BoxRunner

제작 매뉴얼

unity

1. 유니티 에디터

unity

뷰포트에서 가장 많이 쓰이는 컨트롤

화면 컨트롤

회전	Alt + Drag
이동	Drag
확대,축소	Wheel

ctrl+ shift +F

unity

뷰포트에서 가장 많이 쓰이는 컨트롤

오브젝트 컨트롤

이동	W , Drag
회전	E , Drag
스케일	R , Drag

unity

2. 스크립트개념

unity

유니티는 크게
두가지의 개념을 가지고 있습니다.

오브젝트

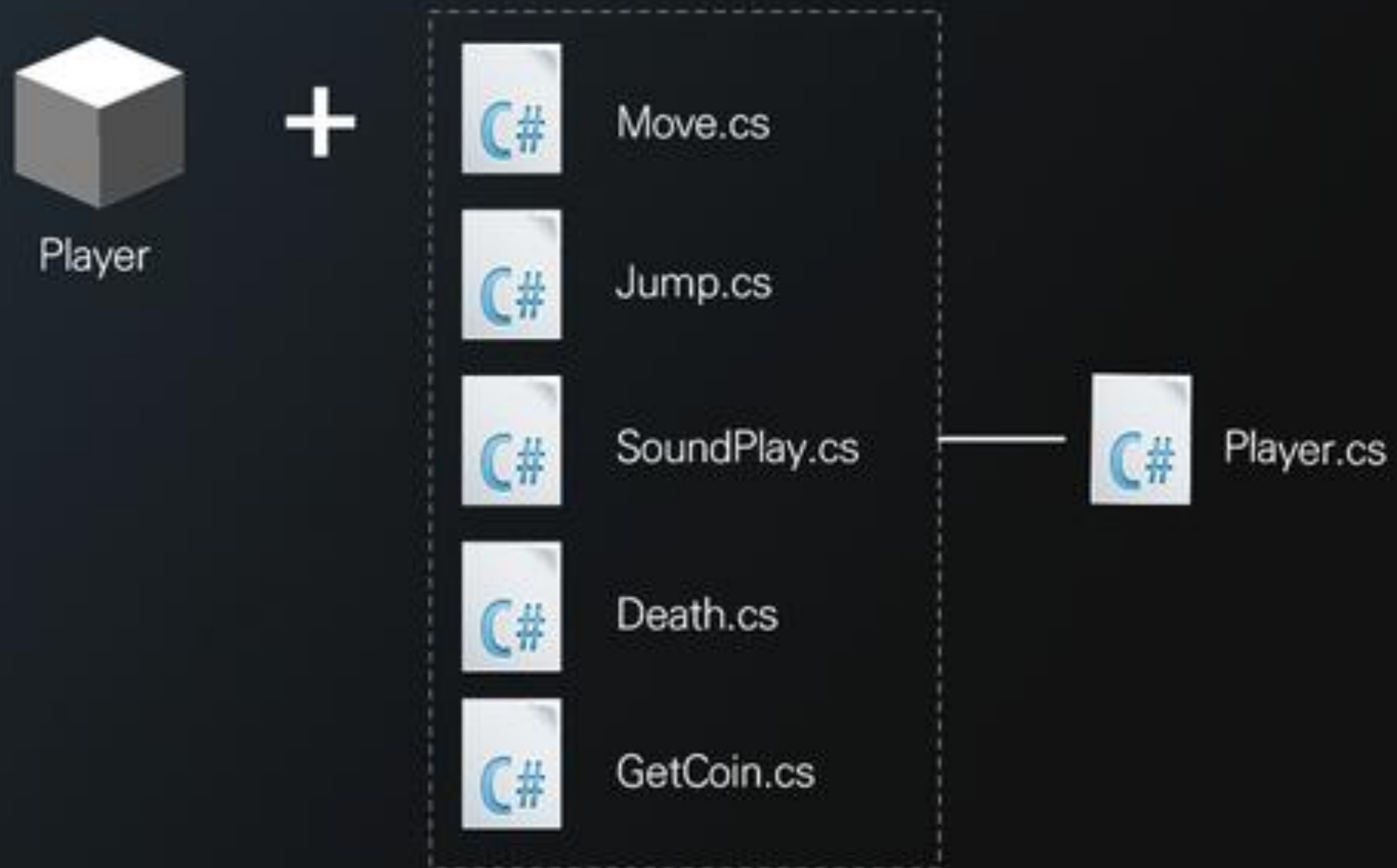
뷰포트상에 보여지는 존재

스크립트

안보이는 곳에서 돌아가고 있는 코드

unity

보통의 구조는 이렇습니다.



unity

스크립트는 크게 **두가지의 개념**을 가지고 있습니다.

변수 함수

unity

변수 명사

이름, 나이, 키, 몸무게,
달리기속도, 체력, 연애횟수,
학년 등등등...

함수 동사

말하다, 때리다, 먹다, 싸다,
부딪히다, 점프하다, 달리다
등등등...

unity

함수

인사

A라는 버튼을 누르면, "안녕하세영"이라고 대답하라.

변수 변수 함수

(실제로 인사 만들어보기, Start와 Update 개념 이해)

함수이름

_____ 을 누르면, _____ 로 _____ 하라

변수 변수 함수

이동 ---- 오른쪽버튼을 누르면, "오른쪽으로 1" 만큼 움직여라.

왼쪽버튼을 누르면, "왼쪽으로 1" 만큼 움직여라.

점프 ---- 점프버튼을 누르면, "위쪽으로 10" 만큼 움직여라.

공격 ---- 공격버튼을 누르면, "-1"로 적의 Hp를 깎아라.

배변 ---- 배의 Hp가 0 이면, X를 분출하라.

⋮

unity

게임 소개

원드러너, 쿠키런 등과 유사한 횡스크롤 무한 러닝게임.
오직 점프 하나만으로 가장 오랫동안 살아남아 많은 동전을 획득하는 게임을 만들어 봅니다.

게임개요

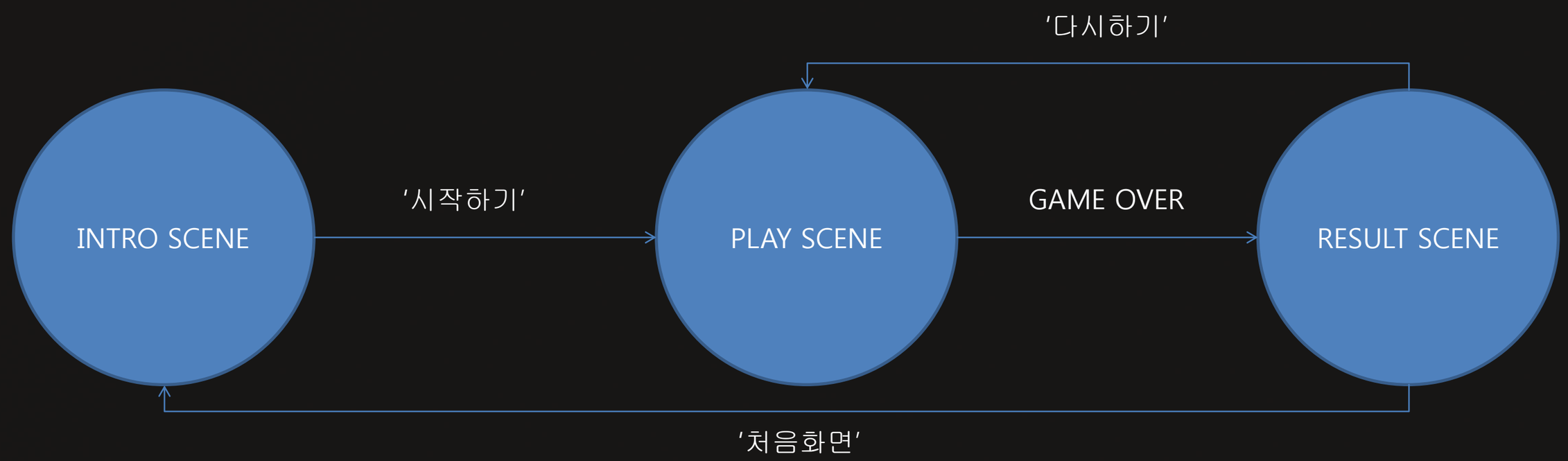
목적 ---------- 코인을 최대한 많이 획득하고,
오랫동안 살아남는다.

레벨디자인요소 -------- 점점 게임진행의 스피드가 빨라진다.

게임오버 조건 ---------- 바닥에 떨어지면 죽는다.

주인공의 기능 ---------- 점프, 2단점프

게임구조


'다시하기'
INTRO SCENE
'시작하기'
PLAY SCENE
GAME OVER
RESULT SCENE
'처음화면'

게임내 전체 요소

배경

원거리	근거리

플레이어

움직임	충돌	2D애니

효과

사운드	페이드	카메라

GUI

게임화면	결과화면	처음화면

좌측에 있는 다양한 요소들이
하나씩 모여 하나의 게임이 됩니다.

unity

제작과정

Step 01. 반복되는 배경화면 만들기

Step 02. 반복되서 생성되는 바닥블럭 만들기

Step 03. 주인공 캐릭터 만들기

Step 04. 카메라 줌인줌아웃 효과적용

Step 05. GUI 와 게임매니져 만들기

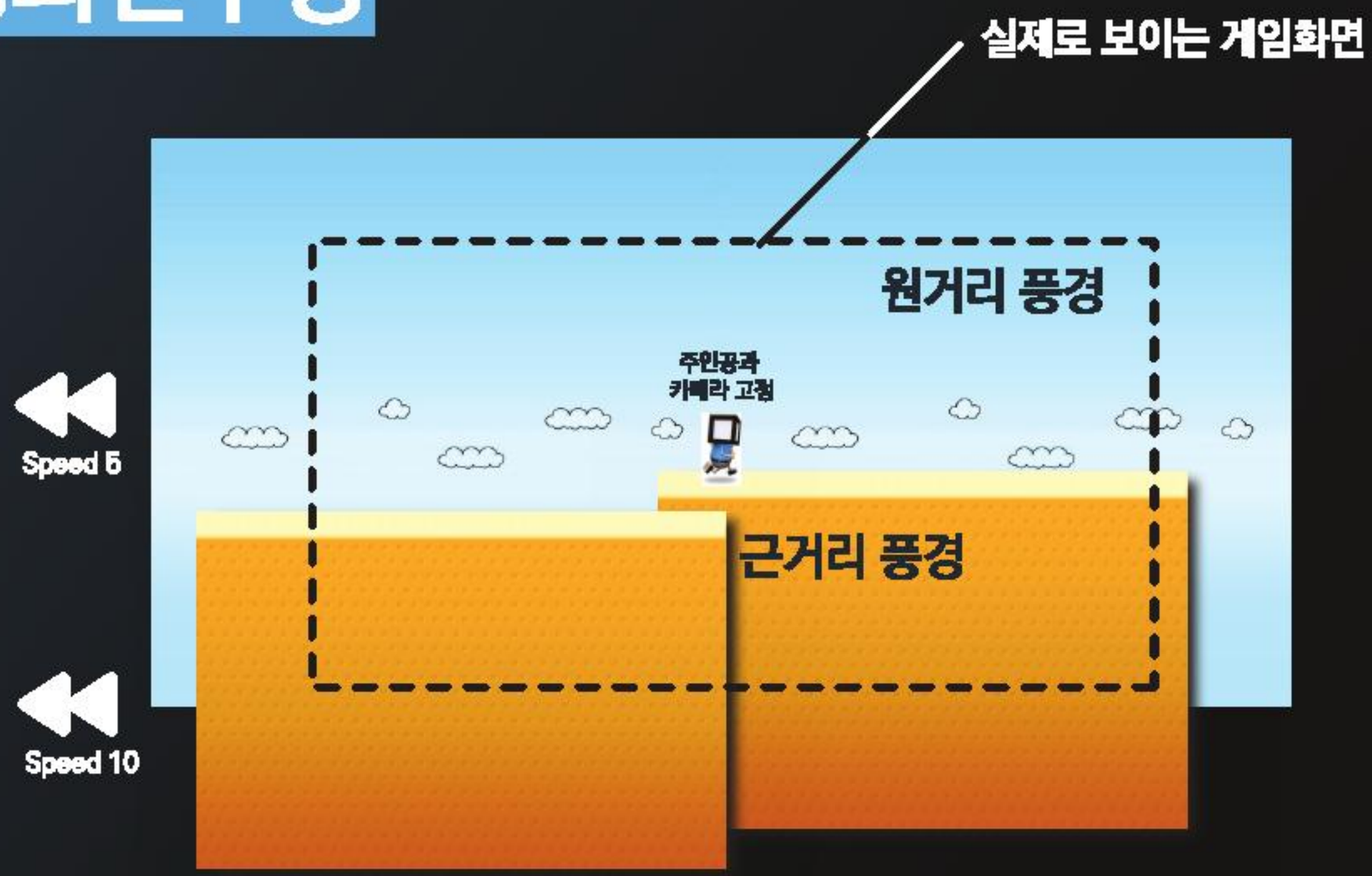
Step 06. Intro씬 만들고 최종 빌드형태로 구성하기

unity

STEP.1 반복되는 배경하늘 만들기

unity

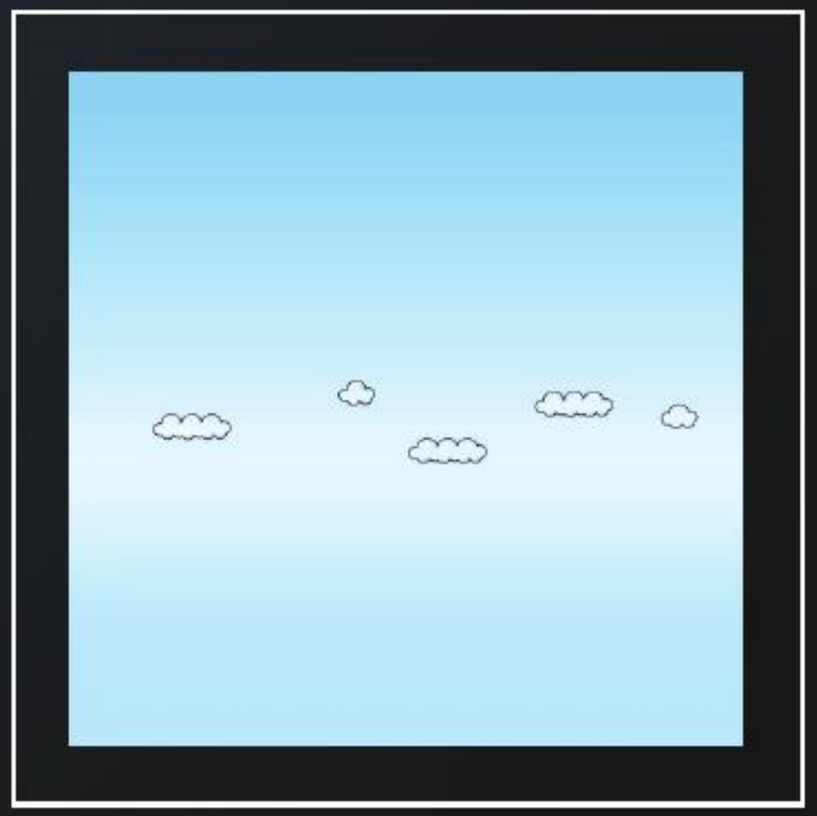
배경화면구성



배경화면을 두가지로 나눠 속도를 조절함으로써
조금 더 게임에 깊이감을 줄 수 있습니다.

STEP.1 반복되는 배경하늘 만들기

일단 게임의 배경으로 쓰이는 하늘을 만들어봅니다.



Object

+

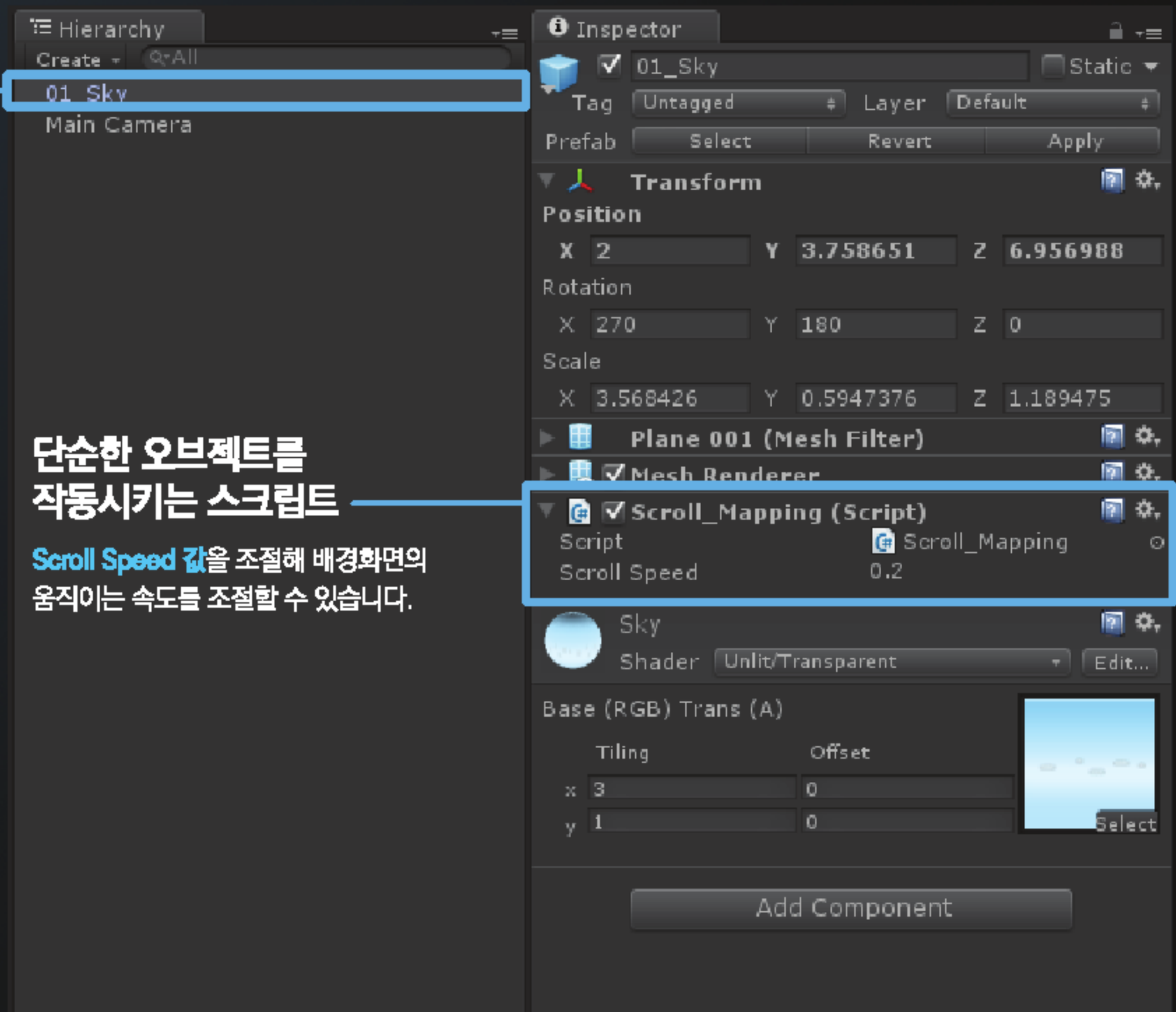
Scroll_Mapping.cs

배경화면 텍스쳐의 Offset값 을 계속,
Speed 만큼 움직여라.

Script

unity

게임화면에 사용하기 위해 배치 된 오브젝트



단순한 오브젝트를 작동시키는 스크립트

Scroll Speed 값을 조절해 배경화면의 움직이는 속도를 조절할 수 있습니다.

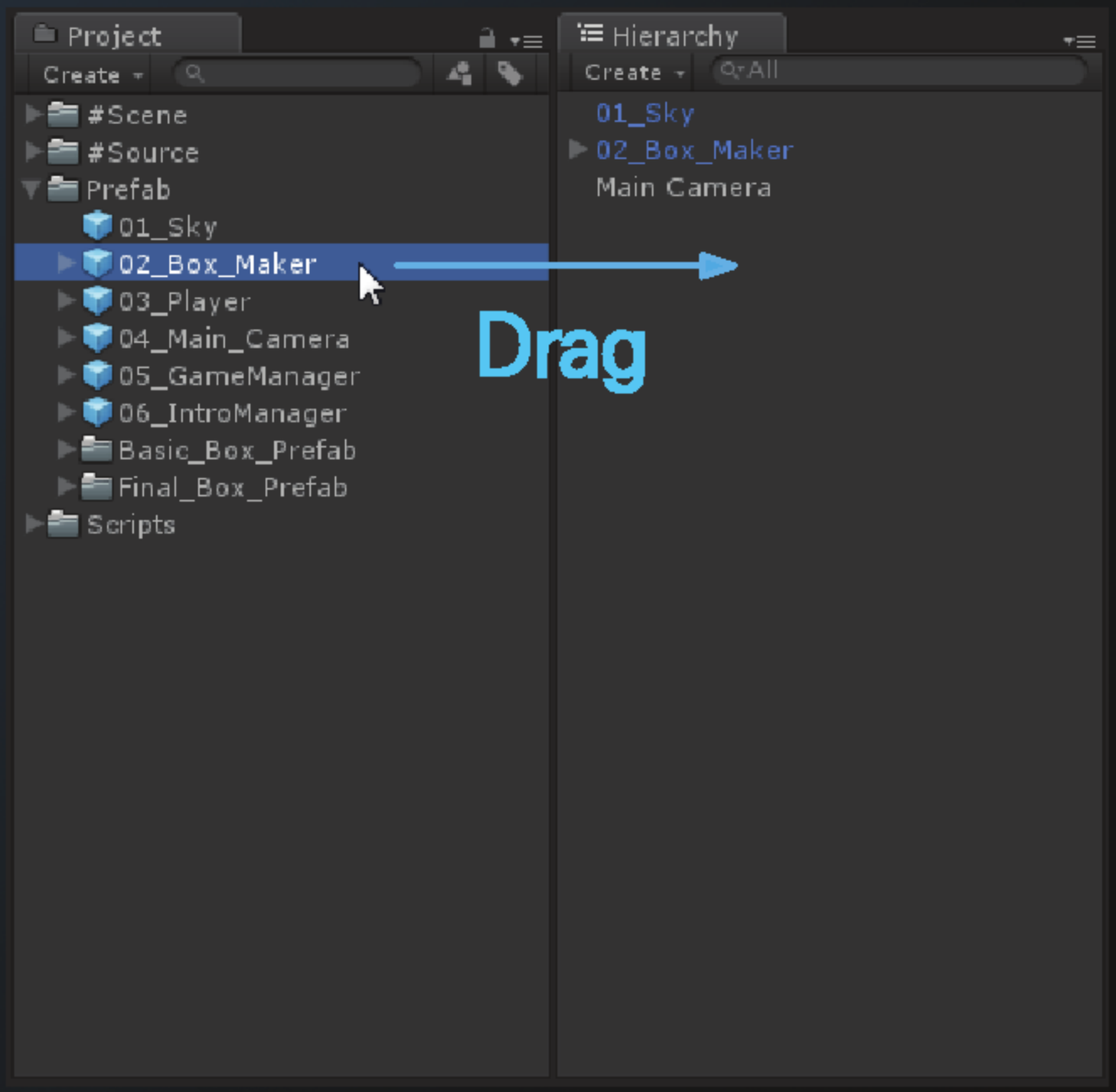
```
using UnityEngine;
using System.Collections;

public class Scroll_Mapping : MonoBehaviour
{
    public float ScrollSpeed = 0.5f;
    float Target_Offset;

    void Update()
    {
        Target_Offset += Time.deltaTime * ScrollSpeed;
        GetComponent<Renderer>().material.mainTextureOffset = new Vector2(Target_Offset, 0);
    }
}
```

배경은 2~3장 정도의 단계를 나눠서 연출해 보세요.

