

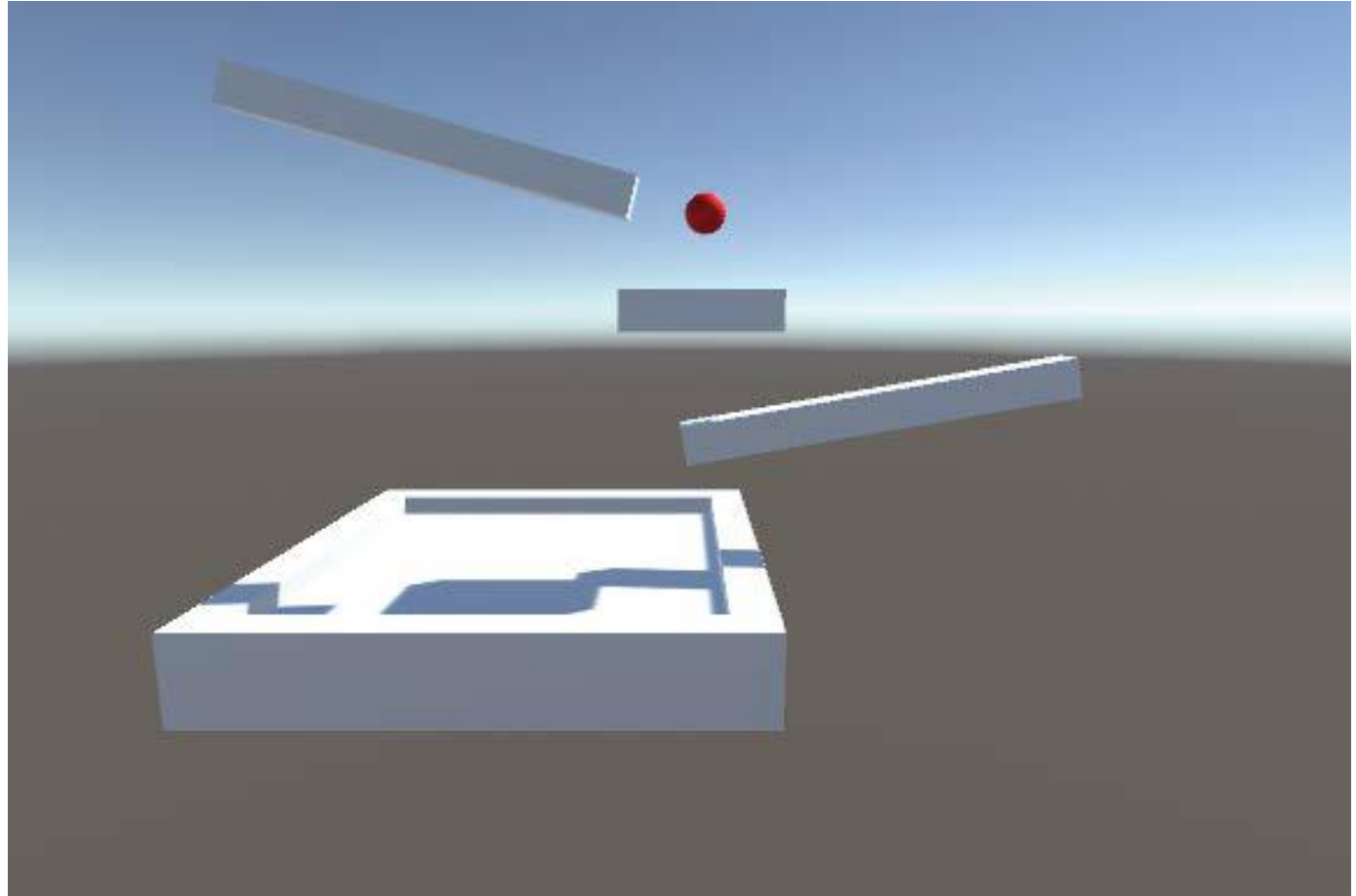


유니티 사용하기

유니티의 기본 조작을 살펴보면서 간단한 '공 굴리기 게임'을 만들어 보기

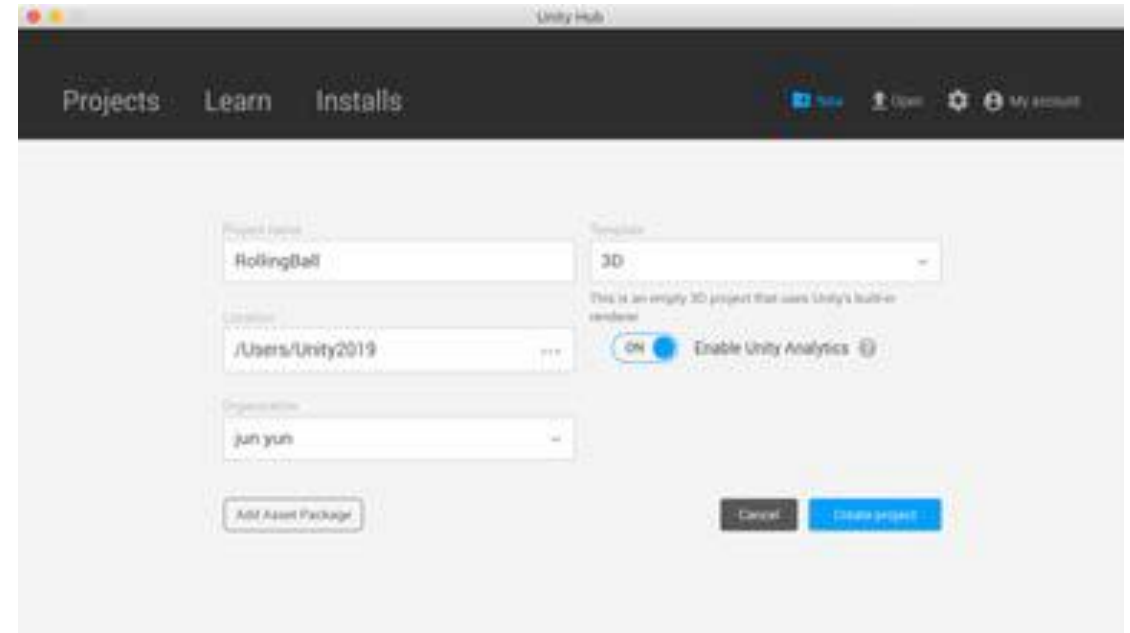
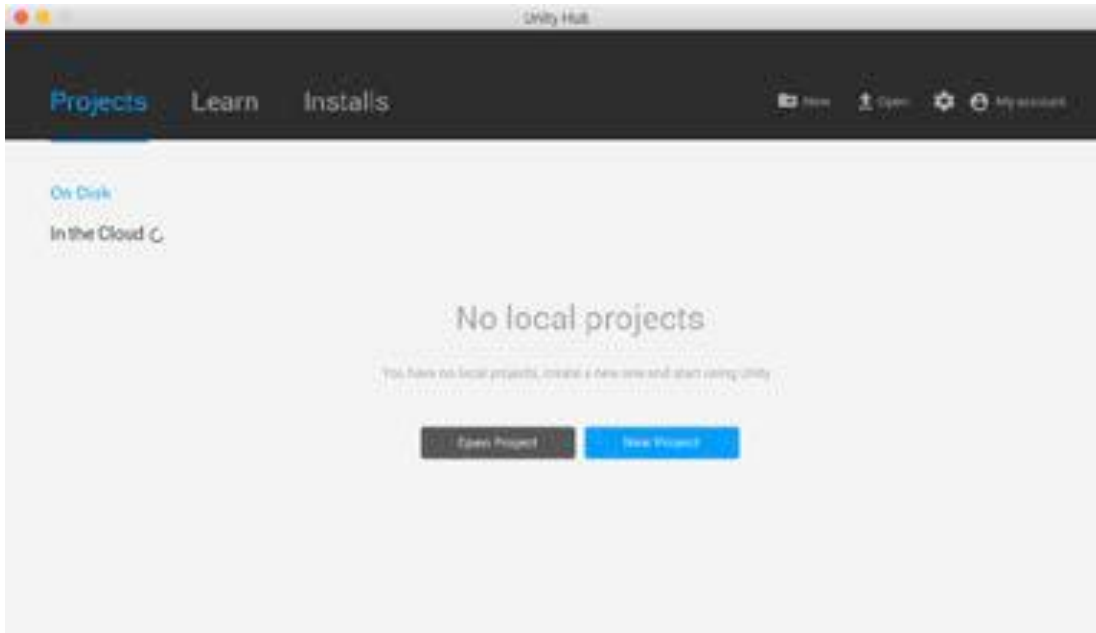
만들 예제

- 오브젝트를 배치하고 위치 설정하기
- 카메라를 설정하여 화면 변경하기
- 오브젝트에 물리적인 동작 추가하기
- 오브젝트의 색 변경하기
- 게임 실행해보기



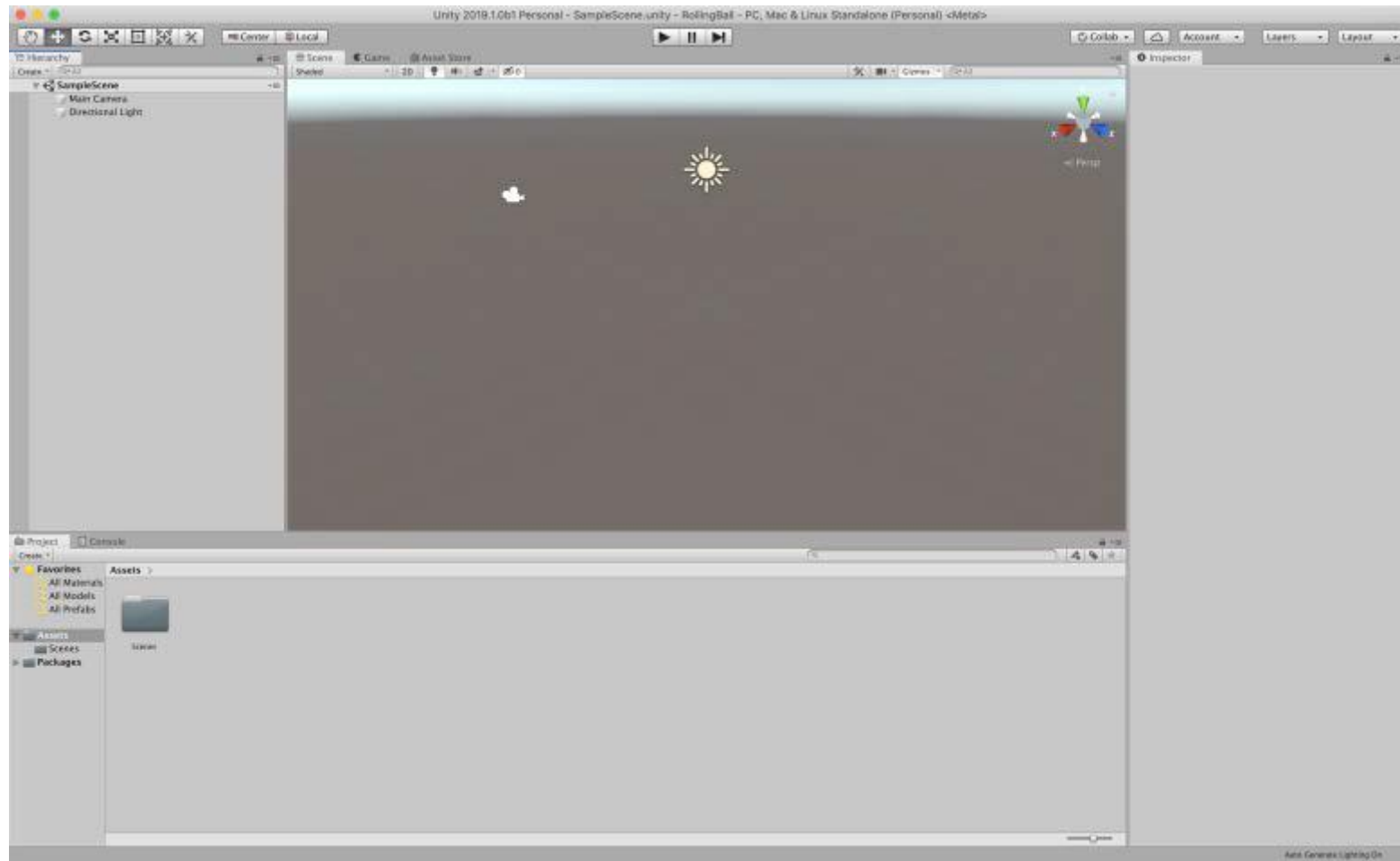
프로젝트 만들기

- 새 프로젝트 만들기: 유니티를 실행하면 나타나는 창에서 Projects 클릭 > New 클릭
- 프로젝트 이름 입력: > RollingBall
- 3D 선택
- 저장 경로 지정하기(임의의 폴더)
- Create project 클릭



