

chapter02

Unity 용어

1. **RigidBody**
2. **PreFab**
3. **Script**
4. **Unity Overview**

<https://github.com/jongukjeong/unity>

01 Rigidbody

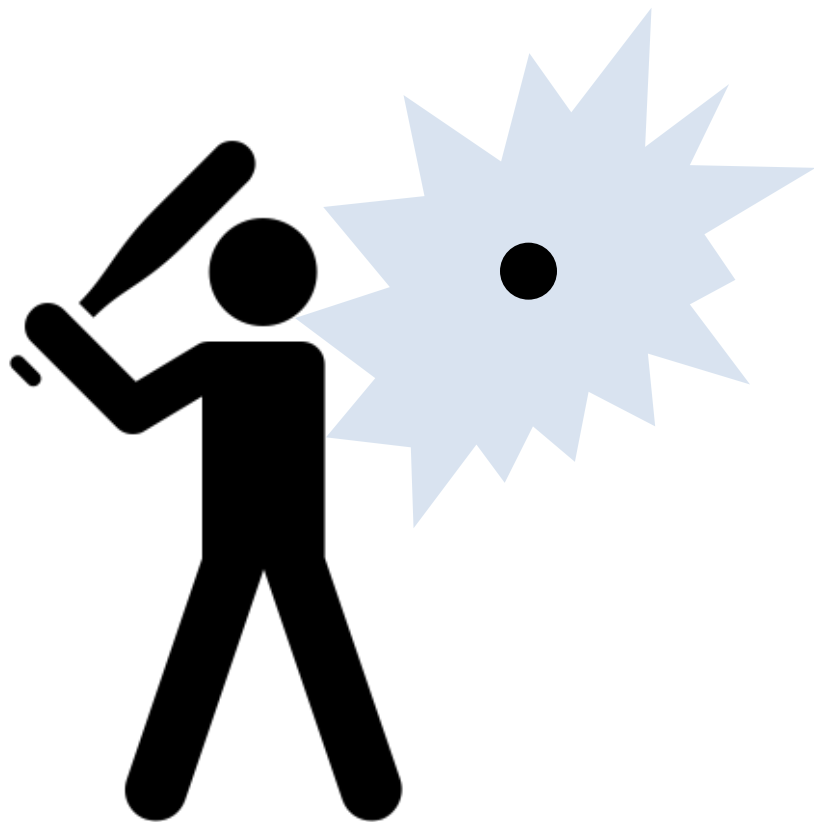
■ 아이템을 먹으려면



충돌을
이해해야한다

01 Rigidbody

■ 홈런을 치려면



두 물체가

