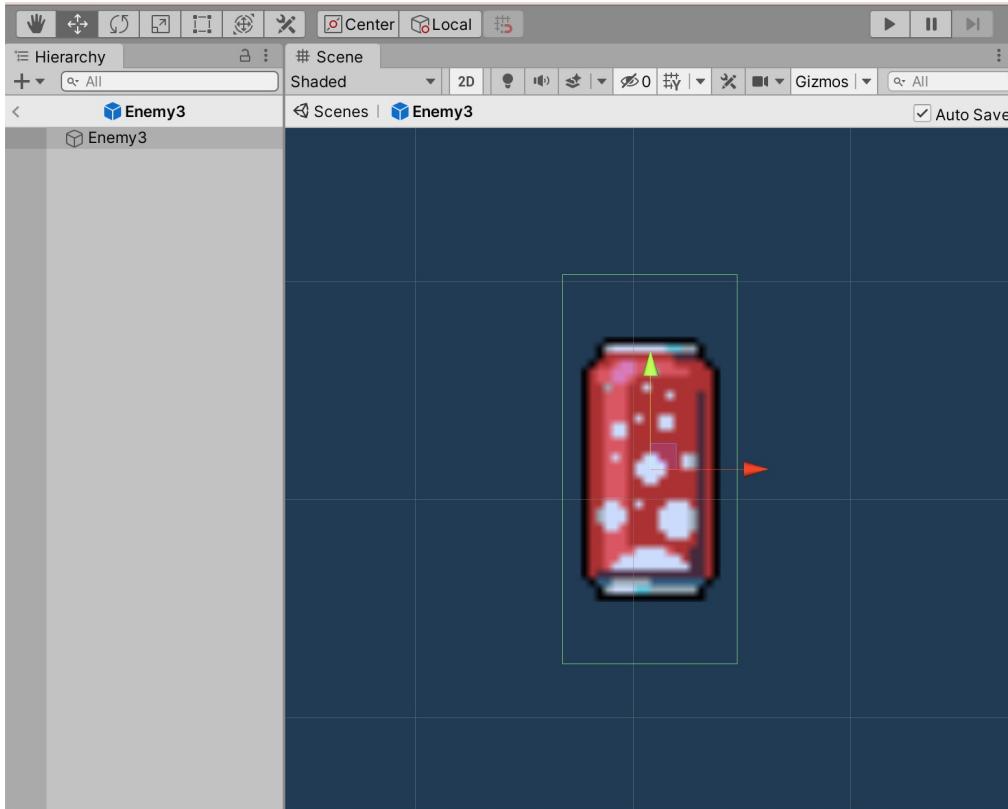
## 02 적 만들기

### ■ Enemy3~7 선택 후 Sprite Renderer 의 Sprite 이미지 교체



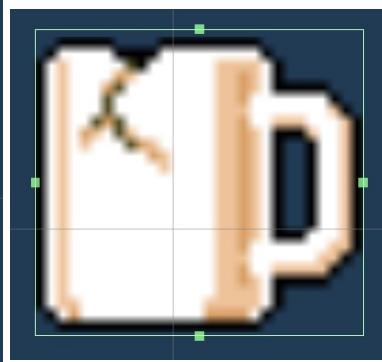
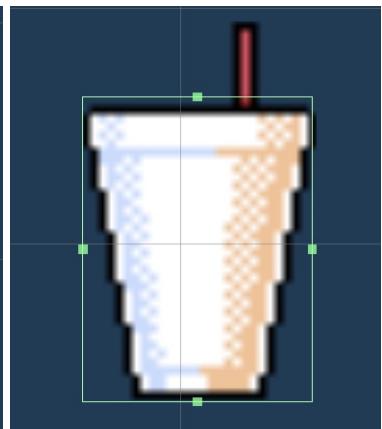
## 02 적 만들기