▲ 새 프로젝트 만들기

프로젝트 만들기

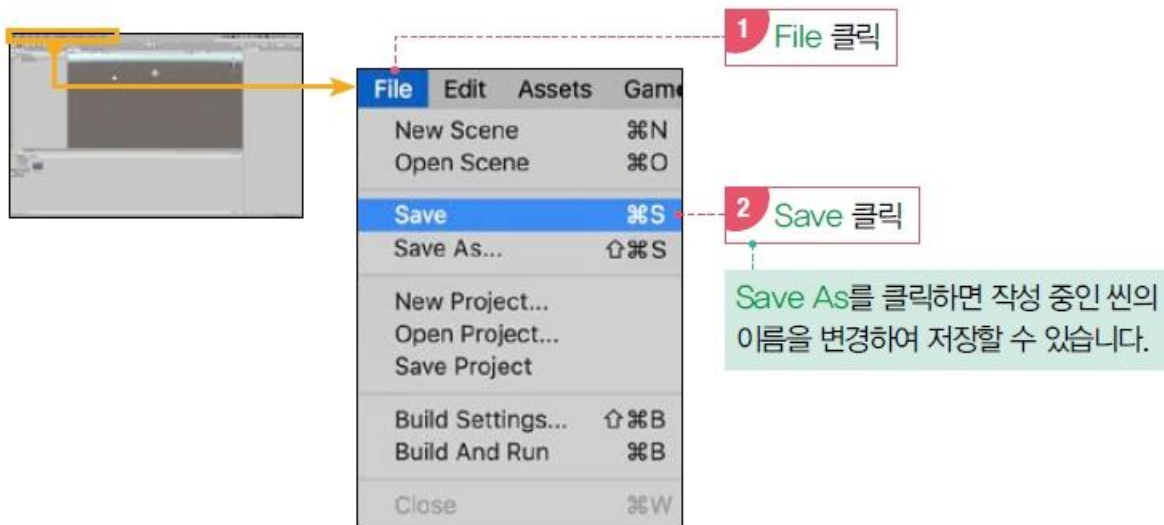


▲ 새로 만들어진 프로젝트

프로젝트 만들기

- 씬 저장: 게임을 구성하는 세계의 데이터로, 유니티의 게임 화면은 씬 단위로 관리됨
- 유니티의 게임 화면 데이터는 하이어라키 창과 씬 뷰에 표시되는 오브젝트를 말함
- Save 클릭: 작성 중인 씬 저장
- 씬을 저장하면 프로젝트 창의 Assets 폴더에 씬 아이콘이 추가됨

단계 1 Save 클릭하기



단계 2 씬 이름 입력하기



재질(Material)

- 머티리얼은 모든 3D 게임제작 프로그램에서 사용하는 공통적인 개념으로 3D 모델의 외형을 설정하기 위한 수단이다.
- 재질을 적용하는 방법
 - 메쉬에 색을 칠하기
 - 메쉬에 서로 다른 텍스처를 적용하기
 - 셰이더 적용하기

재질(Material)

- 유니티 내장 셰이더(Shader)
- Normal
 - 불투명한 질감의 오브젝트를 위해 사용함
- Transparent
 - 완전히 혹은 부분적으로 투명한 물체를 위해 사용함
- Self Illuminated
 - 자체적으로 빛을 발산하는 물체들을 위해 사용함
- Refletive

Material & Shader



◦ Materials Rendering Mode

- Opaque : 기본 값으로 불투명한 텍스처를 표현(Solid 객체에 적합)
- Cutout : 불투명한 부분과 투명한 부분을 동시에 표현(풀, 그물망 등)
- Transparent : 투명한 플라스틱 또는 유리와 같은 재질 표현
- Fade : 불투명한 객체를 부분적으로 페이드 아웃/인을 시킬 수 있어 홀로그램 효과를 구현할 수 있음

Material

- Albedo
 - 빛을 반사하는 정도 (반사율)
- Metallic
 - 오브젝트 표면에 금속의 재질을 표현하기 위한 텍스처
- Normal Map
 - 오브젝트 표면에 굴곡을 표현하기 위한 텍스처

Material

- Height Map

- 텍스처로 높낮이를 표현하는 것으로 보통 지형(Terrain)을 표현

- Occlusion

- 흑백의 텍스처로 간접조명에 의해 생기는 명암을 뚜렷히 표현하여 사물의 체감과 깊이감을 높여줌

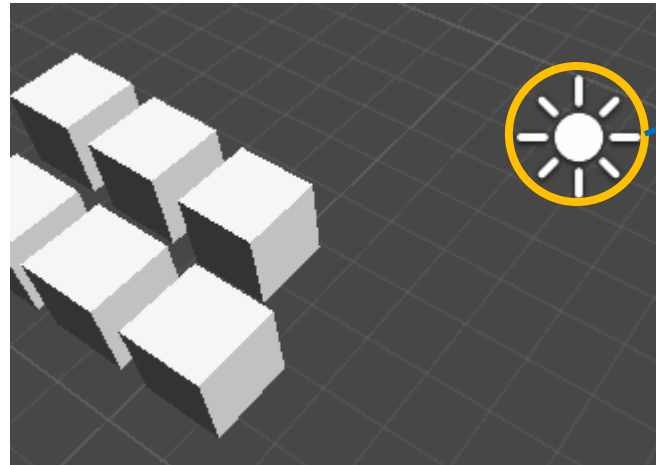
- Emission

- 스스로 빛을 방출하는 속성

- Detail Mask

- 특정 부분만 좀 더 세부적인 텍스처를 표현할 때 사용

Directional Light

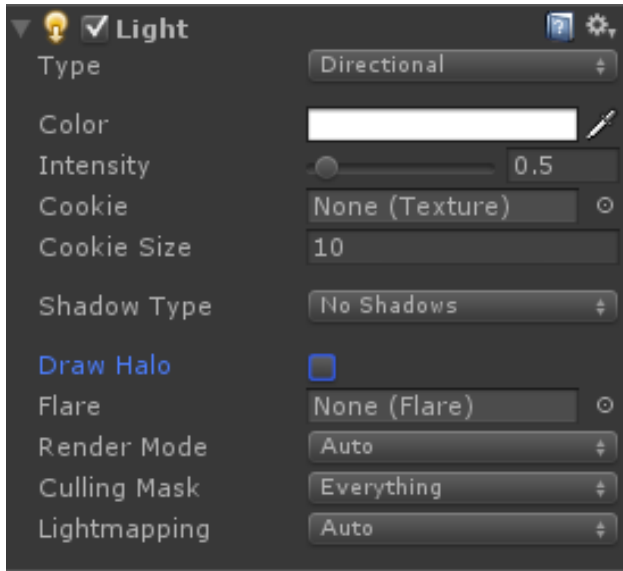


Directional Light
Gizmo

햇빛과 같은 존재로 **방향만 있고 거리감이 없는** 조명이다.

게임 제작 시 기본 조명으로 사용되며 그림자를 만든다.
조명의 밝기와 색상을 설정할 수 있다.

Directional Light



Cookie & Cookie Size : Cookie에 등록된 Texture 모양, Cookie Size 크기의 빛 생성을 설정함

Draw Halo : 빛 주변으로 후광 효과를 설정함

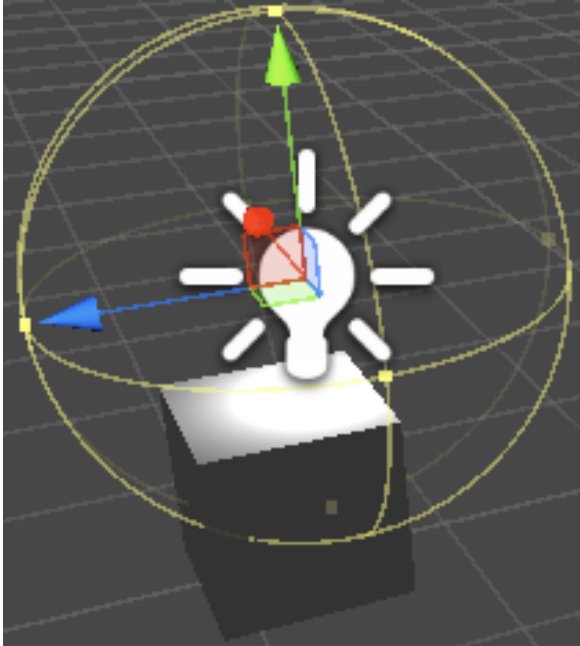
Flare : 빛의 효과에 대한 모습을 설정함

Render Mode : 중요한 빛과 중요하지 않은 빛을 구분할 때 설정함

Culling Mask : Layer에 빛을 적용할지 안할지를 설정함

Light

Point Light



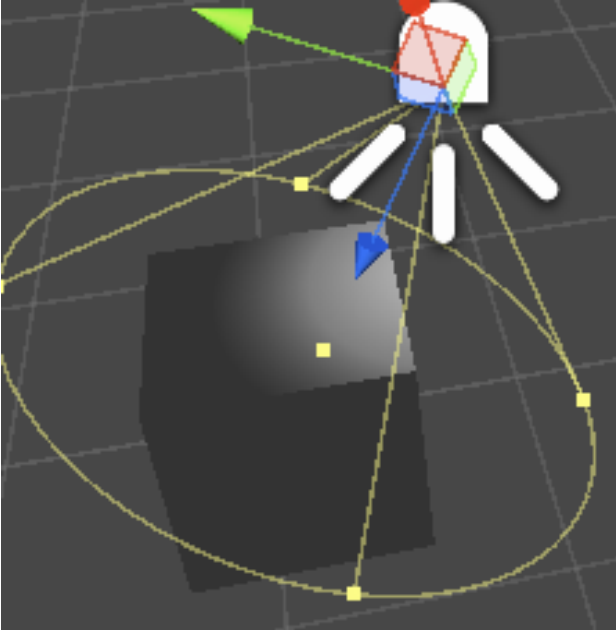
전구의 빛처럼 사방으로 고르게 퍼져가는 조명으로
방향은 없고 **거리**만 **존재**함

Intensity : 조명의 밝기를 조절함

Range : 조명의 거리를 조절함

Light

Spot Light



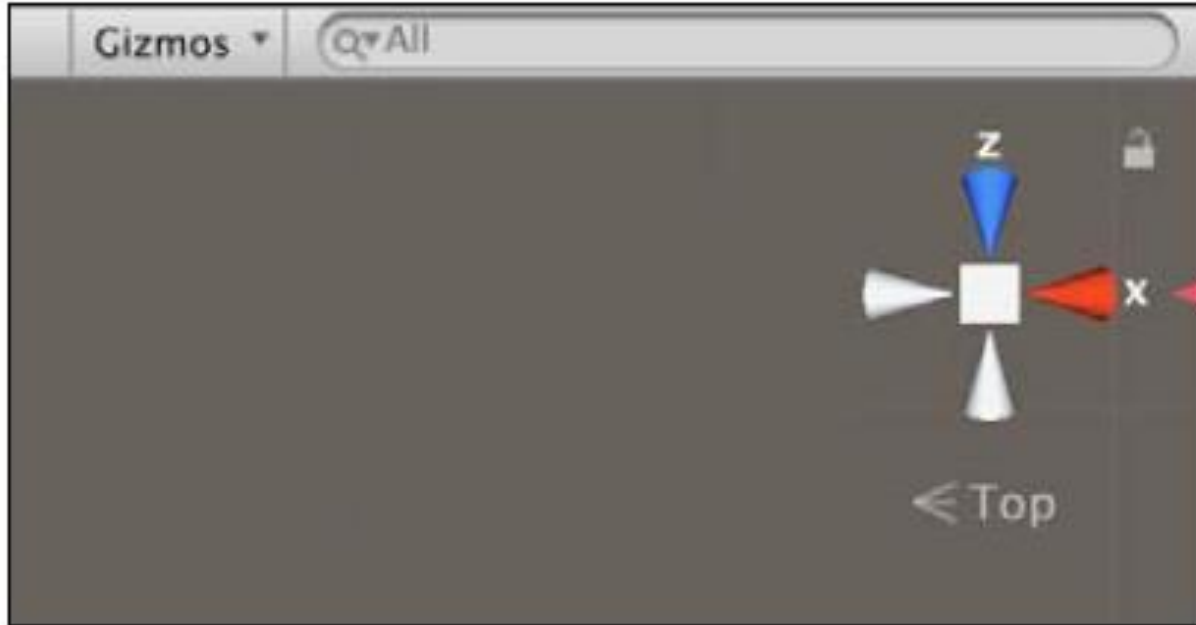
방향과 **거리가 있는 조명**

Spot Angle : 조명의 퍼짐을 조절함

그림자를 만들지 않지만 그림자의 밝기에는 영향을 줌

바닥과 벽 만들기

- 시점 조절: 씬 뷰 오른쪽 상단에 있는 씬 기즈모(Scene Gizmo)의 원뿔 부분 클릭
- 파란 원뿔(z)이 위, 빨간 원뿔(x)이 오른쪽에 오도록 함



z가 위, x가 오른쪽에 오도록 변경하기

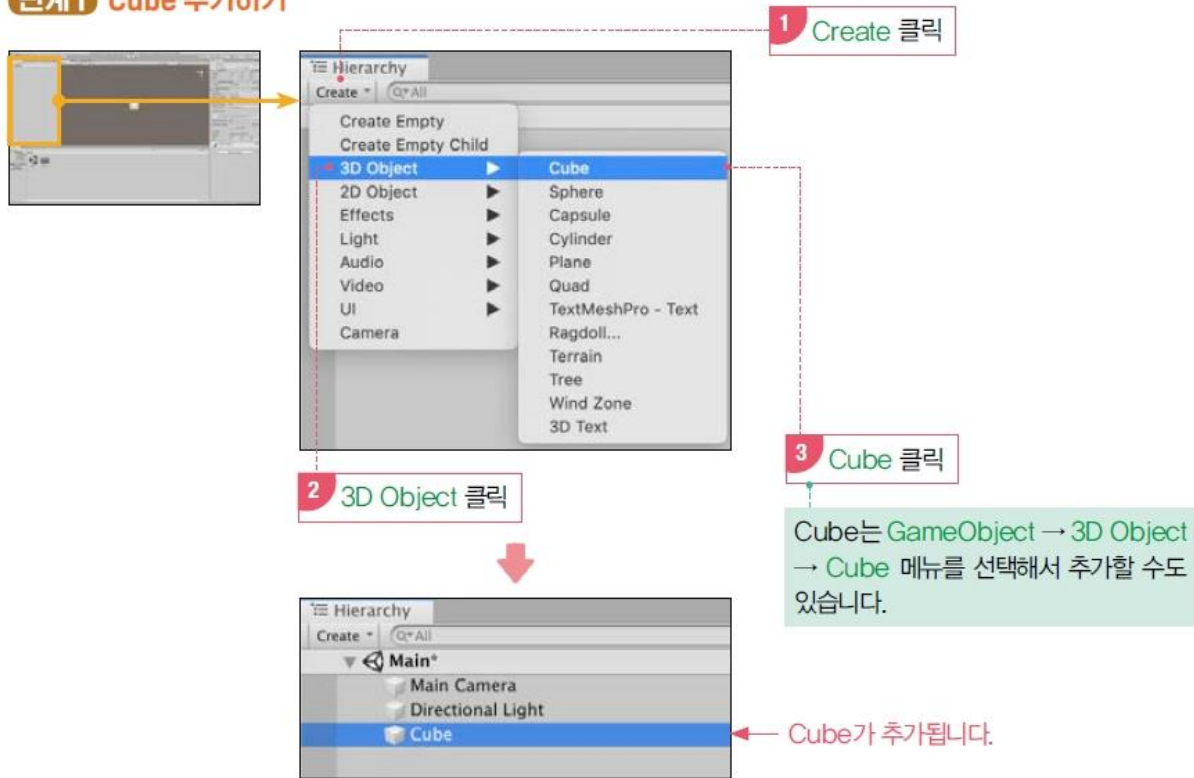
원뿔 부분을 클릭하면 씬의 방향이 회전합니다.