STEP.2 반복생성하는 바닥블럭 만들기

unity

STEP.2 반복생성하는 바닥블럭 만들기



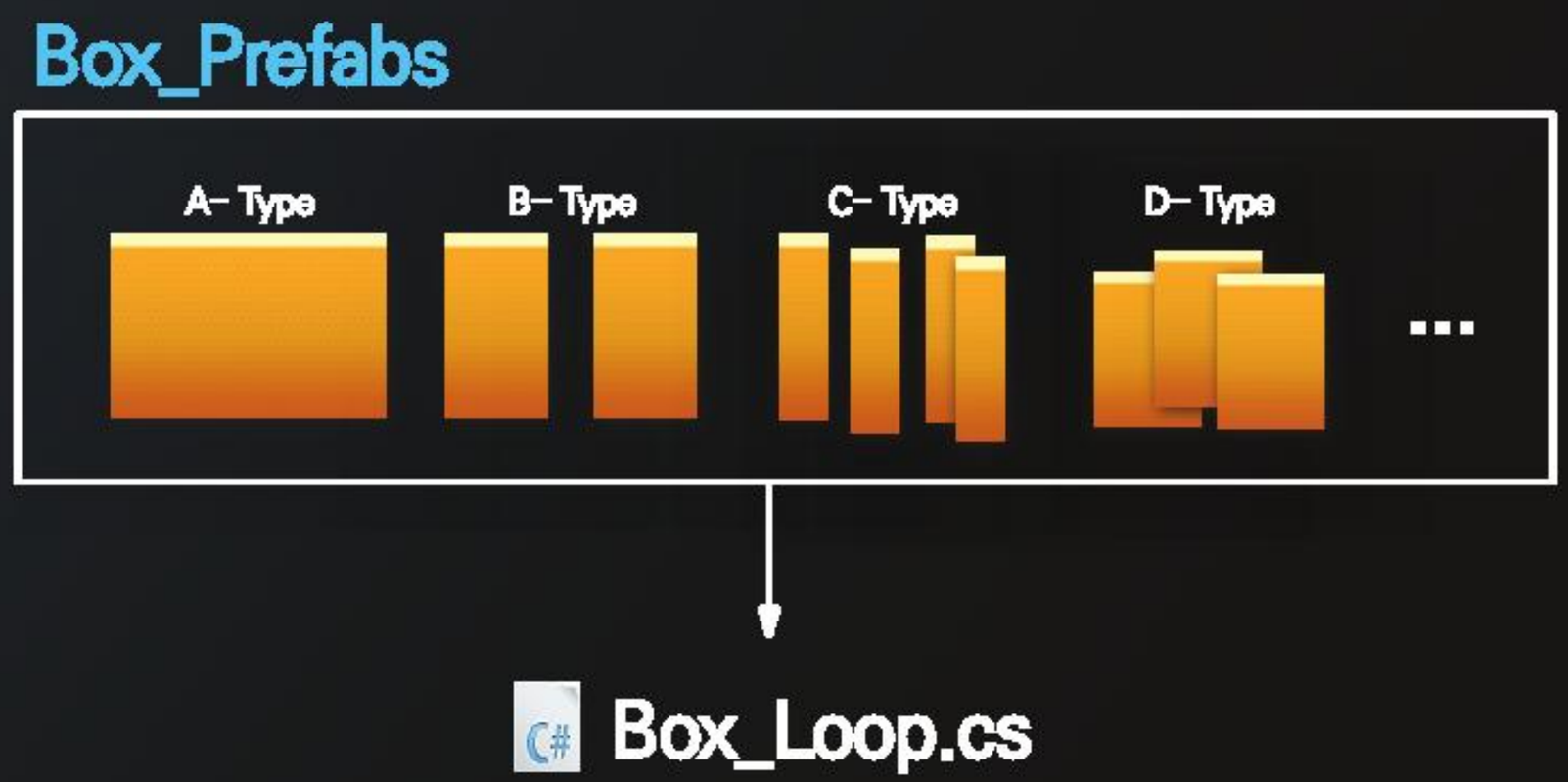
움직이는 하늘을 만들었으니,
다음은 계속 움직이는 바닥을 만들 차례입니다.

Project 뷰의 Prefab폴더 안에 있는
02_Box_Maker 프리팹을 드래그해
Hierachy뷰로 옮깁니다.

플레이 해보면 바닥블럭들이 랜덤하게 생성되
플레이어 쪽으로 진행되며 바닥이 끊이지 않고
계속 변화하는 느낌이 드는 효과가납니다.

02_Box_Maker 프리팹의 구조는 다음과 같습니다.

1. 다양하게 Prefab으로 만들어 둔 바닥블럭들

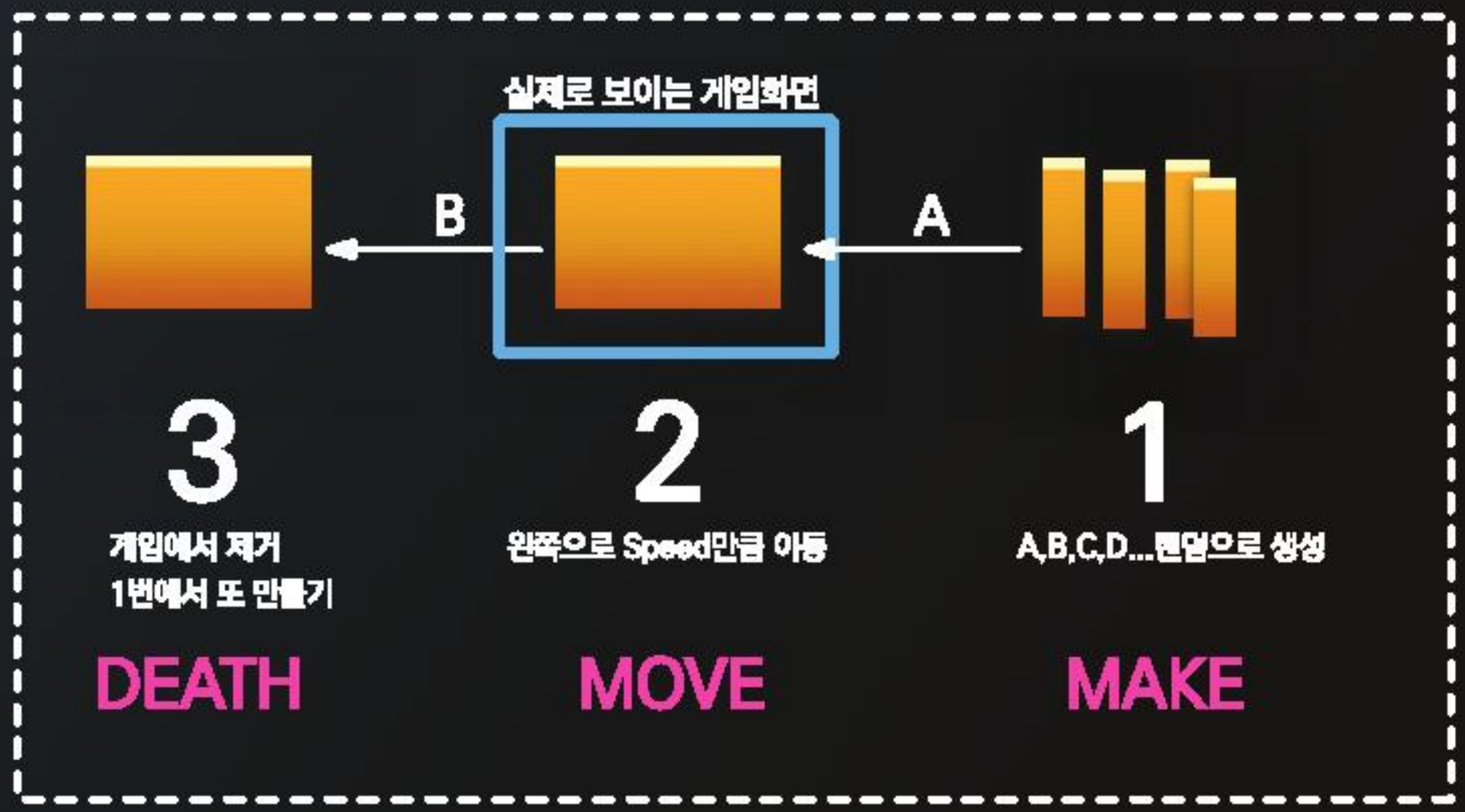


Block Manager
(create->empty object)

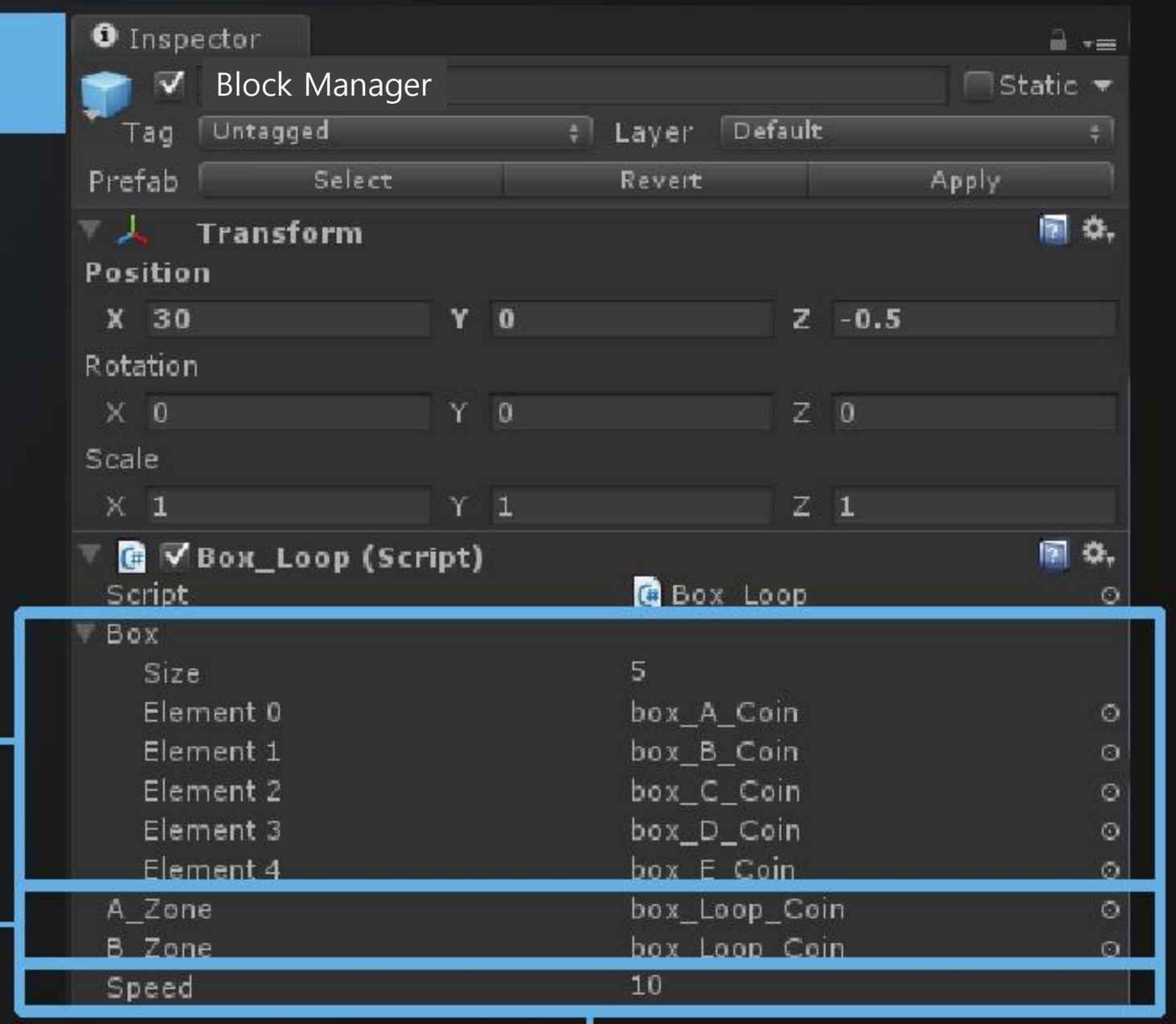
```
using UnityEngine;
using System.Collections;

public class Block_Loop : MonoBehaviour {

    public float Speed = 3;         //블럭들이 움직일 스피드.
    public GameObject Block;        //무한으로 생성할 블럭.
    //public GameObject[] Block;        //무한으로 생성할 블록 목록.
    public GameObject A_Zone;       //최초 시작시, 가운데 있는 블럭.
    public GameObject B_Zone;       //최초 시작시, 오른쪽에 있는 블럭.

    void Update()
    {

        Move();

    }

    void Move()
    {
        A_Zone.transform.Translate(Vector3.left * Speed * Time.deltaTime);
        B_Zone.transform.Translate(Vector3.left * Speed * Time.deltaTime);

        if (B_Zone.transform.position.x <= 0) //B존이 화면 좌측에 들어선 순간 처리할 내용
        {
            Destroy(A_Zone);     //A존을 화면에서 제거

            A_Zone = B_Zone;     //B존을 A존으로 변경
            Make();        //B존으로 될 블럭을 새로 생성
        }

    }

    void Make()
    {
        B_Zone = Instantiate(Block, new Vector3(A_Zone.transform.position.x + 10, A_Zone.transform.position.y, A_Zone.transform.position.z), transform.rotation) as GameObject;
/*
        int A = Random.Range (0, Block.Length);
        B_Zone = Instantiate(Block[A], new Vector3(A_Zone.transform.position.x + 10, A_Zone.transform.position.y, A_Zone.transform.position.z), transform.rotation) as GameObject;
*/
    }

}
```

Box_Loop.cs

2 – 1. Prefab을 카메라 밖에서 랜덤으로 생성하고,
2 – 2. 왼쪽으로 일정속도만큼 움직이고,
2 – 3. 카메라 밖으로 나가면 제거함과 동시에



Box_Loop.cs



우선은 하나의 블록만 사용하겠습니다.

랜덤으로 생성될 프리팹들

반드시 Project 뷰에 있는 프리팹이 연결되어야합니다.

필드에서 동작할 A와 B

반드시 Hierachy 뷰에 있는 오브젝트가 연결되어야합니다.
A는 x포지션이 30인 블럭,
B는 x포지션이 0인 블럭을 연결하면 됩니다.

현재 화면에 배치되어 있는 A/B 블록(존)을 선택합니다. Speed 값을 조절해 블럭들의 움직이는 속도를 조절할 수 있습니다.

STEP.3 플레이어 만들기

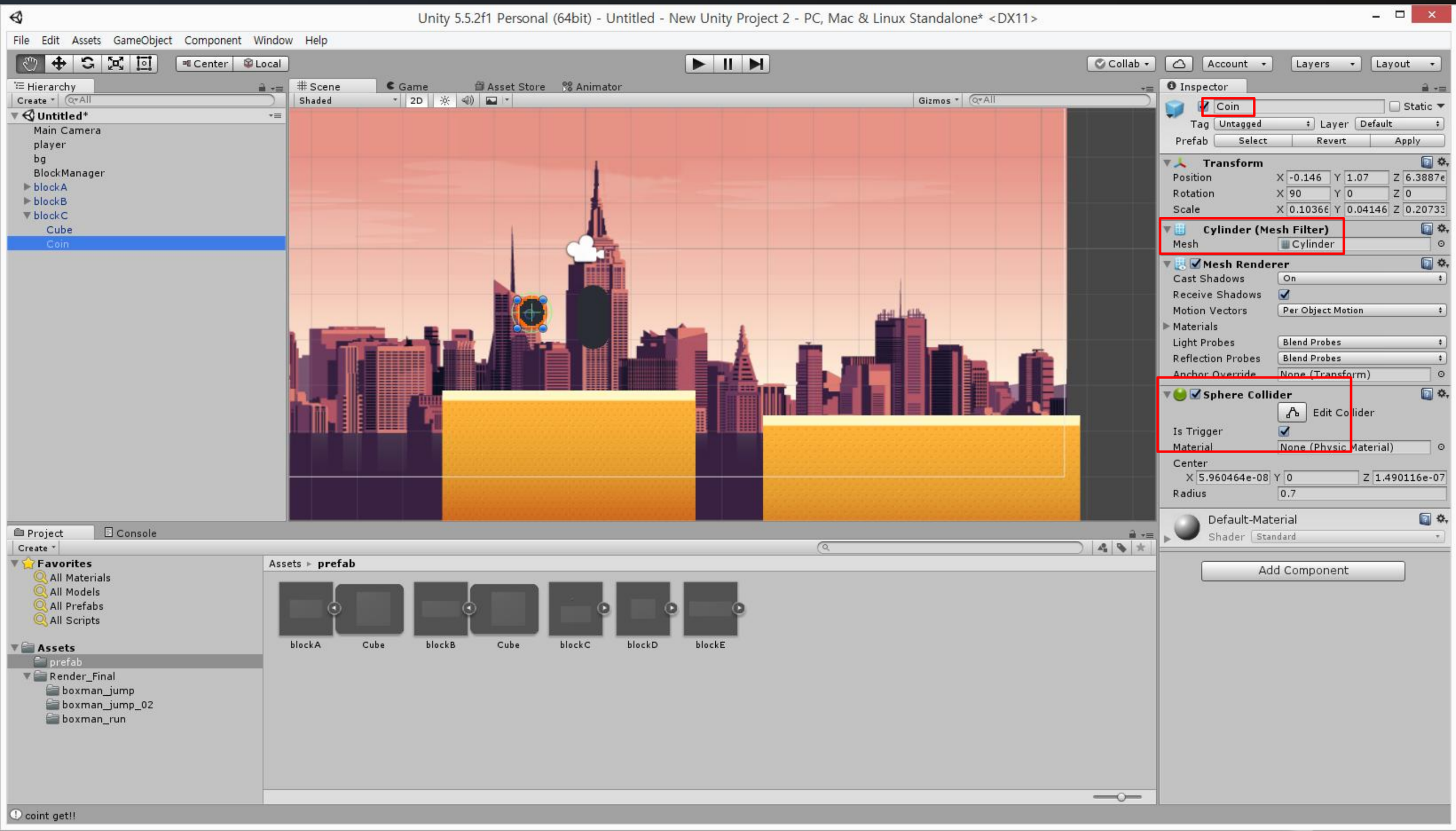
스프라이트를 입히기 전에 먼저 캡슐 오브젝트로 테스트 삼아 만들어 봅니다.

Create->3D->capsule
Size:0.7 정도

```
using UnityEngine;
using System.Collections;

public enum PlayerState
{
    Run,
    Jump,
    D_Jump,
    Death
}

public class Player_Ctrl : MonoBehaviour {

    public PlayerState PS;
    public float Jump_Power = 500f;

    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Space))
        {
            if (PS == PlayerState.Jump)
            {
                D_Jump();
            }
            if (PS == PlayerState.Run)
            {
                Jump();
            }
        }
    }
```

```
void Jump()
    {
        PS = PlayerState.Jump;
        GetComponent<Rigidbody>().AddForce(new Vector3(0,Jump_Power,0));
    }

    void D_Jump()
    {
        PS = PlayerState.D_Jump;
        GetComponent<Rigidbody>().AddForce(new Vector3(0, Jump_Power, 0));
    }

    void Run()
    {
        PS = PlayerState.Run;
    }

    void OnCollisionEnter(Collision collision)
    {
        if (PS != PlayerState.Run)
        {
            Run();
        }
    }        }
```

배경 바닥 블록에 플레이어가 획득할 수 있는 코인을 배치해 봅니다.



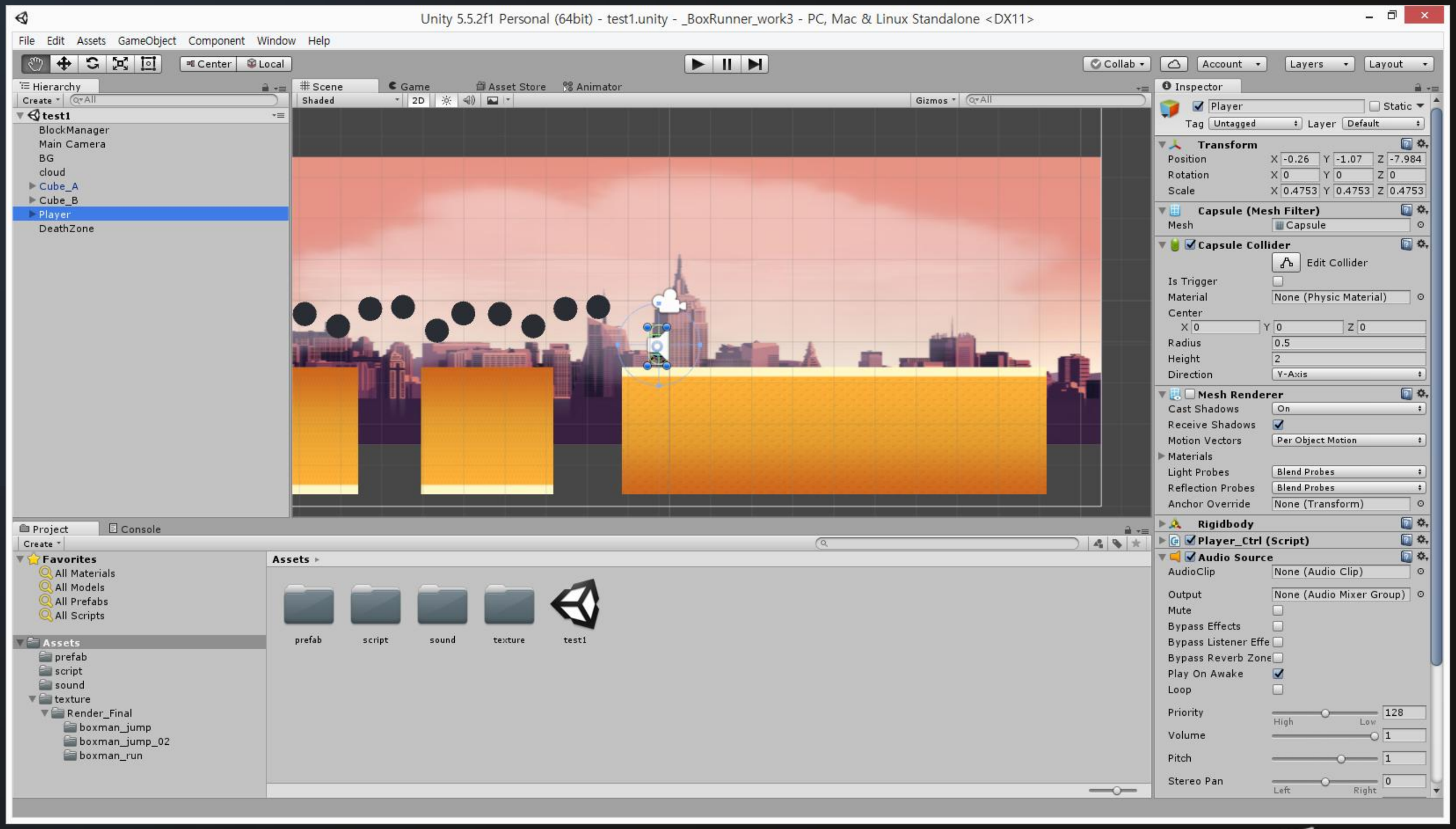
Player_Ctrl.cs에 아래 붉은 색으로 구분된 부분을 추가하면, 맵에 배치되어 있는 Coin을 획득할 수 있습니다.

```
void Run()
{
    PS = PlayerState.Run;
}

void OnCollisionEnter(Collision collision)
{
    if (PS != PlayerState.Run)
    {
        Run();
    }
}

void CoinGet()
{
    Debug.Log ("coint get!!");
}

void OnTriggerEnter(Collider other)
{
    //rigidbody.WakeUp ();
    if (other.gameObject.name == "Coin")
    {
        Destroy (other.gameObject);
        CoinGet ();
    }
}
```

Player_Ctrl.cs에 아래 붉은 색으로 구분된 부분을 추가하면, 맵에 배치되어 있는 DeathZone에 닿았을 때 처리가 될 수 있습니다.

```
void CoinGet()
{
    Debug.Log ("coint get!!");
}

void GameOver()
{
    Debug.Log ("Now You Dead");
    PS = PlayerState.Death;
}

void OnTriggerEnter(Collider other)
{
    //rigidbody.WakeUp ();
    if (other.gameObject.name == "Coin")
    {
        Destroy (other.gameObject);
        CoinGet ();
    }
    if (other.gameObject.name == "DeathZone")
    {
        GameOver ();
    }
}
```

Main Camera에는 AudioListner를, Player에는 Audio Source 를 추가해 줍니다.



STEP.3 플레이어 만들기

Player_OBJ

1. Player_Move

플레이어의 이동상태를 관리하는 스크립트,
키보드와 모바일 터치의 Input을 받아와 플레이어OBJ에 적용합니다.

2. Player_Get

플레이어와 부딪히는 다른 것들에 대한 처리를하는 스크립트.

3. Sprite_Animation

플레이어의 스프라이트 애니메이션을 관리하는 스크립트.
맵핑된 텍스처를 자속적으로 들려주는 역할을 합니다.

