

UNITY LECTURE 04

Rigidbody 기능들 알아보기

RIGIDBODY 란?

Rigidbody는 물리엔진을 적용시켜주는 컴포넌트입니다. 2D의 경우 Rigidbody 2D가 존재하고, 이 컴포넌트를 넣어주는 것만으로도 중력이나 작용&반작용 등 모든 물리효과가 적용됩니다.

주로 물리적인 자연스러운 움직임을 구현하기 위해서 사용되고, 유체역학이나 로프와 같은 복잡한 물리 연산은 지원하지 않습니다. 또한, Collider 편에서 말씀드렸다시피 충돌 확인을 하기 위해선 충돌 하는 쪽이나 받는 쪽 둘 중 하나에 Rigidbody가 있어야합니다.

속성들 - 1

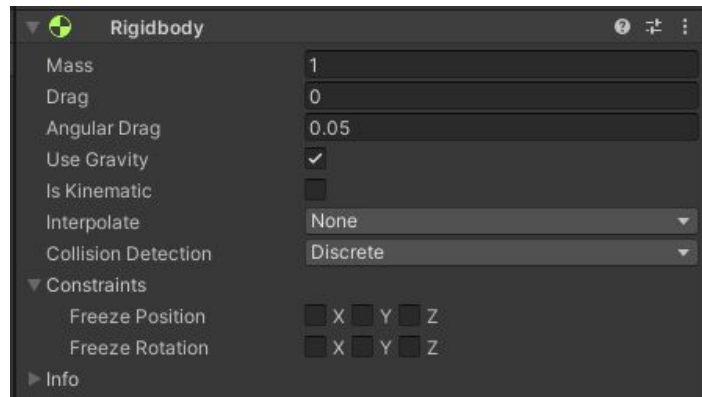
Mass: 무게이고, 기본으로 kg단위를 사용합니다

Drag: 힘으로 움직일 때 공기 저항의 정도입니다.
0이면 공기저항이 없고 무한대면 즉시 정지합니다.

Angular Drag: 오브젝트가 회전할 때 공기 저항입니다.
0이면 공기저항이 없지만 무한대라고 정지하지 않습니다.

Use Gravity: 중력 사용 여부입니다.

Is Kinematic: 물리 효과를 주기는 하지만, 받지는 않습니다. 차가
들이받아도 차만 부서지고 애는 멀쩡합니다.



속성들 - 2

Interpolate: 보간 옵션입니다. 움직임이 부드럽지 않을 때 사용됩니다.

- └ Interpolate: 이전 프레임의 트랜스폼에 맞게 움직임을 부드럽게 처리합니다.

- └ Extrapolate: 다음 프레임의 트랜스폼을 추정해 움직임을 부드럽게 처리합니다.

Collision Detection: 빠르게 움직이는 오브젝트가 충돌 확인보다 빠르게 뚫고 지나가는걸 막을 때 사용됩니다.

- └ Discrete: 불연속 충돌 검사입니다.

- └ Continuous: 연속 충돌 검사입니다. Rigidbody가 없는 정적 Collider나 Continuous Dynamic이 없는 오브젝트에는 불연속 충돌 검사를 합니다.

- └ Continuous Speculative: 추측성 연속 검사로, 위보다 성능이 괜찮습니다.

Constraints: 체크된 X, Y, Z의 물리 효과에 의한 이동/회전을 막습니다.

유용한 메서드

- `AddForce(Vector3, ForceMode)`

주어진 `Vector3`의 x, y, z 방향으로 힘을 가합니다. `ForceMode`에 대해서는 다음 장에서 알아보시다.

- `AddTorque(Vector3, ForceMode)`

`Torque`(회전힘)을 x, y, z 축으로 가합니다. X축 기준으로 회전한다는 것은 X축을 감싸며 돌도록 회전한다는 뜻입니다. 기준점에서 X축 방향으로 긴 꼬챙이를 꽂아 돌린다고 할 수도 있겠죠.

- `AddExplosionForce(float, Vector3, float, float = 0.0f, ForceMode)`

폭발 힘 / 위치 / 범위 / 위쪽 추가속도를 입력받고, 폭발의 위치에서 범위 내에 있는 `Rigidbody`들에게 거리에 따라 힘을 가합니다. 또한, 위쪽 추가속도가 있다면 날아가게 된 오브젝트들이 더 멀리 날아가게 됩니다.

FORCEMODE 클래스

ForceMode는 힘이 가해지는 방법을 바꿀 수 있는 열거형입니다.

- ForceMode.Force : Mass(질량)을 사용해서 연속적인 힘을 가합니다. 기본값입니다.
- ForceMode.Acceleration : Mass(질량)을 무시하고, 연속적인 가속력을 가합니다.
- ForceMode.Impulse : Mass(질량)을 사용해서 순간적인 힘을 가합니다. 예시로 점프가 있습니다.
- ForceMode.VelocityChange : Mass(질량)을 무시하고, 즉시 속력을 바꿉니다. 대시를 생각할 수도 있습니다.

유용한 속성들

스크립트에서만 다룰 수 있는 속성들 또한 존재합니다. 많이쓰이는 3가지만 알아보도록 하죠.

- Vector3 velocity : 현재 Rigidbody의 x,y,z 속도를 지니고 있습니다. 정확히는 위치의 변화율을 지니고 있습니다.
- Vector3 angularVelocity : 현재 Rigidbody의 x,y,z 회전속도를 지니고 있습니다. 라디안/s 를 가지고 있으며, 이 값을 직접 수정하는건 권장되지 않습니다.
- bool isKinematic : true면 Kinematic, false면 아닙니다.

알아둬야 하는 점

물리적인 계산은 모두 FixedUpdate 에서 일어납니다. 이는 Update 에서드와 비슷하지만, 정해진 시간에 예외없이 무조건 호출된다는 점이 다릅니다. 그래서 Rigidbody를 이용한, 또는 직접 제작한 물리 이동이 있다면 이는 FixedUpdate() 에서 해주는 것이 좋습니다.

마치며

궁금한 점은 꼭 질문해주세요!

카카오톡 | 디스코드

sigening | Sigening#6088