▲ 씬의 방향 변경하기

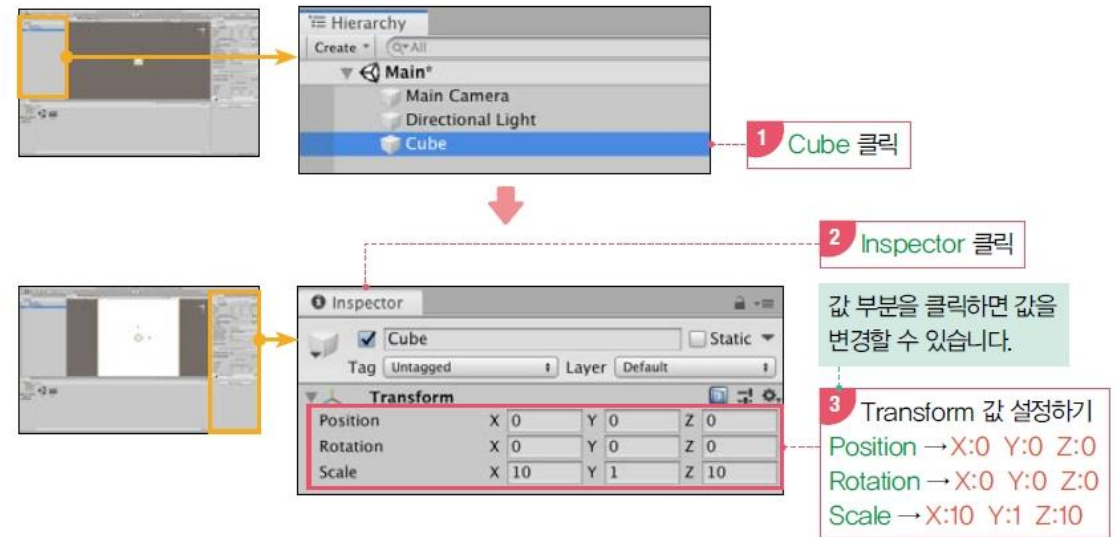
바닥과 벽 만들기

- 바닥 만들기: 유니티에 미리 마련되어 있는 기본 도형 중 Cube를 이용해서 바닥을 만듦
- 1단계 Cube 추가
- 2단계 Cube의 위치, 각도, 크기 설정

단계 1 Cube 추가하기



단계 2 Cube의 위치, 각도, 크기 설정하기



바닥과 벽 만들기

- 바닥 만들기: 유니티에 미리 마련되어 있는 기본 도형 중 Cube를 이용해서 바닥을 만듦
- 3단계 Cube의 이름 변경하기
- 바닥 완성

단계 3 Cube의 이름 변경하기

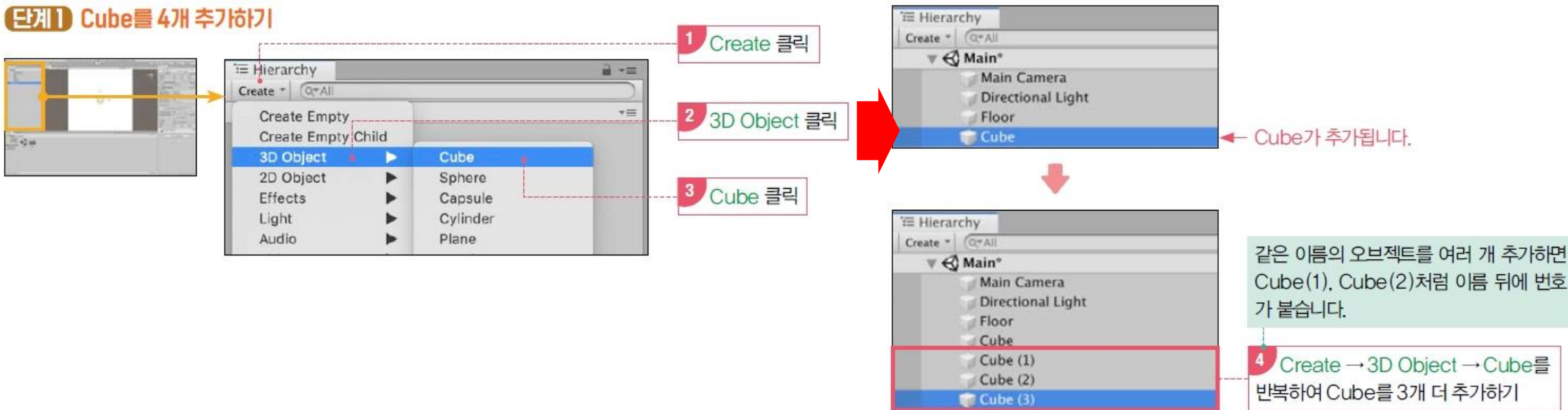
The process is shown in four steps:

- 1 Cube 우클릭**: A screenshot of the Unity Hierarchy panel with 'Cube' selected. A red arrow points to the 'Cube' entry.
- 2 Rename 클릭**: A screenshot of the context menu that appears after right-clicking 'Cube', with 'Rename' highlighted.
- 3 이름 변경하기 → Floor**: A screenshot of the Hierarchy panel where 'Cube' has been renamed to 'Floor'. A text box explains: "오브젝트는 구분하기 쉽도록 이름을 변경해줍니다." (We change the name of the object so it's easy to distinguish).
- Floor**: A screenshot of the Unity scene view showing a yellow cube in the center. A red arrow points to the cube with the label 'Floor'. A text box explains: "하이어라키 창에서 Floor를 더블 클릭하면 Floor가 화면 중앙에 표시됩니다. 씬 뷰에서 오브젝트를 찾기 어려울 때는 하이어라키 창에서 찾아봅니다." (Double-clicking Floor in the Hierarchy window displays Floor in the center of the screen. When it's difficult to find the object in the Scene View, look for it in the Hierarchy window).

바닥과 벽 만들기

- 벽 만들기: 바닥 만들기과 같은 방법으로 Cube를 4개 추가
- 1단계 Cube를 4개 추가

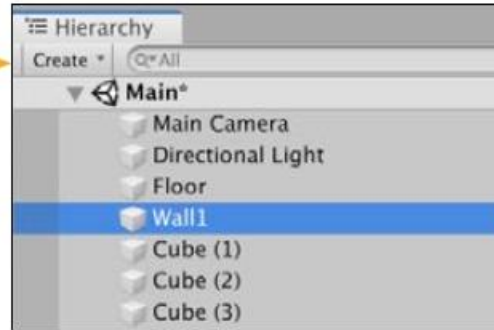
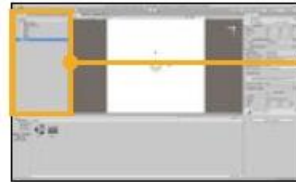
단계 1 Cube를 4개 추가하기



바닥과 벽 만들기

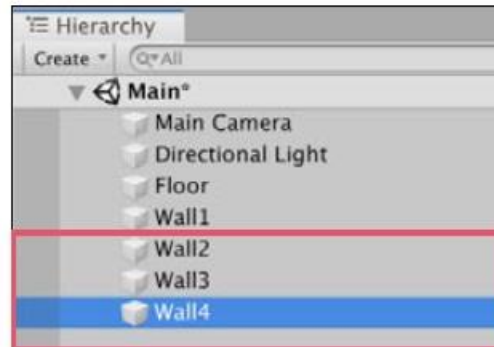
- 벽 만들기: 바닥 만들기과 같은 방법으로 Cube를 4개 추가
- 2단계 Cube의 이름 변경하기

단계 2 Cube의 이름 변경하기



이름을 변경하려면 오브젝트를 우클릭한 뒤 **Rename**을 선택합니다.

1 Cube의 이름 변경하기
→ Wall1



오브젝트를 우클릭한 뒤 **Delete**를 선택하면 오브젝트를 제거할 수 있습니다.

2 나머지 Cube의 이름도 변경하기
Cube(1) → Wall2
Cube(2) → Wall3
Cube(3) → Wall4

▲ 벽 만들기

바닥과 벽 만들기

- 벽 만들기: 바닥 만들기과 같은 방법으로 Cube를 4개 추가
- 3단계 Wall1의 위치, 각도, 크기 설정하기

단계 3 Wall1의 위치, 각도, 크기 설정하기

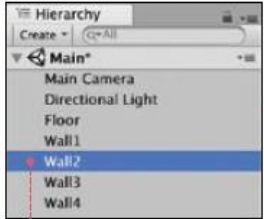
1 Wall1 클릭

2 Transform 값 설정하기
Position → X:5.5 Y:0 Z:0
Rotation → X:0 Y:0 Z:0
Scale → X:1 Y:2 Z:10

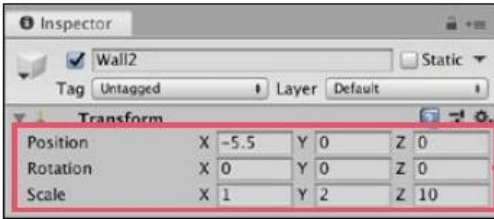
Wall1

바닥과 벽 만들기

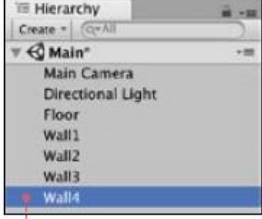
- 벽 만들기: 바닥 만들기과 같은 방법으로 Cube를 4개 추가
- 4단계 Wall2~Wall4의 위치, 각도, 크기 설정하기



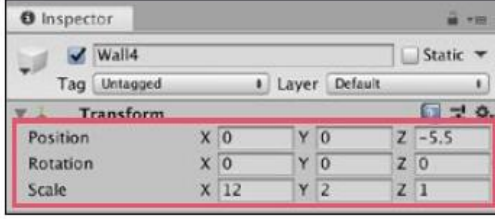
1 Wall2 클릭



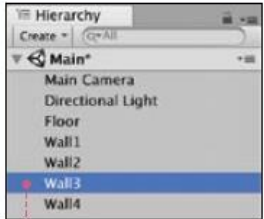
2 Transform 값 설정하기
Position → X:-5.5 Y:0 Z:0
Rotation → X:0 Y:0 Z:0
Scale → X:1 Y:2 Z:10



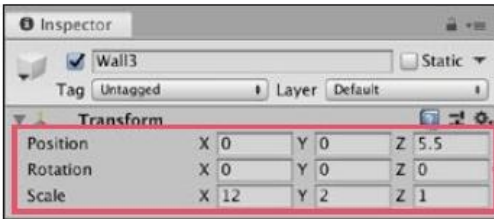
5 Wall4 클릭



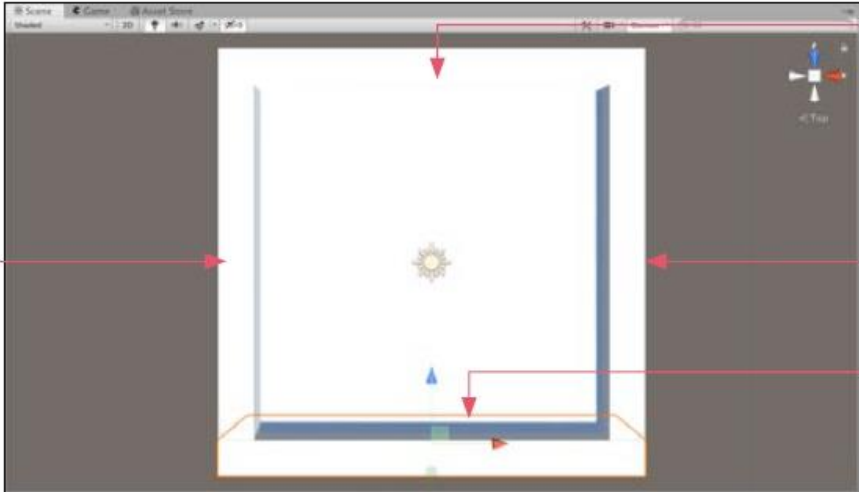
6 Transform 값 설정하기
Position → X:0 Y:0 Z:-5.5
Rotation → X:0 Y:0 Z:0
Scale → X:12 Y:2 Z:1



3 Wall3 클릭



4 Transform 값 설정하기
Position → X:0 Y:0 Z:5.5
Rotation → X:0 Y:0 Z:0
Scale → X:12 Y:2 Z:1



Wall3

Wall2

Wall1

Wall4

▲ 벽 만들기

바닥과 벽 만들기

- 카메라 변경하기: Main Camera의 위치와 각도를 설정하여 목표 지점이 제대로 화면에 나타나게 함
- Main Camera의 위치와 각도 변경하기

