



UNITY -CHAPTER 6-

SOUL SEEK



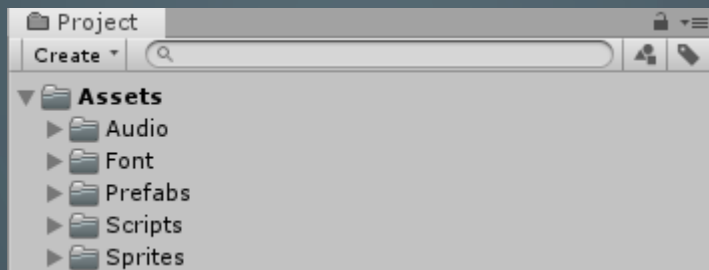
목차

1. 플레이어 제작

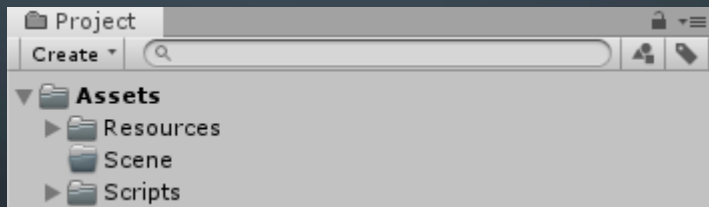
플레이어 제작

1. 준비하기

- **Chapter6_Uni-Run Project**를 열어서 **Asset**들을 확인하자.



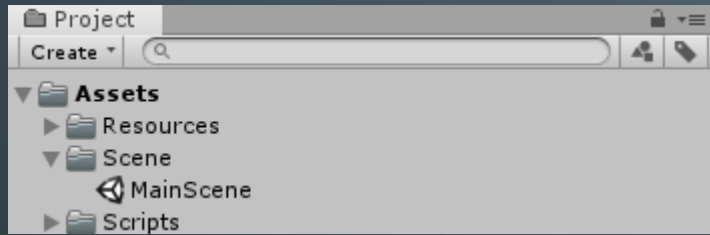
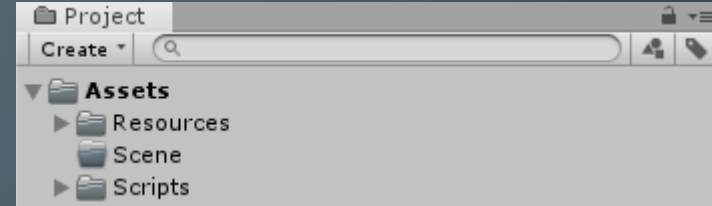
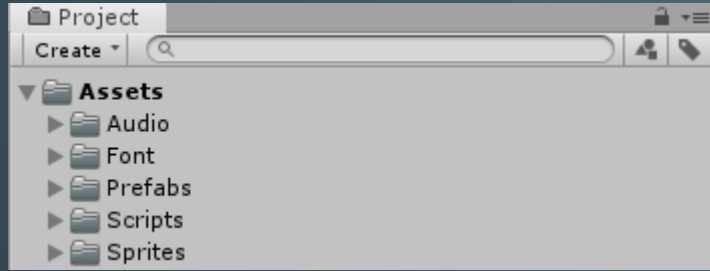
- **Resources Folder**를 이용해서 **Scripts**를 제외한 나머지 부분을 정리하자.
 - 게임 내부에서 사용하고 있는 **Resources**들을 코드 컨트롤에 사용하기 위한 목적
 - **Resources**에 포함된 **Material**, 모델링, **Animation**, **Texture**, **Sprite**와 같은 것들은 다른 **Component**에 추가하여 사용하는 경우가 많다. **Resources Folder**를 구성할 경우 **Build**할 때 **APK** 포함되는 경로이고 기본 경로로 활용하기 좋다.
 - **Solution > Project**의 경로가 **Application**의 기본경로가 되기 때문