Game Object

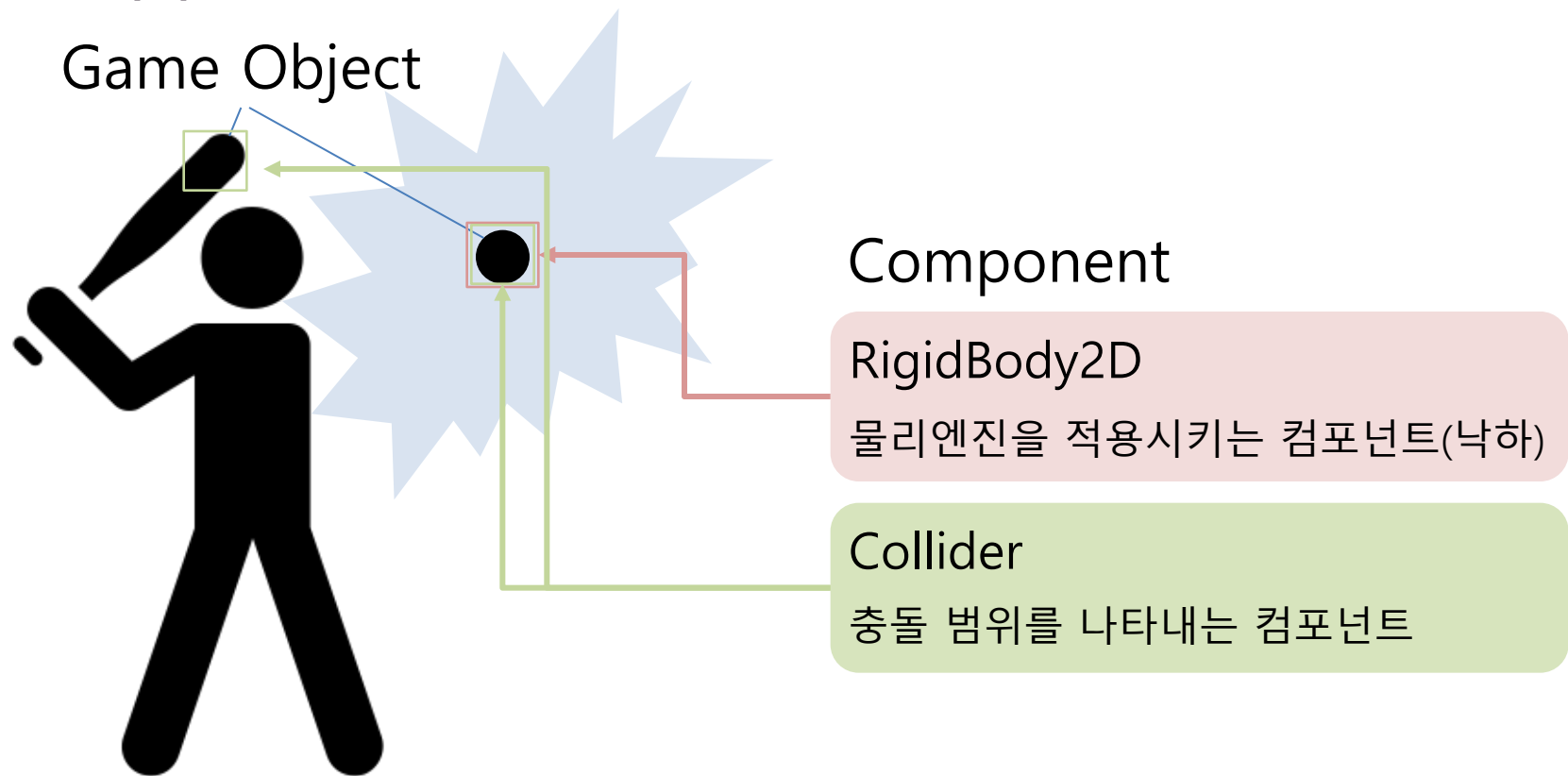
짧은 시간동안

부딪치는 것

Component

01 Rigidbody

■ 홈런을 치려면

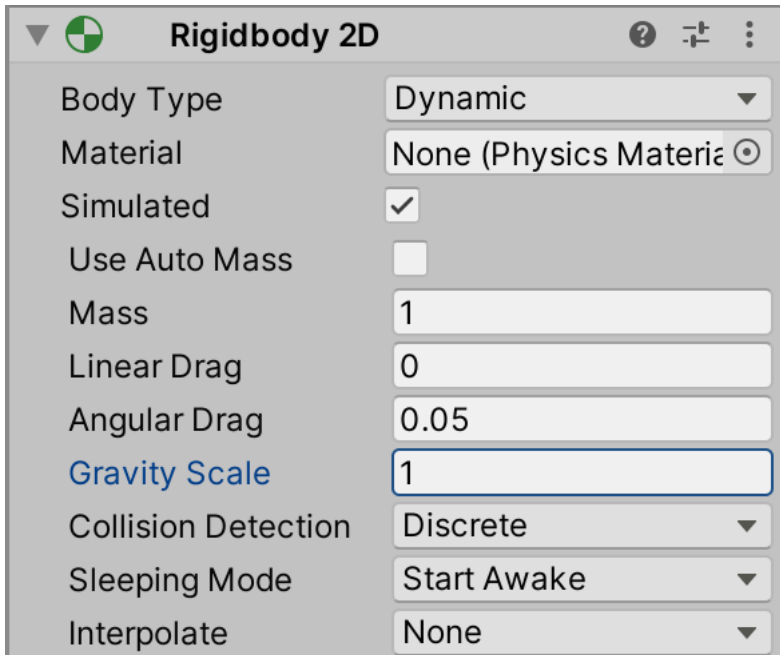


01 Rigidbody

■ 홈런을 치려면

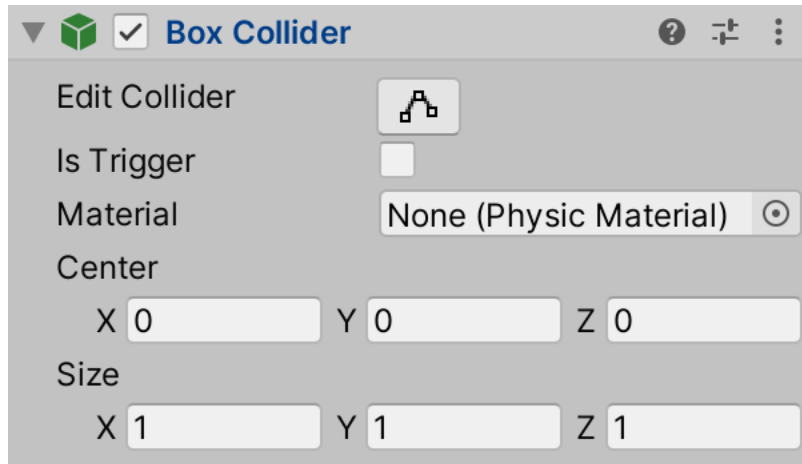
Rigidbody2D

물리엔진을 적용시키는 컴포넌트(낙하)