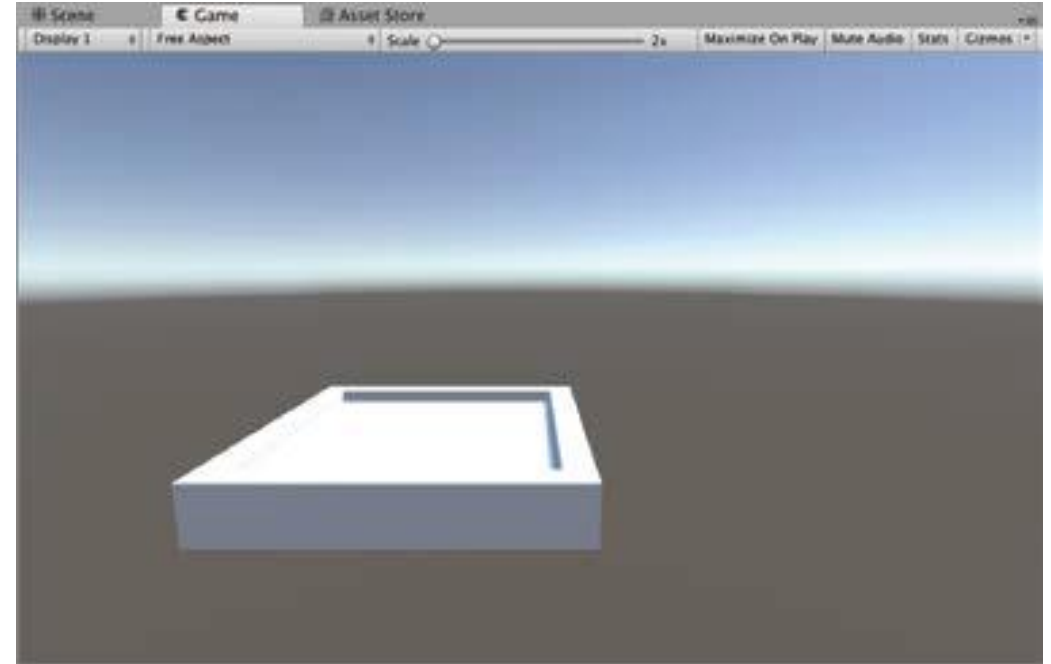
단계1 Main Camera의 위치와 각도 변경하기

1 Main Camera 클릭

2 Position 값 설정하기
→ X:-20 Y:7 Z:-4

3 Rotation 값 설정하기
→ X:10 Y:90 Z:0

| Inspector | | | | | |
|-------------|---|------------|---|---------------|------|
| Main Camera | | | | | |
| Tag | | MainCamera | | Layer Default | |
| Transform | | | | | |
| Position | X | -20 | Y | 7 | Z -4 |
| Rotation | X | 10 | Y | 90 | Z 0 |
| Scale | X | 1 | Y | 1 | Z 1 |

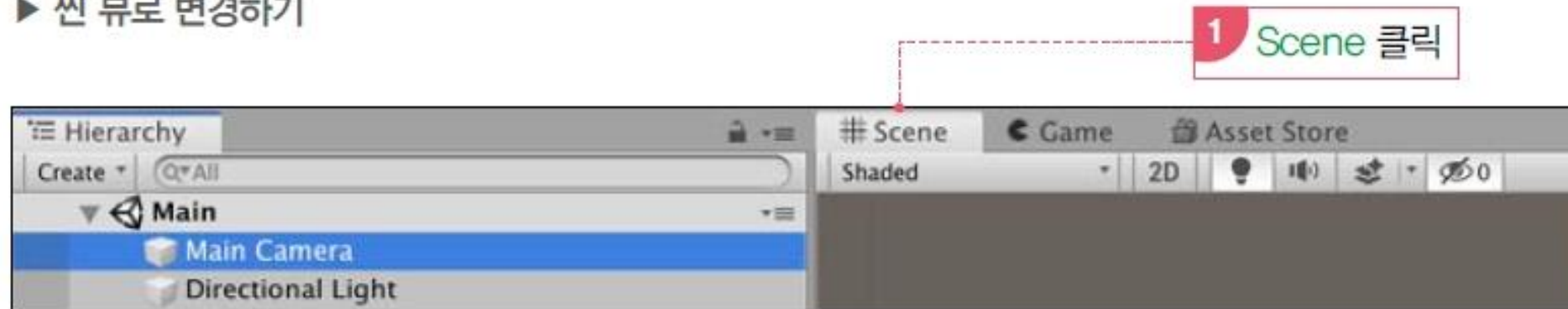


▲ 카메라 변경하기

경사면 만들기

- 경사면을 만들기 전, 화면을 씬 뷰로 변경

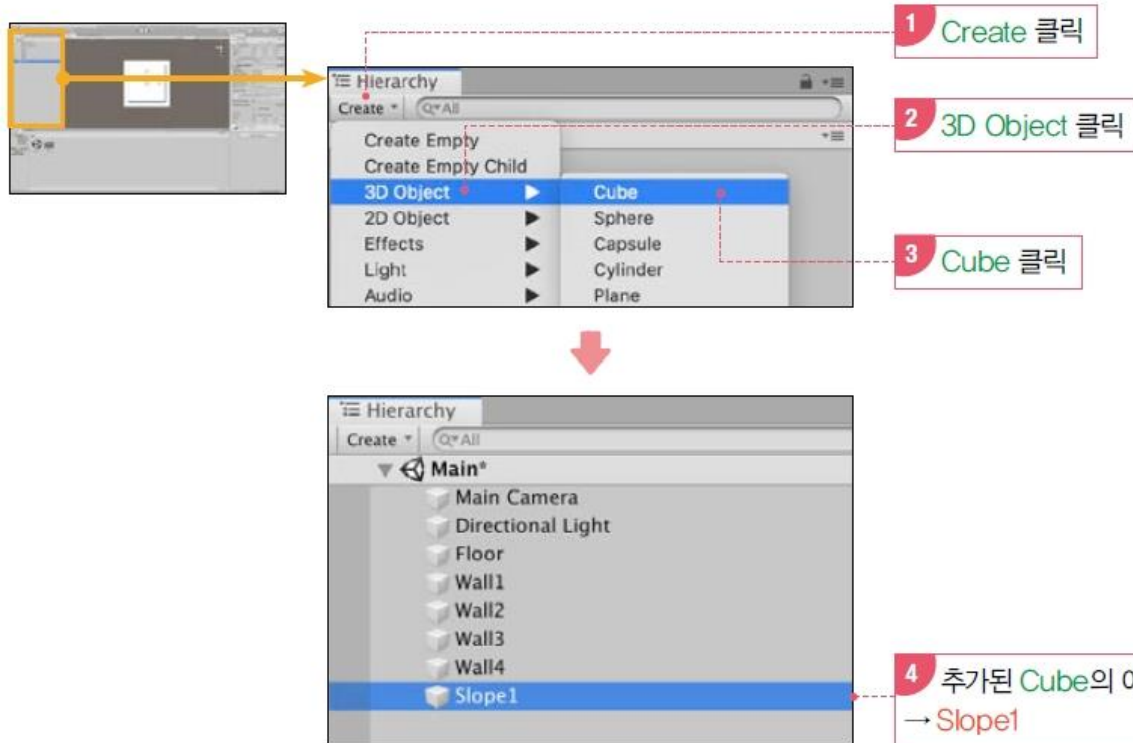
▶ 씬 뷰로 변경하기



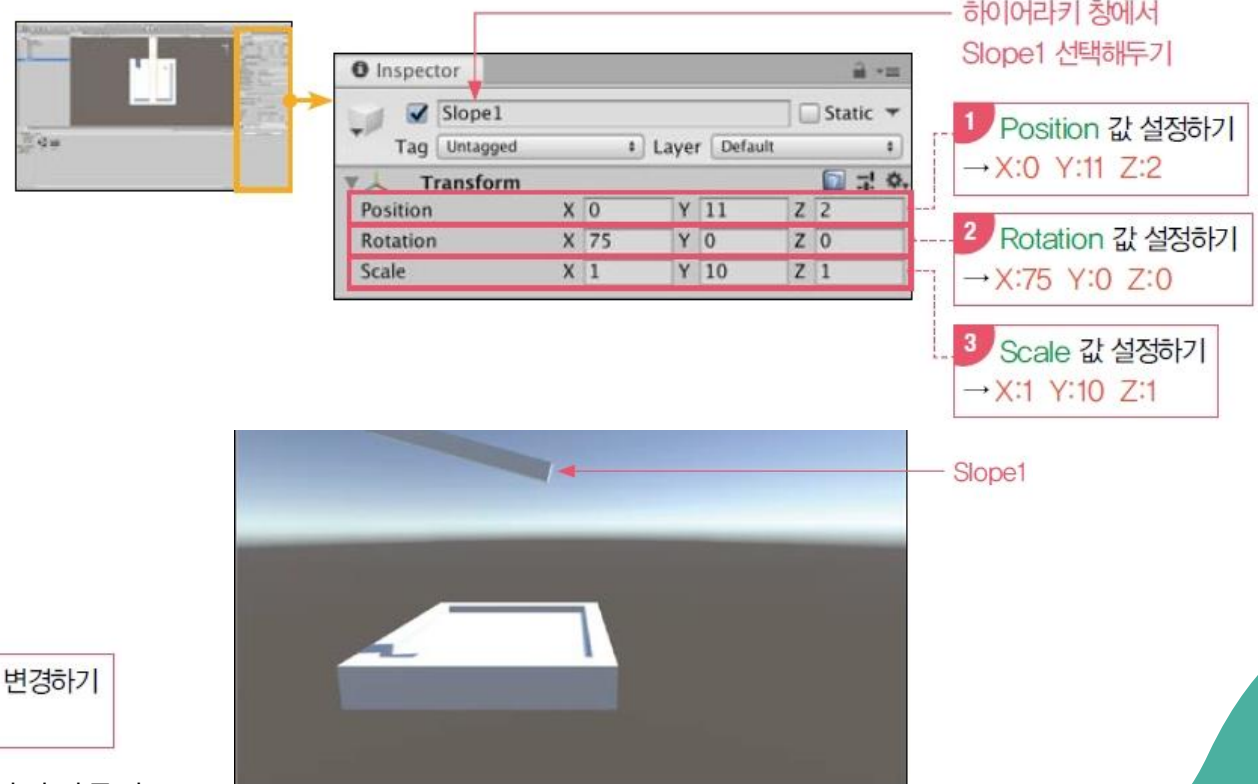
경사면 만들기

- 첫 번째 경사면 만들기: Cube 추가하여 이름 변경, 위치, 각도, 크기 설정
- 단계 1 Slope1 추가하기
- 단계 2 Slope1의 위치, 각도, 크기 설정하기