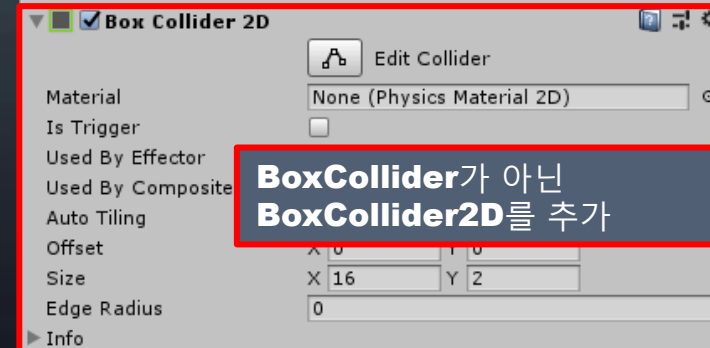
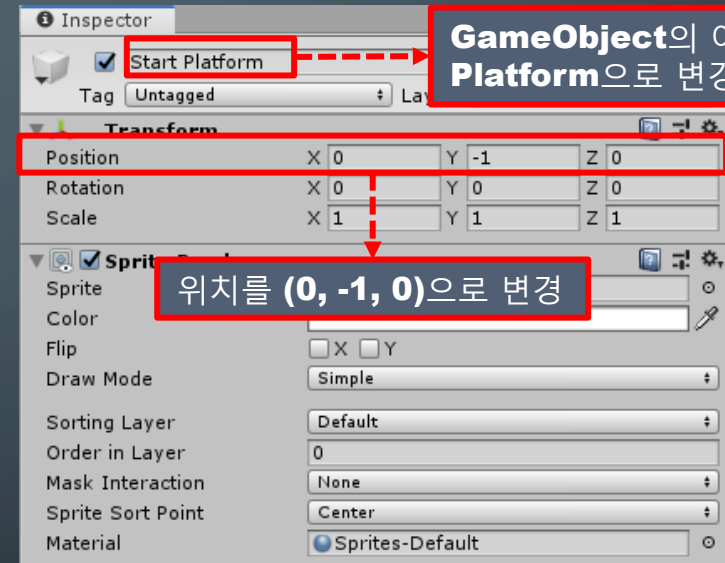
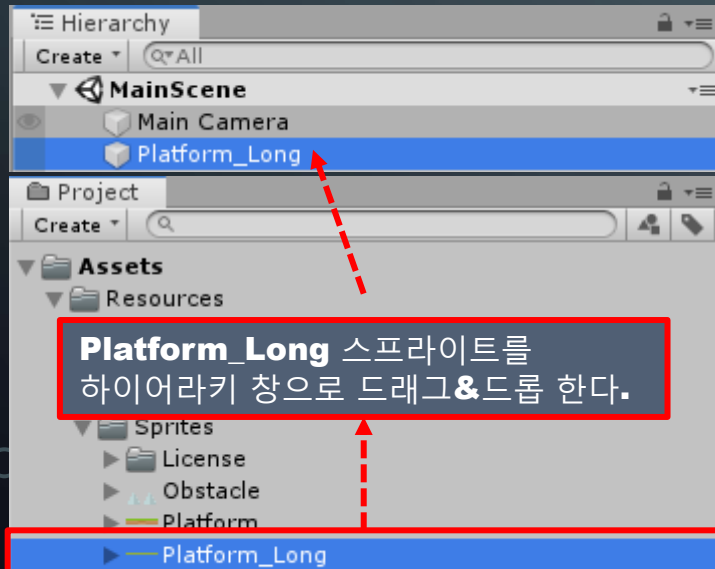
- **Scene**을 제작하고 관리할 **Scene**폴더를 추가해주자.

1. 시작지점 만들기

- **Main Scene**을 만들고 시작 지점을 만들 준비를 하자.



- 준비된 **Asset**을 가지고 시작점을 제작해 보자.

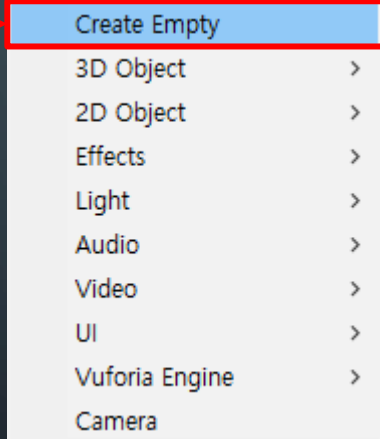


2. 데드존 만들기

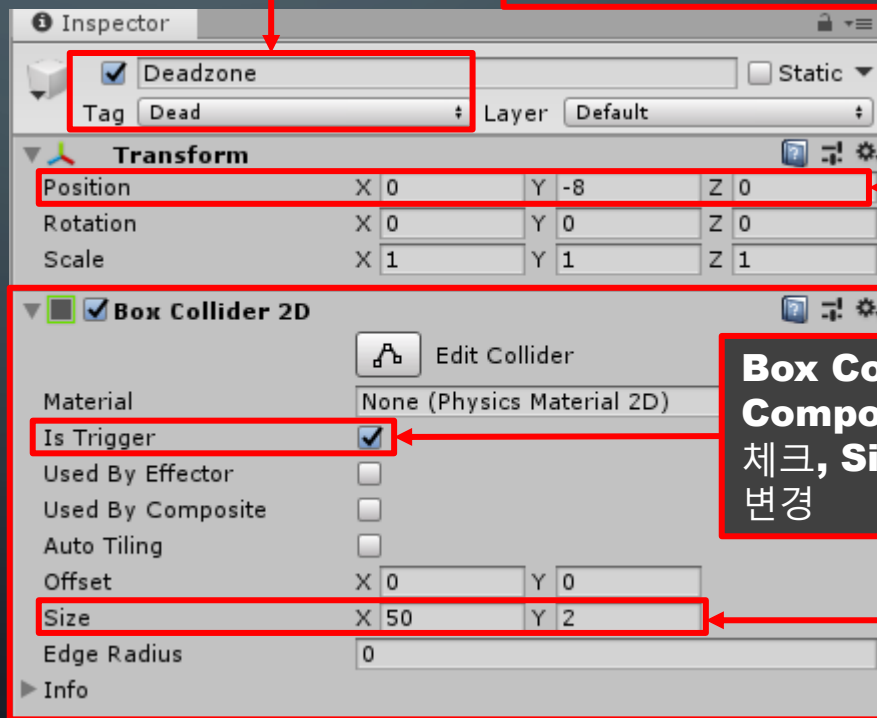
- 일정 이하의 위치로 떨어지면 죽는 구간을 만들자.
→ 오브젝트는 따로 필요 없고 물리적인 충돌보다 **Trigger**를 이용한 이벤트 체크만해서 처리하자.