4. Sound_Player

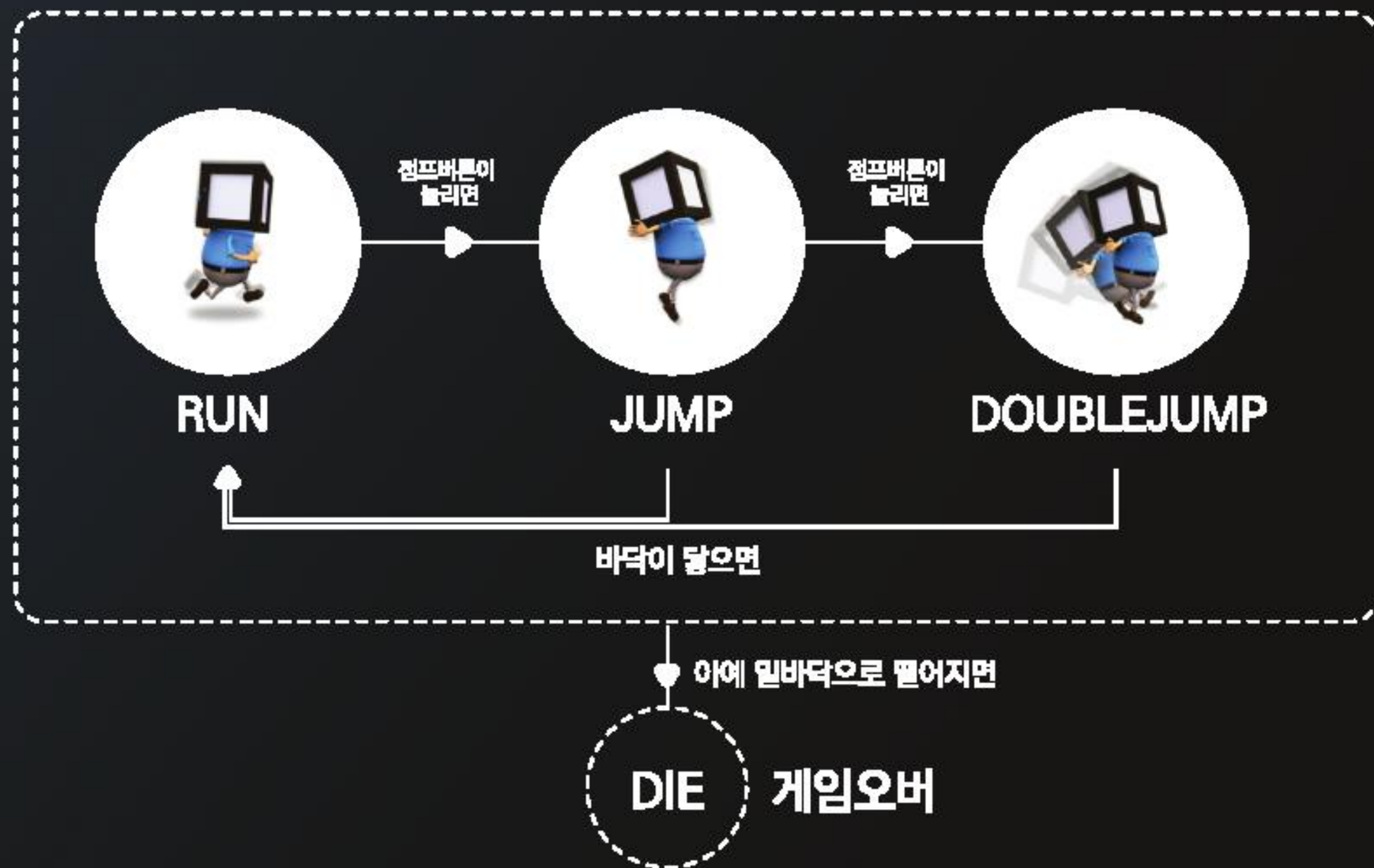
Player_Move 와 Player_Get 으로부터 신호를 받아
가지고 있는 사운드리스트의 사운드를 재생합니다.

unity

1. Player_Move

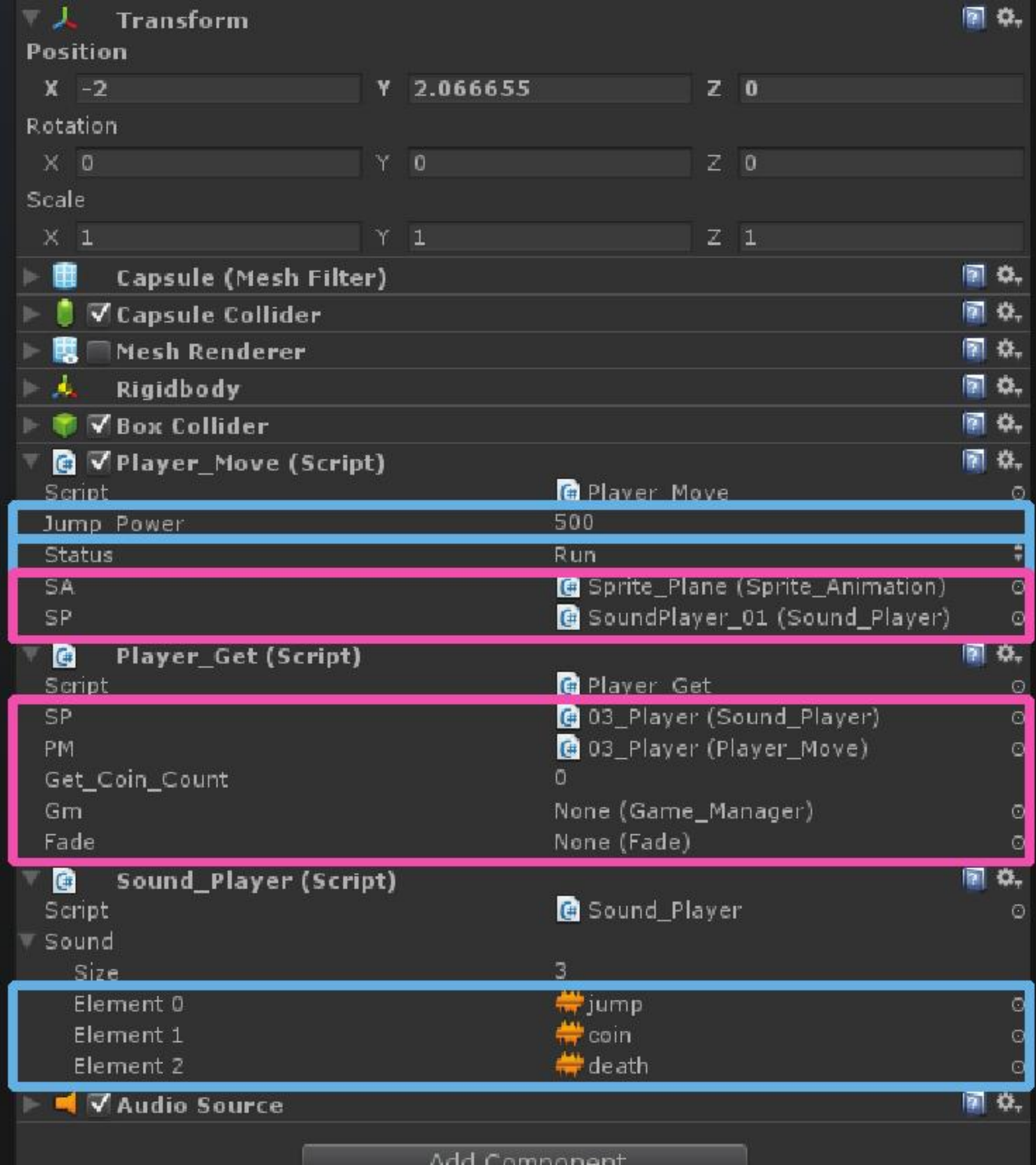
전체적인 주인공의 상태를 관리하는 스크립트입니다.

키보드나 터치 인풋, 그리고 바닥에 부딪힐때의 값 을 받아와 주인공의 상태를 바꿉니다.



점프하는 높이를 결정하는 점프력
현재 주인공의 상태

다른 스크립트들과의
연동을 하기 위해
연결해야하는 부분
(Drag&Drop으로 연결하면 됩니다)



2. Player_Get

주인공에게 특정오브젝트가 부딪혔을 시 그 특정 오브젝트의 Tag를 통해
그에 맞는 반응을 보여주는 스크립트입니다.

OnTriggerEnter(Collider Get)

Get의 tag 가 Coin 일 경우
Get 을 없애고,
획득한 Coin의 갯수에 1을 더한다.

Get의 tag 가 Deathzone 일 경우
플레이어의 상태를 DIE로 바꾼다.

충돌체 이해

유니티에서 사용하는 두가지 충돌체에 대해 간단히
이해하고 사용해봅니다.

unity

충돌체의 종류

C

Collider

실제 물리적으로
충돌하는 역할

T

Trigger

실제 물리적충돌은 일어나지 않지만
충돌했다는 값을 받아 올 수 있는 개념

unity

RigidBody

RigidBody가 붙어 있는 쪽은 물리엔진의 영향을 받게 됩니다.
충돌체크를 위해선 둘중 한명이라도 RigidBody가 붙어있어야 합니다.

(일반 콜라이더로 공 떨어트려보기 실습)

Collider, Collision

name = XXX
Tag = YYY
position = (2,3,5)

충돌된 오브젝트의 이름, 태그, 등등
기본적으로 가지고 있는 데이터를 가지고 와
이용할 수 있습니다.

(트리거 체크로 부딪힌 오브젝트 Tag 출력하기 실습)

3. Sprite_Animation

Player_Move로부터 상태가 변할때 변한다는 메세지를 가져와
그 바뀌는 상태의 애니메이션을 재생하도록 셋팅해놓은 스크립트입니다.

EX) RUN함수가 작동되었을 경우

런이 체크되었을 경우 달리는 스프라이트 [X] 장 을
현재 오브젝트의 매터리얼의 텍스쳐로 계속해서 갈아끼운다.

.
.
.

각 애니메이션의 Frame Count

각 프레임에 들어갈 스프라이트 이미지

애니메이션 플레이 속도

Sprite_Animation (Script)

Script	Sprite_Animation
Run_Ani_count	14
Run_Image	
Jump_Ani_count	18
Jump_Image	
D_Jump_Ani_count	14
D_Jump_Image	
Size	14
Element 0	boxman_jump_02_024
Element 1	boxman_jump_02_025
Element 2	boxman_jump_02_026
Element 3	boxman_jump_02_027
Element 4	boxman_jump_02_028
Element 5	boxman_jump_02_028
Element 6	boxman_jump_02_030
Element 7	boxman_jump_02_031
Element 8	boxman_jump_02_032
Element 9	boxman_jump_02_033
Element 10	boxman_jump_02_034
Element 11	boxman_jump_02_035
Element 12	boxman_jump_02_036
Element 13	boxman_jump_02_037
Ani_Play_Speed	0.02
Run	☐
Jump	☐
D_jump	☐

Sprite

Shader Unlit/Transparent Edit...

Base (RGB) Trans (A)

	Tiling	Offset
x	1	0
y	1	0

Select

4. Sound_Play

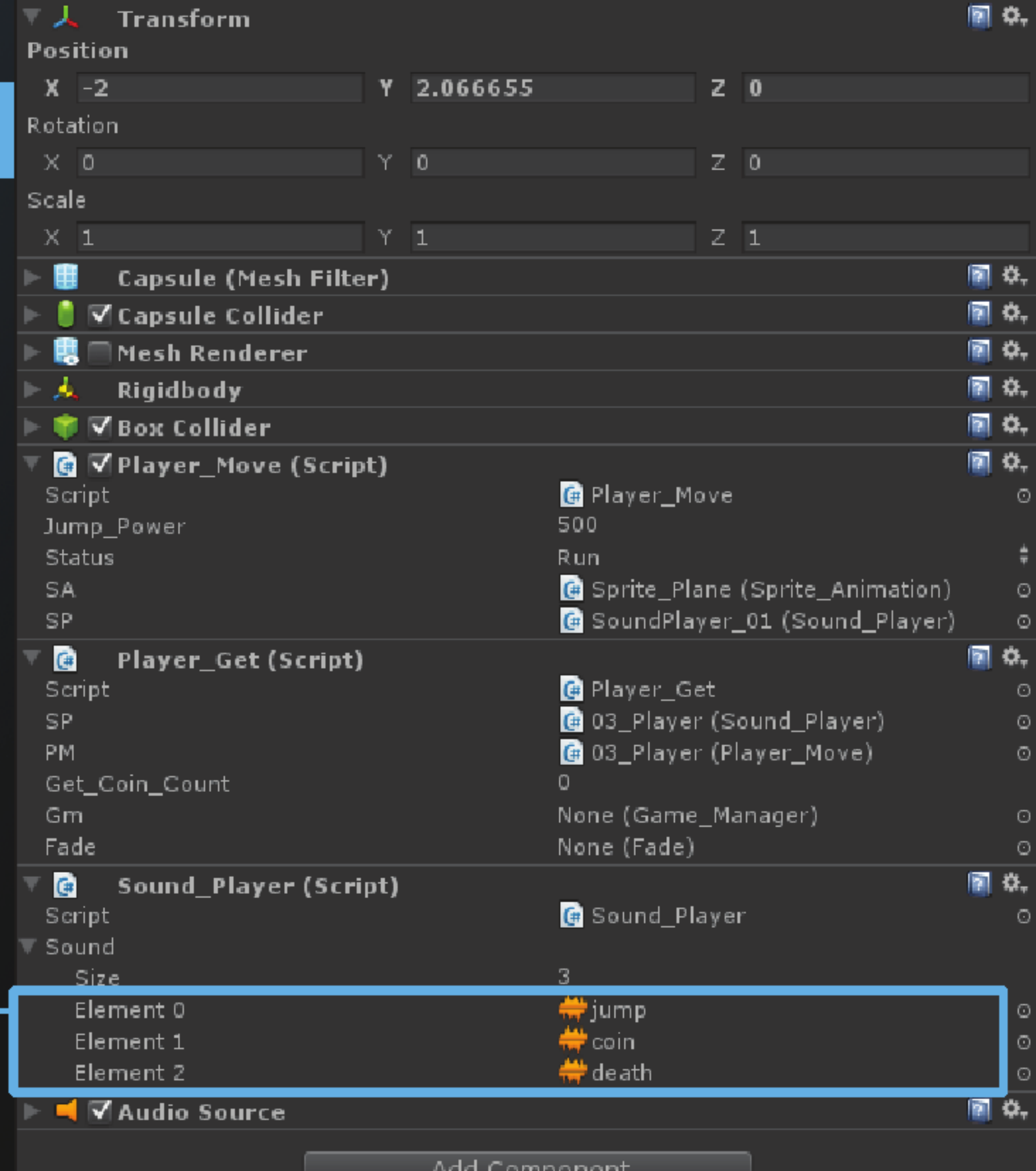
점프했을 때

코인을 먹었을 때

죽었을 때

위의 세가지 상황에 나올
사운드를 미리 셋팅.
그 상황이 되면 이 스크립트로
전달해 그에 맞는 소리를
재생하게 하는 스크립트 입니다.

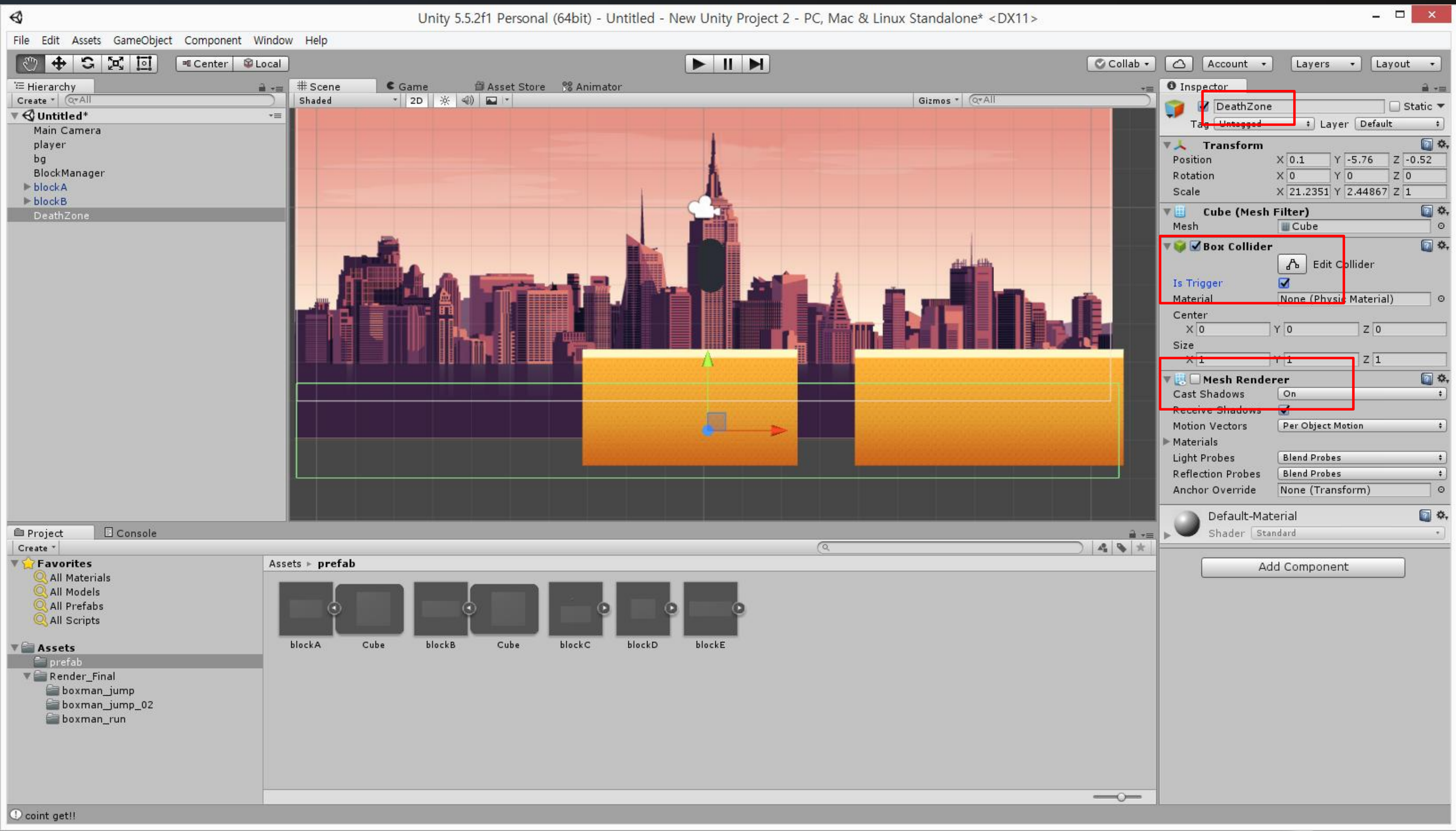
각각에 맞는 소리를
연결하면 됩니다.



Player_Ctrl.cs에 아래 붉은 색으로 구분된 부분을 추가하면, 효과음을 재생할 수 있게 됩니다.

```
using UnityEngine;
using System.Collections;

public enum PlayerState
{
    Run,
    Jump,
    D_Jump,
    Death
}

public class Player_Ctrl : MonoBehaviour {

    public PlayerState PS;
    public float Jump_Power = 500f;

    public AudioClip[] Sound;

    void Update()
    {
        //rigidbody.WakeUp ();
        if (Input.GetKeyDown(KeyCode.Space)&&PS!=PlayerState.Death)
        {
            if (PS == PlayerState.Jump)
            {
                D_Jump();
            }
            if (PS == PlayerState.Run)
            {
                Jump();
            }
        }
    }

    void Jump()
    {
        PS = PlayerState.Jump;
        GetComponent<Rigidbody>().AddForce(new Vector3(0,Jump_Power,0));
        SoundPlay (2);
    }

    void D_Jump()
    {
        PS = PlayerState.D_Jump;
        GetComponent<Rigidbody>().AddForce(new Vector3(0, Jump_Power, 0));
        SoundPlay (2);
    }

    void Run()
    {
        PS = PlayerState.Run;
    }

    void OnCollisionEnter(Collision collision)
    {
        if (PS != PlayerState.Run&&PS!=PlayerState.Death)
        {
            Run();
        }
    }

    void CoinGet()
    {
        SoundPlay (0);
        Debug.Log ("coint get!!");
    }

    void GameOver()
    {
        Debug.Log ("Now You Dead");
        PS = PlayerState.Death;
        SoundPlay (1);
    }

    void SoundPlay(int Num)
    {
        audio.clip = Sound [Num];
        audio.Play ();
    }

    void OnTriggerEnter(Collider other)
    {
        //rigidbody.WakeUp ();
        if (other.gameObject.name == "Coin")
        {
            Destroy (other.gameObject);
            CoinGet ();
        }
        if (other.gameObject.name == "DeathZone")
        {
            GameOver ();
        }
    }
}
```

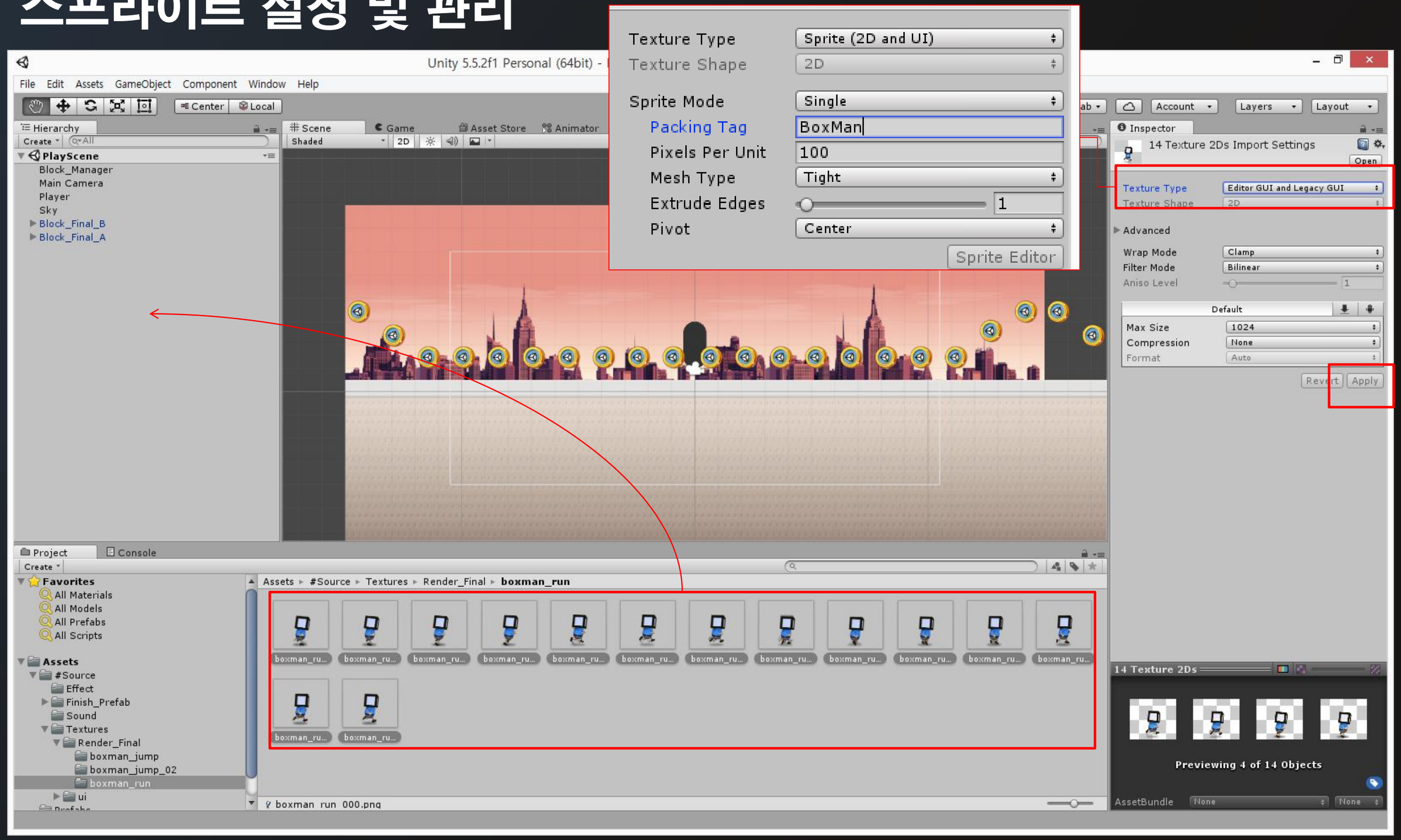
이제 플레이어 캐릭터가 떨어지면 '끝' 이라는 구역 처리를 해 봅니다.