- 충돌영역 지정
- Prefabs 의 Enemy1 더블클릭 후 확인



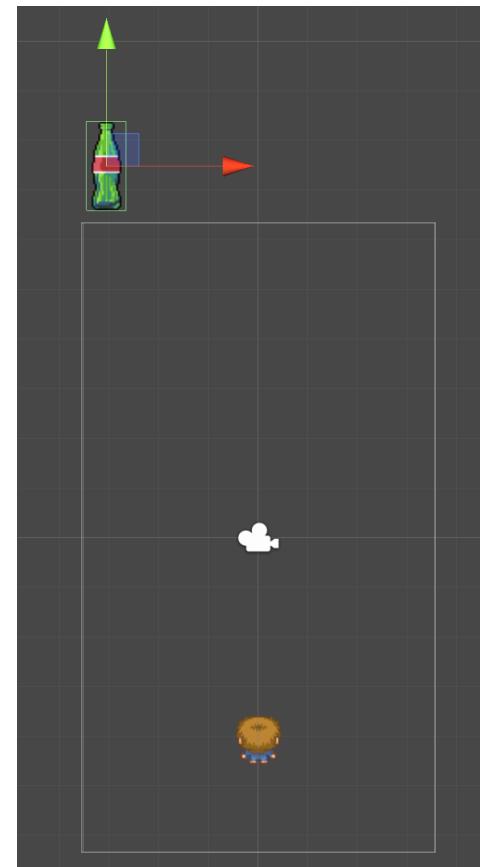
## 02 적 만들기

- 충돌영역 지정
- Prefabs 의 객체를 더블클릭 후 확인



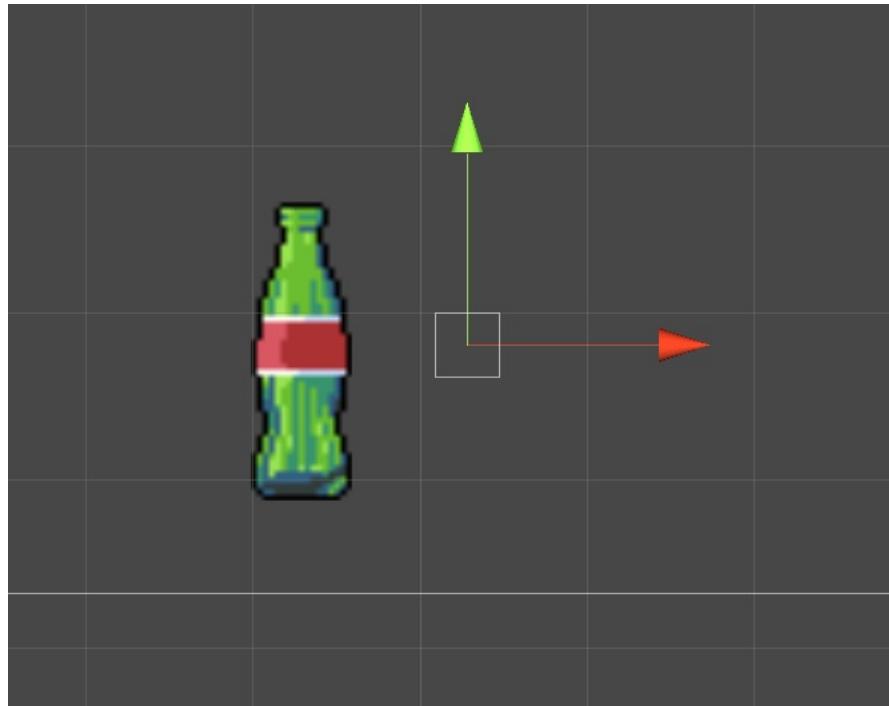
## 03 랜덤으로 적 만들기

### ■ 적이 생성될 위치 확인



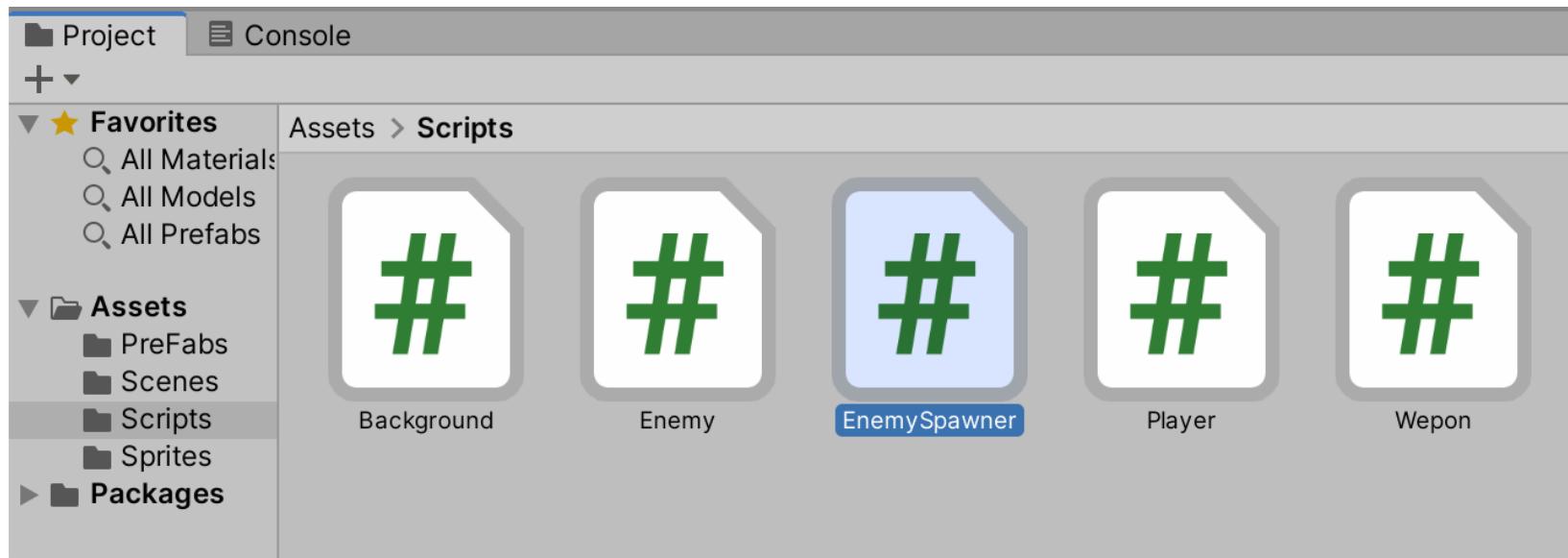
## 03 랜덤으로 적 만들기

### ■ Create Empty 준비 - EnemySpawner



## 03 랜덤으로 적 만들기

### ■ EnemySpawner Script 준비



## 03 랜덤으로 적 만들기

### ■ EnemySpawner Script 수정

```
public class EnemySpawner : MonoBehaviour
{
    [SerializeField]
    private GameObject[] enemies;
    private float[] arrPosX = {-2.2f, -1.1f, 0.0f, 1.1f, 2.2f};