- 빈 게임 오브젝트를 생성한다.(하이어라키 창에서 **Create > Create Empty** 클릭 or 우 클릭 > **Create Empty**)
- 생성된 게임 오브젝트의 이름은 **DeadZone**, 태그는 **Dead**, 위치는 **(0, -8, 0)**으로 변경
- Box Collider 2D Component** 추가
- Box Collider 2D Component is Trigger** 체크, **Size**는 **(50, 2)**로 변경

빈 게임 오브젝트 생성



이름은 **Deadzone**, 태그는 **Dead**, 위치는 **(0, -8, 0)**으로 변경



Box Collider 2D Component의 **is Trigger** 체크, **Size**는 **(50, 2)**로 변경

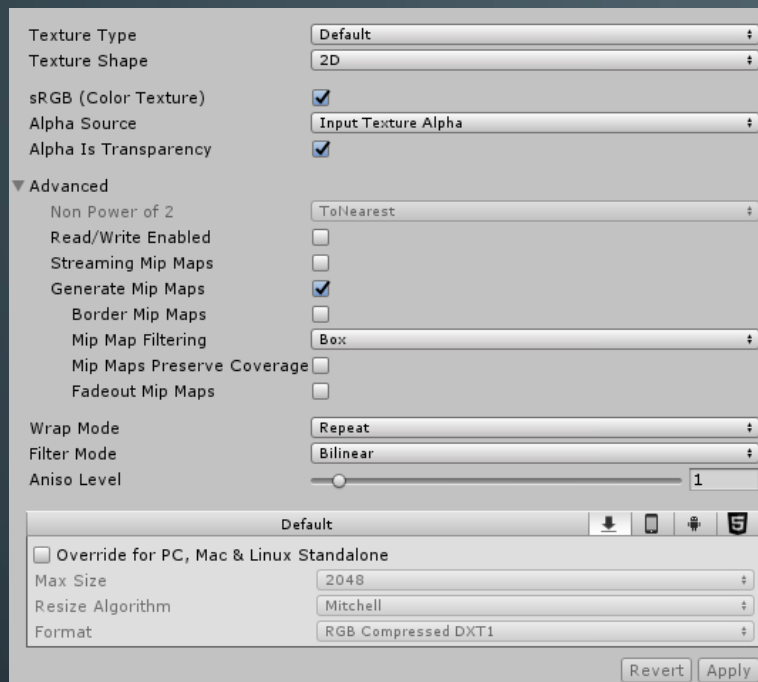
4. TEXTURE

- **GameObject**를 덮거나 래핑하는 이미지를 말한다.
- 쓰임에 따라 여러 **type**의 **Texture**를 설정 할 수 있으며 그 쓰임새의 설정이 우선시 된다.
- 최소한의 쓰임에 **Type**과 옵션만 기억하자(**Advanced**는 잘 안 쓰는 옵션들이 감춰져 있다.)
- **BMP, EXR, GIF, HDR, IFF, JPG, PICT, PNG, TGA, TIFF**
- **A**채널을 사용하지 않는 **Texture**는 포맷형식을 그에 맞게 설정하는 것이 가장 바람직하다.
- **2**의 제곱수 크기의 **Texture**를 사용한다.

Texture Type

- **Default** : 기본 설정이다. **Import** 그대로의 속성
- **Normal Map** : **Shader**에서 일반 적인 **Texture**에 더 디테일 감을 주기 위해 쓰이는 **Texture** 형식
- **GUI(Editor GUI and Legacy GUI)** : **HUD or GUI** 컨트롤에서 사용하기 위한 타입
- **Sprite(2D and UI)** : **2D**에서 사용하는 **Sprite**형식으로 사용하기 위한 타입(**UGUI**에선 필수 타입)
- **Cursor** : 커스텀 커서형식으로 사용 할 때 설정한다.
- **Cookie** : 광원 쿠키에 사용되는 기본 파라미터로 설정하기 위한 타입.
- **Lightmap** : **LightMap**으로 사용하는 경우 설정하는 타입
- **Single Channel** : 채널 하나만 사용할 경우(**R, G, B, A**중하나만..)