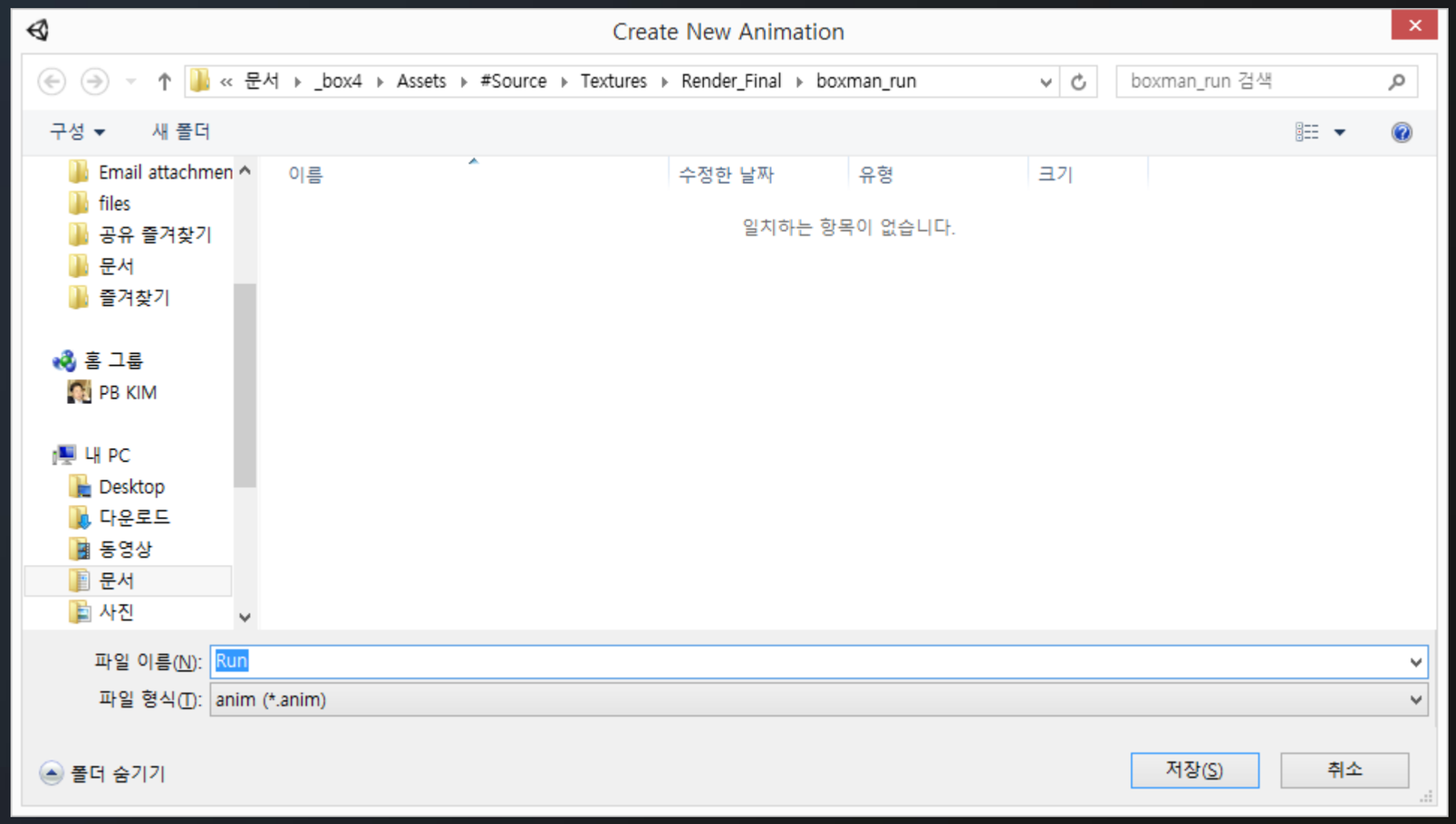


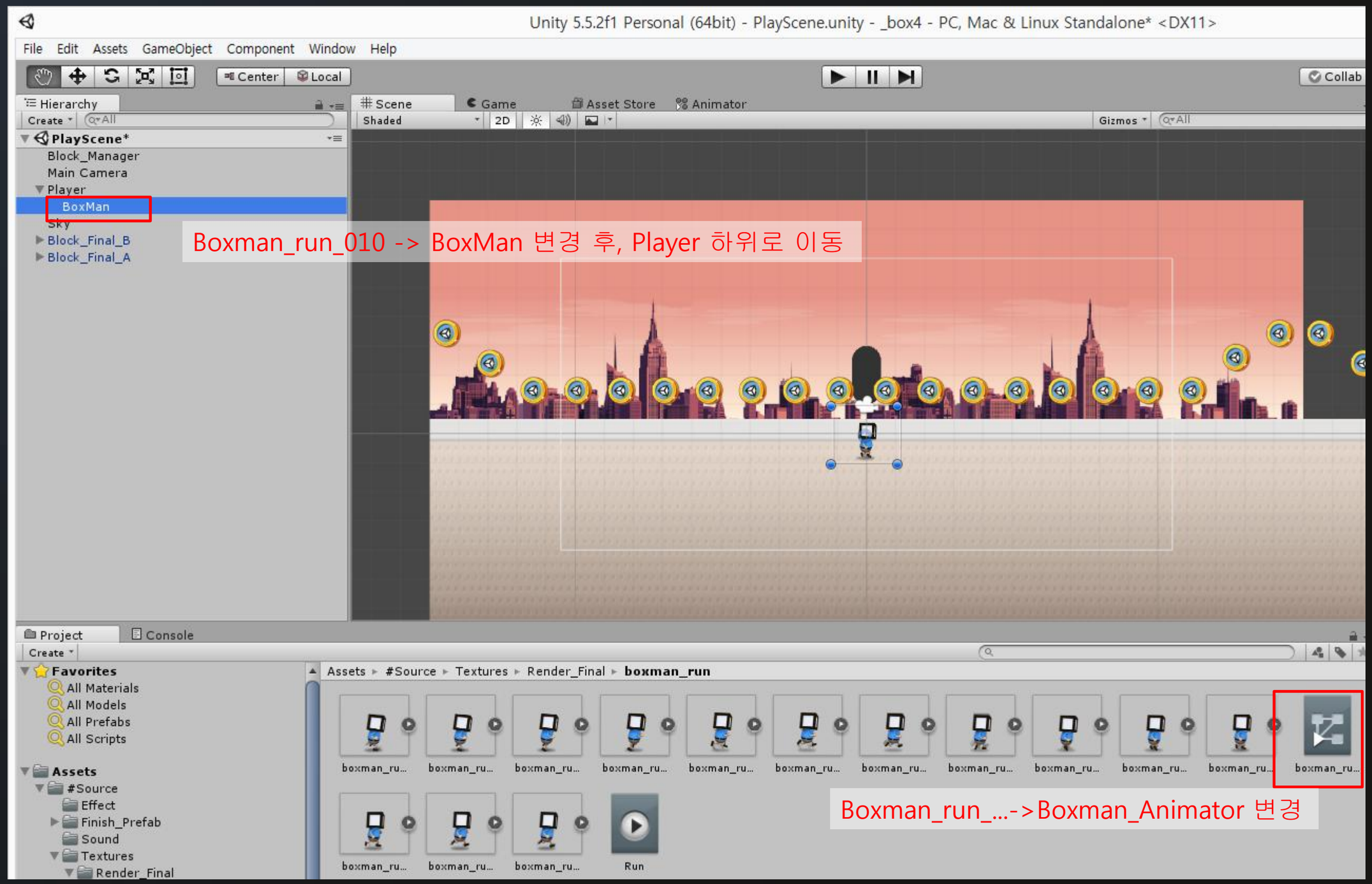
만약 정상 동작하지 않을 경우, UNITY version 문제일 수 있으므로, 아래와 같이 수정합니다.

```
void SoundPlay(int Num)
{
        GetComponent<AudioSource>().clip = Sound[Num];
        GetComponent<AudioSource>().Play();
}
```

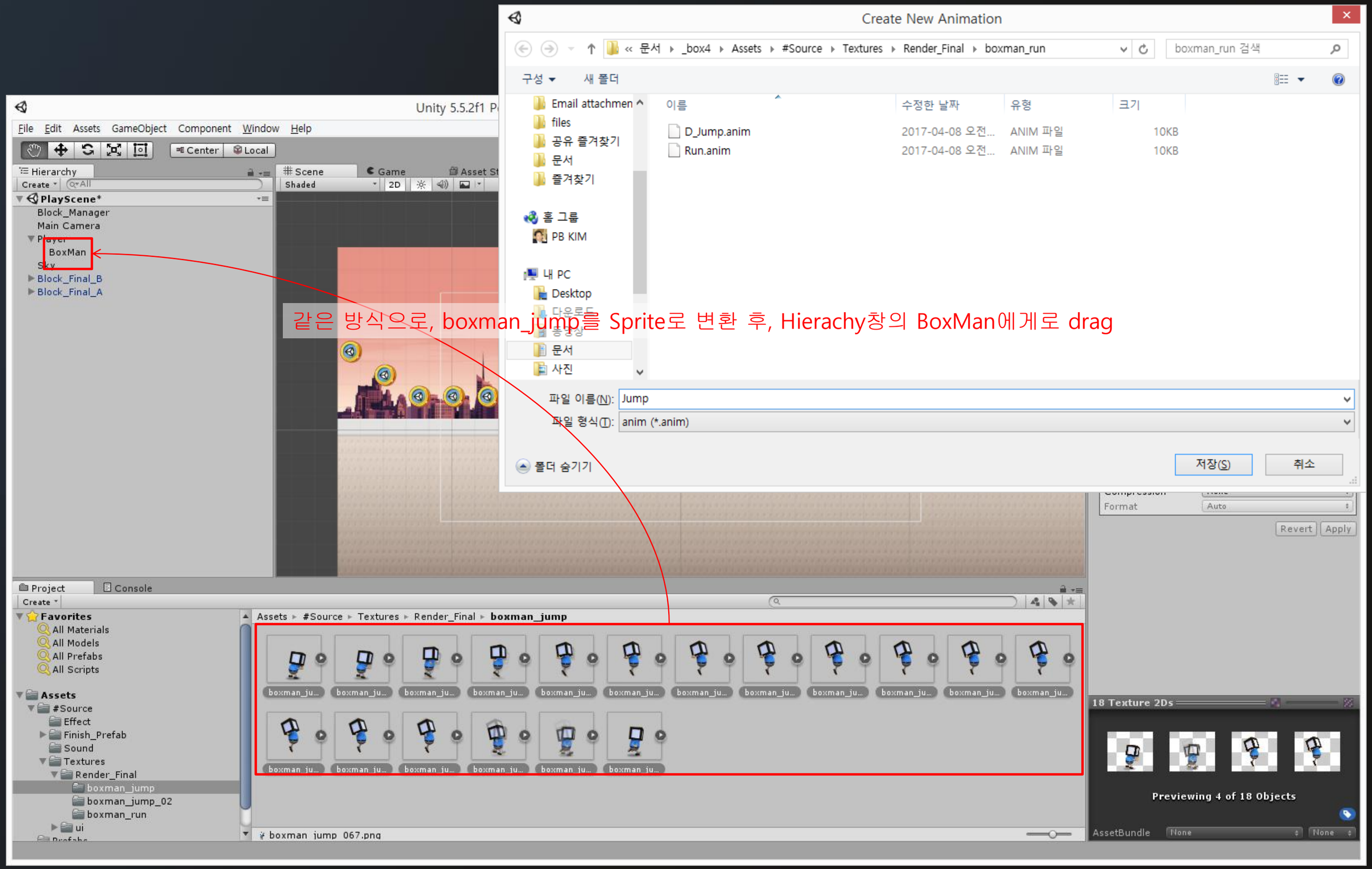
스프라이트 설정 및 관리


Texture Type
Sprite (2D and UI)
Texture Shape
2D
Sprite Mode
Single
Packing Tag
BoxMan
Pixels Per Unit
100
Mesh Type
Tight
Extrude Edges
1
Pivot
Center
Sprite Editor
Unity 5.5.2f1 Personal (64bit) -
File Edit Assets GameObject Component Window Help
Center
Local
Hierarchy
Create
PlayScene
Block_Manager
Main Camera
Player
Sky
Block_Final_B
Block_Final_A
Scene
Game
Asset Store
Animator
Shaded
2D
Inspector
14 Texture 2Ds Import Settings
Open
Texture Type
Editor GUI and Legacy GUI
Texture Shape
2D
Advanced
Wrap Mode
Clamp
Filter Mode
Bilinear
Aniso Level
1
Default
Max Size
1024
Compression
None
Format
Auto
Revert
Apply
Account
Layers
Layout
Project
Console
Favorites
All Materials
All Models
All Prefabs
All Scripts
Assets
#Source
Effect
Finish_Prefab
Sound
Textures
Render_Final
boxman_jump
boxman_jump_02
boxman_run
ui
Assets ▸ #Source ▸ Textures ▸ Render_Final ▸ boxman_run
boxman_run_000.png
14 Texture 2Ds
Previewing 4 of 14 Objects
AssetBundle
None
None


Create New Animation
« 문서 ▸ _box4 ▸ Assets ▸ #Source ▸ Textures ▸ Render_Final ▸ boxman_run
boxman_run 검색
구성
새 폴더
Email attachmen
files
공유 즐겨찾기
문서
즐겨찾기
홈 그룹
PB KIM
내 PC
Desktop
다운로드
동영상
문서
사진
이름
수정한 날짜
유형
크기
일치하는 항목이 없습니다.
파일 이름(N):
Run
파일 형식(T):
anim (*.anim)
폴더 숨기기
저장(S)
취소


Unity 5.5.2f1 Personal (64bit) - PlayScene.unity - _box4 - PC, Mac & Linux Standalone* <DX11>
File Edit Assets GameObject Component Window Help
Center Local
Collab
Hierarchy
Scene
Game
Asset Store
Animator
Create
Shaded
2D
Gizmos
PlayScene*
Block_Manager
Main Camera
Player
BoxMan
Sky
Block_Final_B
Block_Final_A
Boxman_run_010 -> BoxMan 변경 후, Player 하위로 이동
Project
Console
Favorites
All Materials
All Models
All Prefabs
All Scripts
Assets
#Source
Effect
Finish_Prefab
Sound
Textures
Render_Final
Assets ▸ #Source ▸ Textures ▸ Render_Final ▸ boxman_run
Run
Boxman_run_...->Boxman_Animator 변경

같은 방식으로, boxman_jump를 Sprite로 변환 후, Hierachy창의 BoxMan에게로 drag



Unity 5.5.2f1 Personal (64bit) - PlayScene.unity - _box4 - PC, Mac & Linux Standalone* <DX11>

File Edit Assets GameObject Component Window Help

Jump는 1회만 동작해야 되므로, Loop Time 을 uncheck.

Inspector

Jump

Open

Length 1.500 12 FPS

Loop Time

Loop Pose

Cycle Offset 0

Curves Pos: 0 Quaternion: 0 Euler: 0 Scale: 0 Muscles: 0
Generic: 0 PPtr: 1
Curves Total: 1, Constant: 0 (0.0%) Dense: 0 (0.0%)
Stream: 1 (100.0%)
2.6 KB

Jump 0.10

No model is available for preview.
Please drag a model into this Preview Area.

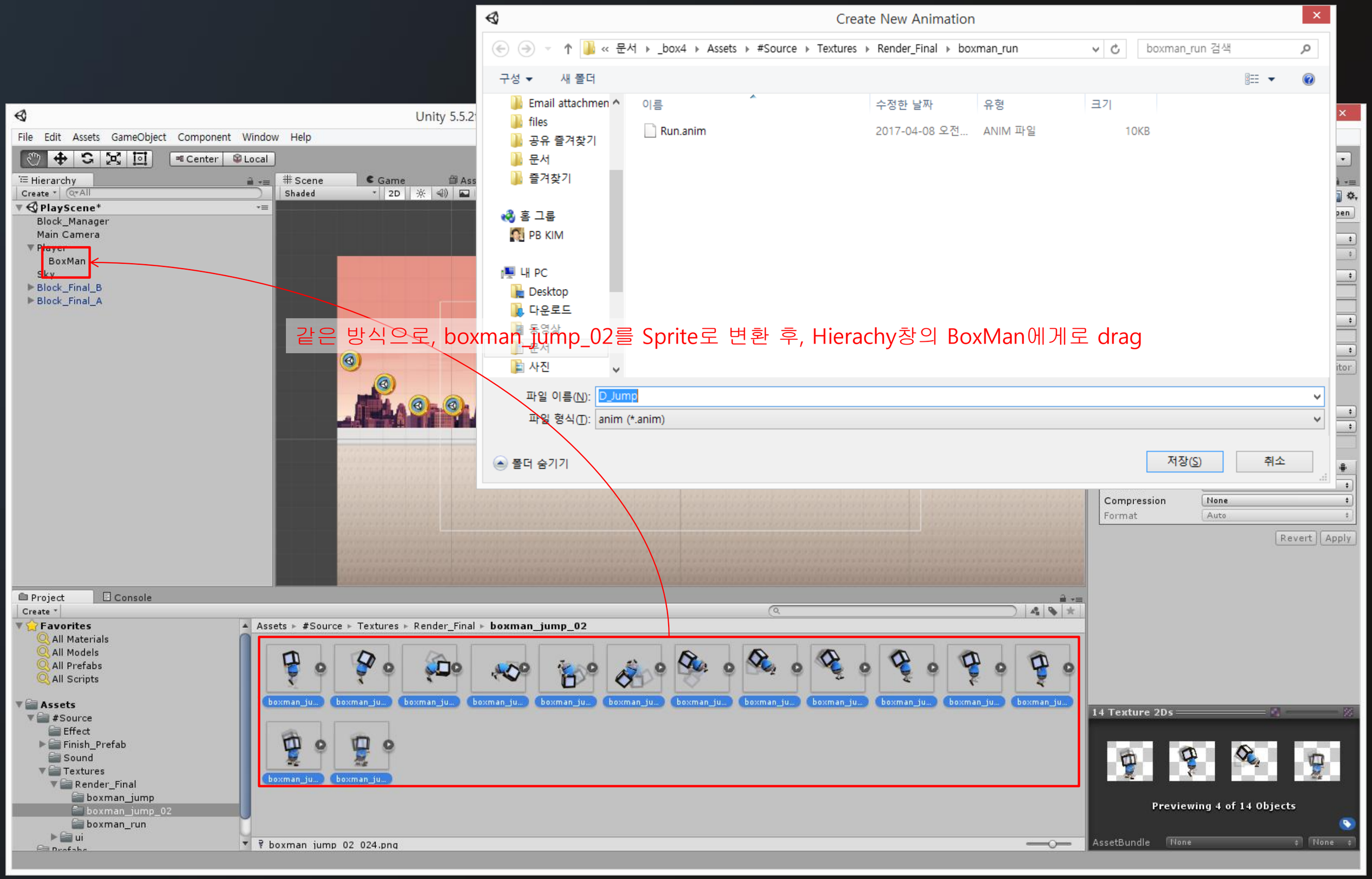
0:00 (000.0%) Frame 0

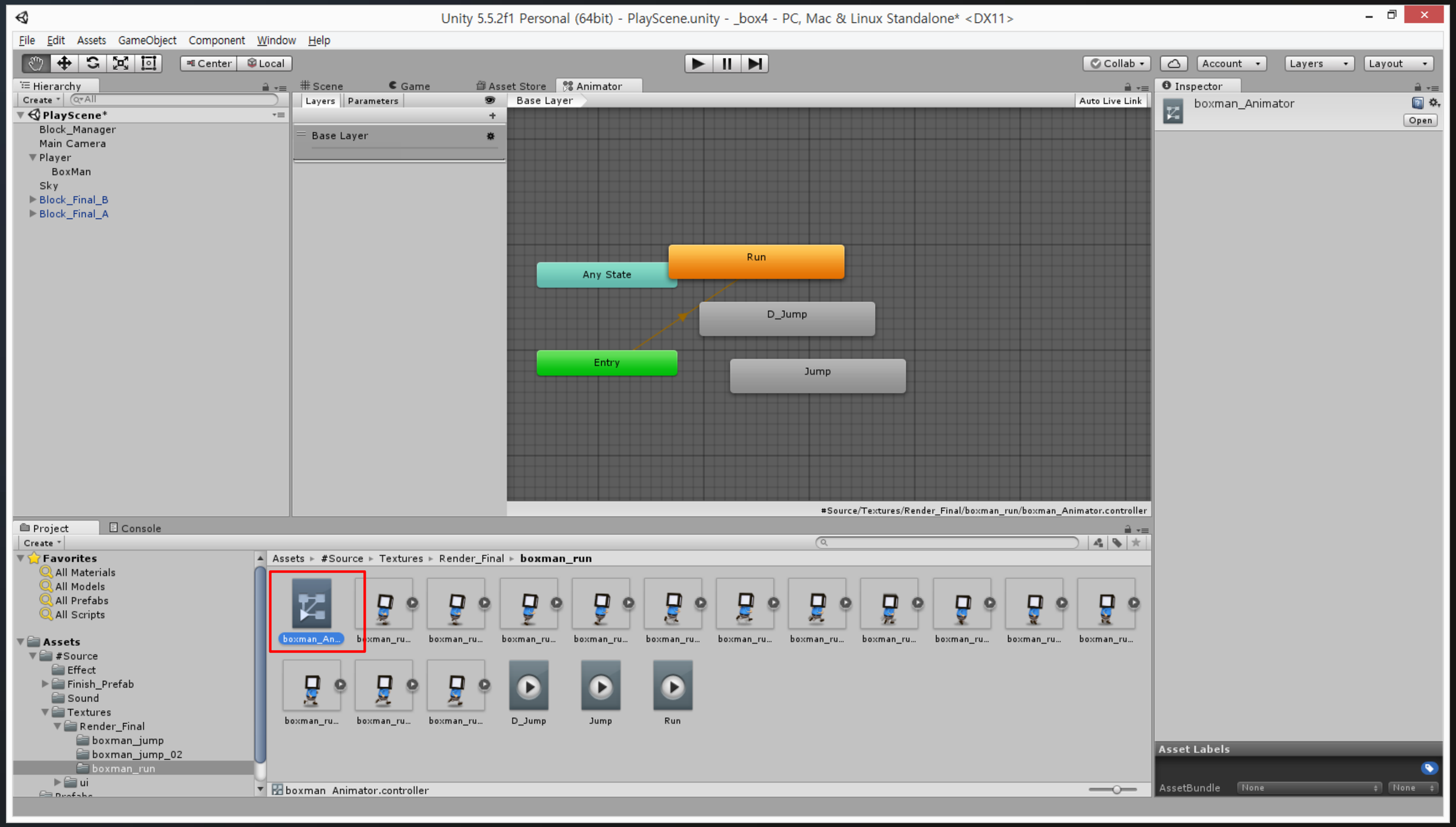
AssetBundle None None

Assets ▸ #Source ▸ Textures ▸ Render_Final ▸ boxman_run

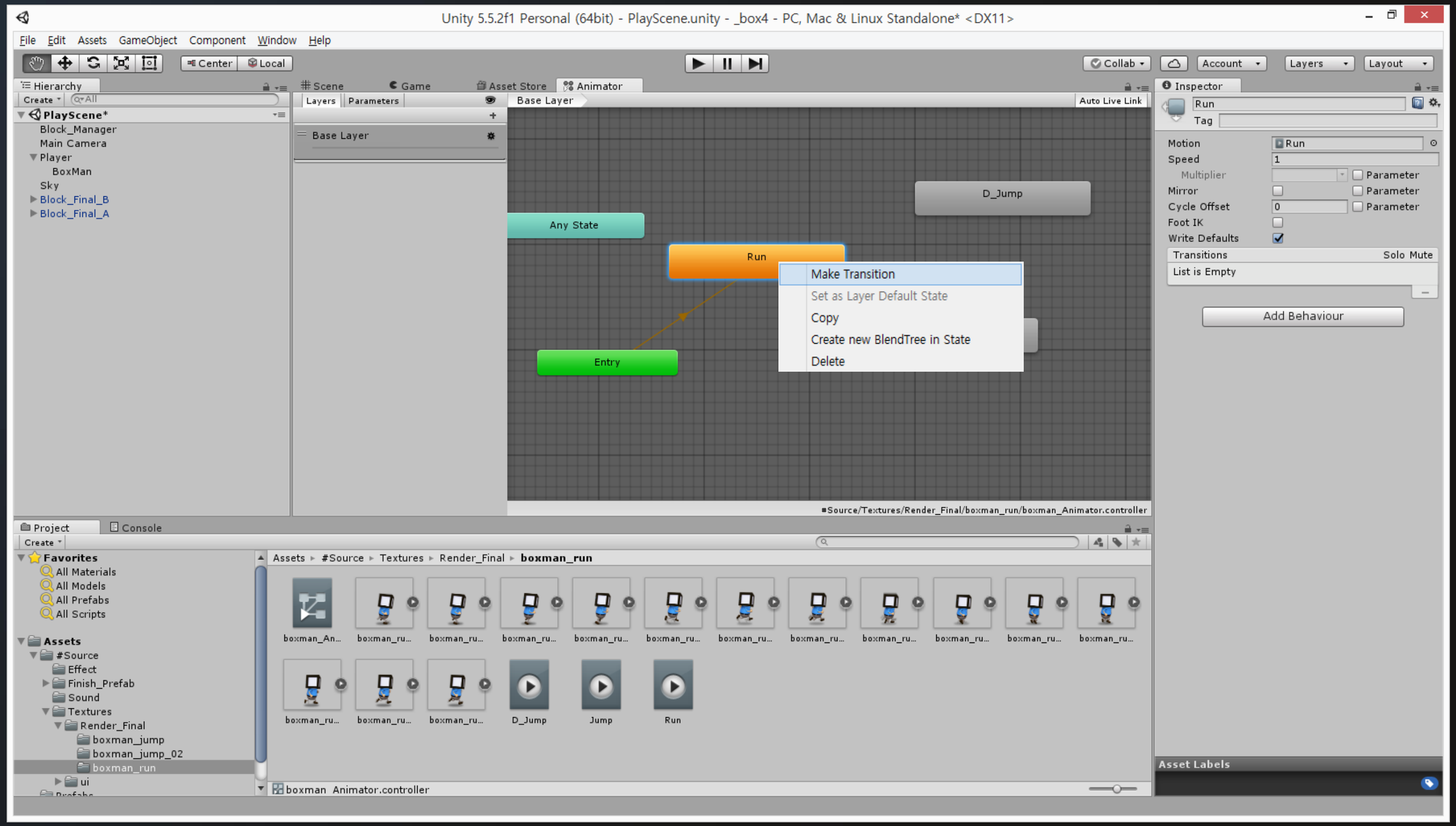
Jump.anim

같은 방식으로, boxman_jump_02를 Sprite로 변환 후, Hierachy창의 BoxMan에게로 drag




Unity 5.5.2f1 Personal (64bit) - PlayScene.unity - _box4 - PC, Mac & Linux Standalone* <DX11>
File Edit Assets GameObject Component Window Help
Center Local
Collab Account Layers Layout
Hierarchy
Create All
PlayScene*
Block_Manager
Main Camera
Player
BoxMan
Sky
Block_Final_B
Block_Final_A
Scene Game Asset Store Animator
Layers Parameters
Base Layer
Base Layer
Auto Live Link
Any State
Run
D_Jump
Entry
Jump
#Source/Textures/Render_Final/boxman_run/boxman_Animator.controller
Inspector
boxman_Animator
Open
Project Console
Create
Favorites
All Materials
All Models
All Prefabs
All Scripts
Assets
#Source
Effect
Finish_Prefab
Sound
Textures
Render_Final
boxman_jump
boxman_jump_02
boxman_run
ui
Assets ▸ #Source ▸ Textures ▸ Render_Final ▸ boxman_run
boxman_An...
boxman_ru...
D_Jump
Jump
Run
boxman_Animator.controller
Asset Labels
AssetBundle None None