단계 1 Slope1 추가하기



단계 2 Slope1의 위치, 각도, 크기 설정하기

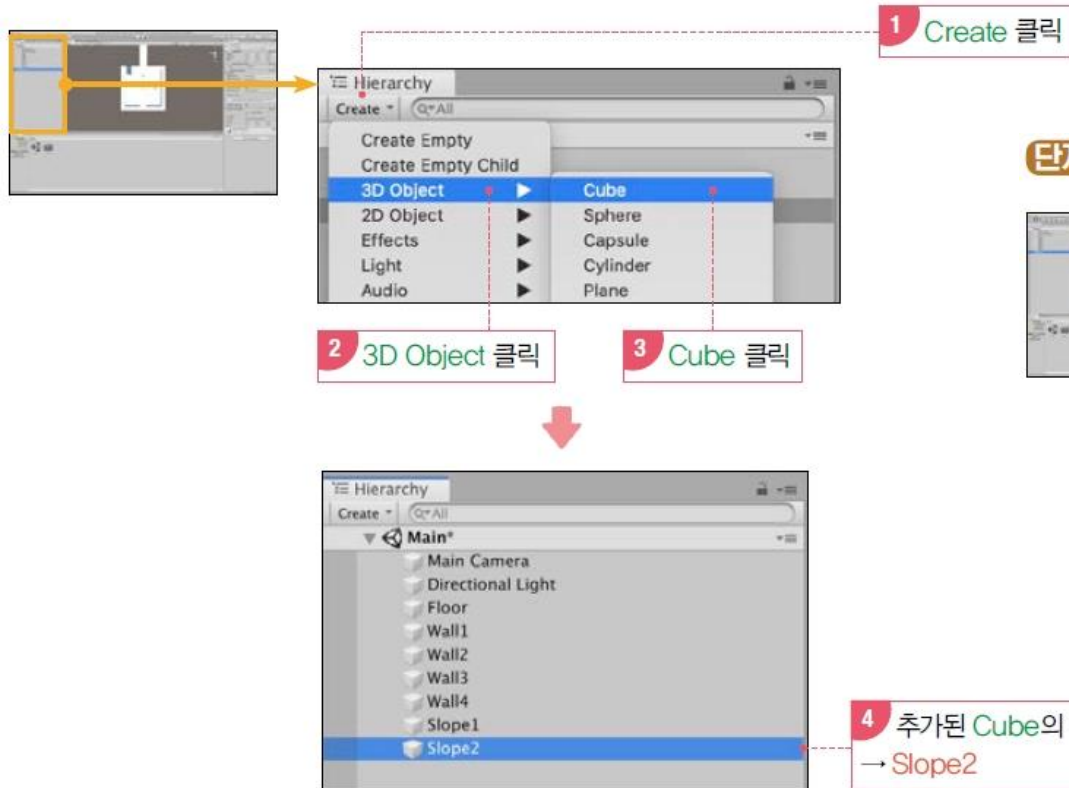


▲ 경사면 만들기

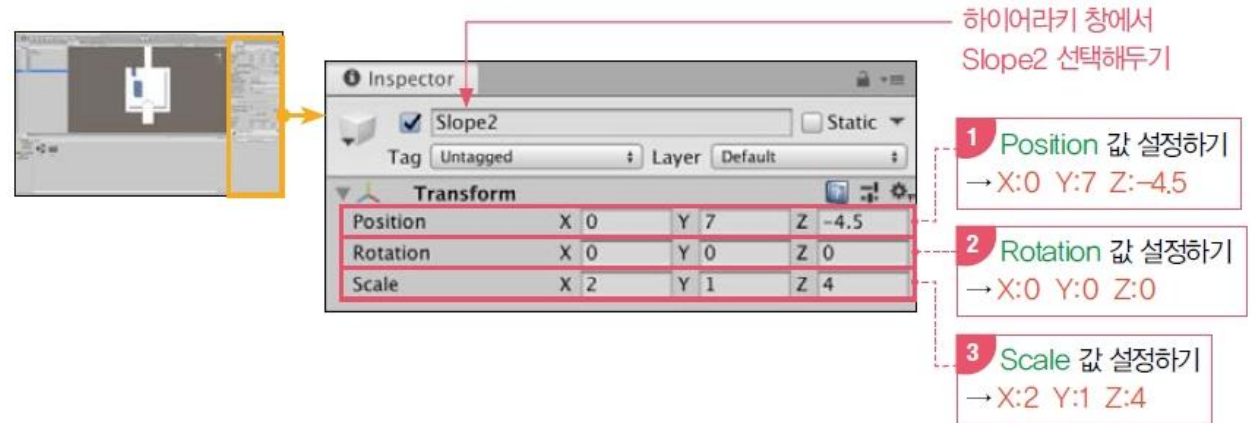
경사면 만들기

- 두 번째 경사면 만들기
- 단계 1 Slope2 추가하기
- 단계 2 Slope2의 위치, 각도, 크기 설정하기

단계 1 Slope2 추가하기



단계 2 Slope2의 위치, 각도, 크기 설정하기

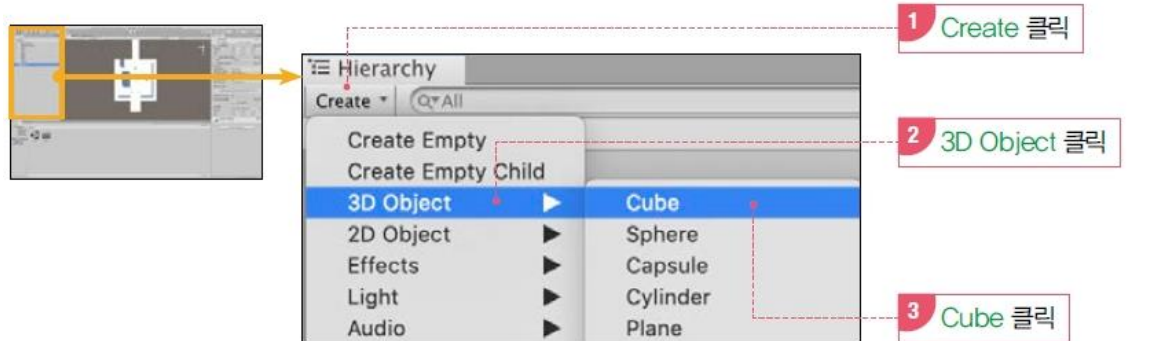


▲ 경사면 만들기

경사면 만들기

- 세 번째 경사면 만들기
- 단계 1 Slope3 추가하기
- 단계 2 Slope3의 위치, 각도, 크기 설정하기


단계 1 Slope3 추가하기



1 Create 클릭

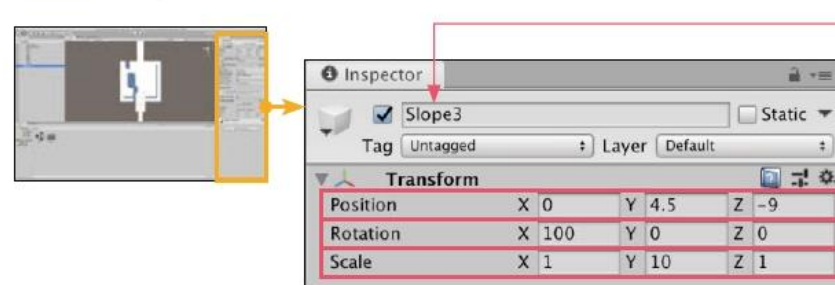
2 3D Object 클릭

3 Cube 클릭



4 추가된 Cube의 이름 변경하기
→ Slope3

단계 2 Slope3의 위치, 각도, 크기 설정하기



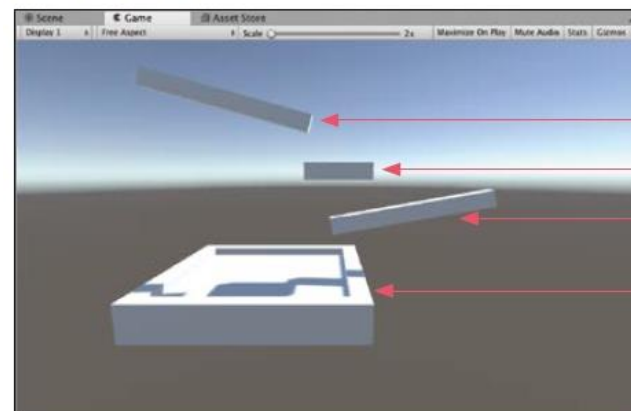
하이어라키 창에서 Slope3 선택해두기

1 Position 값 설정하기
→ X:0 Y:4.5 Z:-9

2 Rotation 값 설정하기
→ X:100 Y:0 Z:0

3 Scale 값 설정하기
→ X:1 Y:10 Z:1

▶ 스테이지 완성



Slope1

Slope2

Slope3

Floor

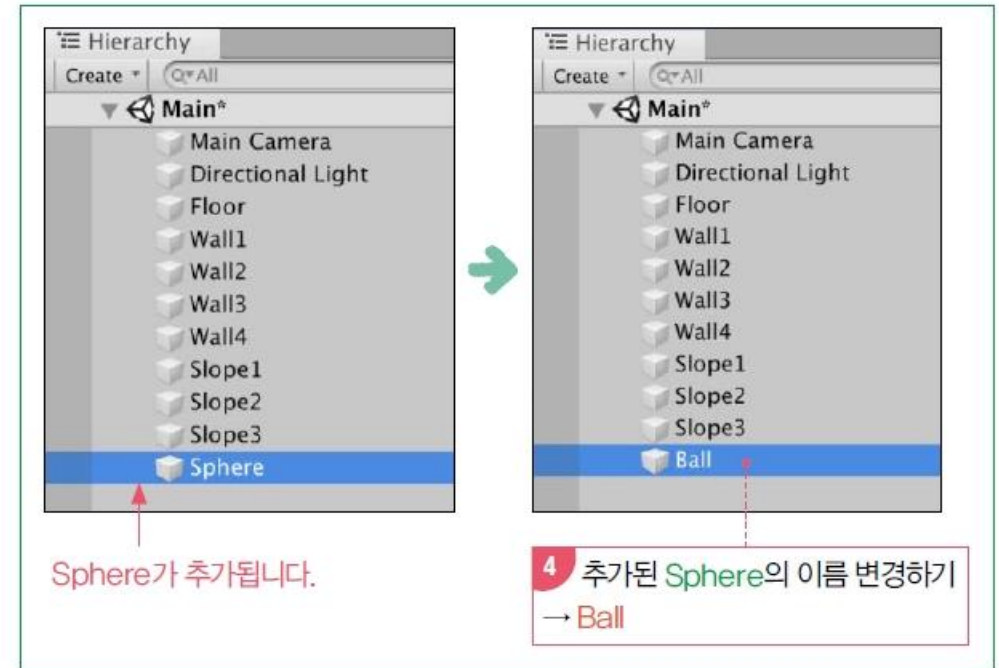
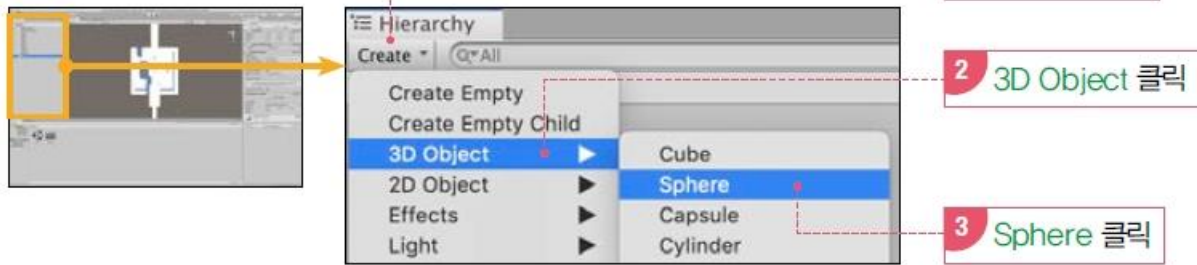
Wall1 ~ Wall4

▲ 경사면 만들기

공 만들기

- 공 추가하기: 유니티 기본 도형 가운데 Sphere를 사용
- 단계 1 Sphere 추가하기
- 단계 2 Ball의 위치와 크기 설정하기

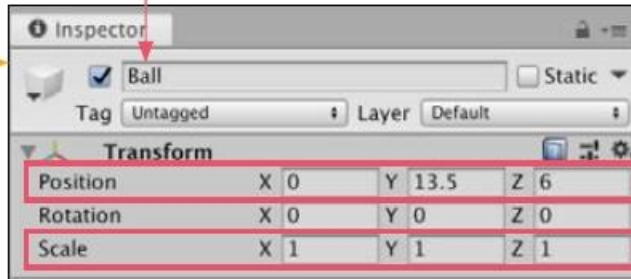
단계1 Sphere 추가하기



공 만들기

- 공 추가하기: 유니티 기본 도형 가운데 Sphere를 사용
- 단계 1 Sphere 추가하기
- 단계 2 Ball의 위치와 크기 설정하기

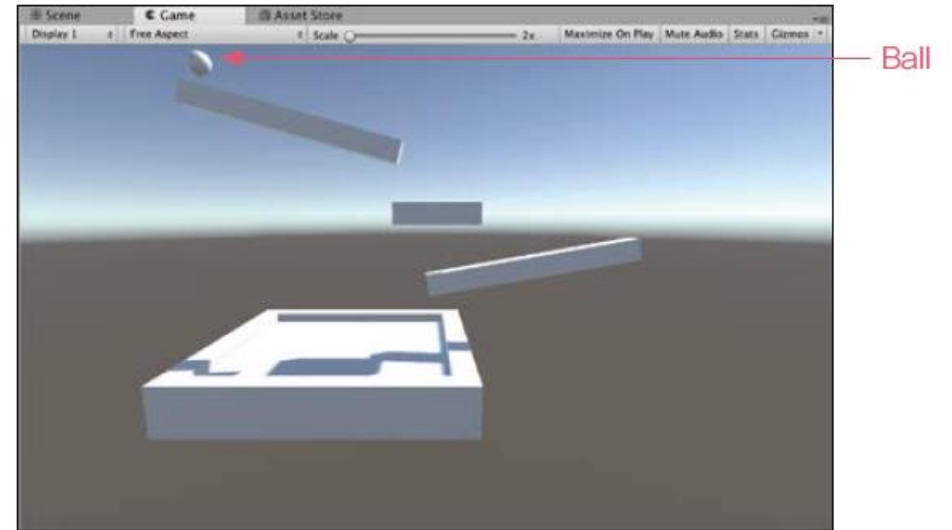
단계 2 Ball의 위치와 크기 설정하기



하이어라키 창에서
Ball 선택해두기

1 Position 값 설정하기
→ X:0 Y:13.5 Z:6

2 Scale 값 설정하기
→ X:1 Y:1 Z:1



공 만들기

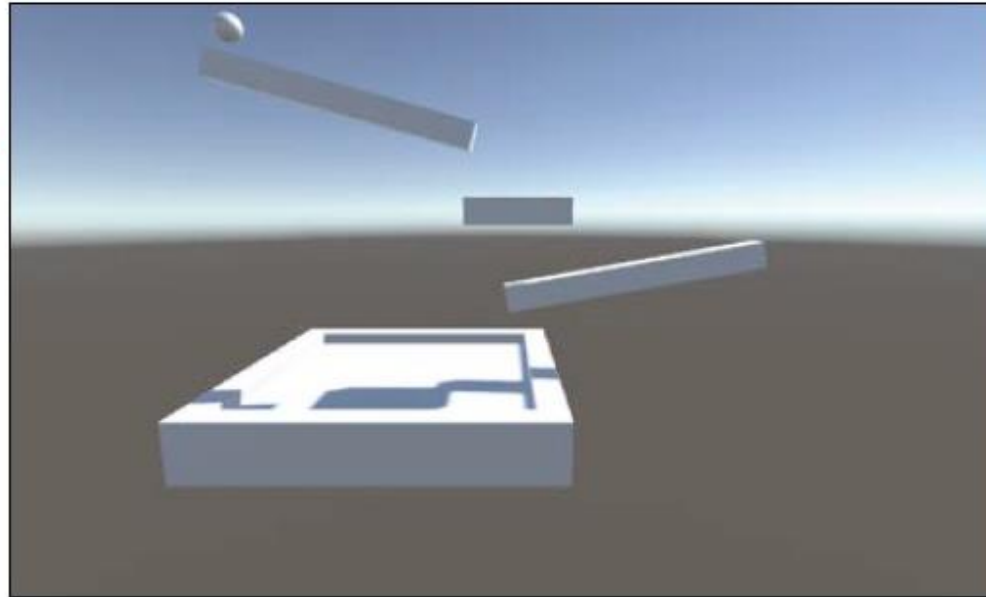
- **게임 실행하기:** 실행 도구의 플레이 버튼 클릭하여 게임 실행
- 다만 지금 상태로는 공이 굴러가지 않음
- 유니티에서 공을 굴리거나 상자를 떨어뜨리는 등의 물리적 동작을 위해서는 리지드바디 (Rigidbody)라는 컴포넌트 설정이 필요

▶ 게임 실행하기



1 플레이 버튼 클릭

왼쪽 : 게임을 실행/중지합니다.
가운데: 게임을 일시 중지합니다.
오른쪽: 게임을 한 프레임씩 움직입니다.