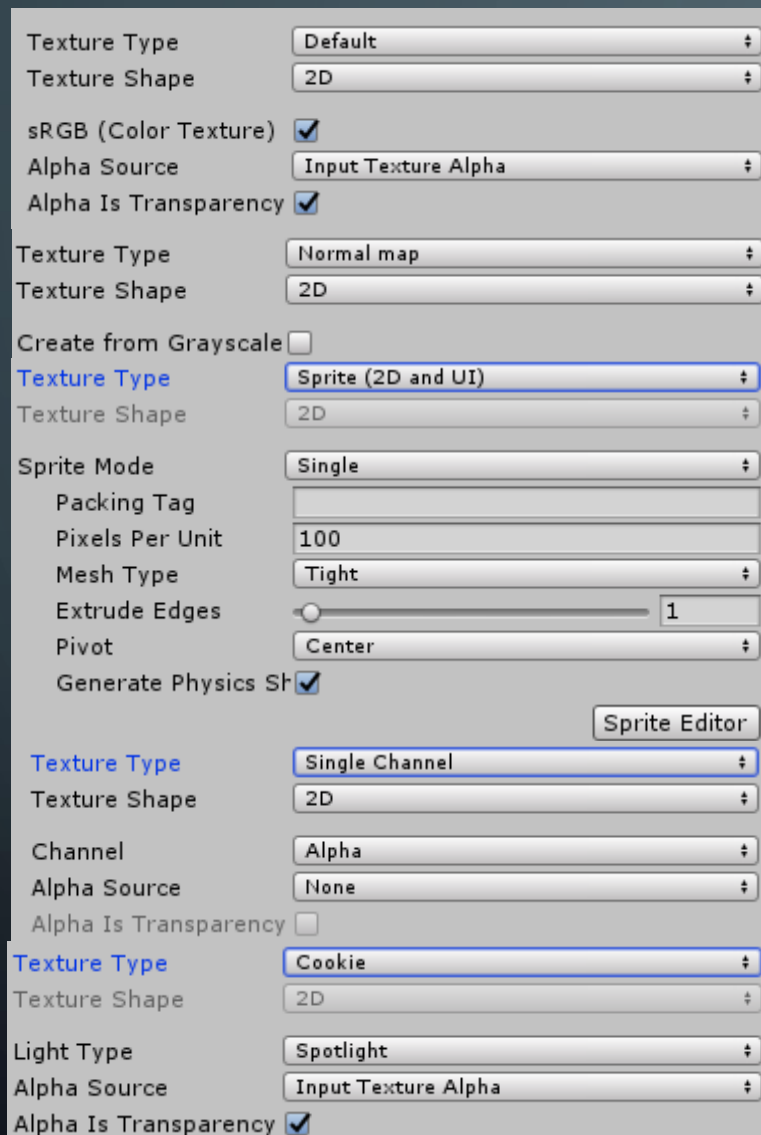
4. TEXTURE



공통 프리퍼티

- **Texture Type** : 텍스처 타입을 설정.
- **Texture Shape** : 텍스처 모양을 정의 한다. **2D, or Cube**
- **Non Power of 2 : NPOT**(2의 제곱크기가 아닌 경우)인 경우 확장해서 2의 제곱으로 확장해서 바꿔줄 경우 설정.
- **Read/Write Enabled : Texture2D.SetPixels, Texture2D.GetPixels** 및 기타 **Texture2D** 함수같은 스크립트 함수에 액세스 할 수 있게 설정.(메모리 값이 두배로 증가하므로 필요할때만 사용)
- **Generate Mip Map : MipMap**을 생성할 경우 체크 화면에 가깝고 멀고에 따른 텍스처 세밀도를 표현하기 위해 작은 이미지를 미리 만들어 두는 것, 기본적으로 체크 되어 있다.
- **Border Mip Maps : MipMap**의 심도를 제한한다(**Cookie**타입에 사용 그래서 기본적으로 논 클릭.)
- **MipMapFiltering : MipMap**에 사용되는 최적화 필터링. **Box** – 페이드 아웃 하는 가장 간단한 방법 레벨이 작을 수록 평탄해 진다. **Kaiser** – 크기가 작아질 때 **MipMap**에 대해 샤프닝 알고리즘이 실행. 먼거리에서 보면 너무 흐릿한 경우 이 옵션을 사용하는 것이 좋다.
- **Fade Out Map : Mip**레벨이 변할 때 마다 **MipMap**이 회색으로 페이드 된다. **Deatail map**일 경우에만 사용된다.
- **Warp Mode** : 바둑판 형식으로 배열해서 보여 줄때 동작 방식을 설정.
- **Filter Mode : 3D** 변환을 통한 텍스처 변환을 할 때 필터링 방식. **Point** – 가까이에서 보면 블럭 현상이 나타난다. **Bilinear** – 가까이에서 흐릿하게 보인다. **Trilinear** – 서로 다른 **MIP**레벨 간에 흐릿해 보인다.
- **Aniso Level** : 텍스처를 가파른 각도에서 볼 때의 품질을 설정한다. 바닥 및 지면 표현에 효과적이다.

4. TEXTURE



Default

- **sRGB(Color Texture)** : 텍스처가 감마 공간에 저장 되도록 설정, **HDR** 컬러 텍스처가 아닌 경우 항상 선택되어 있어야 한다. **Shader**의 특정값의 경우 대신 사용하기 때문에 체크 해제를 해주어야 한다.
- **Alpha Source : Alpha** 채널이 생성되는 방법을 지정하는데 사용.

Normal map

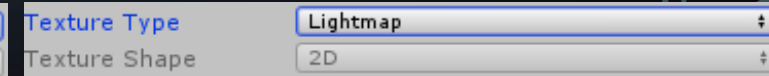
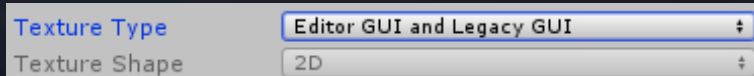
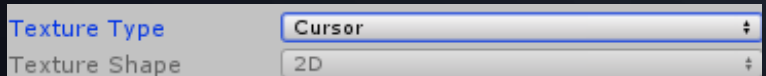
- **Create from Grayscale : Hight map** 기반의 **Normal Map**을 만들 때 체크한다.