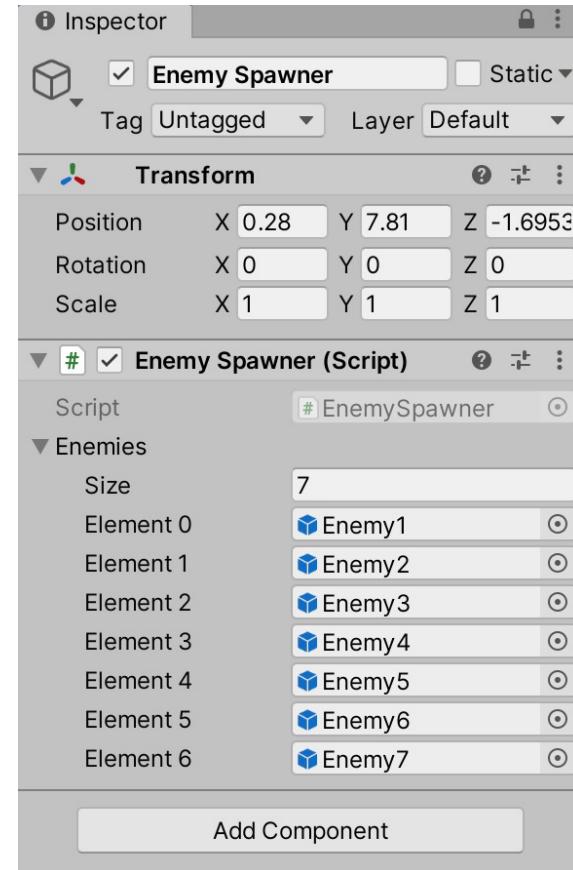
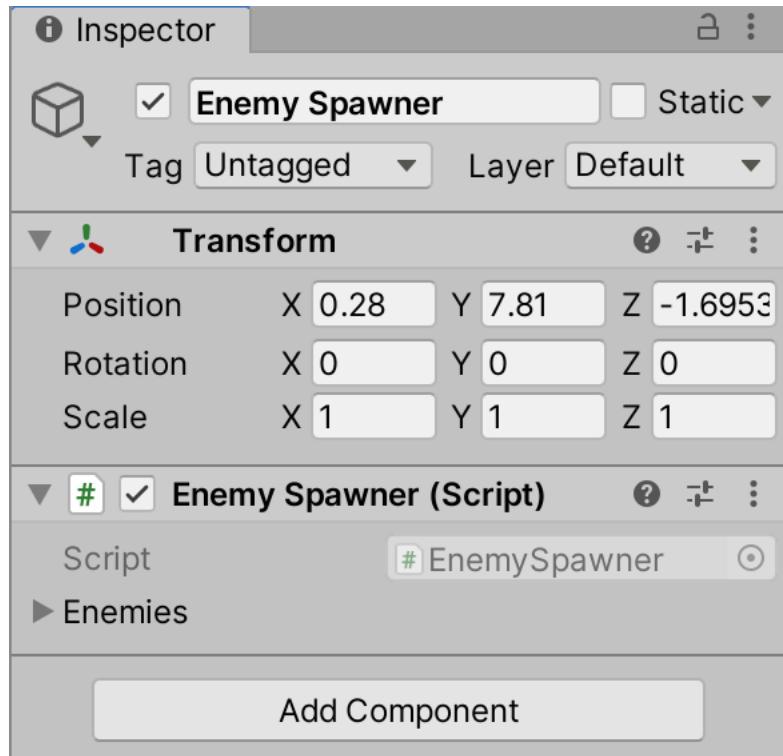
    // Start is called before the first frame update
    void Start()
    {

    }

    void SpawnEnemy(){
    }
}
```

## 03 랜덤으로 적 만들기

### ■ EnemySpawner Script 의 enemies 배열에 요소(Prefabs) 추가



## 03 랜덤으로 적 만들기

### ■ EnemySpawner Script 수정

```
[SerializeField]
private GameObject[] enemies;
private float[] arrPosX = {-2.2f, -1.1f, 0.0f, 1.1f, 2.2f};

// Start is called before the first frame update
void Start()
{
    int index = Random.Range(0, enemies.Length);
    SpawnEnemy(arrPosX[0], index);
}

void SpawnEnemy(float posX, int index){
    Vector3 spawnPos = new Vector3(posX, transform.position.y, transform.position.z);
    Instantiate(enemies[index], spawnPos, Quaternion.identity);
}
```

## 03 랜덤으로 적 만들기

### ■ Play 해보기



## 03 랜덤으로 적 만들기

### ■ EnemySpawner Script 수정

```
// Start is called before the first frame update
void Start()
{
    foreach(float posX in arrPosX){
        int index = Random.Range(0, enemies.Length);
        SpawnEnemy(posX, index);
    }
}
```