▲ 게임 실행하기

물리 엔진

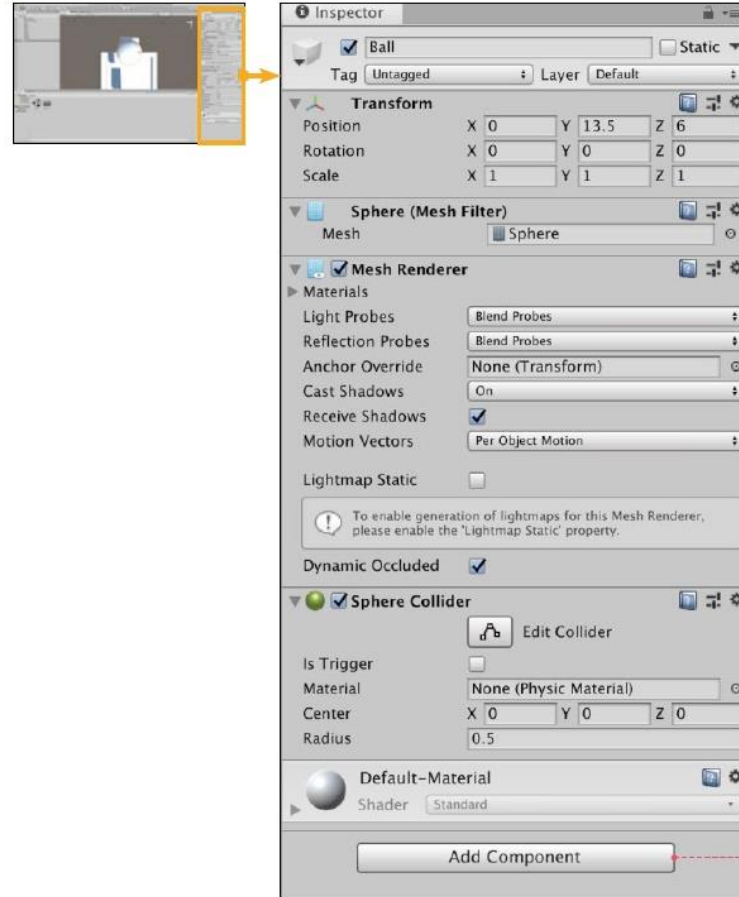
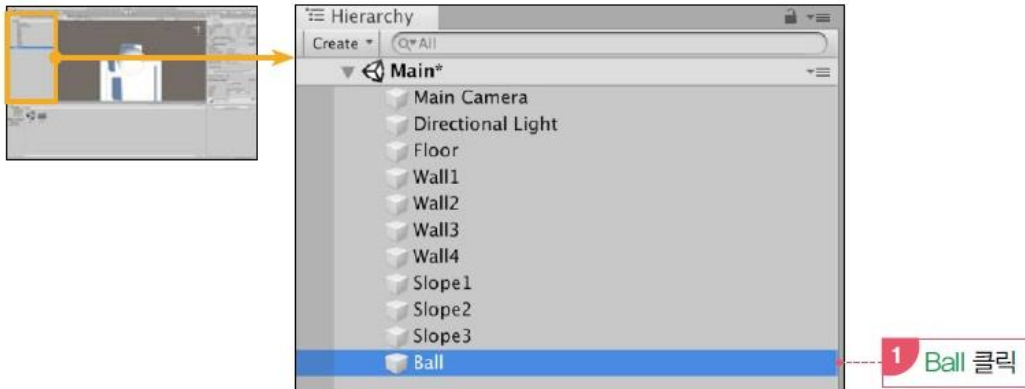
- 리지드 바디[Rigidbody] : 오브젝트에 물리적인 움직임을 적용할 때 사용

| 옵션 | 설명 |
|---------------------|--|
| Mass | 질량 설정 |
| Drag | 저항 설정 |
| Angular Drag | 회전 저항 설정 |
| Use Gravity | 중력 적용 |
| Is Kinematic | 트랜스폼 값으로만 오브젝트 이동 (Collider 무시) |
| Interpolate | Rigidbody의 움직임이 어색할경우에 적용 Interpolate : 이전 프레임과 시간값 기준으로 물체 위치 보간 Extrapolate : 강체의 속력을 활용해 위치 보간 |
| Collision Detection | Discrete : 분리된 프레임 별로 충돌 체크. 계산량이 적음 Continuous : 경로 예측으로 충돌 예측 탐지. 계산량이 많음 Continuous Dynamic : 가장 부하가 많으며, 빠르게 움직이는 물체의 충돌을 감지하는데 적합 |
| Constraints | 물리 축을 고정 하는 기능 |

중력 설정하기

- 리지드바디가 설정된 오브젝트는 물리 법칙에 따라 동작: 중력의 영향을 받게 됨
- **Ball에 리지드바디 설정하기**: 하이어라키 창에서 Ball 선택 > 인스펙터 창에서 Add Component 클릭

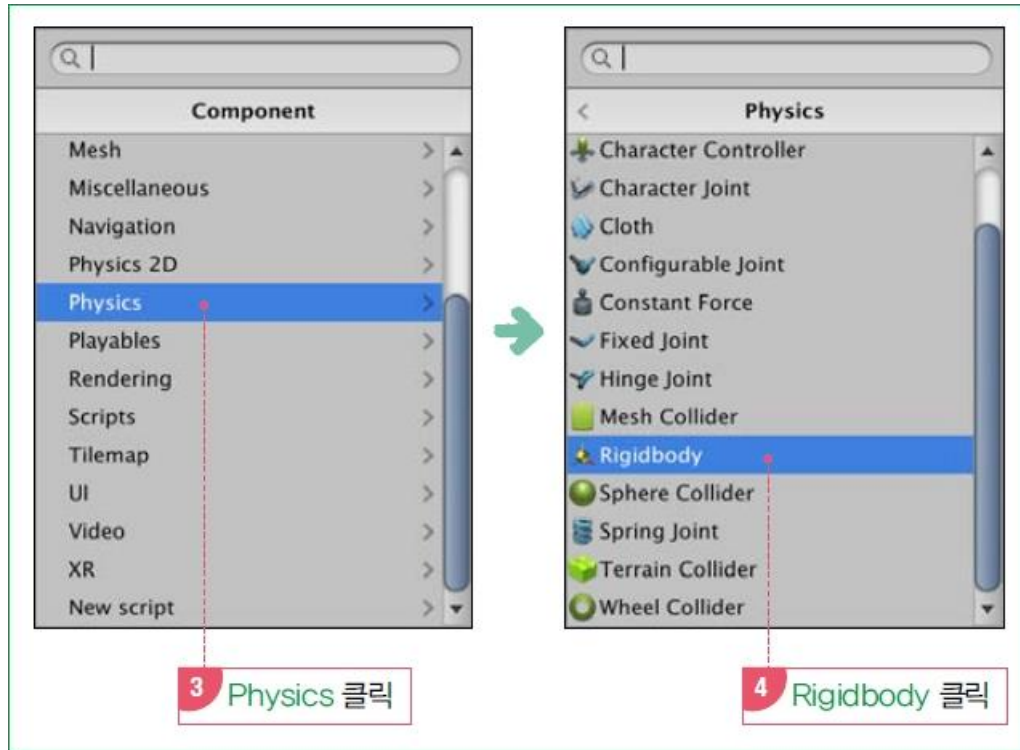
단계 1 Ball에 Rigidbody 추가하기



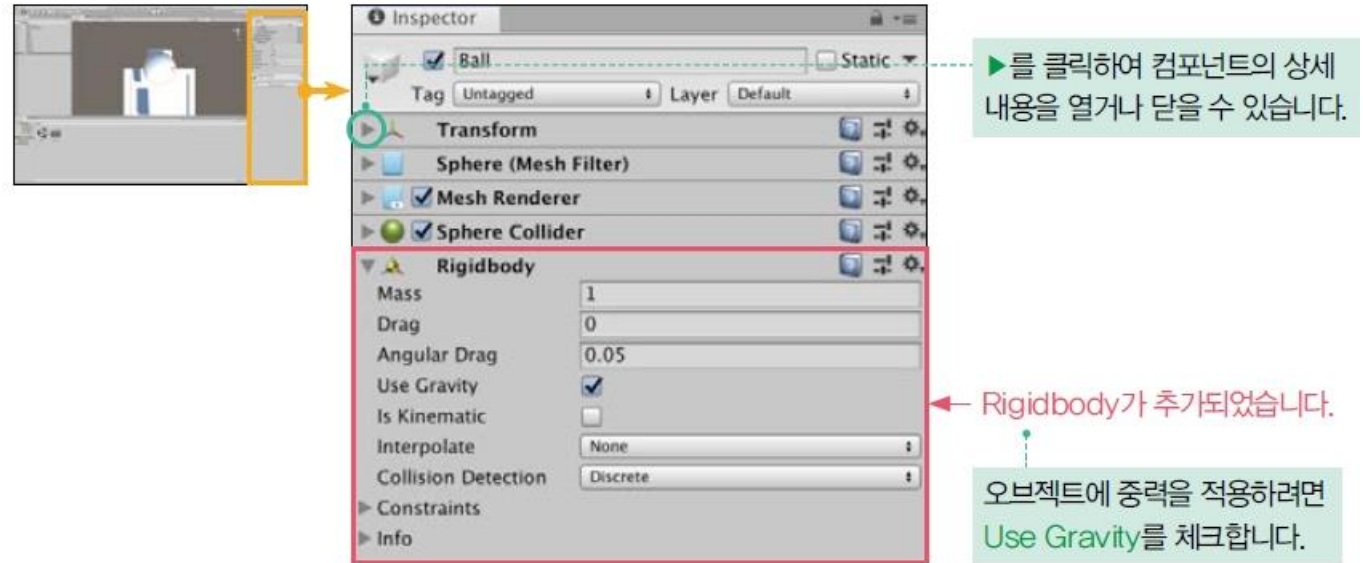
▲ 중력 설정하기

중력 설정하기

- Rigidbody가 Ball에 추가됨: 이렇게 오브젝트에 컴포넌트를 추가하는 것을 어태치라고 함



단계 2 Rigidbody 확인하기



중력 설정하기

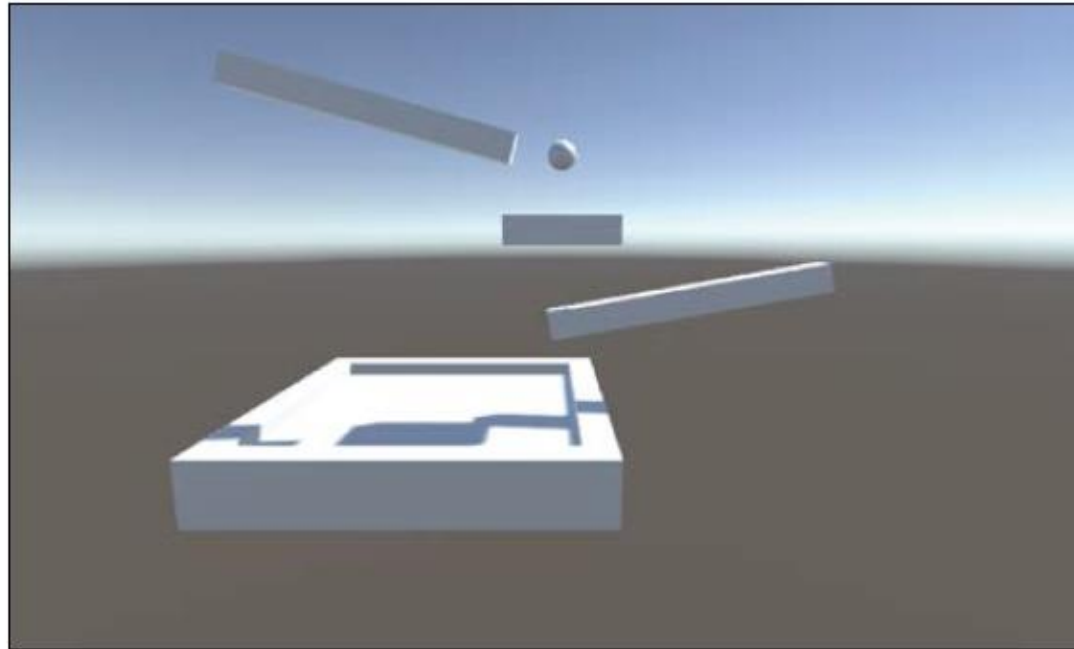
- 추가한 리지드바디가 제대로 동작하는지 확인하기
- 실행 도구의 플레이 버튼 클릭하여 게임 실행 > 확인 후 다시 플레이 버튼 클릭하여 실행 중지

▶ 공이 제대로 굴러가는 모습



1 플레이 버튼 클릭

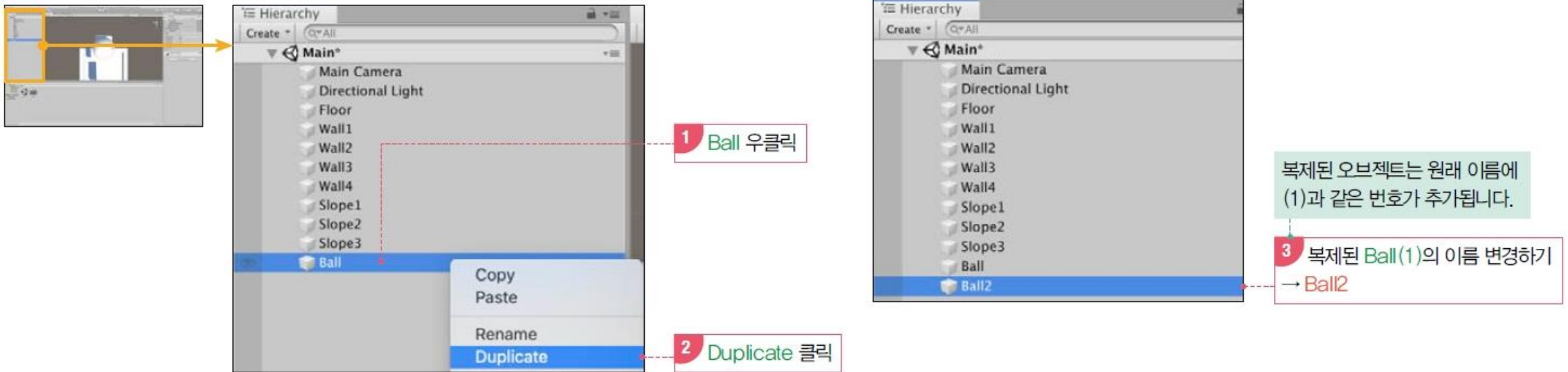
게임 동작을 확인한 후에는 다시
플레이 버튼을 클릭하여 실행을
중지합니다.



