

5주차 정리

개요

게임 오브젝트 간 충돌 처리와 물리 시뮬레이션 학습

리지드바디와 콜라이더 설정에 대한 이해

간단한 물리 시뮬레이션 구현

개요

- ▶ 게임 오브젝트 간 충돌 처리와 물리 시뮬레이션 학습
- ▶ 리지드바디와 콜라이더 설정에 대한 이해
- ▶ 간단한 물리 시뮬레이션 구현

▼ 게임 오브젝트 간 충돌 처리와 물리 시뮬레이션 학습

#1 충돌을 감지할 오브젝트 설정

- 충돌 함수를 처리하기 위해서는 충돌하는 두 물체 중 한가지는 리지드바디로 설정되어 있어야 한다.
- 설정 방법은 유니티 환경에서 오브젝트를 클릭하여 inspector 창을 열고 add component에서 rigid를 검색하면 나온다.
- 이때 3D 오브젝트는 Rigidbody로, 2D 오브젝트는 Rigidbody 2D 로 설정해야한다.
- 두 오브젝트 중 움직이는 것에 리지드바디를 설정하는것이 좋다고 한다.
- 주의할 점은 바닥을 놓지 않고 실행하면 큐브가 수직낙하한다. 이때 공중에 가만히 멈춰두고싶다면 is kinematic을 체크하여 중력과 충돌에 반응하지 않도록 한다.
- 충돌에 반응하면서 해당 위치를 벗어나지 않는 것을 원한다면 Constraint의 freeze Position을 설정하면 된다.

#2 충돌 이벤트 처리 함수

- 충돌 함수는 크게 두 종류가 있는데, 각각 이러하다.

1. **Collision** : 오브젝트간 물리적 연산을 하며 충돌감지

2. **Trigger** : 오브젝트간 물리적 연산을 하지 않고 충돌만 감지

- 쉽게 말하자면 collision은 충돌이 일어나면 튕겨나가고 Trigger는 통과해서 지나친다.

다음으로 충돌함수의 세 유형을 정리해보자.

1) 충돌이 시작될 때 호출

```
void OnCollisionEnter(Collision col)
```

2) 충돌이 지속되고 있을 때 매 프레임마다 호출

```
void OnCollisionStay(Collision col)
```

3) 충돌이 끝날 때, 즉 물체가 서로 떨어질때 호출

```
void OnCollisionExit(Collision col)
```

이번에는 Collision을 다룰 예정이므로 Collision으로 설명하였지만, Trigger도 함수명만 다를 뿐 유형 자체는 똑같다.

▼ 리지드바디와 콜라이더 설정에 대한 이해

#Rigid body(리지드바디 : 강체)

- 리지드바디 컴포넌트는 유니티의 물리 엔진과 상호 작용할 수 있습니다.
- 리지드바디 컴포넌트로 중력과 같은 힘을 적용할 게임 오브젝트를 구별하고 적용합니다.
- 예를 들어 자동차 오브젝트에 가속 버튼을 누르게 되면 자동차가 가는 방향에 속력이 더 빨라지는 것을 볼 수 있습니다.
- 이러한 물리적 힘을 적용할 수 있게 해주는 것이 리지드바디 컴포넌트입니다.
- **PlayerObject > Add Component > Rigidbody 2D** 컴포넌트를 추가하게 되면 PlayerObject에 Rigidbody 2D 컴포넌트가 추가가 된것을 볼 수 있습니다.
- Box Collider 2D 컴포넌트와 마찬가지로 Rigidbody 2D 컴포넌트의 이름 좌측에 화살표를 눌러서 Rigidbody 2D 컴포넌트를 펼칩니다.

1. **Body Type > Dynamic(동적)**

2. *Mass(질량)* > 1

3. *Linear Drag(선 항력)* > 0

4. *Angular Drag(각 항력)* > 0

5. *Gravity Scale(중력 크기)* > 0

#Collider(콜라이더 : 충돌체)

- 콜라이더는 게임오브젝트에 추가하는 컴포넌트입니다.
- 유니티에서는 콜라이더를 통하여 오브젝트가 충돌이 일어났는지 확인합니다.
- 오브젝트를 어림한 모양을 만드는 "프리미티브 콜라이더" 를 사용을 하고 "프리미티브 콜라이더"는 프로세서의 부담을 덜어주는 장점을 가지고 있습니다.

1. Box Collider 2D 컴포넌트의 이름 왼쪽에 보시면 아래 화살표 모양을 눌러 Box Collider 2D를 펼칩니다.

2. Box Collider 2D > Edit Collider > 도구 모양을 클릭하면 Scene뷰에 GameObject에 초록색테두리가 보이게됩니다.

3. 초록색 테두리를 Object의 모양대로 확대/축소합니다.