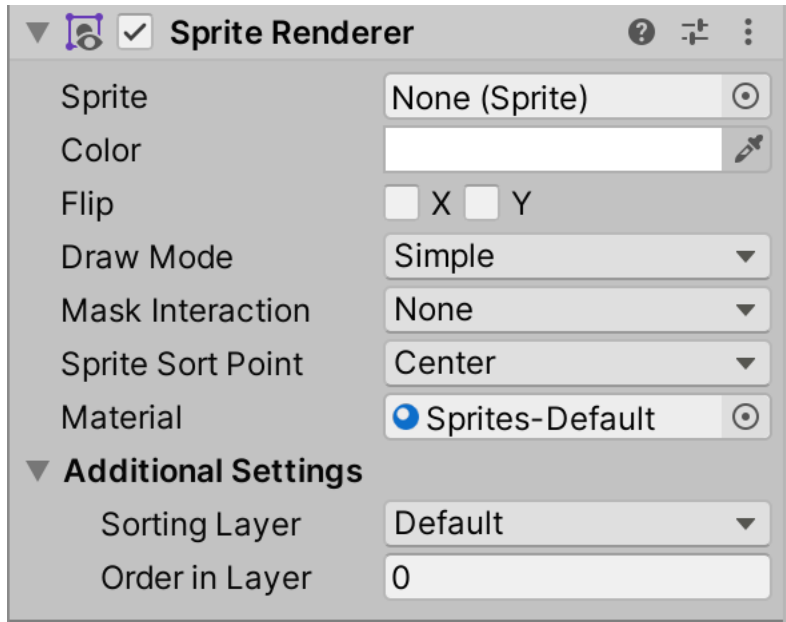
Collider

충돌 범위를 나타내는 컴포넌트



01 Rigidbody