# 04 적 무한생성 하기

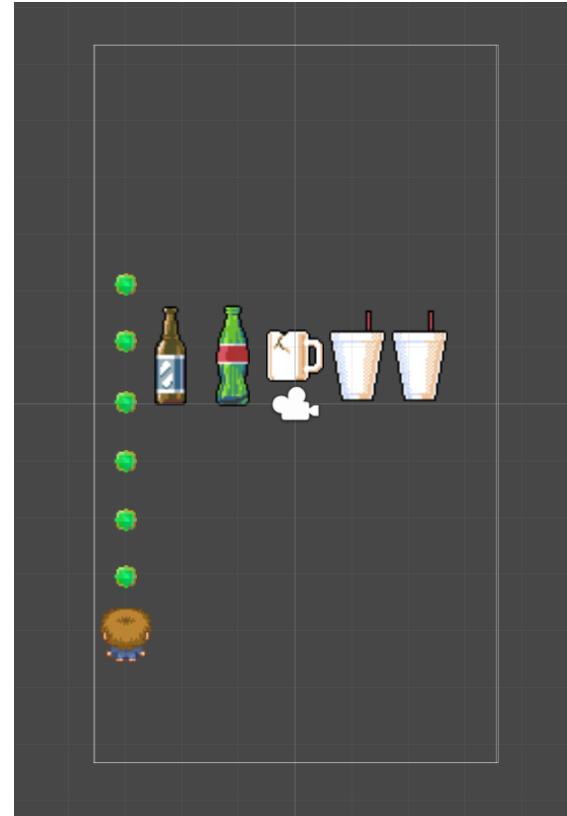
## ■ Script 수정

```
[SerializeField]
private GameObject[] enemies;
private float[] arrPosX = {-2.2f, -1.1f, 0.0f, 1.1f, 2.2f};
[SerializeField]
private float spawnInterval = 1.5f;
// Start is called before the first frame update
void Start()
{
    StartEnemyRoutine();
}

void StartEnemyRoutine(){
    StartCoroutine("EnemyRoutine");
}

IEnumerator EnemyRoutine(){
    yield return new WaitForSeconds(3f);
    while(true){
        foreach(float posX in arrPosX){
            int index = Random.Range(0, enemies.Length);
            SpawnEnemy(posX, index);
        }
        yield return new WaitForSeconds(spawnInterval);
    }
}

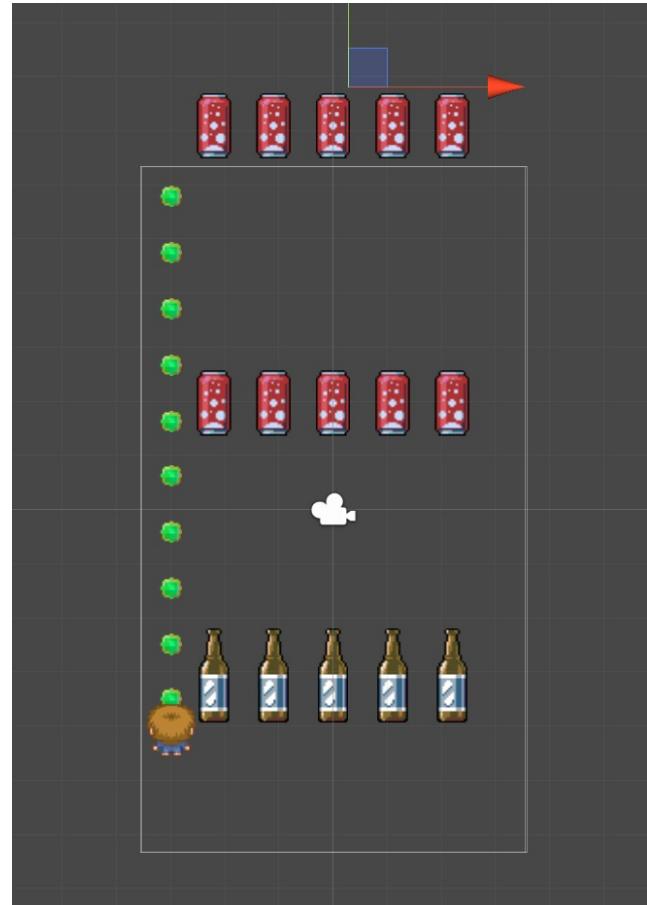
void SpawnEnemy(float posX, int index){
    Vector3 spawnPos = new Vector3(posX, transform.position.y, transform.position.z);
    Instantiate(enemies[index], spawnPos, Quaternion.identity);
}
```