메카님으로 anim 파일 연결하기



Unity 5.5.2f1 Personal (64bit) - PlayScene.unity - _box4 - PC, Mac & Linux Standalone* <DX11>

File Edit Assets GameObject Component Window Help

Center Local

Collab Account Layers Layout

Hierarchy

Create All

PlayScene*
- Block_Manager
- Main Camera
- Player
  - BoxMan
- Sky
- Block_Final_B
- Block_Final_A

Scene Game Asset Store Animator

Layers Parameters

Base Layer

Base Layer Auto Live Link

Any State
Entry
Run
Jump
D_Jump

#Source/Textures/Render_Final/boxman_run/boxman_Animator.controller

Inspector

D_Jump

Tag

Motion D_Jump

Speed 1

Multiplier Parameter

Mirror Parameter

Cycle Offset 0 Parameter

Foot IK

Write Defaults

Transitions Solo Mute

D_Jump -> Run

Add Behaviour

Asset Labels

Project Console

Create

Favorites
- All Materials
- All Models
- All Prefabs
- All Scripts

Assets
- #Source
  - Effect
  - Finish_Prefab
  - Sound
  - Textures
    - Render_Final
      - boxman_jump
      - boxman_jump_02
      - boxman_run
  - ui

Assets ▸ #Source ▸ Textures ▸ Render_Final ▸ boxman_run

boxman_An... boxman_ru... boxman_ru... boxman_ru... boxman_ru... boxman_ru... boxman_ru... boxman_ru... boxman_ru... boxman_ru... boxman_ru... boxman_ru...

boxman_ru... boxman_ru... boxman_ru... D_Jump Jump Run

boxman_Animator.controller

Unity 5.5.2f1 Personal (64bit) - PlayScene.unity - _box4 - PC, Mac & Linux Standalone* <DX11>

File Edit Assets GameObject Component Window Help

Center Local

Collab Account Layers Layout

Hierarchy

Create All

PlayScene*

- Block_Manager
- Main Camera
- Player
  - BoxMan
- Sky
- Block_Final_B
- Block_Final_A

Scene Game Asset Store Animator

Shaded 2D Gizmos All

Inspector

Player Static

Tag Untagged Layer Default

Transform

Position	X 0	Y 2	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Capsule (Mesh Filter)

Mesh: Capsule

Capsule Collider

Edit Collider

Is Trigger: ☐

Material: None (Physic Material)

Center: X 0 Y 0 Z 0

Radius: 0.5

Height: 2

Direction: Y-Axis

Mesh Renderer

Cast Shadows: On

Receive Shadows: ☑

Motion Vectors: Per Object Motion

Materials

Light Probes: Off

Reflection Probes: Blend Probes

Anchor Override: None (Transform)

Rigidbody

Mass: 1

Drag: 0

Angular Drag: 0.05

Use Gravity: ☑

Is Kinematic: ☐

Interpolate: None

Collision Detection: Discrete

Constraints

Player_Ctrl (Script)

Script: Player_Ctrl

PS: Run

Jump_Power: 700

Default-Diffuse

Shader: Legacy Shaders/Diffuse

Add Component

Project Console

Create

Favorites

- All Materials
- All Models
- All Prefabs
- All Scripts

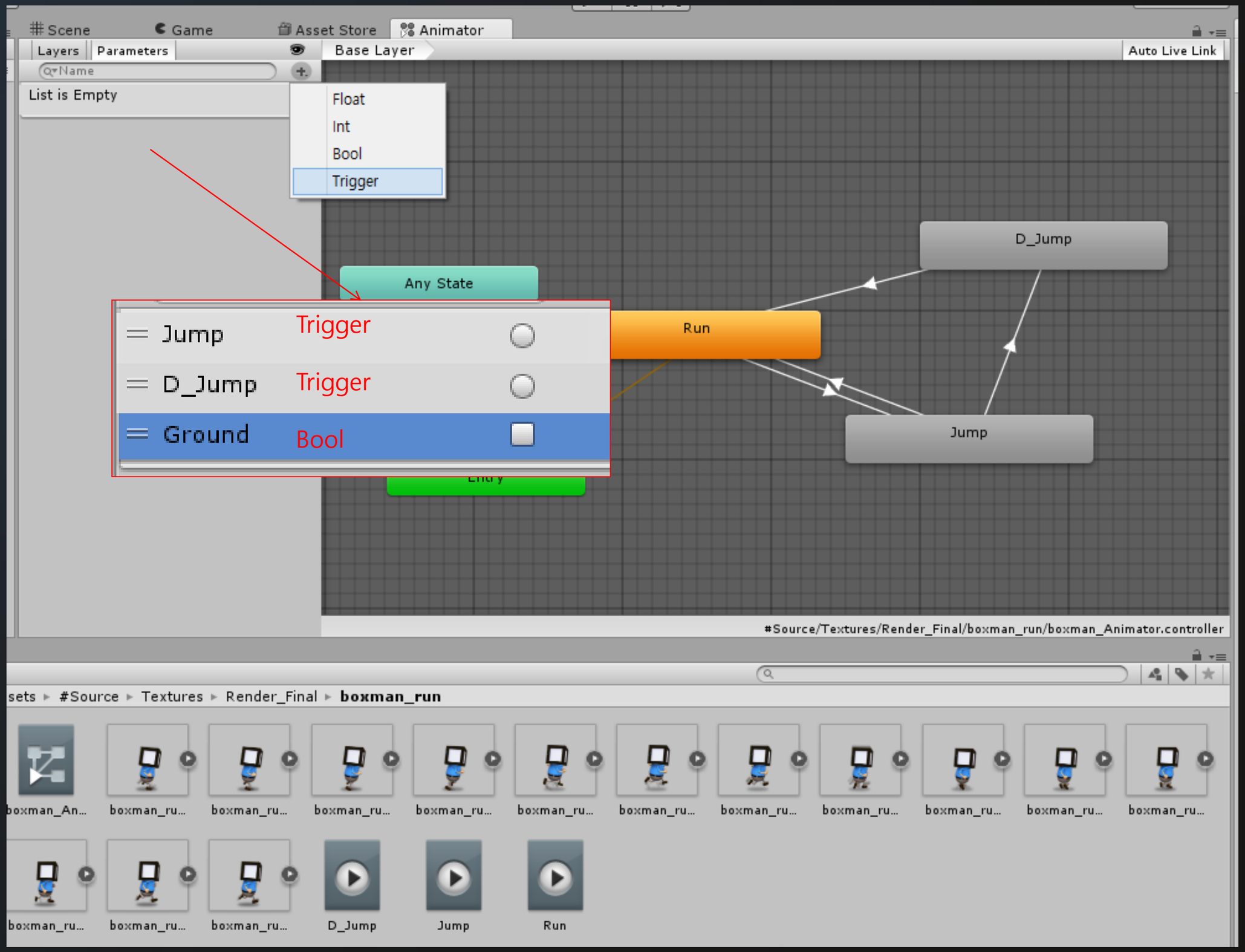
Assets

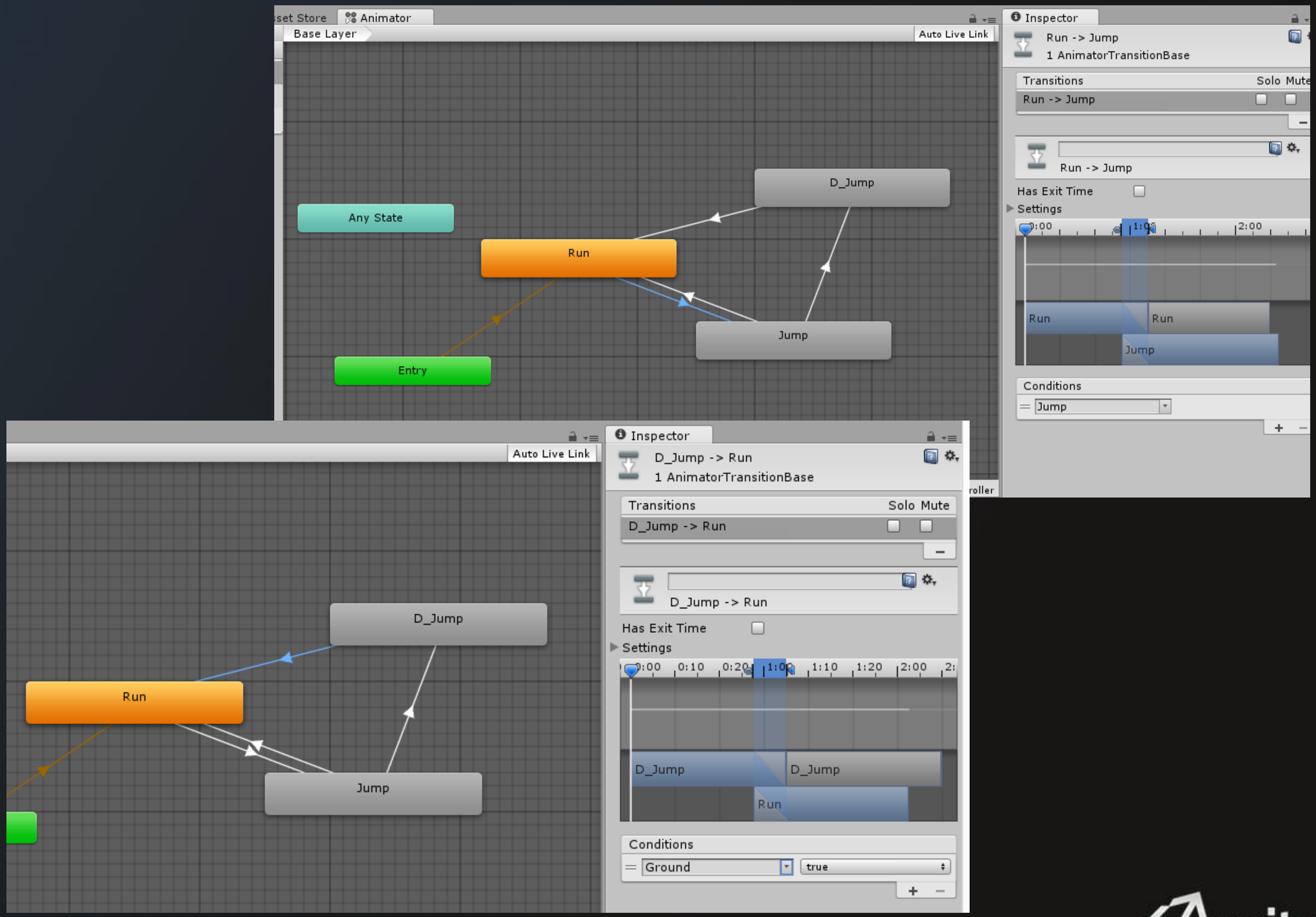
- #Source
  - Effect
  - Finish_Prefab
  - Sound
  - Textures
    - Render_Final
      - boxman_jump
      - boxman_jump_02
      - boxman_run
  - ui

Assets ▸ #Source ▸ Textures ▸ Render_Final ▸ boxman_run

boxman_An... boxman_ru... boxman_ru... boxman_ru... boxman_ru... boxman_ru... boxman_ru... boxman_ru... boxman_ru... boxman_ru... boxman_ru... boxman_ru... boxman_ru... boxman_ru... boxman_ru... D_Jump Jump Run

unity


Scene
Game
Asset Store
Animator
Layers
Parameters
Base Layer
Auto Live Link
Name
List is Empty
Float
Int
Bool
Trigger
Any State
Jump
Trigger
D_Jump
Trigger
Ground
Bool
Run
D_Jump
Jump
Entry
#Source/Textures/Render_Final/boxman_run/boxman_Animator.controller
sets ▸ #Source ▸ Textures ▸ Render_Final ▸ boxman_run
boxman_An...
boxman_ru...
D_Jump
Jump
Run


Animator
Base Layer
Auto Live Link
Any State
Entry
Run
D_Jump
Jump
Inspector
Run -> Jump
1 AnimatorTransitionBase
Transitions
Solo
Mute
Run -> Jump
Has Exit Time
Settings
Conditions
Jump
Auto Live Link
Run
D_Jump
Jump
Inspector
D_Jump -> Run
1 AnimatorTransitionBase
Transitions
Solo
Mute
D_Jump -> Run
Has Exit Time
Settings
Conditions
Ground
true

```
public class Player_Ctrl : MonoBehaviour {

    public PlayerState PS;
    public float Jump_Power = 500f;
//    public AudioClip[] Sound;

    public Animator animator;
```

```
 void Jump()
        {
                PS = PlayerState.Jump;
                GetComponent<Rigidbody>().AddForce(new Vector3(0,Jump_Power,0));
//                SoundPlay (2);

                animator.SetTrigger("Jump");
                animator.SetBool("Ground",false);
            }

        void D_Jump()
        {
                PS = PlayerState.D_Jump;
                GetComponent<Rigidbody>().AddForce(new Vector3(0, Jump_Power, 0));
//                SoundPlay (2);

                animator.SetTrigger("D_Jump");
                animator.SetBool("Ground",false);
            }


        void Run()
        {
                PS = PlayerState.Run;
                animator.SetBool("Ground",true);
            }
```

Unity 5.5.2f1 Personal (64bit) - PlayScene.unity - _box4 - PC, Mac & Linux Standalone* <DX11>

File Edit Assets GameObject Component Window Help

Center Local | Collab | Account | Layers | Layout

Hierarchy

Create All

- PlayScene*
  - Block_Manager
  - Main Camera
  - Player
    - BoxMan
  - Sky
  - Block_Final_B
  - Block_Final_A

Scene | Game | Asset Store | Animator

Display 1 | Free Aspect | Scale 1x | Maximize On Play | Mute Audio | Stats | Gizmos

Inspector

Transform

Position	X 0	Y 2	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Capsule (Mesh Filter)

Mesh: Capsule

Capsule Collider

Edit Collider

Is Trigger: ☐

Material: None (Physic Material)

Center: X 0 Y 0 Z 0

Radius: 0.5

Height: 2

Direction: Y-Axis

Mesh Renderer

Cast Shadows: On

Receive Shadows: ☑

Motion Vectors: Per Object Motion

Materials

Light Probes: Off

Reflection Probes: Blend Probes

Anchor Override: None (Transform)

Rigidbody

Mass: 1

Drag: 0

Angular Drag: 0.05

Use Gravity: ☑

Is Kinematic: ☐

Interpolate: None

Collision Detection: Discrete

Constraints

Player_Ctrl (Script)

Script: Player_Ctrl

PS: Run

Jump_Power: 700

Animator: BoxMan (Animator)

Default-Diffuse

Shader: Legacy Shaders/Diffuse

Add Component

Project | **Console**

Clear | Collapse | Clear on Play | Error Pause

10 0 0

```
coint get!!
UnityEngine.Debug:Log(Object)
coint get!!
UnityEngine.Debug:Log(Object)
coint get!!
UnityEngine.Debug:Log(Object)
coint get!!
UnityEngine.Debug:Log(Object)
coint get!!
UnityEngine.Debug:Log(Object)
coint get!!
UnityEngine.Debug:Log(Object)
```

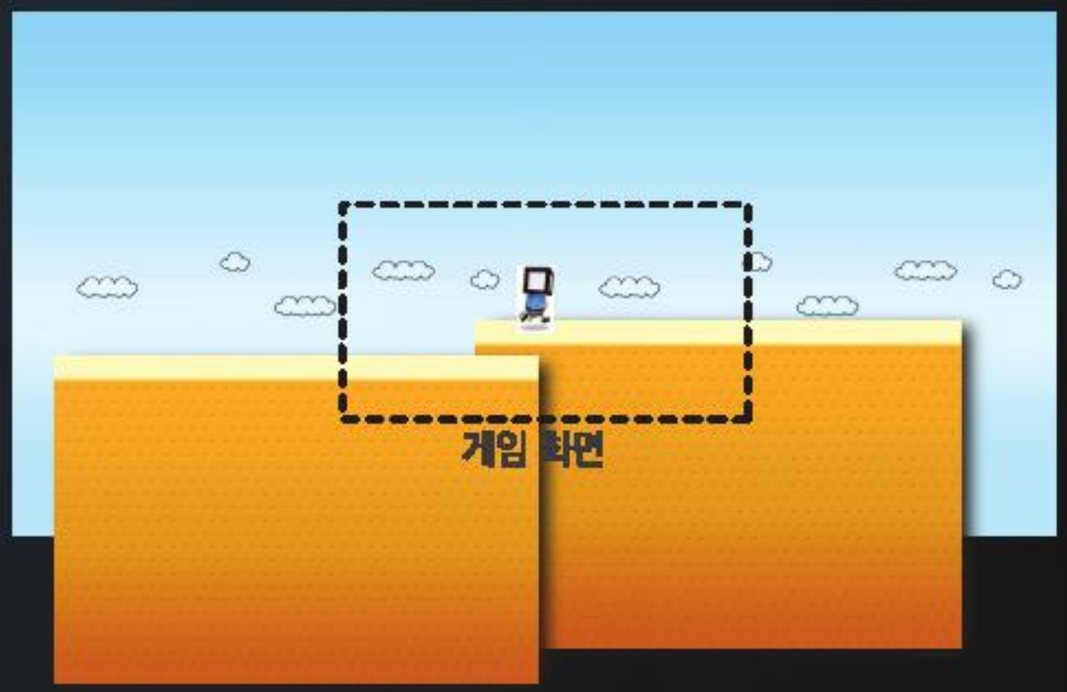
coint get!!

unity

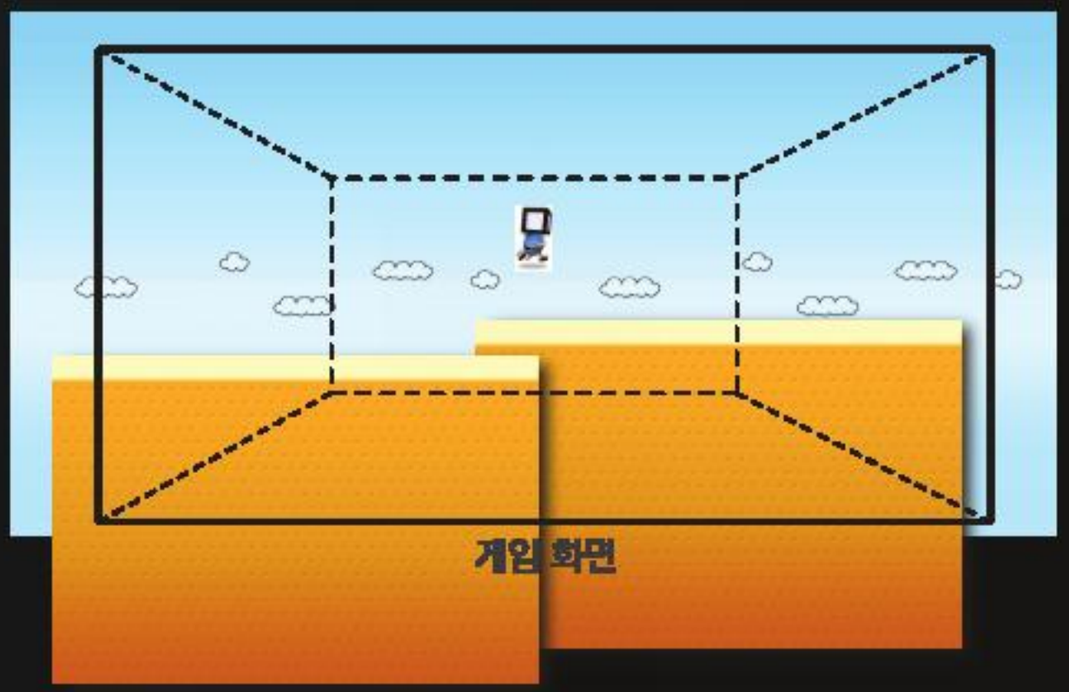
STEP.4 카메라 변경

Camera_Zoom

주인공이 높이 점프하는 경우 화면 바깥으로 나가버리는 사태가 발생합니다.
그걸 막기 위해 **주인공의 Y축의 값**을 기준으로 **카메라를 줌인 줌아웃** 하는
스크립트를 카메라에 적용시키도록합니다.



플레이어의 Y 축이 돌아오면 카메라를 줌인한다.



플레이어의 Y 축이 높아지면 카메라를 줌아웃한다.

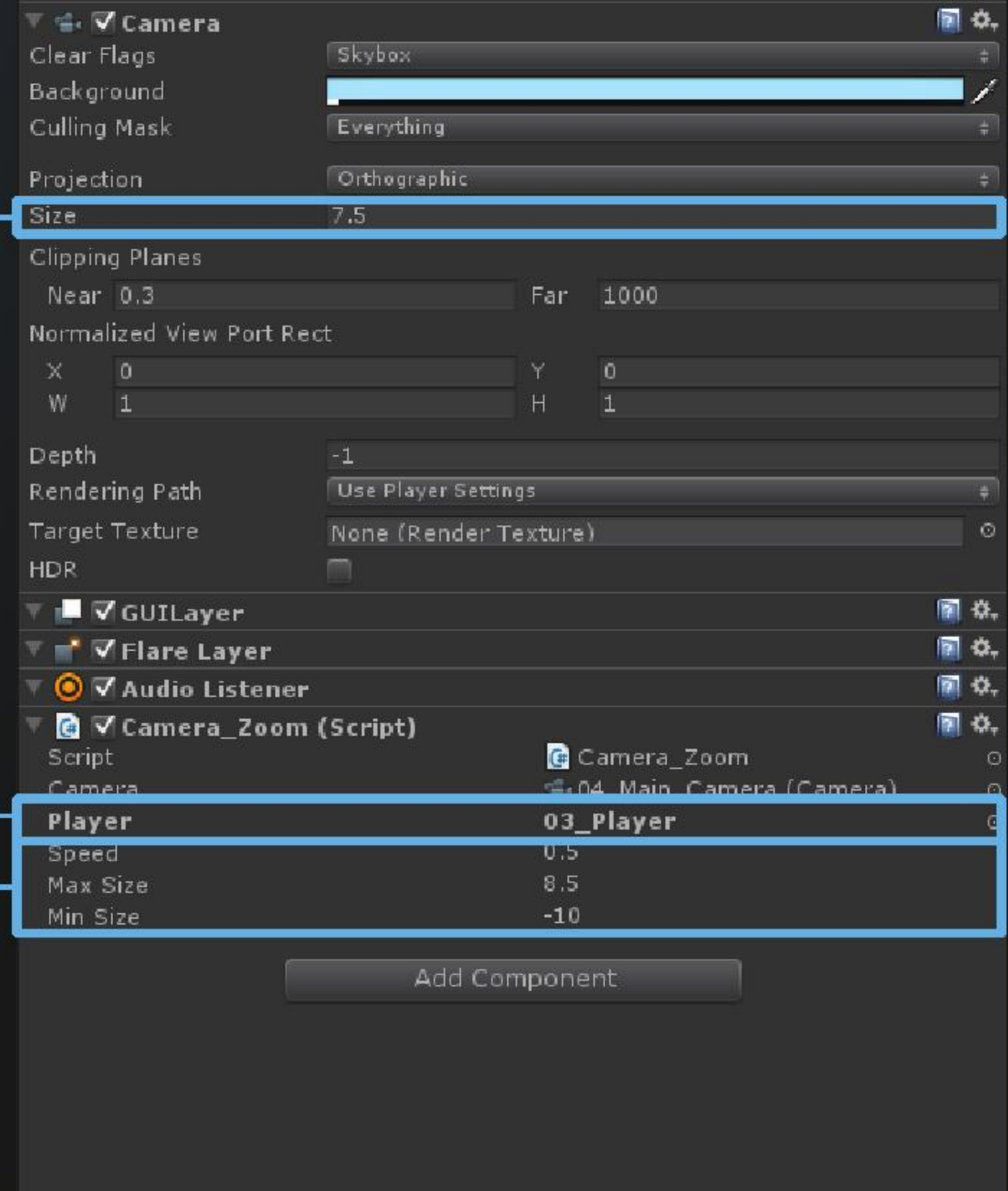
카메라 스크립트 작성

```
using UnityEngine;
using System.Collections;

public class CameraZoom : MonoBehaviour {

    public Camera camera;
    public GameObject player;

    public float speed = 0.5f;
    float cameraSize = 5f;

    public float MaxSize = 10f;
    public float MinSize = -5f;

    void Update () {

        //camera.orthographicSize = 5f + player.transform.position.y;
        cameraSize = 5f + player.transform.position.y;

        if(cameraSize>=MaxSize){
            cameraSize = MaxSize;
        }
        if(cameraSize<=MinSize){
            cameraSize = MinSize;
        }

        camera.orthographicSize = Mathf.Lerp(camera.orthographicSize, cameraSize, Time.deltaTime / speed);
    }
}
```

실제로는 이 **Size 값을 조정함**으로써
카메라를 줌인줌아웃합니다.
(우리가 조절할 필요는 없습니다.)