Sprite(2D and UI)

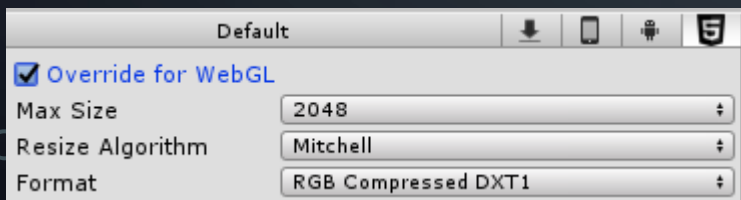
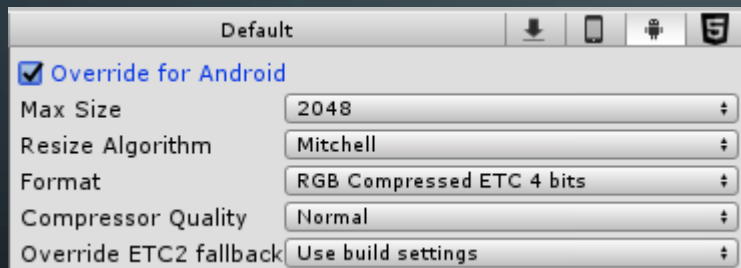
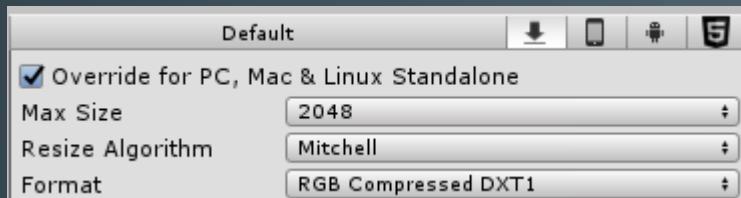
- **Sprite mode** : 이미지에서 추출되는 방법을 지정하는데 사용 **single** – 단일 이미지, **multiple** – 여러 스프라이트를 함께 모아 두는 용도로 사용할 때, 애니메이션 프레임 같은 걸로 이용 할 때.
- **Packing Tag : Sprite Atlas** 이름을 사용해서 지정한다.
- **Pixels Per Unit** : 월드 공간에서 하나의 거리 단위에 대응하는 스프라이트 이미지의 너비 / 높이 픽셀 수 이다.
- **Mesh Type** : 스프라이트를 표현할 때 생성하는 단면 메쉬를 설정한다. **FullRect** – 매핑할때 사각형을 이용한다. **32 * 32**는 무조건 **fullrect**로 지정된다. **Tight** – 픽셀 하나하나의 **Alpha**값 기반으로 메시 형태로 만든다.

SingleChannel

- **Channel – Red, Alpha**중 하나를 선택해서 사용.
- **Alpha Source – Red**일 경우 비활성화, **None – Alpha**값이 존재해도 사용하지 않음, **Input Texture Alpha** – 텍스처가 제공하는 경우 입력 텍스처의 알파를 사용, **From Gray Scale** – 입력 텍스처 **RGB** 값의 평균을 사용하여 알파 생성.



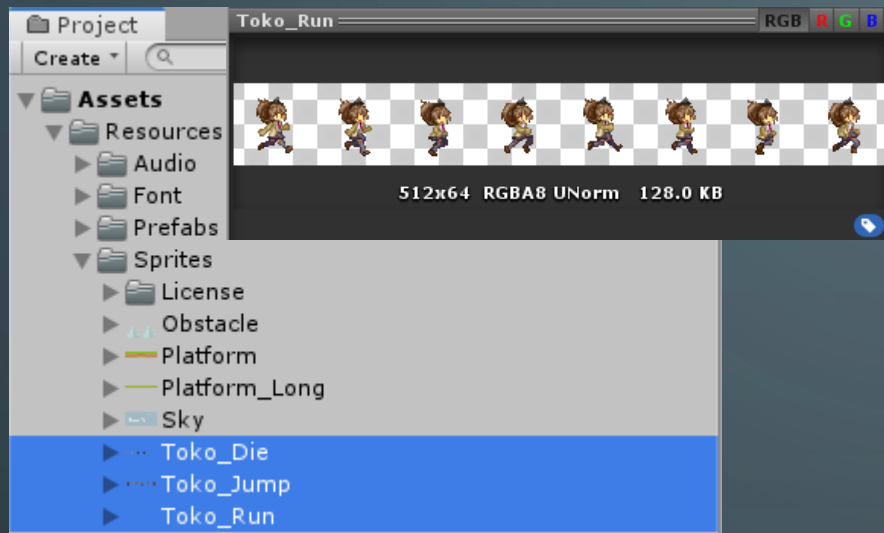
4. TEXTURE



- **Max Size** : 해당 텍스처의 최대 크기를 제어하는 옵션
- **Resize Algorithm** : **Resizing** 작업을 수행 할 알고리즘을 선택한다.
- **Format** : 압축 포맷형식을 지정합니다. 모바일 버전 빌드 시 용량이나 재원을 줄일 수 있는 방법이니 이 포맷을 **Documentation**을 통해 알아 두는 것이 좋다. (**Alpha** 채널이 없는 텍스처에 **RGBA** 압축설정을 하지 말자.)
- **Compressor Quality** : 압축 퀄리티를 말한다.
- **Override ETC2 fallback** : **ETC2** 실패 시 돌려놓을 상태를 말한다.