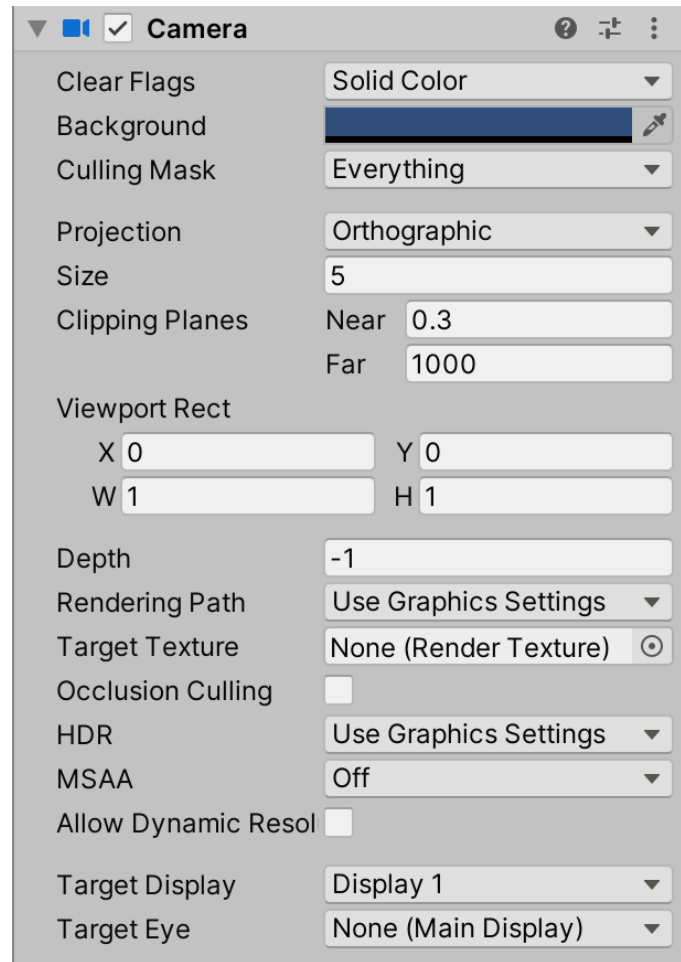
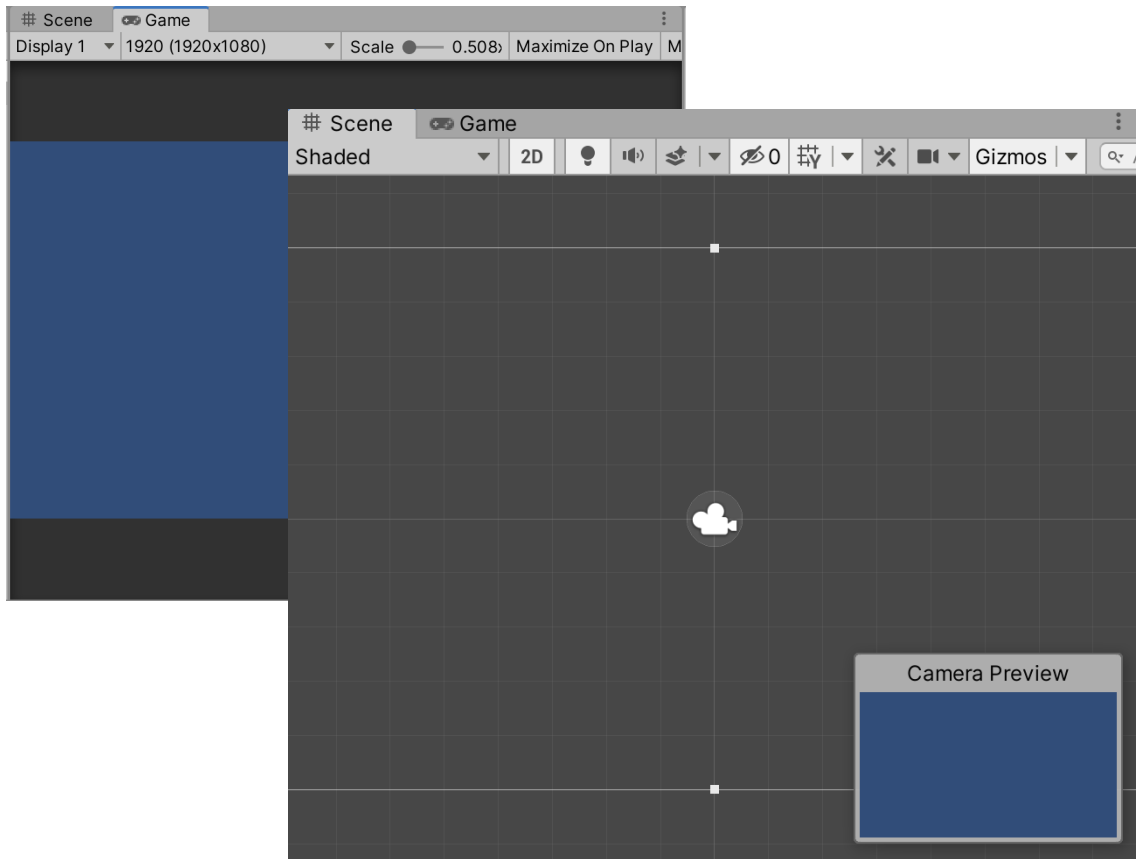
■ Sprite Renderer