게임 발전시키기

- 공을 여러 개 배치하기: 이미 있는 공을 복제해서 공의 개수 늘리기
- 단계 1 Ball 복제하기: 하이어라키 창에서 Ball 우클릭 > Duplicate

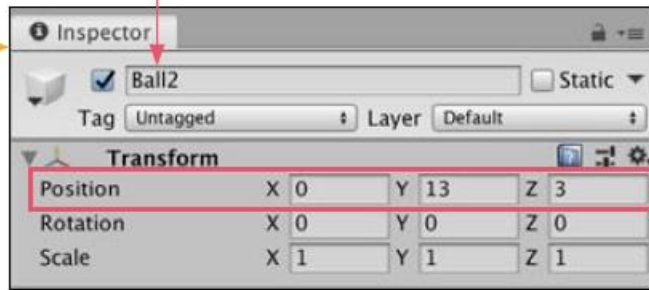
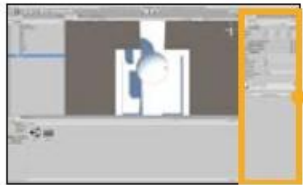
단계 1 Ball 복제하기



게임 발전시키기

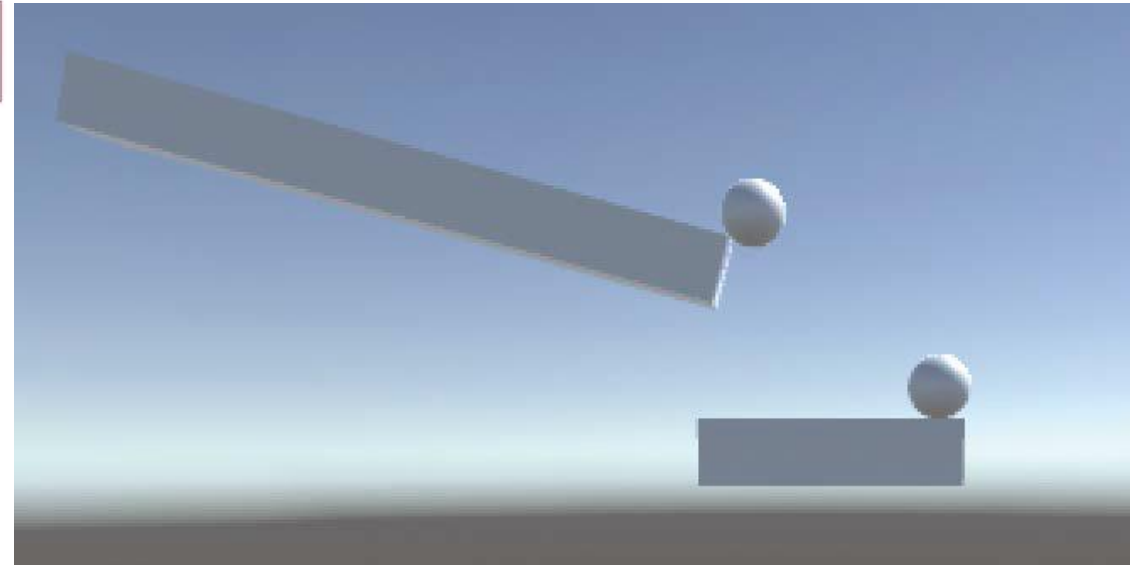
- 단계 2 Ball2의 위치 설정하기

단계 2 Ball2의 위치 설정하기



하이어라키 창에서
Ball2 선택해두기

1 Position 값 설정하기
→ X:0 Y:13 Z:3

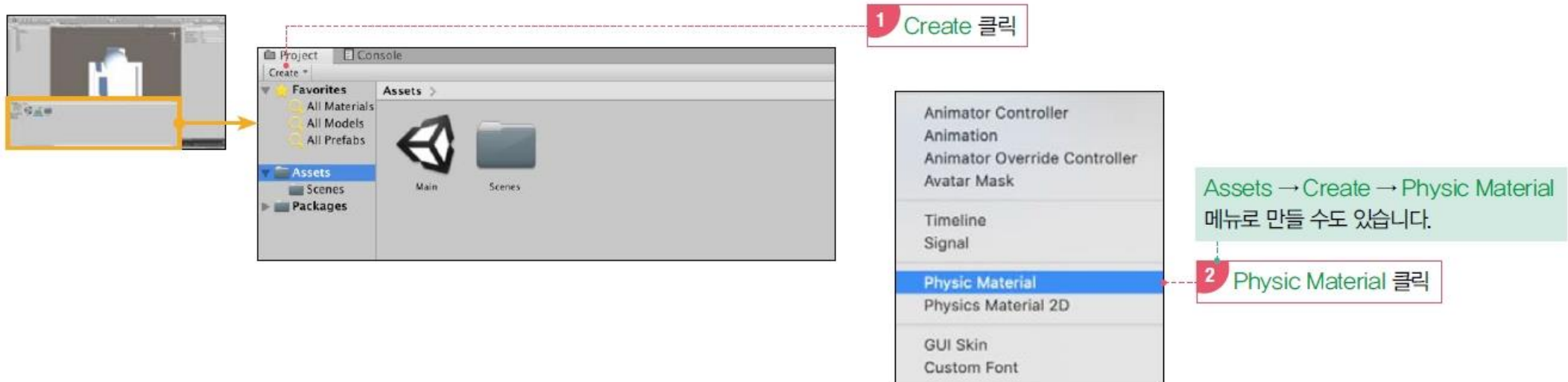


▲ Ball 복제하기

게임 발전시키기

- 탄성 설정하기: 공이 현실의 움직임과 똑같이 동작하게 하려면 Physic Material을 사용
- Physic Material을 Collider 컴포넌트에 추가: 오브젝트끼리 접촉했을 때 마찰계수나 반발계수와 같은 물리 동작 설정이 가능
- 단계 1 Physic Material 추가하기

단계 1 Physic Material 추가하기



Physics Material

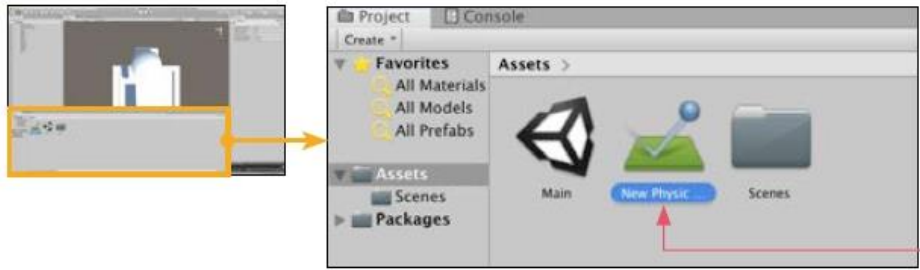
- **Physics Material** : 마찰, 탄성, 충돌에 관한 효과를 적용할 때 사용

| 옵션 | 설명 |
|------------------|--|
| Dynamic Friction | 이동 중일 때 사용되는 마찰력으로 0에 가까울수록 얼음 같은 느낌이 나고 1에 가까울수록 빠르게 정지함 |
| Static Friction | 오브젝트가 정지해 있을 때 사용되는 마찰력으로 0이면 얼음 같은 느낌이 되고 1에 가까울수록 매우 강하게 움직임 |
| Bounciness | 반동하는 방법으로 1이 되면 설정된 에너지의 손실 없이 반동 |

게임 발전시키기

- 단계 2 Physic Material의 이름 변경하기
- 단계 3 Bounce 값 변경하기

단계 2 Physic Material의 이름 변경하기



New Physic Material이 추가됩니다.

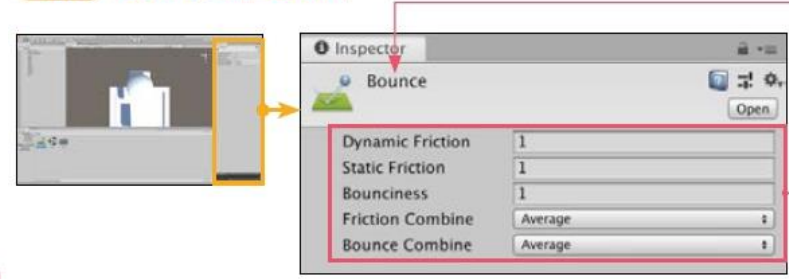


Physic Material은 프로젝트의 Assets 폴더에 추가됩니다.

1 New Physic Material의 이름 변경하기
→ Bounce

식별하기 쉽도록 이름을 변경합니다. 임의의 이름으로 변경해도 상관없습니다.

단계 3 Bounce 값 변경하기



프로젝트 창에서 Bounce 선택해두기

1 값 변경하기

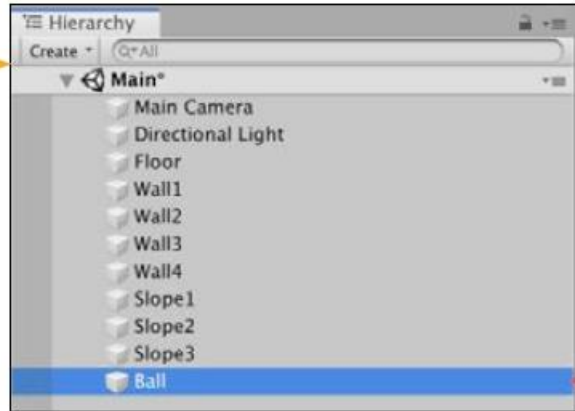
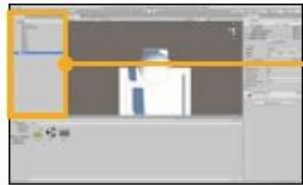
| | |
|------------------|---------|
| Dynamic Friction | 1 |
| Static Friction | 1 |
| Bounciness | 1 |
| Friction Combine | Average |
| Bounce Combine | Average |

Dynamic Friction → 1
Static Friction → 1
Bounciness → 1
Friction Combine → Average
Bounce Combine → Average

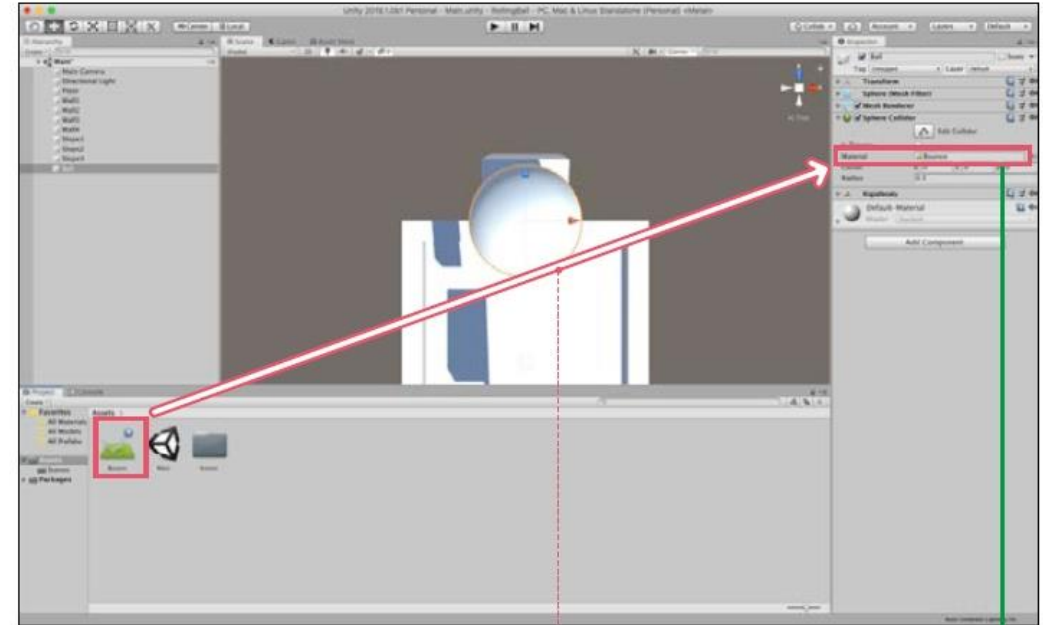
게임 발전시키기

- 단계 4 Ball에 Bounce 어태치하기

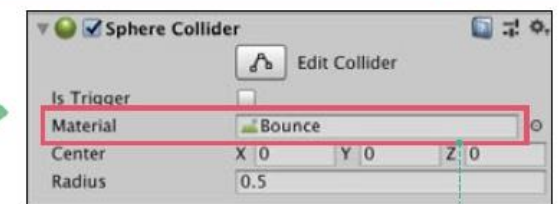
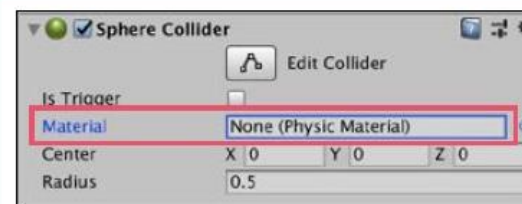
단계 4 Ball에 Bounce 어태치하기