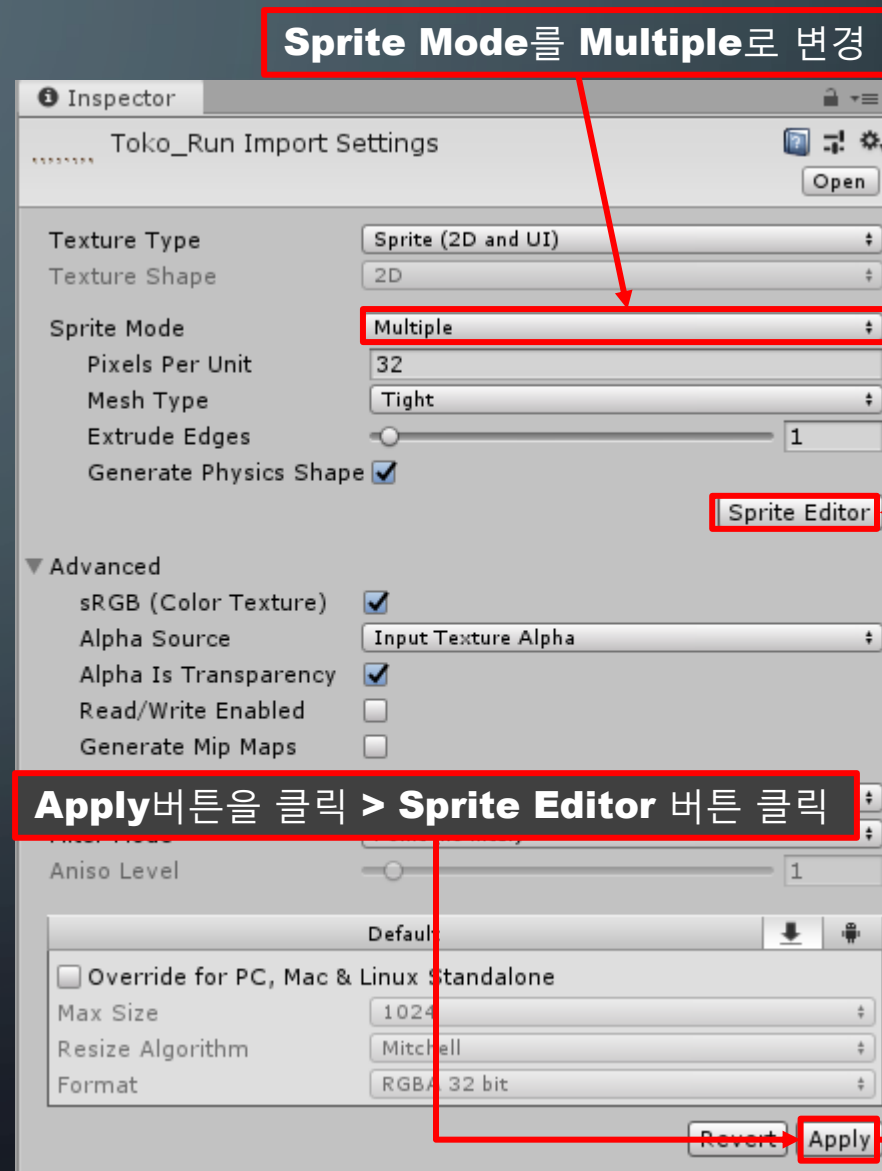
5. 캐릭터 스프라이트 편집

- 준비된 캐릭터 스프라이트를 살펴보자.



- Toko_Run**을 시트를 나눠서 사용할 수 있게 편집해 보자.

- 프로젝트 창의 **Sprites** 폴더에서 **Toko_Run Sprite** 선택
- 인스펙터 창에서 **Sprite Mode**를 **Multiple**로 변경
- Apply** 버튼을 클릭해서 변경사항을 적용한다.

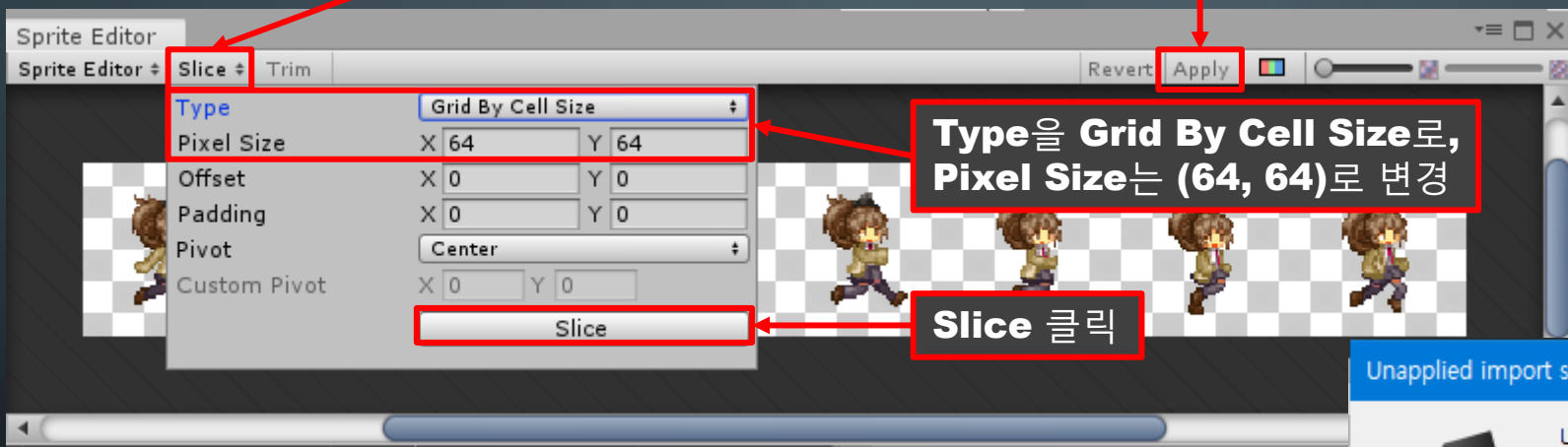


5. 캐릭터 스프라이트 편집

- **Sprite Mode**를 변경하고 저장한 다음에 **Sprite Editor** 버튼을 눌러 편집창을 활성화 하고 **Sprite**를 나누자.