4. 다시 Edit Collider > 도구모양을 클릭하여서 적용합니다.

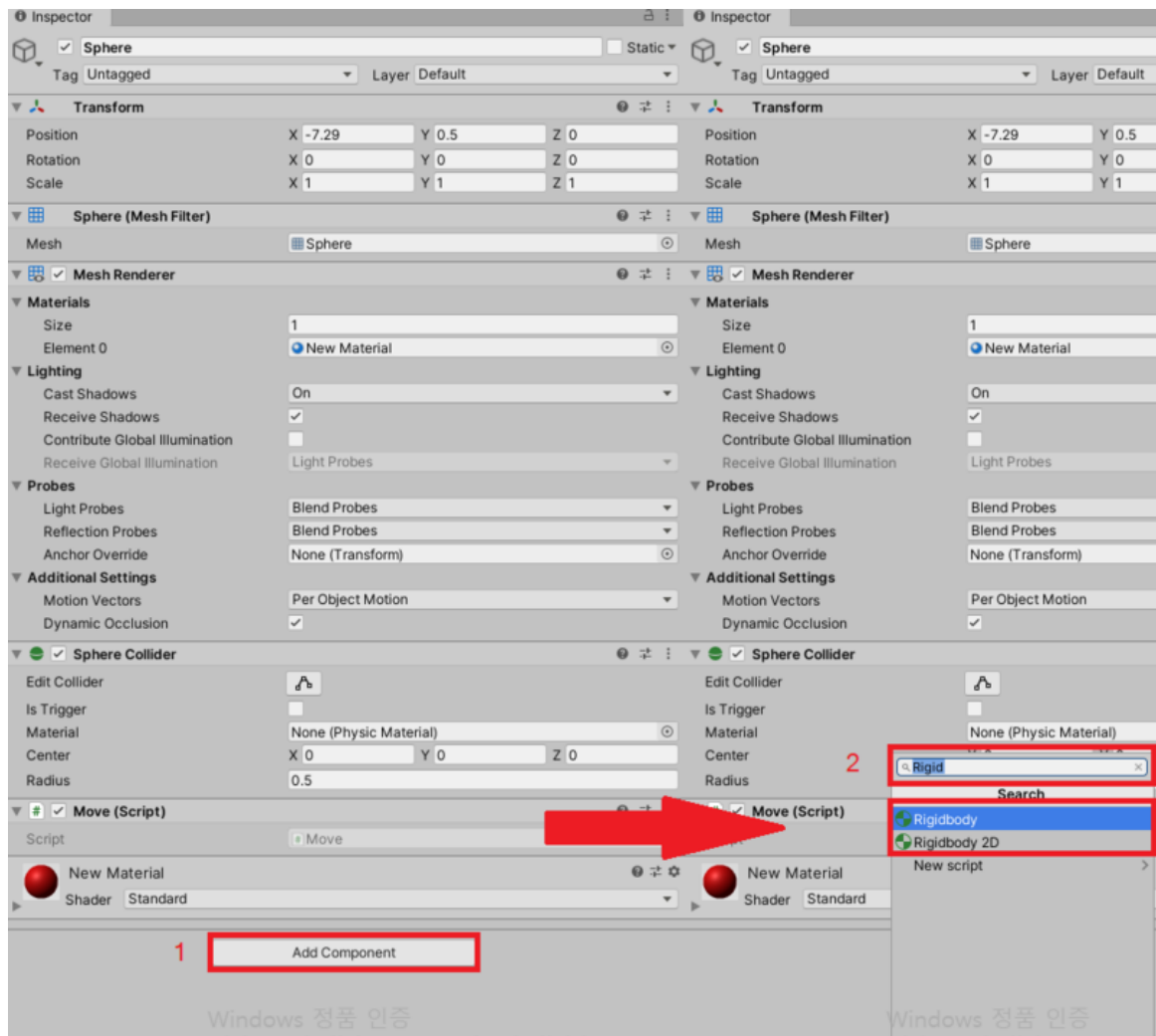
▼ 간단한 물리 시뮬레이션 구현

게임에서도 중력, 탄성 같은 물리 현상을 구현할 수 있다

그리고 그러려면 Rigid body에 대해 알아야 한다

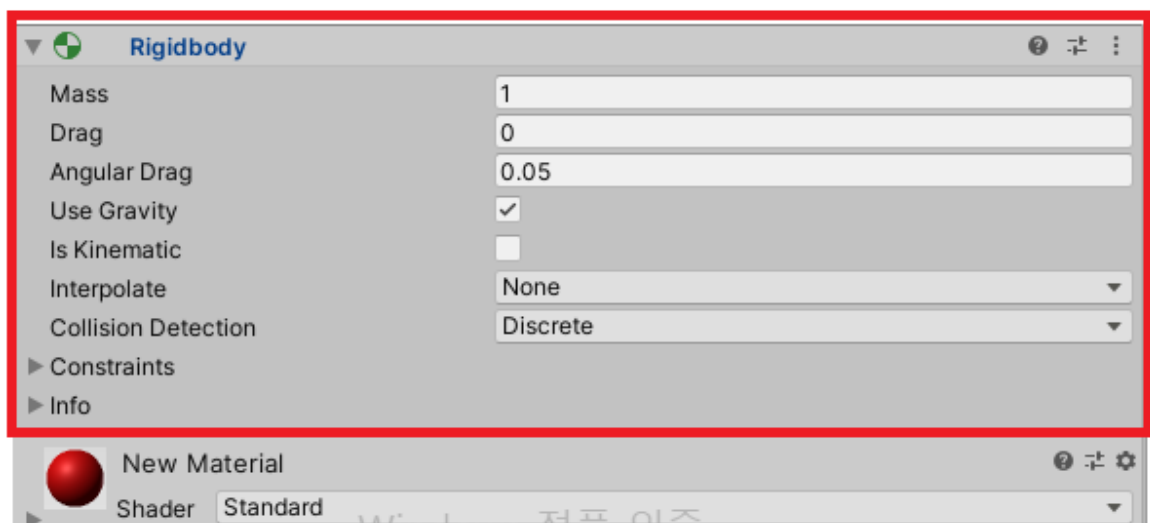
Rigid body는 강체라는 뜻이 있는데

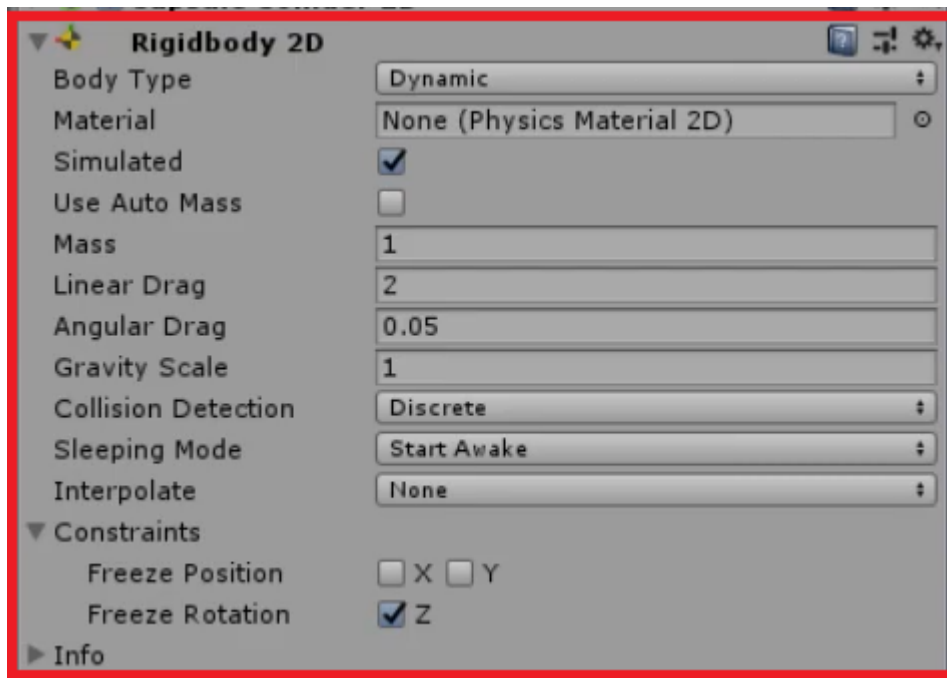
유니티에 물리 효과를 받으려면 애를 추가해줘야 한다



Inspector 창 아래에 있는 Add Component를 클릭한 뒤, 요소를 넣어주는데
3D 이면 Rigidbody로, 2D 이면 Rigidbody 2D로 넣어줘야 한다

3





그러면 이렇게 Inspector 창에 Rigidbody가 추가된다
 왼쪽이 3D 일 때, 오른쪽이 2D 일 때 추가되는 사항이다
 3D와 2D의 기능이 거의 똑같으면서 약간 다르다

Rigid body에 어떤 값들이 있는지 보면

Mass : 물체의 질량

(Linear) Drag : 물체가 받는 공기 저항

Angular Drag : 물체가 회전할 때 받는 공기 저항

Gravity : 중력

Is Kinematic: 외부 물리 효과 무시