플레이어의 좌표에 따라
카메라의 사이즈가 변경되기 때문에
03_Player 오브젝트를 꼭 연결합니다.

카메라의
줌인줌아웃 스피드와
최대줌인 사이즈와
최소 줌아웃 사이즈를
정해놓을 수 있습니다.



Unity 5.5.2f1 Personal (64bit) - test1.unity - _BoxRunner_work3 - PC, Mac & Linux Standalone <DX11>

File Edit Assets GameObject Component Window Help

Center Local

Collab Account Layers Layout

Hierarchy

Create All

test1

BlockManager

Main Camera

BG

cloud

Cube_A

Cube_B

Player

DeathZone

Scene Game Asset Store Animator

Shaded 2D Gizmos All

Camera Zoom (Script)

Script	CameraZoom
Camera	Main Camera (Camera)
Player	Player
Speed	0.5
Max Size	10
Min Size	-5

Inspector

Main Camera Static

Tag MainCamera Layer Default

Transform

	X	Y	Z
Position	0	0	-10
Rotation	0	0	0
Scale	1	1	1

Camera

Clear Flags Skybox

Background

Culling Mask Everything

Projection Orthographic

Size 5

Clipping Planes Near 0.3 Far 1000

Viewport Rect X 0 Y 0 W 1 H 1

Depth -1

Rendering Path Use Graphics Settings

Target Texture None (Render Texture)

Occlusion Culling

HDR

Target Display Display 1

GUI Layer

Flare Layer

Audio Listener

Camera Zoom (Script)

Script	CameraZoom
Camera	Main Camera (Camera)
Player	Player
Speed	0.5
Max Size	10
Min Size	-5

Add Component

Project Console

Create

Favorites

All Materials

All Models

All Prefabs

All Scripts

Assets

prefab

script

sound

texture

Render_Final

boxman_jump

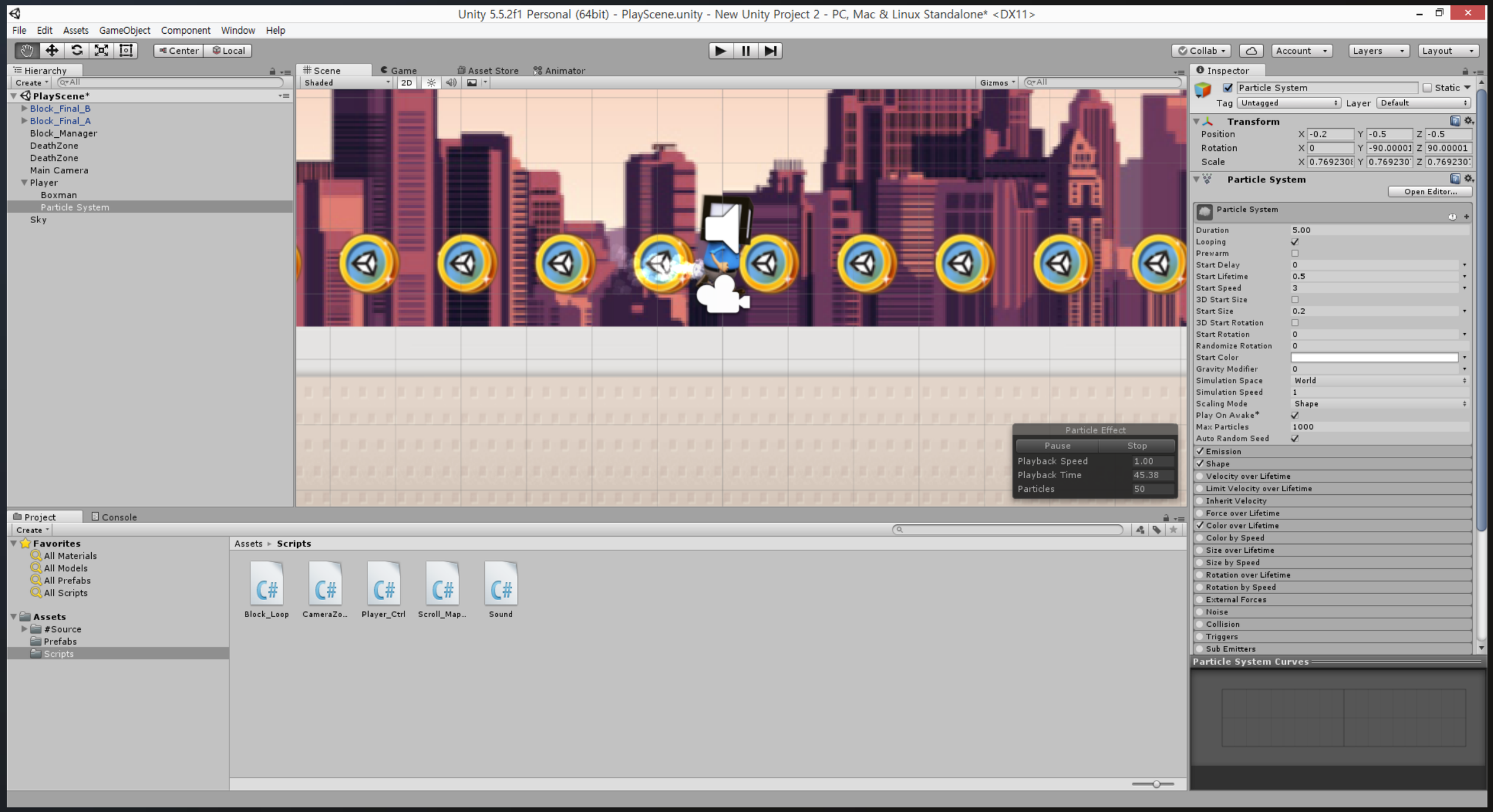
boxman_jump_02

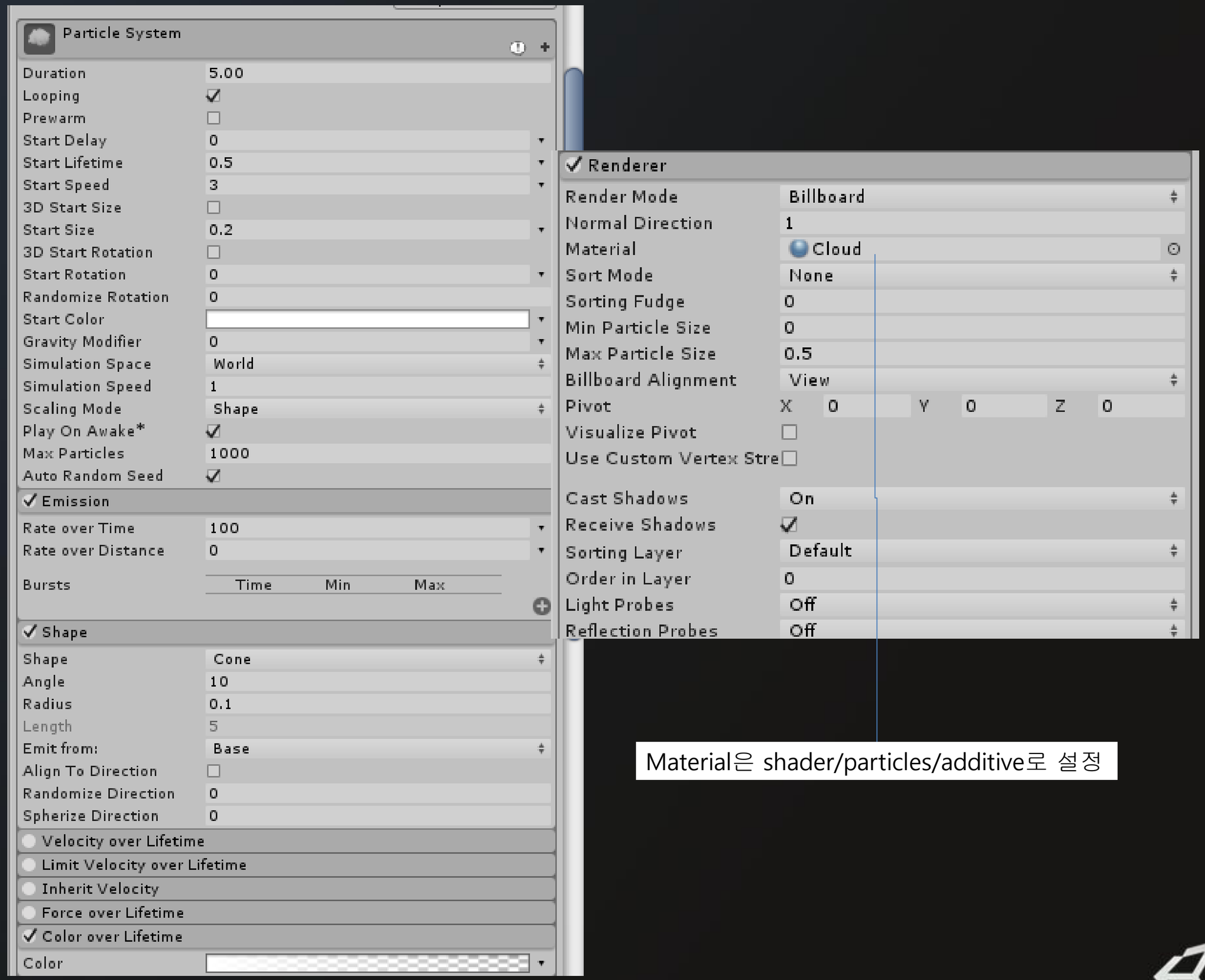
boxman_run

Assets ▸ script

Block_Loop CameraZo... Player_Ctrl Scroll_Map...

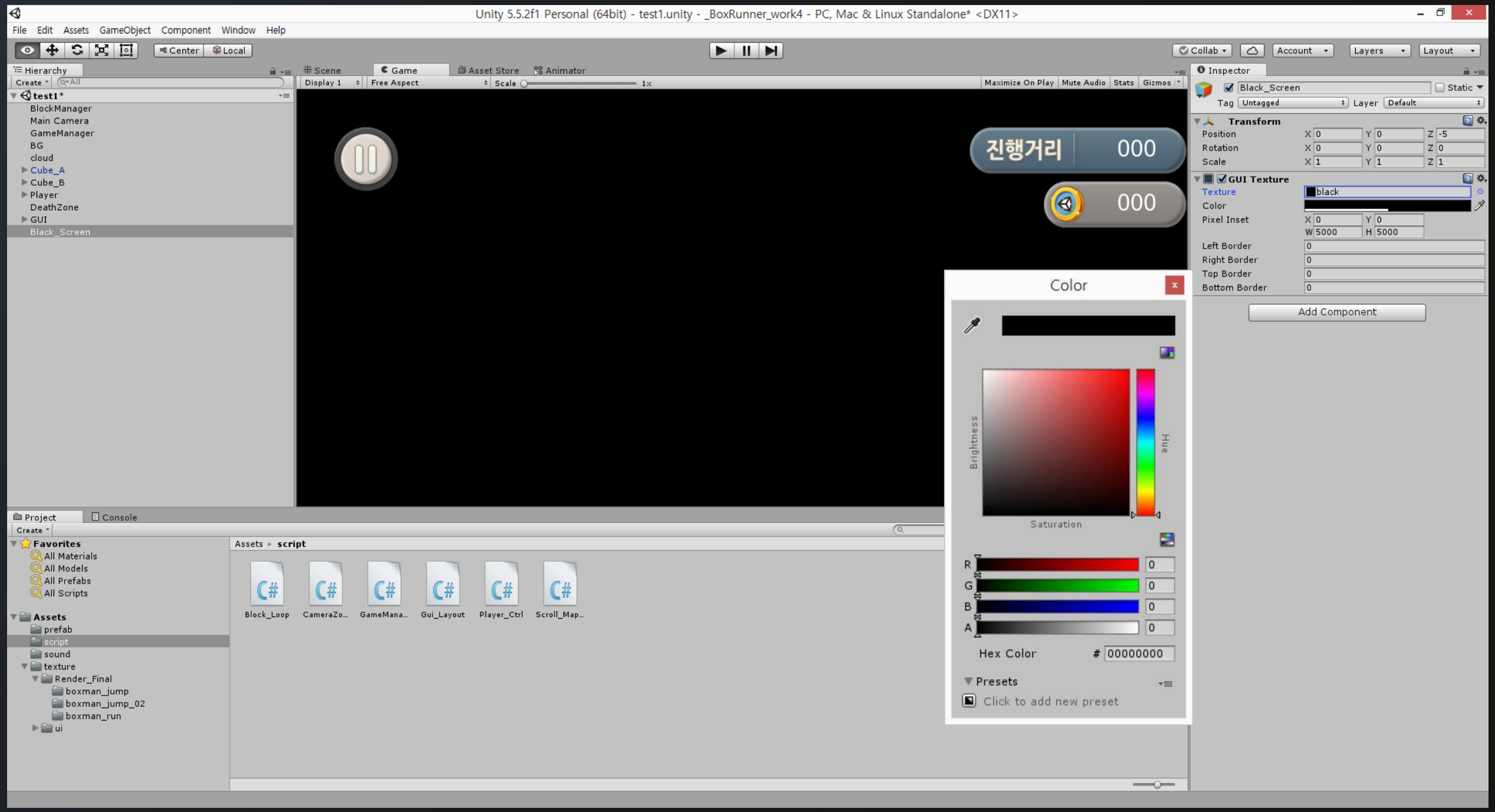
캐릭터 뒤에 따르는 연기 효과



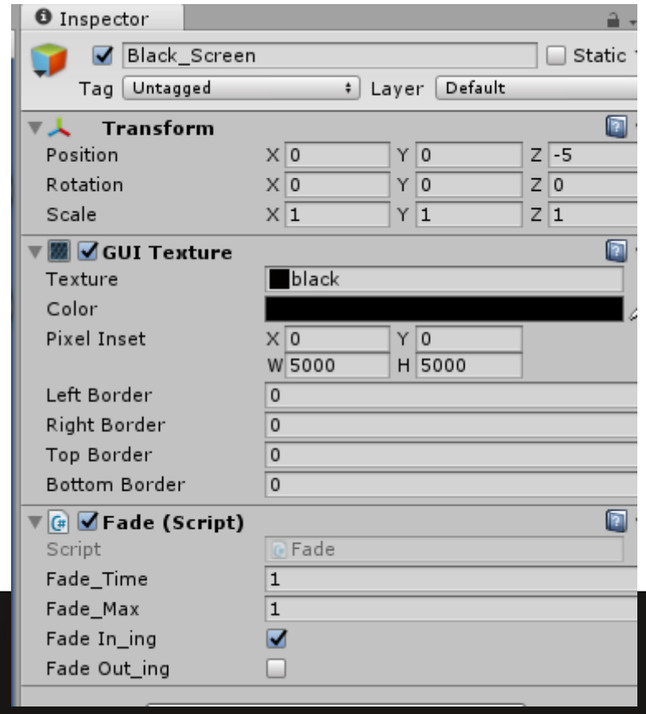


Material은 shader/particles/additive로 설정

Fade-in, Fade-out



```
using UnityEngine;
using System.Collections;
public class Fade : MonoBehaviour{
	GUITexture Black_screen;
	public float Fade_Time = 2f;
	public float Fade_Max = 1f;
	float _time;
	public bool FadeIn_ing = true;
	public bool FadeOut_ing;
	void Start ()
	{
		Black_screen = GetComponent<GUITexture> ();
	}
	void Update ()
	{
		if (FadeIn_ing) {
			_time += Time.deltaTime;
			Black_screen.color = Color.Lerp (new Color (0, 0, 0, Fade_Max), new Color (0, 0, 0, 0), _time / Fade_Time);
		}
		if (FadeOut_ing) {
			_time += Time.deltaTime;
			Black_screen.color = Color.Lerp (new Color (0, 0, 0, 0), new Color (0, 0, 0, Fade_Max), _time / Fade_Time);
		}
		if (_time >= Fade_Time) {
			_time = 0;

			FadeOut_ing = false;
			if(FadeIn_ing==true){
				Destroy(this.gameObject);
		}
	}
	public void FadeIn ()  {
		FadeIn_ing = true;
	}
	public void FadeOut (){
		FadeOut_ing = true;
	}
}
```



과제

코인을 획득할 때 마다 이펙트가 터지도록 하려면 어떻게 해야 할까?

HINT : Instantiate(effect, this.transform.position, Quaternion.identity);

STEP.5 게임매니저 만들기

unity

Game_Manager

만들어진 게임내의 요소들을 컨트롤하는 매니저가 역할을 하는 스크립트입니다.
다른 스크립트로부터 데이터를 가져오기도하고, 내보내기도 하며,
GUI와 관련된 다양한 기능들을 하는 게임의 총괄담당자 같은 역할을 합니다.

- Game_Manager
  - 게임 진행 상태
    - Play
    - Pause
    - End
  - 전체 게임스피드
  - 게임진행거리
  - 획득한 돈

unity

GUI구성

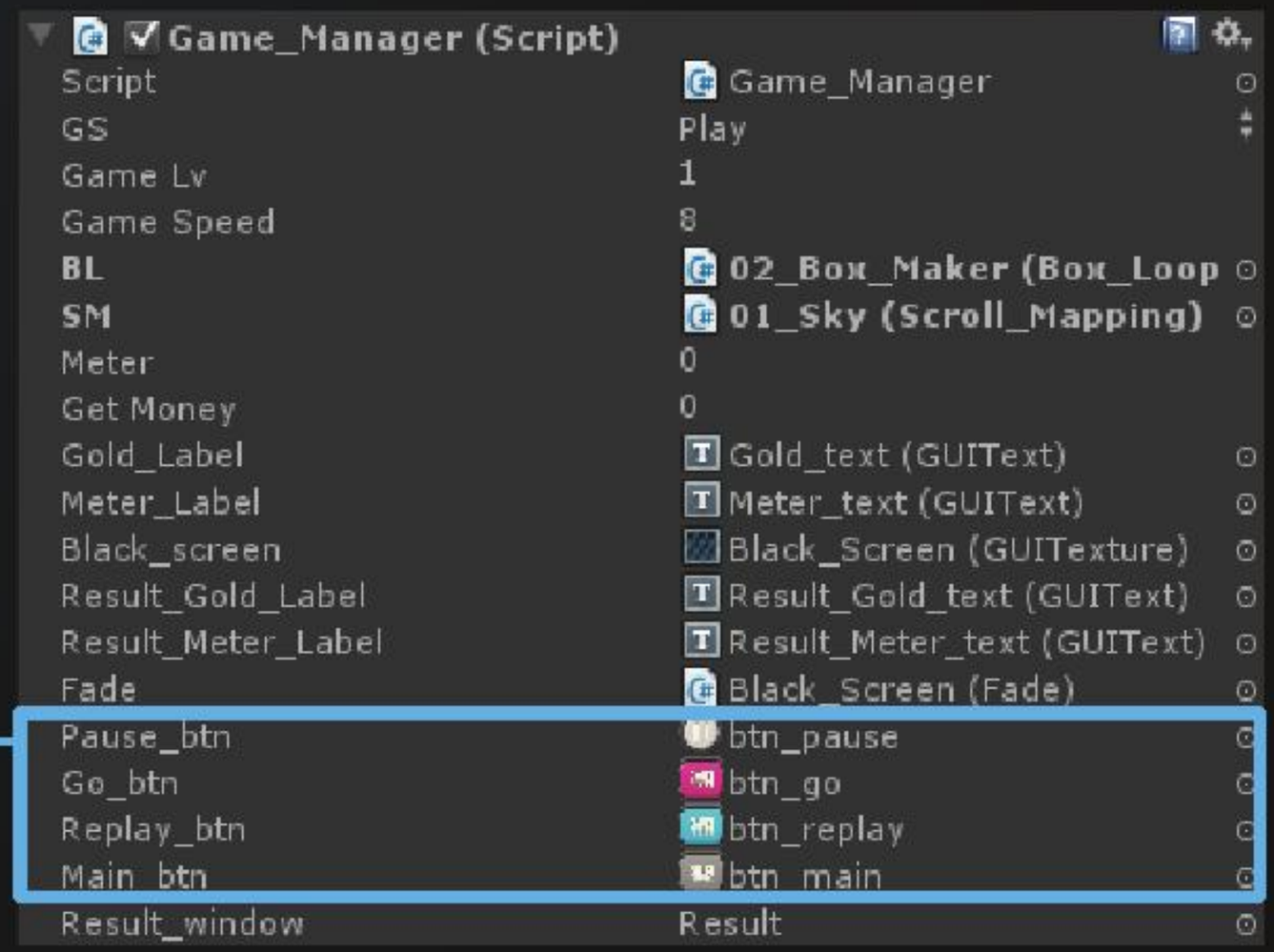
플레이 화면

결과 화면

GUI는 버튼형과 표기형 두가지로 구분해서
작업을 진행했습니다.

버튼형 GUI

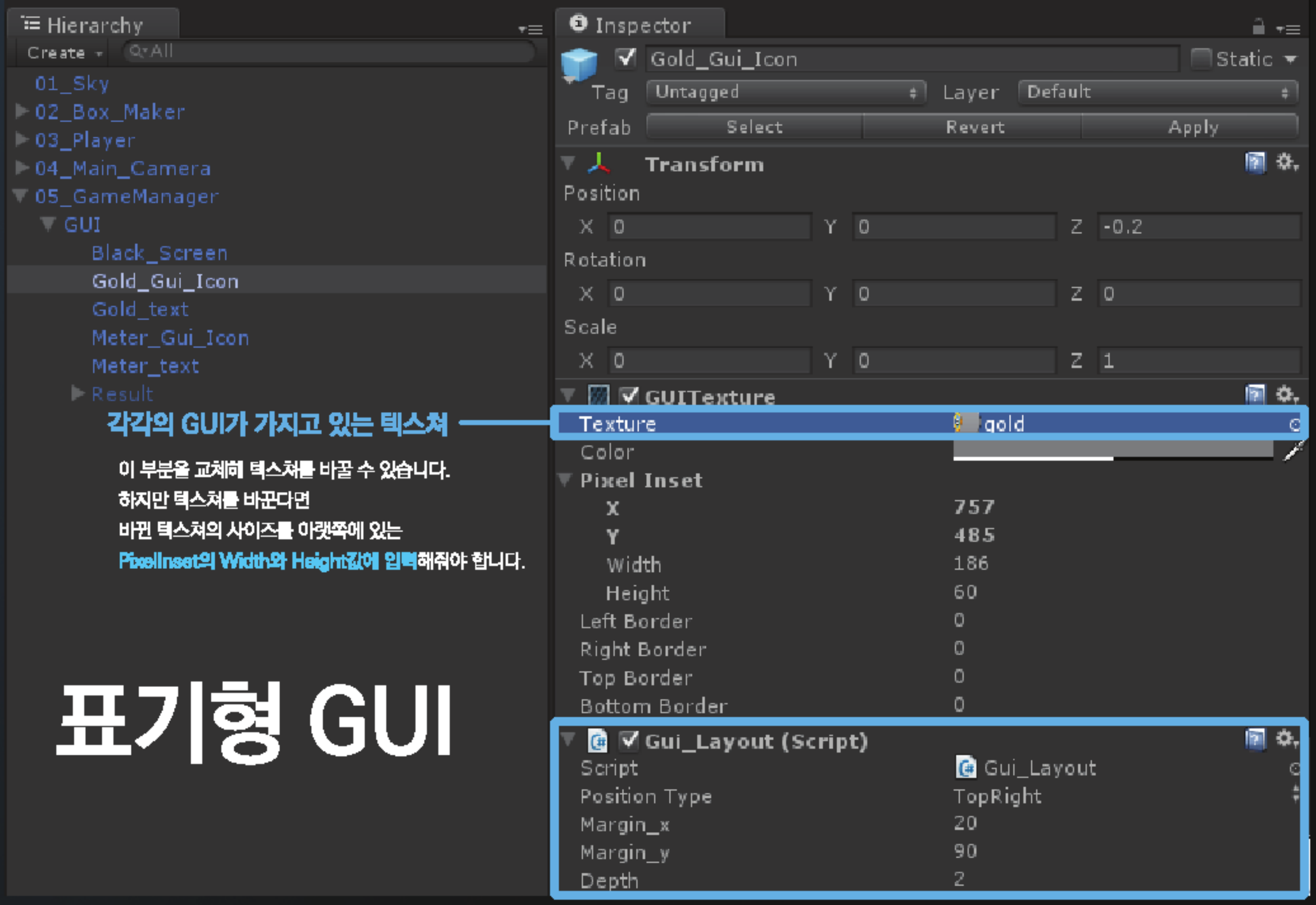
05_GameManager 의
Game_Manager 스크립트에는
GUI 버튼의 이미지를 선택하는
부분이 만들어져 있습니다.