1. **Sprite Editor** 창에서 **Slice** 버튼을 클릭
2. **Type**은 **Grid by Cell Size**로, **Pixel Size**는 (64, 64)로 변경
3. **Slice** 클릭
4. **Apply** 버튼을 클릭해 저장 > **Sprite Editor** 창 닫기

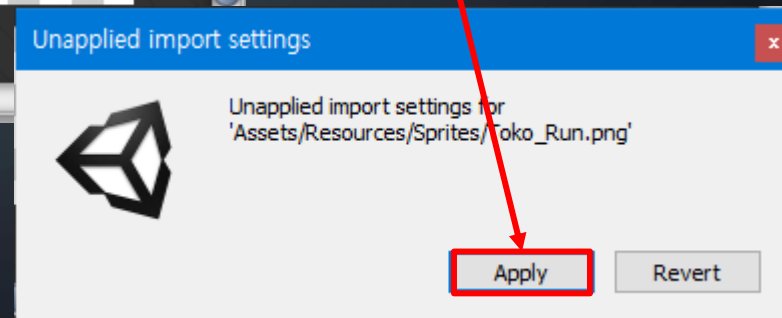
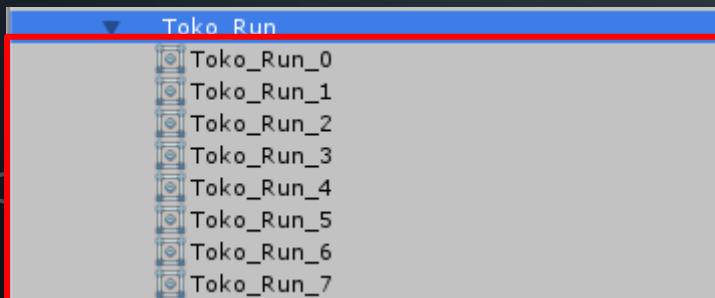
Sprite Editor 창에서 **Slice** 버튼 클릭



Sprite Editor 창에서 **Slice** 버튼 클릭

Type을 **Grid By Cell Size**로,
Pixel Size는 (64, 64)로 변경

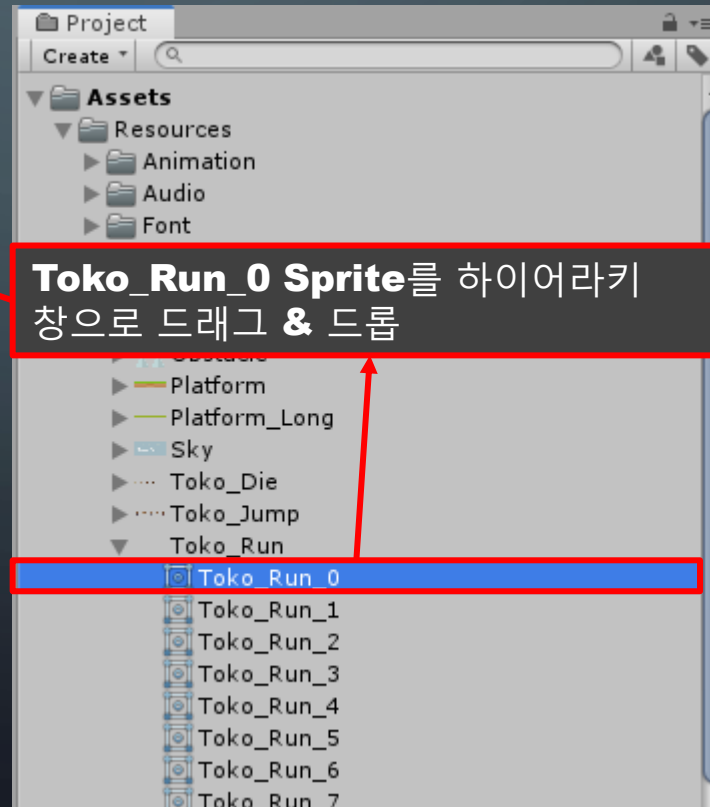
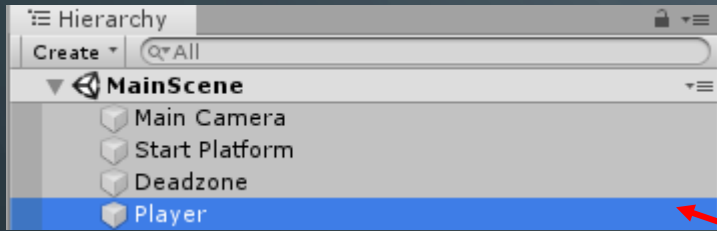
Slice 클릭