1 Ball 클릭



2 Bounce를 Sphere Collider의 Material로 드래그 앤 드롭

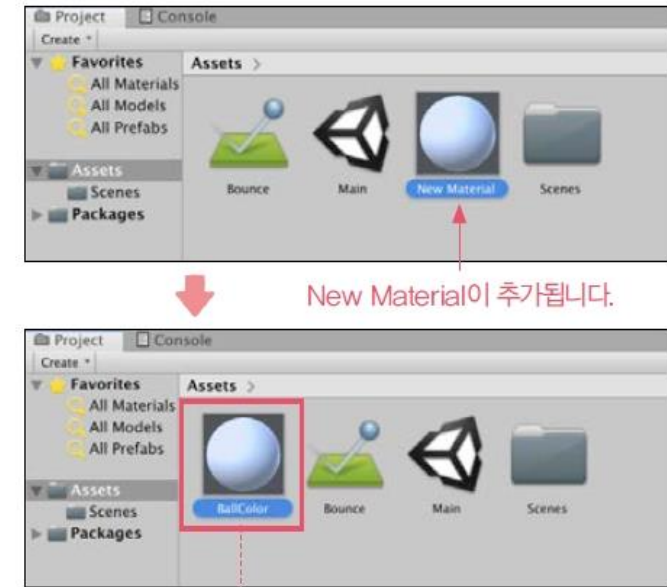
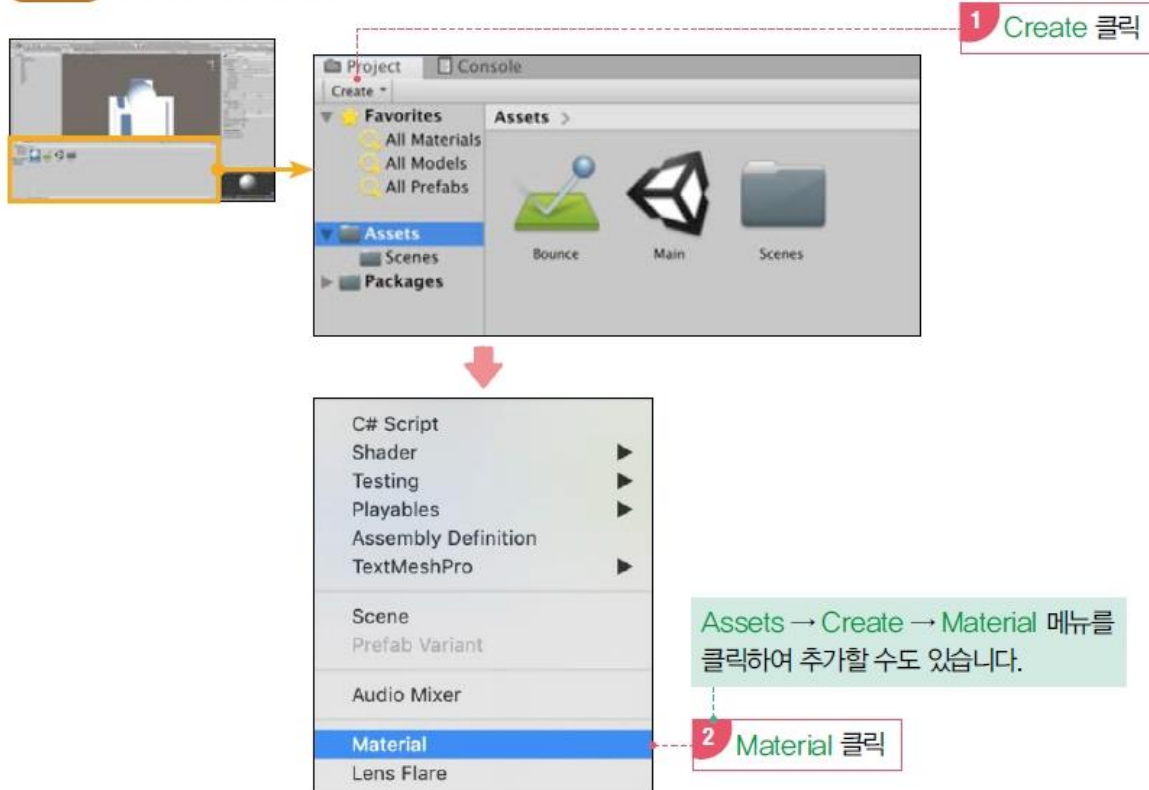


Bounce를 Material(None으로 표시되어 있는 부분)로 드래그 앤 드롭합니다. 오브젝트에 Physic Material을 적용하려면 이처럼 Material에 설정합니다.

게임 발전시키기

- 공의 색 변경하기: 오브젝트의 외관을 변경하려면 머티리얼(Material)을 만들어서 이를 오브젝트에 어태치
- Ball의 색을 빨간색으로 변경하기: 프로젝트 창에 Material을 추가하고 오브젝트로 드래그 앤 드롭하여 어태치

단계 1 Material 추가하기



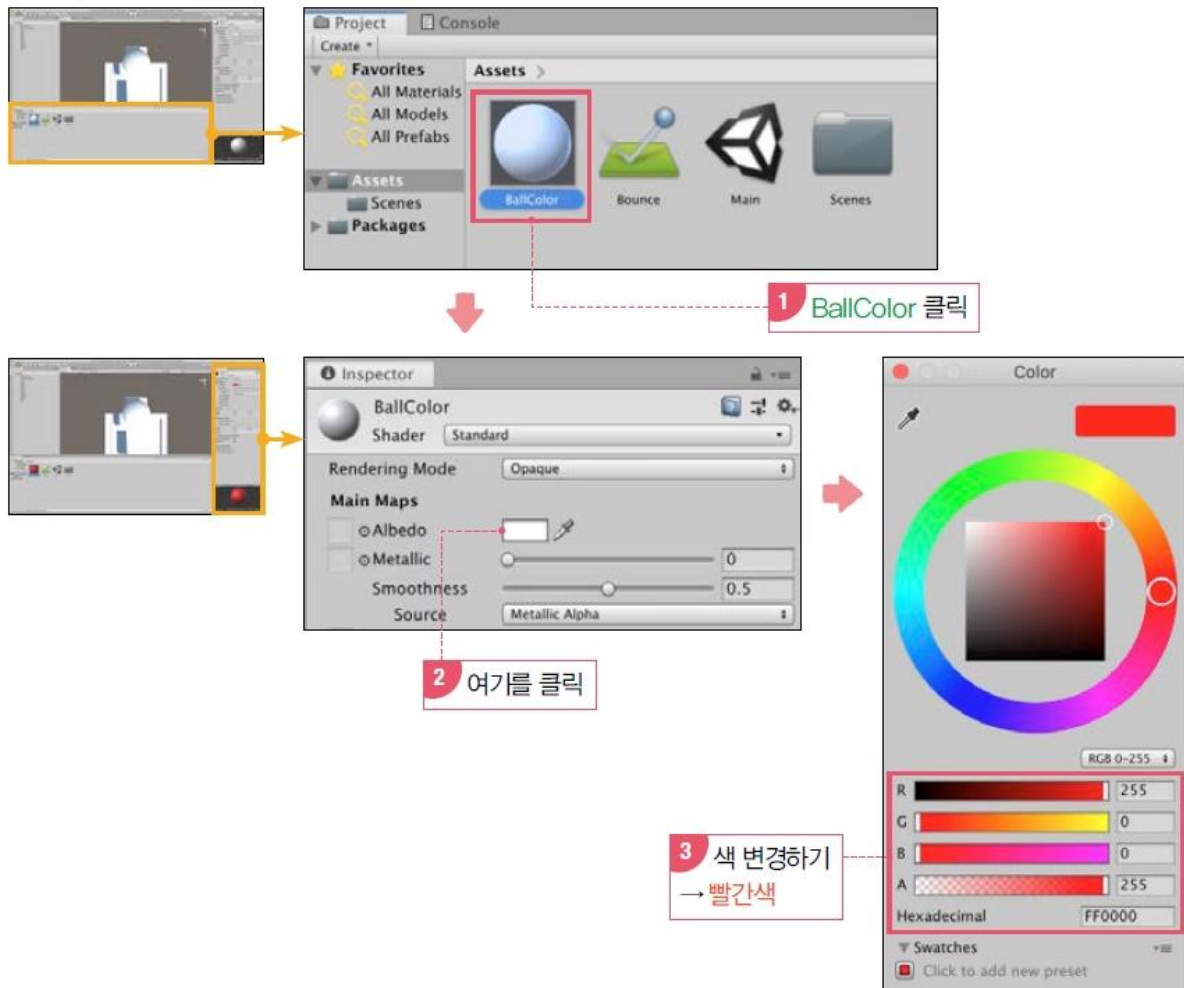
Material은 프로젝트의 Assets 폴더에 만들어집니다.

3 New Material의 이름 변경하기 → BallColor

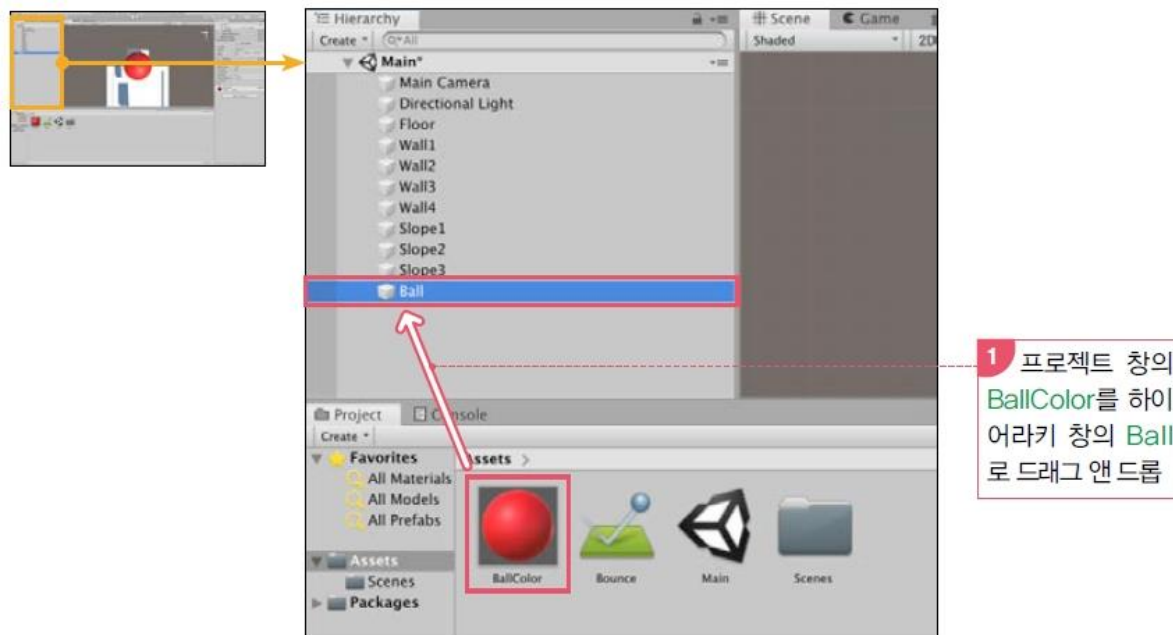
게임 발전시키기

- 단계 2 BallColor의 색 설정하기 / 단계 3 Ball에 BallColor 어태치하기

단계 2 BallColor의 색 설정하기

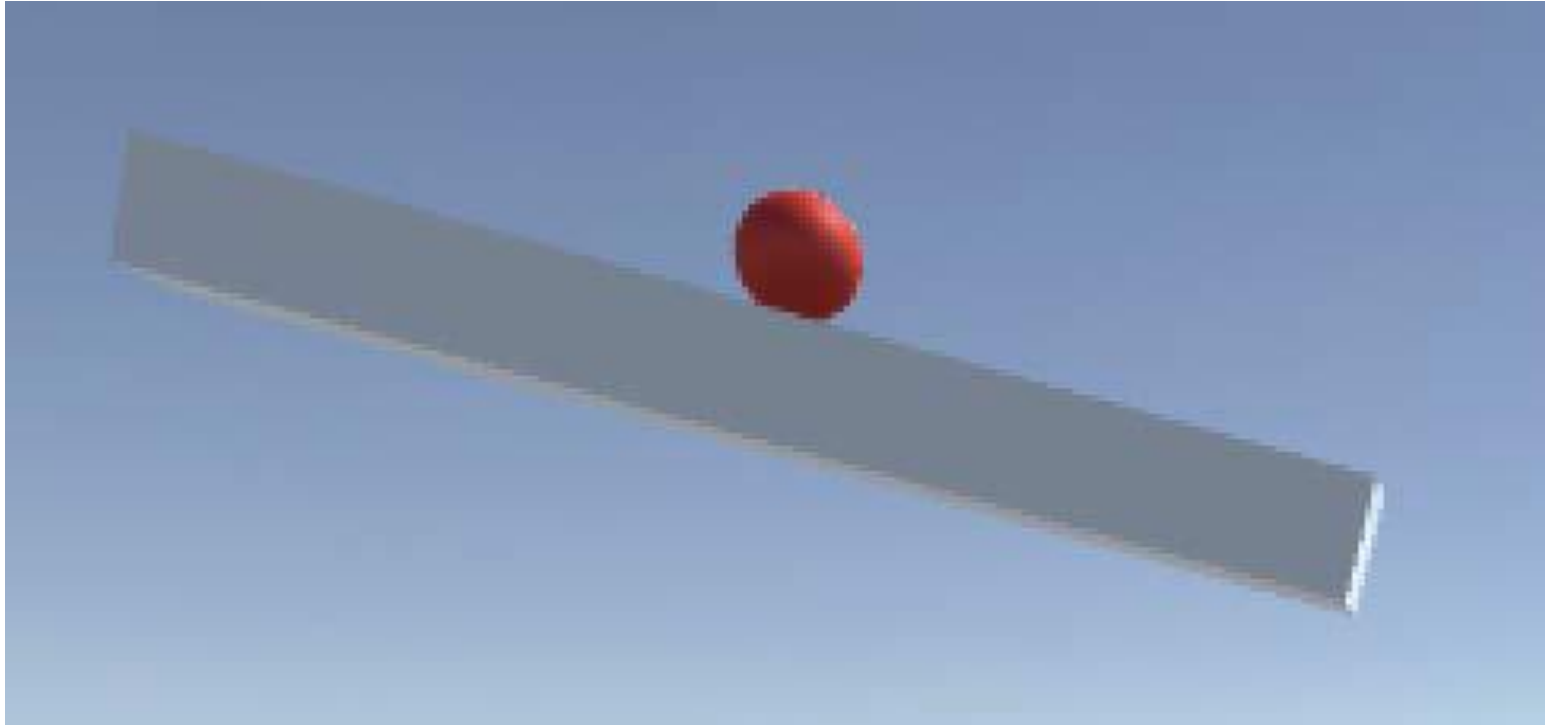


단계 3 Ball에 BallColor 어태치하기



게임 발전시키기

- 공 색깔 변경 완료
- 머티리얼은 색뿐만 아니라 Metallic 등 질감도 설정할 수 있음



게임 발전시키기

- 최종 완성된 모습