GUI버튼으로 사용하기 위한 텍스쳐들

기본적으로 셋팅해두었지만
자신이 사용하고 싶은 이미지로 언제든 교체 할 수 있습니다.

05_Game_Manager 오브젝트



각각의 GUI가 가지고 있는 텍스쳐

이 부분을 교체해 텍스쳐를 바꿀 수 있습니다.
하지만 텍스쳐를 바꾼다면
바뀐 텍스쳐의 사이즈를 아랫쪽에 있는
PixelInset의 Width와 Height값에 입력해줘야 합니다.

표기형 GUI

05_GameManager 의 하위요소로 되어있는 각각의 GUI 를
눌러 하나하나씩 텍스쳐를 바꿔 줄수도 있습니다.

unity

```
using UnityEngine;
using System.Collections;

public enum GameState
{
    Play,
    Pause,
    End
}

public class GameManager : MonoBehaviour
{
    public GameState GS;

    public GUIText Text_Meter;
    public GUIText Text_Gold;
    public GameObject Final_GUI;
    public GUIText Final_Meter;
    public GUIText Final_Gold;
    public GameObject Pause_GUI;

    public float Speed;
    public float Meter;
    public int Gold;

    public void Update()
    {

        if(GS == GameState.Play){
            Meter +=Time.deltaTime*Speed;
            Text_Meter.text = string.Format("{0:N0}m", Meter);
        }
    }

    public void CoinGet()
    {
        Gold+=1;
        Text_Gold.text = string.Format("{0}", Gold);
    }

    public void GameOver()
    {

        Final_Meter.text = string.Format("{0:N1}", Meter);
        Final_Gold.text = string.Format("{0}", Gold);

        GS = GameState.End;
        Final_GUI.SetActive(true);

    }

    public void Replay()
    {

        Time.timeScale = 1f;

        Application.LoadLevel("PlayScene");

    }

    public void MainGo()
    {

        Application.LoadLevel("IntroScene");
    }

    public void Pause()
    {

        GS = GameState.Pause;
        Time.timeScale =0f;
        Pause_GUI.SetActive(true);

    }

    public void UnPause()
    {
        GS = GameState.Play;
        Time.timeScale =1f;
        Pause_GUI.SetActive(false);

    }
}
```

```
using UnityEngine;
using System.Collections;
public enum PlayerState{
                Run,
                Jump,
                D_Jump,
                Death
                }
public class Player_Ctrl : MonoBehaviour {
                public PlayerState PS;
                public float Jump_Power = 500f;
                public AudioClip[] Sound;
                public Animator animator;

                public GameManager GM;
```

```
void CoinGet()
{
        SoundPlay(0);
        if(GM !=null)
        {
                GM.CoinGet();
        }
}

void GameOver()
{
        PS = PlayerState.Death;
        SoundPlay(1);
        GM.GameOver();
}
```

Gui_Layout

오브젝트형 GUI의 레이아웃을 설정하기 위한 스크립트로,
어떤화면의 스크린에도 맞출 수 있도록 제작되었습니다.

셋팅할 변수들 : 피봇 포인트, X_margin, X_margin

1. 화면 사이즈를 구한다.
2. 피봇 포인트를 정한다.
3. 각각의 X,Y 마진값만큼 여백을 밀어준다.

OnGUI 사용법.

```
OnGUI(){
    if (GS == GameState.Play)
    {
        if (GUI.Button(new Rect(20, 20, Pause_btn.width, Pause_btn.height), Pause_btn, guiRectStyle))
        {
            Black_screen.color = new Color(0, 0, 0, 0.4f);
            GS = GameState.Pause;
            Time.timeScale = 0;
        }
    }
}
```

게임이 이 상태일때는

이렇게 생긴 GUI를 만들고

누르면 이렇게 됩니다.

unity

버튼으로 만들 두가지 기능

일시정지(해제)

특정버튼(GUIButton)을 누르면,
시간이 멈추고,
게임의 상태는 일시정지가 된다.

일시정지

```
Time.timeScale = 0;
GS = GameState.Pause;
```

리플레이(메인화면)

특정버튼(GUIButton)을 누르면,
특정씬(리플레이의 경우 이씬) 을 불러온다.

리플레이

```
Application.LoadLevel("[PlayScene]");
```

unity

Game_Manager 의 함수

Update — 게임상태가 플레이중일때,
MeterCheck를 계속 돌린다.

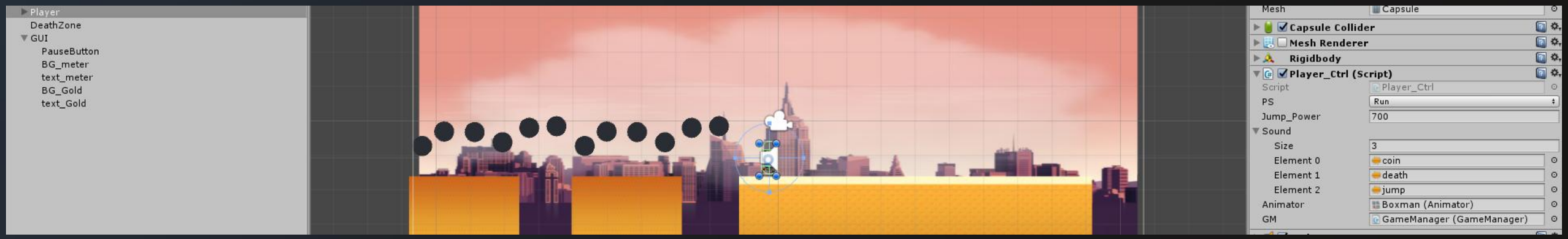
MeterCheck — 게임의 진행속도만큼 진행거리에 계속 더한다.
진행거리를 진행거리 GUIText에 반영한다.

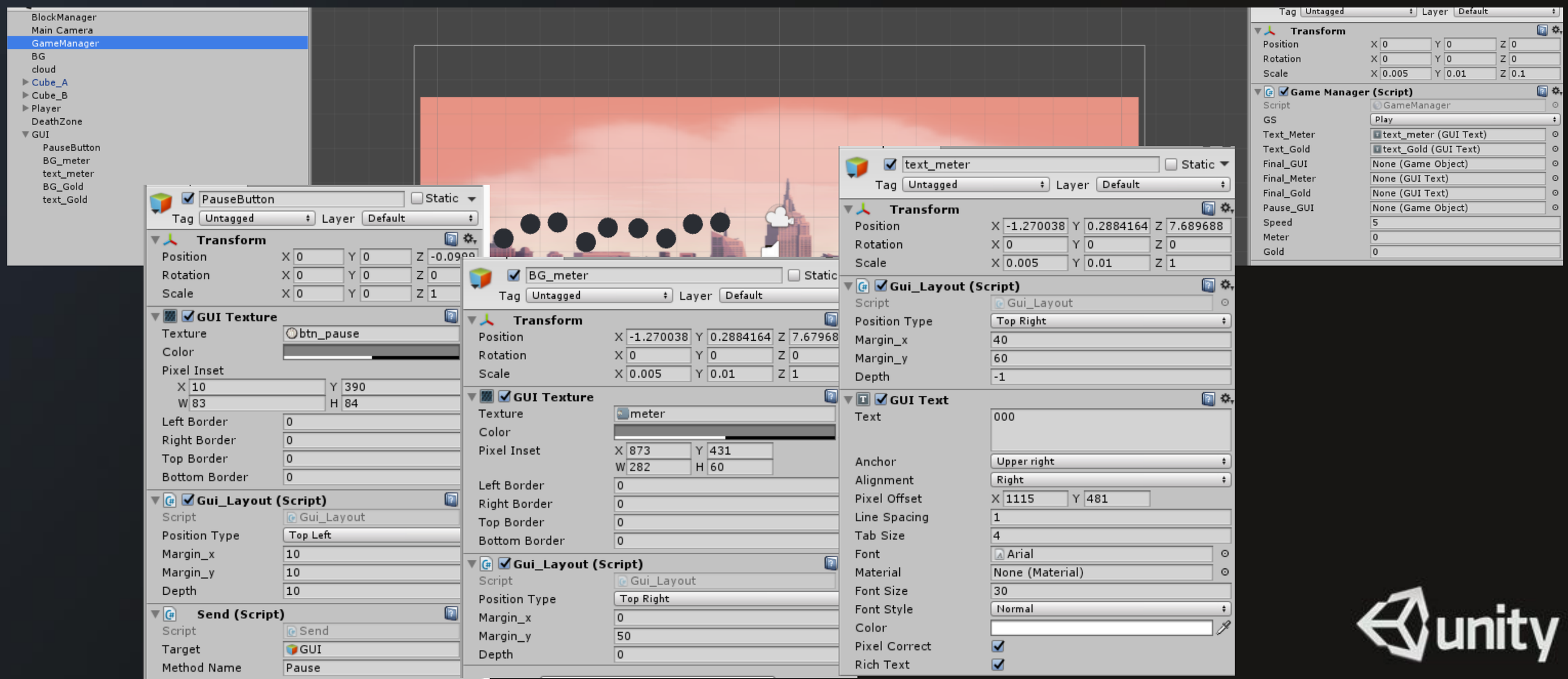
진행한 거리가 100m가 될때마다
LevelUp

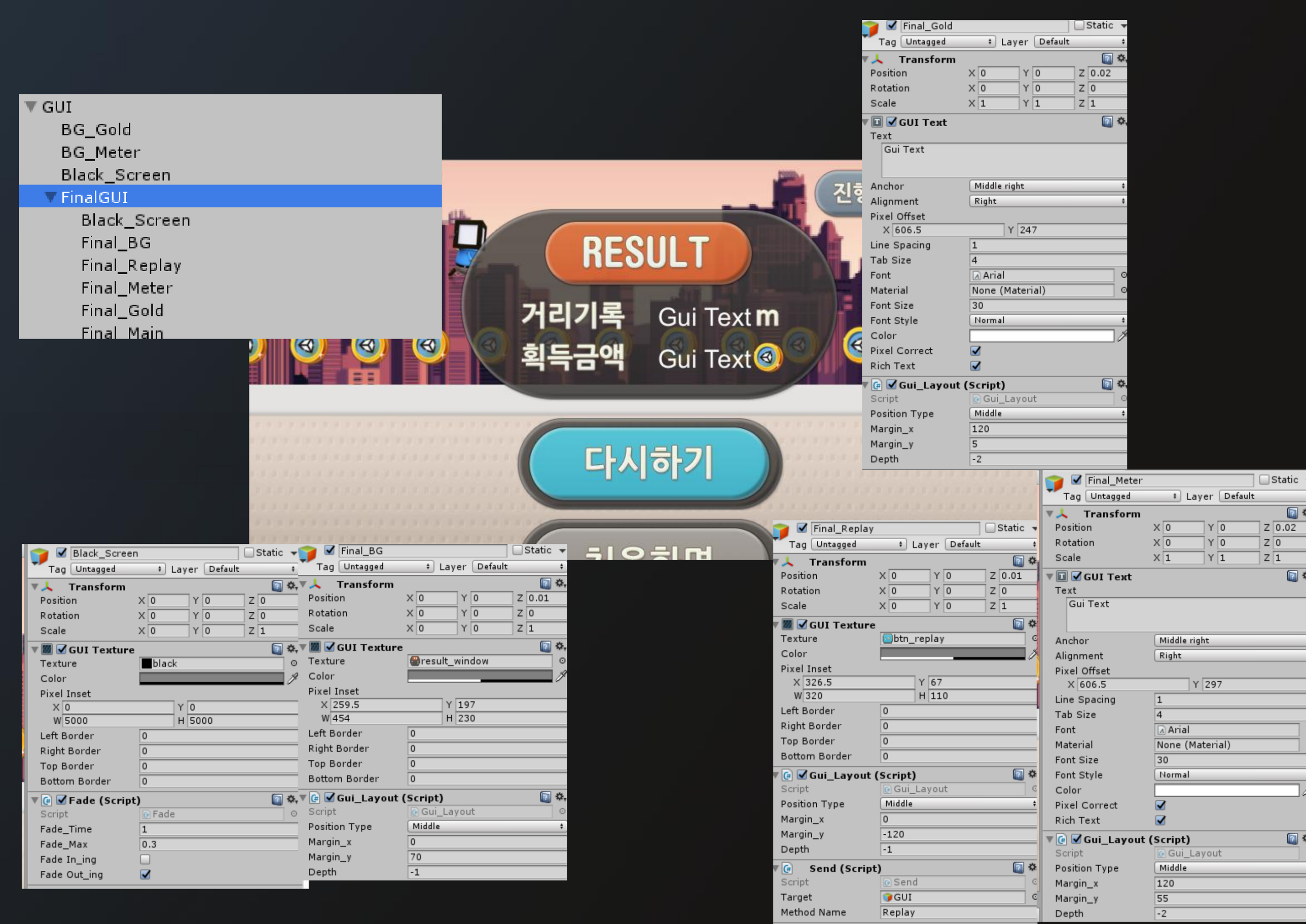
LevelUp — 블럭과 배경의 이동속도가 빨라진다.

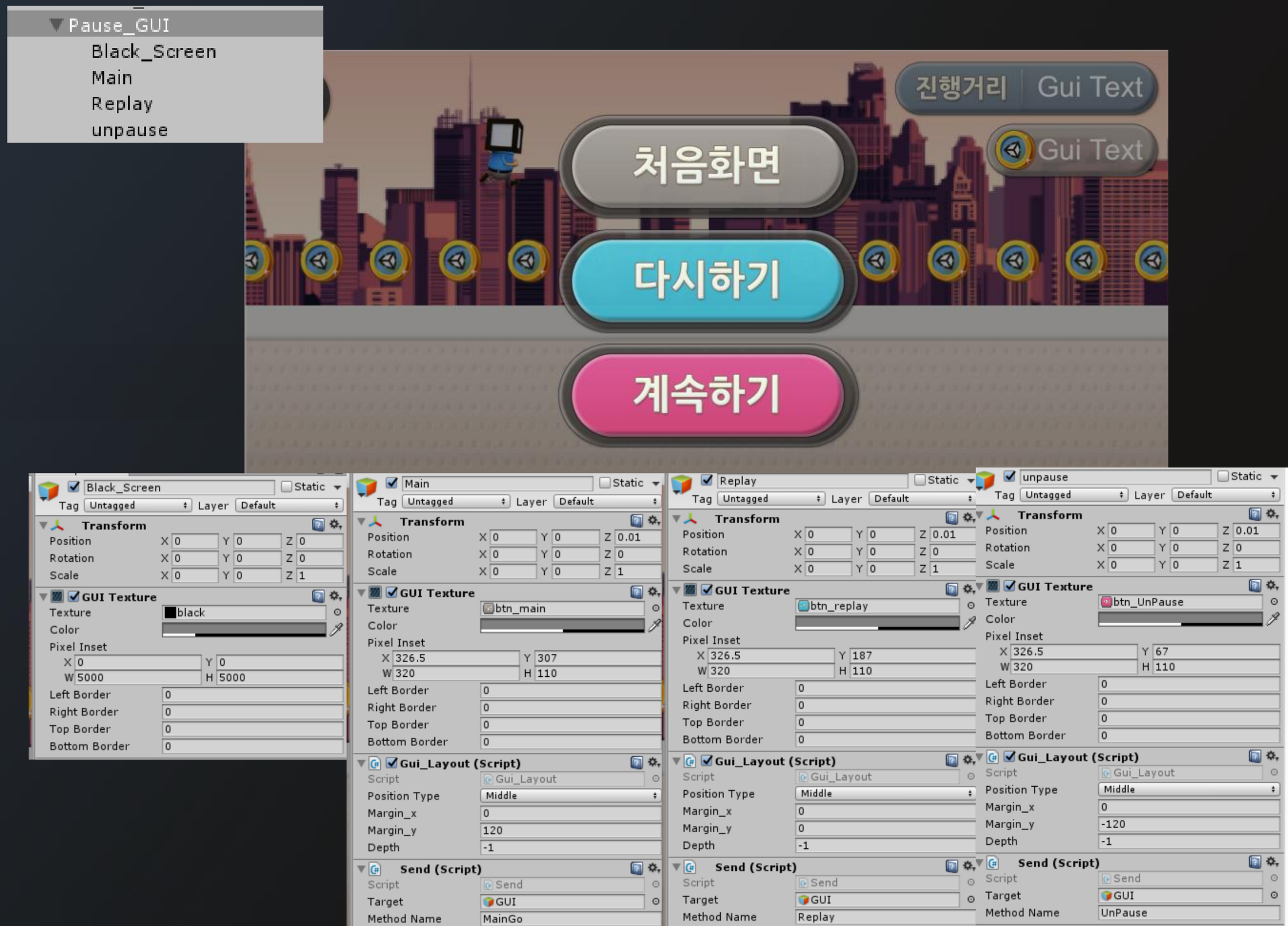
unity

```
using UnityEngine;
using System.Collections;

[ExecuteInEditMode]
public class Gui_Layout : MonoBehaviour
{
    public enum positionType
    {
        TopLeft,
        TopMiddle,
        TopRight,
        MiddleLeft,
        Middle,
        MiddleRight,
        BottomLeft,
        BottomMiddle,
        BottomRight
    }

    public positionType _positionType = positionType.Middle;
    public float margin_x;
    public float margin_y;
    public int _depth;
    float screenX;
    float screenY;
    GUIText _gui_text;
    GUITexture _gui_texture;
    float _guiWidth;
    float _guiHeight;

    bool TextureIN=false;

    void Awake ()
    {
        #if !(UNITY_EDITOR)

        _gui_text = GetComponent<GUIText> ();
        _gui_texture = GetComponent<GUITexture> ();
        screenX = Screen.width;
        screenY = Screen.height;



        if (_gui_texture != null) {
            _guiWidth = _gui_texture.pixelInset.width;
            _guiHeight = _gui_texture.pixelInset.height;
        }
```


Player
DeathZone
GUI
PauseButton
BG_meter
text_meter
BG_Gold
text_Gold
Mesh Capsule
Capsule Collider
Mesh Renderer
Rigidbody
Player_Ctrl (Script)
Script Player_Ctrl
PS Run
Jump_Power 700
Sound
Size 3
Element 0 coin
Element 1 death
Element 2 jump
Animator Boxman (Animator)
GM GameManager (GameManager)


BlockManager
Main Camera
GameManager
BG
cloud
Cube_A
Cube_B
Player
DeathZone
GUI
PauseButton
BG_meter
text_meter
BG_Gold
text_Gold
Tag Untagged Layer Default
Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 0.005 Y 0.01 Z 0.1
Game Manager (Script)
Script GameManager
GS Play
Text_Meter text_meter (GUI Text)
Text_Gold text_Gold (GUI Text)
Final_GUI None (Game Object)
Final_Meter None (GUI Text)
Final_Gold None (GUI Text)
Pause_GUI None (Game Object)
Speed 5
Meter 0
Gold 0
PauseButton Static
Tag Untagged Layer Default
Transform
Position X 0 Y 0 Z -0.0999
Rotation X 0 Y 0 Z 0
Scale X 0 Y 0 Z 1
GUI Texture
Texture btn_pause
Color
Pixel Inset
X 10 Y 390
W 83 H 84
Left Border 0
Right Border 0
Top Border 0
Bottom Border 0
Gui_Layout (Script)
Script Gui_Layout
Position Type Top Left
Margin_x 10
Margin_y 10
Depth 10
Send (Script)
Script Send
Target GUI
Method Name Pause
BG_meter Static
Tag Untagged Layer Default
Transform
Position X -1.270038 Y 0.2884164 Z 7.67968
Rotation X 0 Y 0 Z 0
Scale X 0.005 Y 0.01 Z 1
GUI Texture
Texture meter
Color
Pixel Inset
X 873 Y 431
W 282 H 60
Left Border 0
Right Border 0
Top Border 0
Bottom Border 0
Gui_Layout (Script)
Script Gui_Layout
Position Type Top Right
Margin_x 0
Margin_y 50
Depth 0
text_meter Static
Tag Untagged Layer Default
Transform
Position X -1.270038 Y 0.2884164 Z 7.689688
Rotation X 0 Y 0 Z 0
Scale X 0.005 Y 0.01 Z 1
Gui_Layout (Script)
Script Gui_Layout
Position Type Top Right
Margin_x 40
Margin_y 60
Depth -1
GUI Text
Text 000
Anchor Upper right
Alignment Right
Pixel Offset X 1115 Y 481
Line Spacing 1
Tab Size 4
Font Arial
Material None (Material)
Font Size 30
Font Style Normal
Color
Pixel Correct
Rich Text


GUI
BG_Gold
BG_Meter
Black_Screen
FinalGUI
Black_Screen
Final_BG
Final_Replay
Final_Meter
Final_Gold
Final_Main
RESULT
거리기록
Gui Text m
획득금액
Gui Text
다시하기
Final_Gold
Static
Tag Untagged
Layer Default
Transform
Position X 0 Y 0 Z 0.02
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
GUI Text
Text
Gui Text
Anchor Middle right
Alignment Right
Pixel Offset
X 606.5 Y 247
Line Spacing 1
Tab Size 4
Font Arial
Material None (Material)
Font Size 30
Font Style Normal
Color
Pixel Correct
Rich Text
Gui_Layout (Script)
Script Gui_Layout
Position Type Middle
Margin_x 120
Margin_y 5
Depth -2
Black_Screen
Static
Tag Untagged
Layer Default
Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 0 Y 0 Z 1
GUI Texture
Texture black
Color
Pixel Inset
X 0 Y 0
W 5000 H 5000
Left Border 0
Right Border 0
Top Border 0
Bottom Border 0
Fade (Script)
Script Fade
Fade_Time 1
Fade_Max 0.3
Fade In_ing
Fade Out_ing
Final_BG
Static
Tag Untagged
Layer Default
Transform
Position X 0 Y 0 Z 0.01
Rotation X 0 Y 0 Z 0
Scale X 0 Y 0 Z 1
GUI Texture
Texture result_window
Color
Pixel Inset
X 259.5 Y 197
W 454 H 230
Left Border 0
Right Border 0
Top Border 0
Bottom Border 0
Gui_Layout (Script)
Script Gui_Layout
Position Type Middle
Margin_x 0
Margin_y 70
Depth -1
Final_Replay
Static
Tag Untagged
Layer Default
Transform
Position X 0 Y 0 Z 0.01
Rotation X 0 Y 0 Z 0
Scale X 0 Y 0 Z 1
GUI Texture
Texture btn_replay
Color
Pixel Inset
X 326.5 Y 67
W 320 H 110
Left Border 0
Right Border 0
Top Border 0
Bottom Border 0
Gui_Layout (Script)
Script Gui_Layout
Position Type Middle
Margin_x 0
Margin_y -120
Depth -1
Send (Script)
Script Send
Target GUI
Method Name Replay
Final_Meter
Static
Tag Untagged
Layer Default
Transform
Position X 0 Y 0 Z 0.02
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
GUI Text
Text
Gui Text
Anchor Middle right
Alignment Right
Pixel Offset
X 606.5 Y 297
Line Spacing 1
Tab Size 4
Font Arial
Material None (Material)
Font Size 30
Font Style Normal
Color
Pixel Correct
Rich Text
Gui_Layout (Script)
Script Gui_Layout
Position Type Middle
Margin_x 120
Margin_y 55
Depth -2


Pause_GUI
Black_Screen
Main
Replay
unpause
진행거리
Gui Text
Gui Text
처음화면
다시하기
계속하기
Black_Screen
Static
Tag Untagged
Layer Default
Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 0 Y 0 Z 1
GUI Texture
Texture black
Color
Pixel Inset
X 0 Y 0
W 5000 H 5000
Left Border 0
Right Border 0
Top Border 0
Bottom Border 0
Main
Static
Tag Untagged
Layer Default
Transform
Position X 0 Y 0 Z 0.01
Rotation X 0 Y 0 Z 0
Scale X 0 Y 0 Z 1
GUI Texture
Texture btn_main
Color
Pixel Inset
X 326.5 Y 307
W 320 H 110
Left Border 0
Right Border 0
Top Border 0
Bottom Border 0
Gui_Layout (Script)
Script Gui_Layout
Position Type Middle
Margin_x 0
Margin_y 120
Depth -1
Send (Script)
Script Send
Target GUI
Method Name MainGo
Replay
Static
Tag Untagged
Layer Default
Transform
Position X 0 Y 0 Z 0.01
Rotation X 0 Y 0 Z 0
Scale X 0 Y 0 Z 1
GUI Texture
Texture btn_replay
Color
Pixel Inset
X 326.5 Y 187
W 320 H 110
Left Border 0
Right Border 0
Top Border 0
Bottom Border 0
Gui_Layout (Script)
Script Gui_Layout
Position Type Middle
Margin_x 0
Margin_y 0
Depth -1
Send (Script)
Script Send
Target GUI
Method Name Replay
unpause
Static
Tag Untagged
Layer Default
Transform
Position X 0 Y 0 Z 0.01
Rotation X 0 Y 0 Z 0
Scale X 0 Y 0 Z 1
GUI Texture
Texture btn_UnPause
Color
Pixel Inset
X 326.5 Y 67
W 320 H 110
Left Border 0
Right Border 0
Top Border 0
Bottom Border 0
Gui_Layout (Script)
Script Gui_Layout
Position Type Middle
Margin_x 0
Margin_y -120
Depth -1
Send (Script)
Script Send
Target GUI
Method Name UnPause
unity

```
using UnityEngine;using System.Collections;
public class Send : MonoBehaviour
{
        public GameObject Target;
        public string MethodName;
        public void OnMouseDown()
        {
        Target.SendMessage(MethodName);
        }
}
```