## 04 적 무한생성 하기

### ■ Script 수정

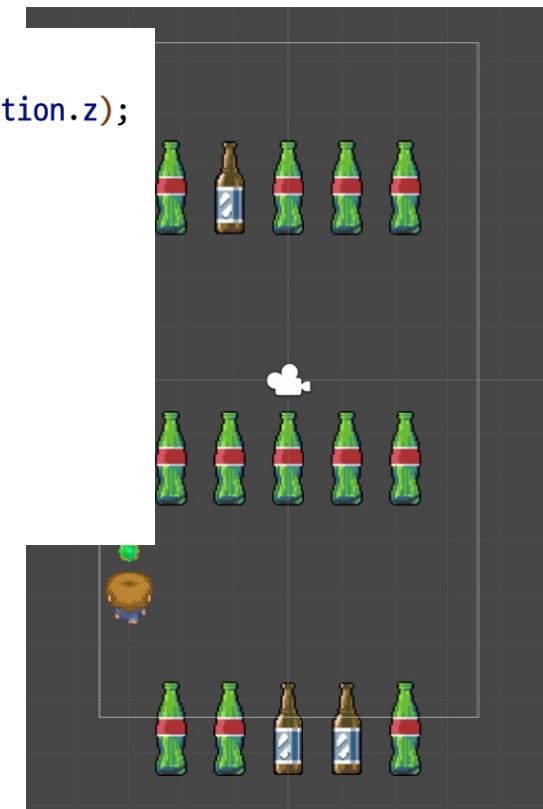
```
IEnumerator EnemyRoutine(){
    yield return new WaitForSeconds(3f);
    int spawnCount = 0;
    int enemyIndex = 0;
    while(true){
        foreach(float posX in arrPosX){
            SpawnEnemy(posX, enemyIndex);
        }
        spawnCount++;
        if(spawnCount % 10 == 0){
            enemyIndex++;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



## 04 적 무한생성 하기

### ■ Script 수정 ■ 오류 방어 코드 및 적 랜덤 변경

```
void SpawnEnemy(float posX, int index){  
    Vector3 spawnPos = new Vector3(posX, transform.position.y, transform.position.z);  
    if(Random.Range(0,5) == 0){ // 0,1,2,3,4 -> 0 20% 확률  
        index++;  
    }  
    if(index >= enemies.Length){  
        index = enemies.Length - 1;  
    }  
    Instantiate(enemies[index], spawnPos, Quaternion.identity);  
}
```



# 04 적 무한생성 하기

## ■ Script 수정 ■ 적 이동속도 변경

The screenshot shows the Unity Editor interface. On the left is the Project window, displaying a file tree for a project named 'StarShoot'. The 'Assets/Scripts' folder contains several C# scripts: Background.cs, Enemy.cs (which is currently selected), EnemySpawner.cs, Player.cs, Wepon.cs, ridgidbody.cs, and a few others. Below these are .vscode, Assets, PreFab, Scenes, and Scripts folders. The Scripts folder contains Background.cs, Enemy.cs, EnemySpawner.cs, Player.cs, and Wepon.cs. At the bottom are Logs, Packages, and Assembly-CSharp.csproj. The right side of the screen is the Editor window, showing the code for 'Enemy.cs'. The code defines a MonoBehaviour class 'Enemy' with a private float variable 'moveSpeed' set to 10f, and a private float variable 'minY' set to -7f. It includes a SetMoveSpeed method and an Update method that moves the transform downwards and destroys the object if it goes below the minY value.

```
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     // Update is called once per frame
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 }
```

# 04 적 무한생성 하기

## ■ Script 수정 ■ 적 이동속도 변경

탐색기 ...

열려 있는 편집기

Assets > Scripts > EnemySpawner.cs

```

20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
    
```

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;
    }
    yield return new WaitForSeconds(spawnInterval);
}
}

void SpawnEnemy(float posX, int index, float moveSpeed){
    Vector3 spawnPos = new Vector3(posX, transform.position.y, transform.position.z);
    if(Random.Range(0,5) == 0){ // 0,1,2,3,4 -> 0 ~ 20% 확률
        index++;
    }
    if(index >= enemies.Length){
        index = enemies.Length - 1;
    }
    GameObject enemyObject = Instantiate(enemies[index], spawnPos, Quaternion.identity);
    Enemy enemy = enemyObject.GetComponent<Enemy>();
    enemy.SetMoveSpeed(moveSpeed);
}
    
```

## 04 적 무한생성 하기

### ■ Play 해서 확인해보기