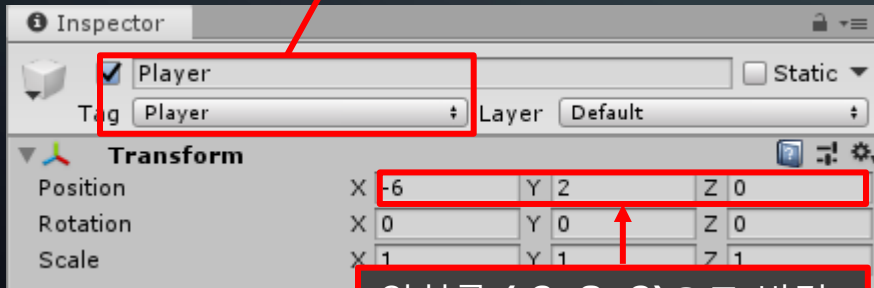
6. 캐릭터 GAMEOBJECT 만들기

- **2D** 캐릭터를 생성하자.
→ 물리 상호작용, 조작을 감지하고 점프, 사망, 애니메이션 재생과 제어, 효과음 재생 기능 추가

1. 프로젝트창에서 **Toko_Run Sprite** 펼치기 > **Toko_Run_0** 스프라이트를 하이어라키 창으로 드래그 & 드롭
2. 생성된 **Toko_Run_0 GameObject**의 이름은 **Player**, Tag는 **Player**로 변경
3. **Player GameObject**의 **Position**을 **(-6, 2, 0)**으로 변경



이름은 **Player**, 태그는 **Player**로 변경

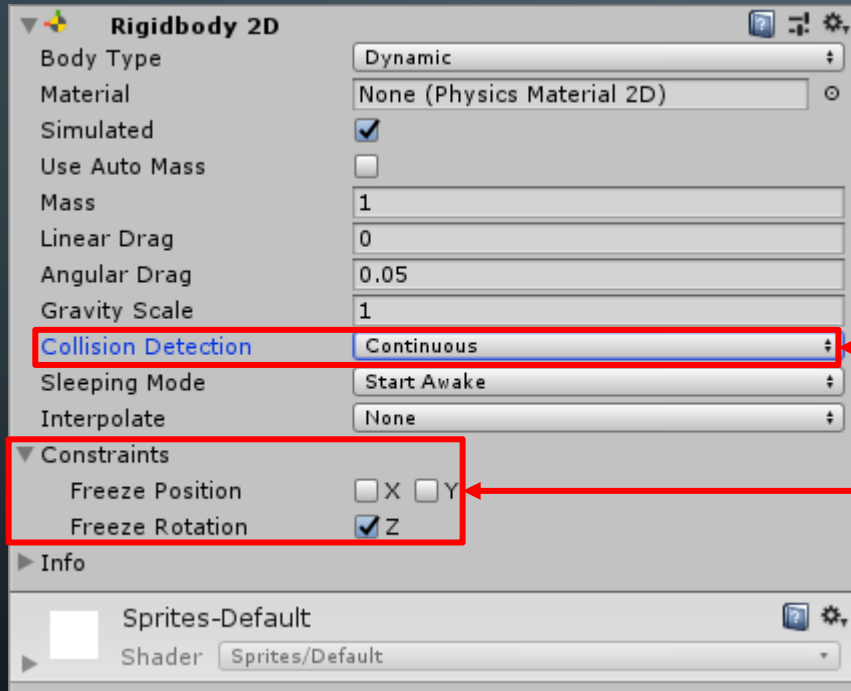


위치를 **(-6, 2, 0)**으로 변경

6. 캐릭터 GAMEOBJECT 만들기

- **Rigidbody2D Component** 추가

1. **Player Game Object**에 **Rigidbody2D Component** 추가
2. **Rigidbody2D Component**의 **Collision Detection**을 **Continuous**로 변경
3. **Rigidbody2D Components** 펼치기 > **Freeze Rotation**의 **Z** 체크



Player GameObject에 **Rigidbody2D Component** 추가

Collision Detection을 **Continuous**로 변경

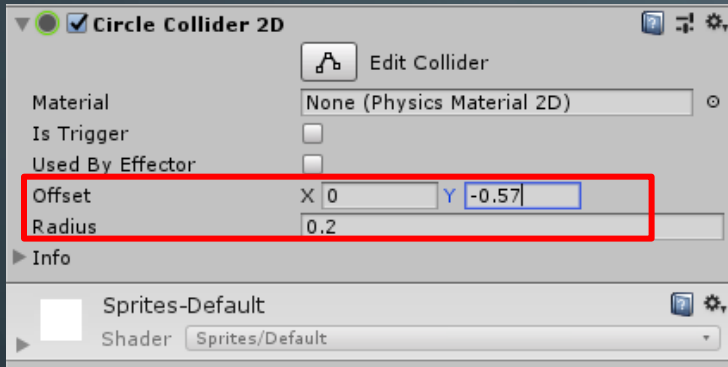
Constraints 펼치기 > **Freeze Rotation**의 **Z** 체크

6. 캐릭터 GAMEOBJECT 만들기

• Circle Collider2D 추가

1. Player GameObject에 Circle Collider 2D 추가

2. Circle Collider 2D Component의 Offset을 (0, -57), Radius를 0.2로 변경



Audio Source 추가

1. Player GameObject에 Audio Source Component 추가

2. Audio Source Component에서 AudioClip 필드 옆의 선택 버튼 >

창에서 jump 오디오 클립 더블 클릭

3. Audio Source Component의 Play On Awake 체크를 해제