Hierarchy
Create All
IntroScene*
Block_Manager
Main Camera
Player
Sky
Start
Title

Scene | Game | Asset Store
Display 1 | Free Aspect | Scale 1x | Maximize on Play | Mute audio | Stats | Gizmos

Start | Static
Tag Untagged | Layer Default

Transform

Position	X 0	Y 0	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 0	Y 0	Z 1

GUI Texture

Texture	btn_start
Color	
Pixel Inset	
X 326.5	Y 87
W 320	H 110
Left Border	0
Right Border	0
Top Border	0
Bottom Border	0

Gui_Layout (Script)

Script	Gui_Layout
Position Type	Middle
Margin_x	0
Margin_y	-100
Depth	0

Scene Move (Script)

Script	SceneMove
Scene Name	PlayScene

Title | Static
Tag Untagged | Layer Default

Transform

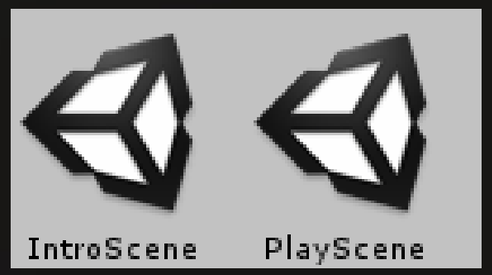
Position	X 0	Y 0	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 0	Y 0	Z 1

GUI Texture

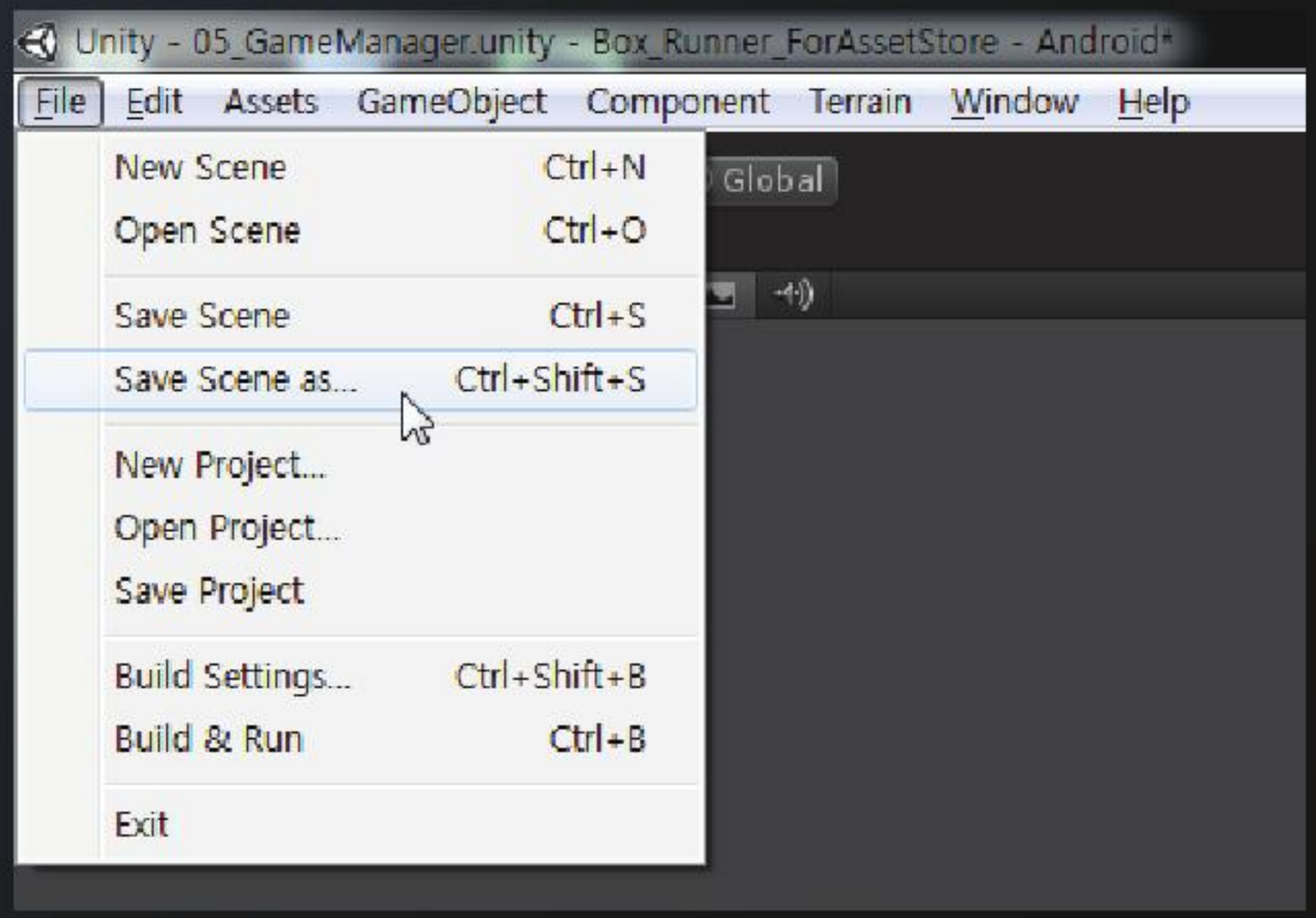
Texture	title
Color	
Pixel Inset	
X 179	Y 271
W 615	H 142
Left Border	0
Right Border	0
Top Border	0
Bottom Border	0

Gui_Layout (Script)

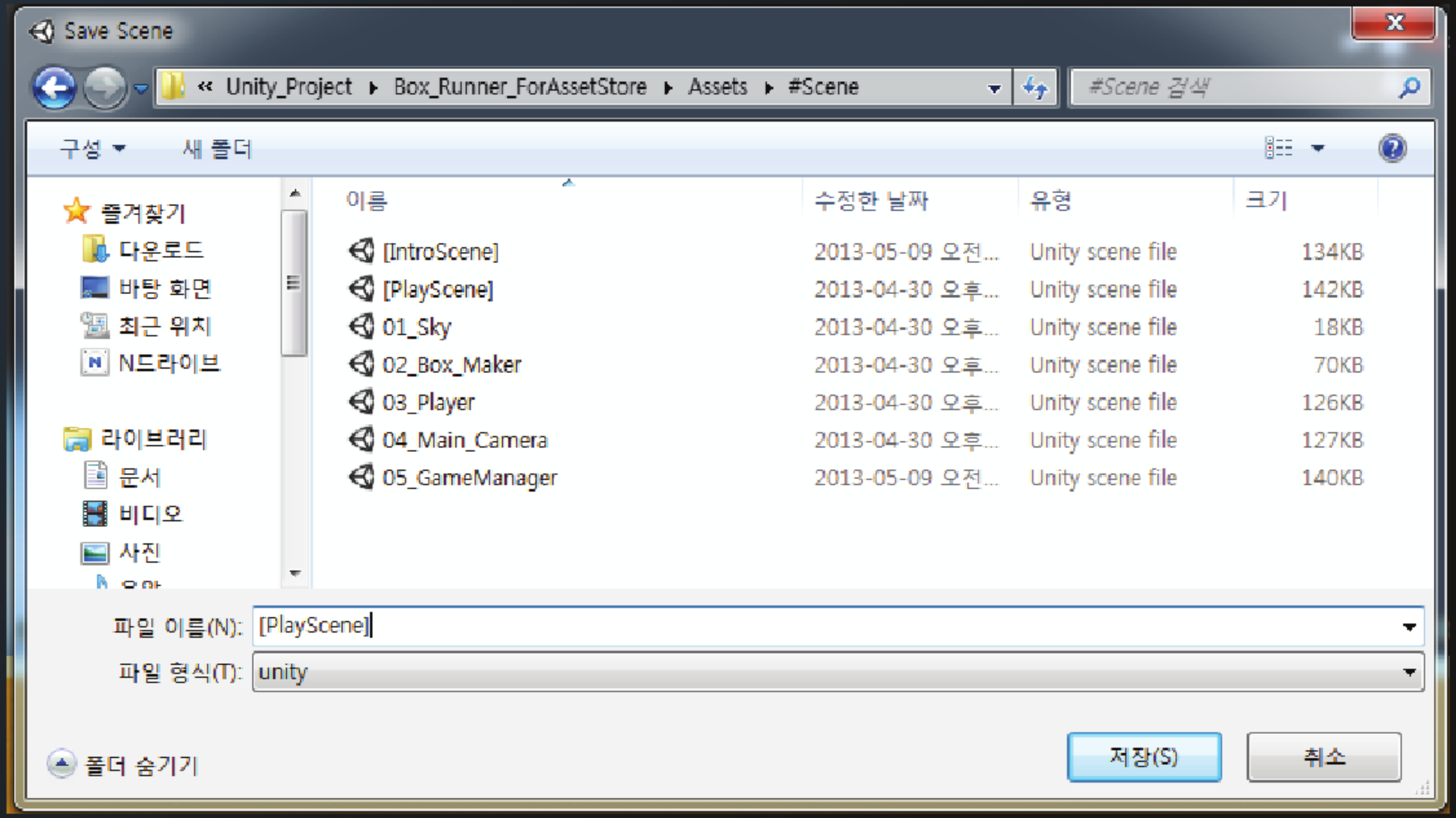
Script	Gui_Layout
Position Type	Middle
Margin_x	0
Margin_y	100
Depth	0



unity



지금까지 한 것을 저장하기 위해
유니티 상단메뉴의 Save Scene as를 눌러 저장할 준비를 합니다.
씬이 동작이 잘 안되는 분은 **05_GameManager 씬을 열고 Save Scene as** 를 합니다.



스크립트에서 [PlayScene] 이라는 이름의 씬을 불러오게 짜두었기 때문에
#Scene 폴더 안에 [PlayScene] 이라는 이름으로 저장합니다.

과제

생성되는 블록을 여러 개 유형 중에서 랜덤으로 선택되도록 하려면 어떻게 해야 할까?

HINT :
GameObject[] Block
Block[A]
int A = Random.Range(0, Block.Length);

STEP.6 Intro씬 제작과 최종빌드

unity

Mac, Window

별다른 필요 없이 그냥 빌드해서 테스트 가능.

iOS

개발자 등록이 안되어 있으면 기기에 넣어 테스트 하는 것이 불가.

Android

처음에 약간의 셋팅이 필요하지만, 셋팅 후에는
아무 제약없이 원하는대로 빌드해서 기기에 넣어 테스트가 가능.

unity

For Android Build

각각의 플랫폼에 따라 셋팅해줘야하는 상황이 다르므로
각각의 PC운영체제에 맞춰 셋팅하도록 합니다.

Windows

Android SDK 설치
JDK 설치
환경변수 설정

Mac

Android SDK 설치

unity

내컴퓨터 (오른쪽키) 〉 속성 〉 고급시스템설정 〉 시스템변수 〉 Path

C:\Program Files (x86)\Java\jdk1.7.0_03\bin;

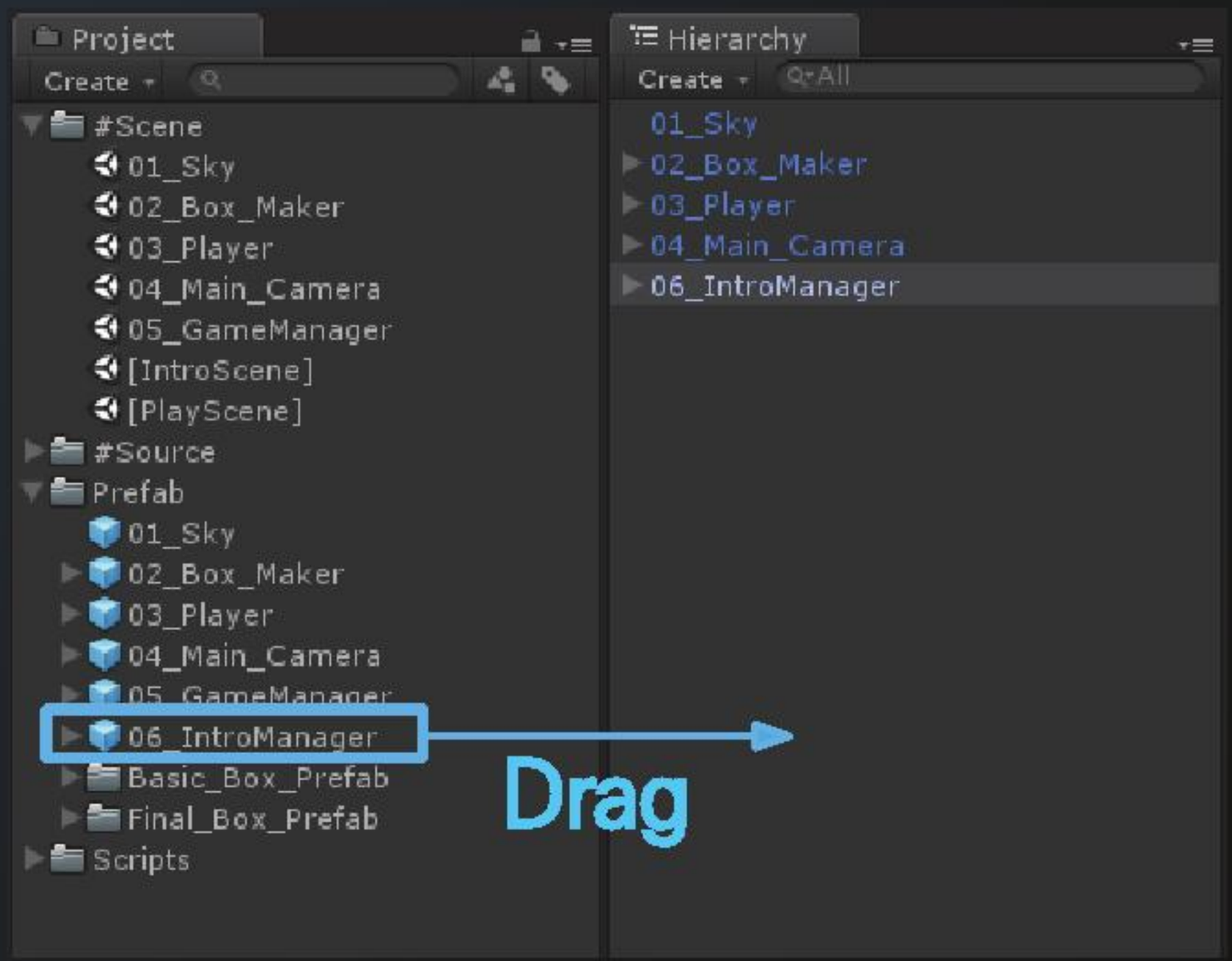
unity

STEP.6 Intro씬 만들고 최종 빌딩

[PlayScene]을 복사합니다.

[PlayScene]을 누르고 컨트롤 + D 를 누르면 복제됩니다.

복제된 씬의 이름을 [IntroScene]으로 바꿉니다.



타이틀이 들어간 화면을 만들기 위해
[IntroScene] 으로 들어갑니다.

현재씬에 있는
05_GameManager을 지우고

Project 뷰의 Prefab폴더 안에 있는
06_IntroManager 프리팹을 드래그해
Hierachy뷰로 옮깁니다.

플레이를 해보면
타이틀이 나오게 될겁니다.

1.

Hierarchy

Create ▾ Q:All

- 01_Sky
- 02_Box_Maker
  - box_Loop_Coin
  - box_Loop_Coin
- 03_Player
- 04_Main_Camera
- 06_IntroManager

Inspector

02_Box_Maker Static

Tag: Untagged Layer: Default

Prefab: Select | Revert | Apply

Transform

Position

X 30 Y 0 Z -0.5

Rotation

X 0 Y 0 Z 0

Scale

X 1 Y 1 Z 1

Box_Loop (Script)

Script: Box_Loop

Box

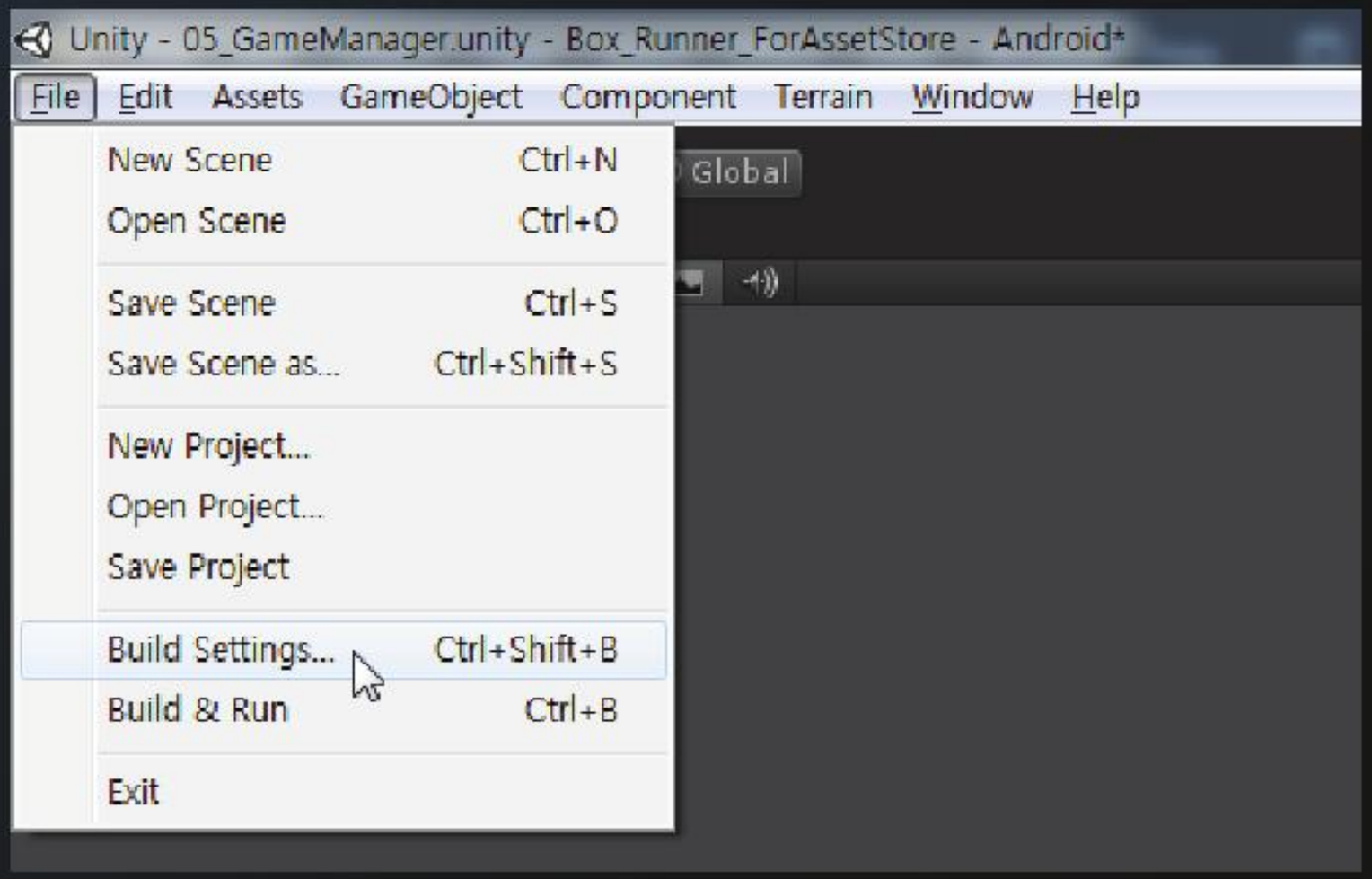
Size	5
Element 0	box_Loop_Coin
Element 1	box_Loop_Coin
Element 2	box_Loop_Coin
Element 3	box_Loop_Coin
Element 4	box_Loop_Coin
A_Zone	box_Loop_Coin
B_Zone	box_Loop_Coin
Speed	10

Add Component

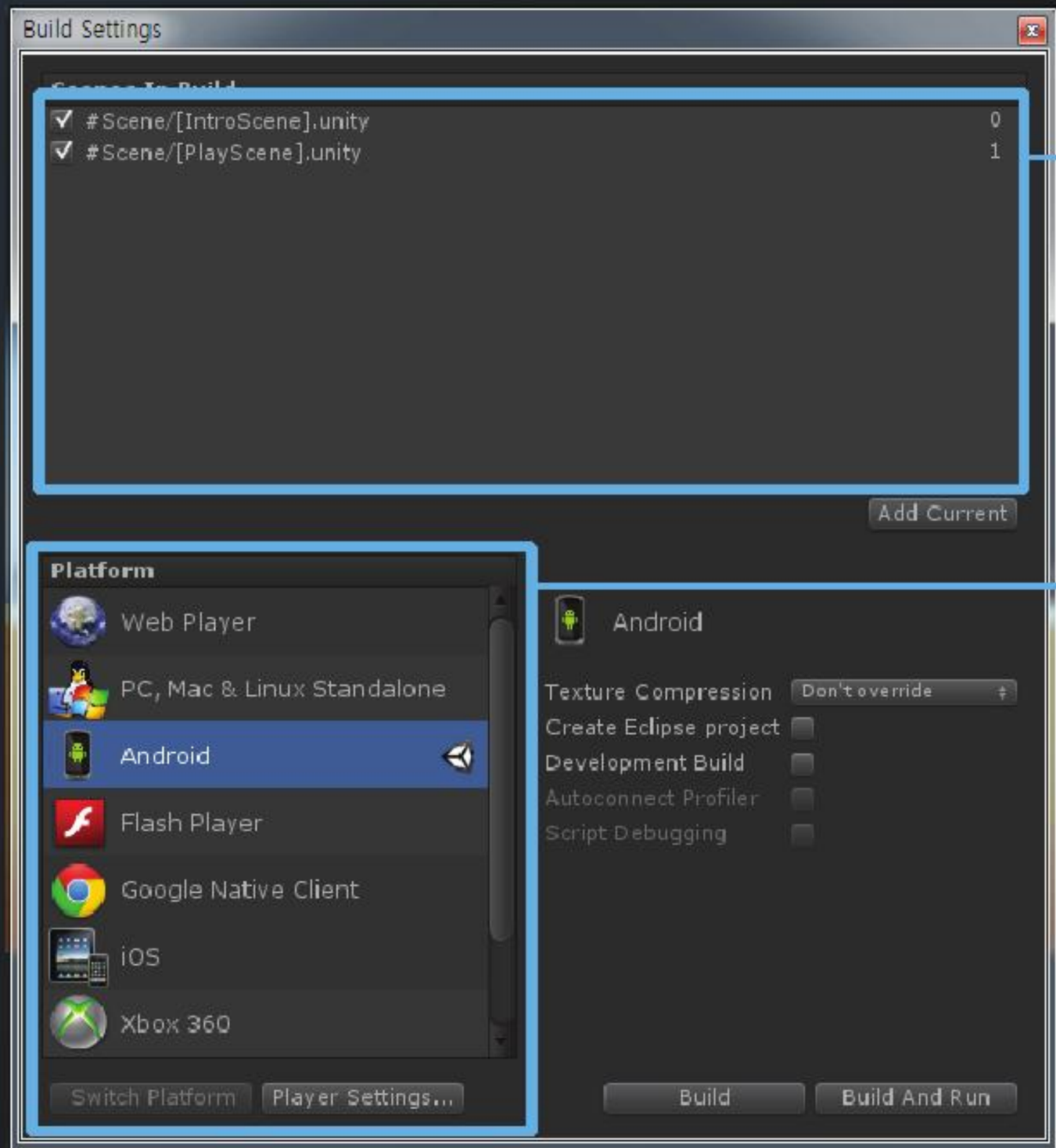
2.

Project 뷰에 있는 Prefab 폴더안의 BasicBox 폴더 안의 Box_Loop_Coin으로 모두 교체합니다.

타이틀씬에서는 주인공이 바닥에 빠지거나 벽에 부딪혀 죽으면 안되므로 반복되서 생성되는 타일을 통일해줍니다.



유니티에서는 최종적으로 사용할 씬을 셋팅해놓아야 빌드가 가능합니다.
상단메뉴의 File – BuildSetting으로 들어갑니다.



최종으로 빌드할 씬들.

프로젝트 뷰에 있는
[IntroScene]과 [PlayScene]을
드래그해서 올려놓습니다.

물론 IntroScene이 0번에 올라가게 올려둡니다.

최종으로 빌드할 플랫폼

다양한 플랫폼이 존재하고 있고
하단의 Switch 플랫폼을 통해
원하는 플랫폼으로 변경할 수 있습니다.