이미지를 선택하고 색을 선택하거나
이미지 반전 여부등을 선택후 씬뷰나 게임뷰
에서 다른 오브젝트들과의 레이어 순서를 결
정 하는 컴포넌트

01 RidgidBody

■ Camera



02 Prefab

■ Prefab 이란



게임 오브젝트를 애셋(Asset)으로 만든 상태

게임 오브젝트를 파일로 만든 상태

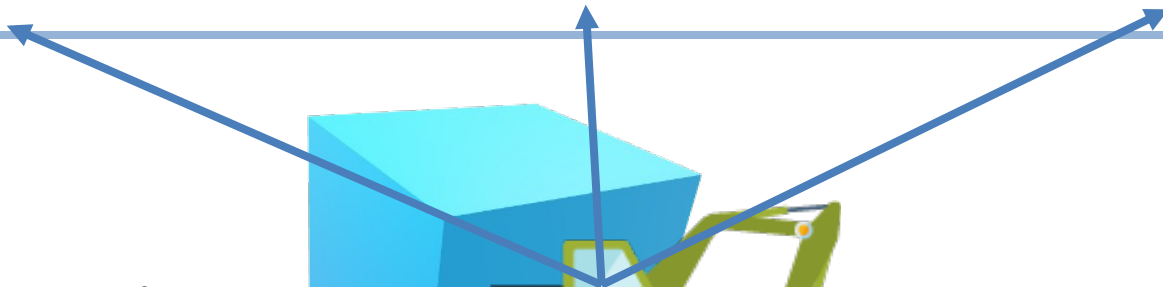
02 Prefab

■ Prefab 이란

Scene



Prefab



■ Unity 에서 Script 는



제어와 조작

항상 그 중심에
스크립트가
있습니다

03 Script

■ Unity 에서 Script 는



Unity 는 (게임)오브젝트를 제어하기 위한 컴포넌트를
준비 하지 않았... (그건 개발자가 직접...)

지금까지 보았던 컴포넌트들은 이미 누군
가 미리 준비 해둔 프로그램 이었던것 ...

유니티가 제공하지 않으니 나머지는 우리
가 직접 개발 해야 한다!!!

03 Script

■ Unity 에서 Script 는

내가 만든 캐릭터를 움직이게 하자



```
if(→가 눌러져 있다면){  
    해당 GameObject 오른쪽으로 이동 시킨다.  
    (물리엔진 적용)  
}
```

03 Script

■ Unity 에서 쓸수 있는 Script 는

- UnityScript (javascript like language)
- C#
- Boo

유니티 개발자

"유니티는 C++로 작성되었으나 다음과 같은 예외가 있습니다.

.NET API를 노출시켜서 여러분들이 C++로 개발할 필요 없게 만든 것 뿐입니다. 자바스크립트나 C# 또는 Boo를 이용해서 코딩을 할 수 있는 이유가 바로 .NET API 덕분입니다.

에디터 프로그램의 UI는 C#으로 작성했습니다. 유니티를 사용하시는 게임 개발자들에게 공개된 API와 거의 동일한 API를 쓰지요. (아직 공개되지 않는 API를 조금 쓰기도 합니다만 그리 많진 않습니다.)

이것이 바로 "UnityEngine.dll 파일을 다른 C# 프로젝트에서 사용할 수 있나요?"라는 질문에 대한 답이 언제나 "아니요"인 이유입니다. UnityEngine.dll에는 거의 아무 기능도 들어있지 않습니다.

이건 그저 C#/자바스크립트 호출들을 C++ 엔진에 연결해줄 뿐입니다.

C++ 엔진이 없이는 그냥 속빈강정이랄까요?"

- 유니티 내부는 **C++**이여서 C#보단 성능이 빠릅니다. 단, C#/C++ 컨텍스트 스위치 되는 부분에서 약간의 성능저하는 있을 수 있지만 언리얼엔진에 비해 크게 문제는 안됩니다
- 유니티 에디터는 **C#**으로 작성되어 있습니다. (에디터에서 리플렉션을 자동으로 지원해주는것도 이때문일듯...)

03 Script

■ Unity 에서 Script 적용 예

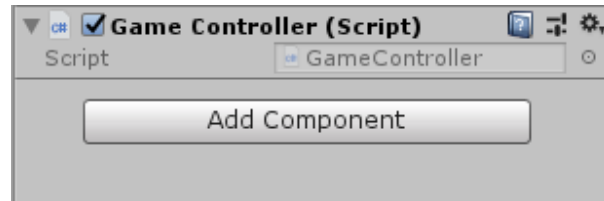
```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class GameController : MonoBehaviour {
6
7      // Use this for initialization
8      void Start () {
9
10     }
11
12     // Update is called once per frame
13     void Update () {
14
15     }
16 }
17

```

연결된 GameObject가 생성될때 실행되는 함수

매 프레임마다 실행되는 함수



유니티에서 클래스는 컴포넌트의 역할

04 Unity Overview

