

3D콘텐츠 이론 및 활용

유길상

3주. 유니티 컴포넌트

1,2교시 : 컴포넌트 활용

3,4교시 : 스크립트의 구성

학습 목표

- 매트리얼을 생성하고 활용할 수 있다.
- 오브젝트 배치하고 다양한 속성을 변경할 수 있다.
- 오브젝트에 물리 동작 컴포넌트를 추가할 수 있다.
- 중력 및 바운스 처리를 통해 물체의 움직임을 부여할 수 있다.
- 프리팹을 이해하고 활용할 수 있다.

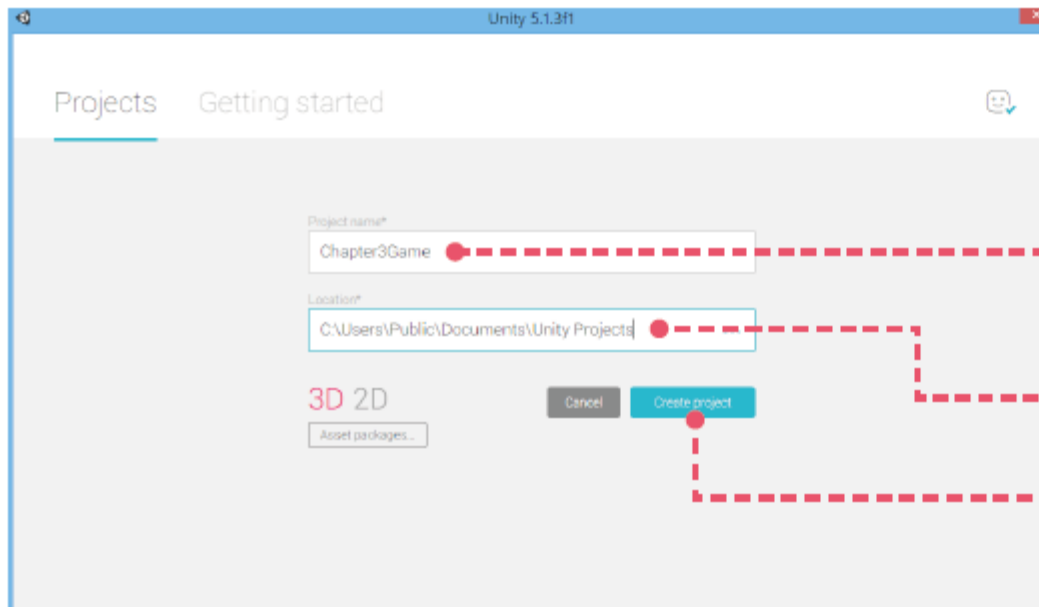
학습 내용

- 오브젝트 배치 실습
- 매트리얼 만들기
- 강체 [Rigid body]의 이해와 활용
- Physic Material 처리
- 프리팹 생성 및 활용

1. 오브젝트를 배치하고 위치 변경하기

1) 새 프로젝트 만들기

- 프로젝트 이름 설정



프로젝트 이름은 알파벳으로 설정하길 추천합니다. 또한 나머지 설정은 기본 값으로 진행합니다.

1 프로젝트 이름 입력하기
→ Chapter3Game

2 저장 경로 지정하기(임의의 폴더)

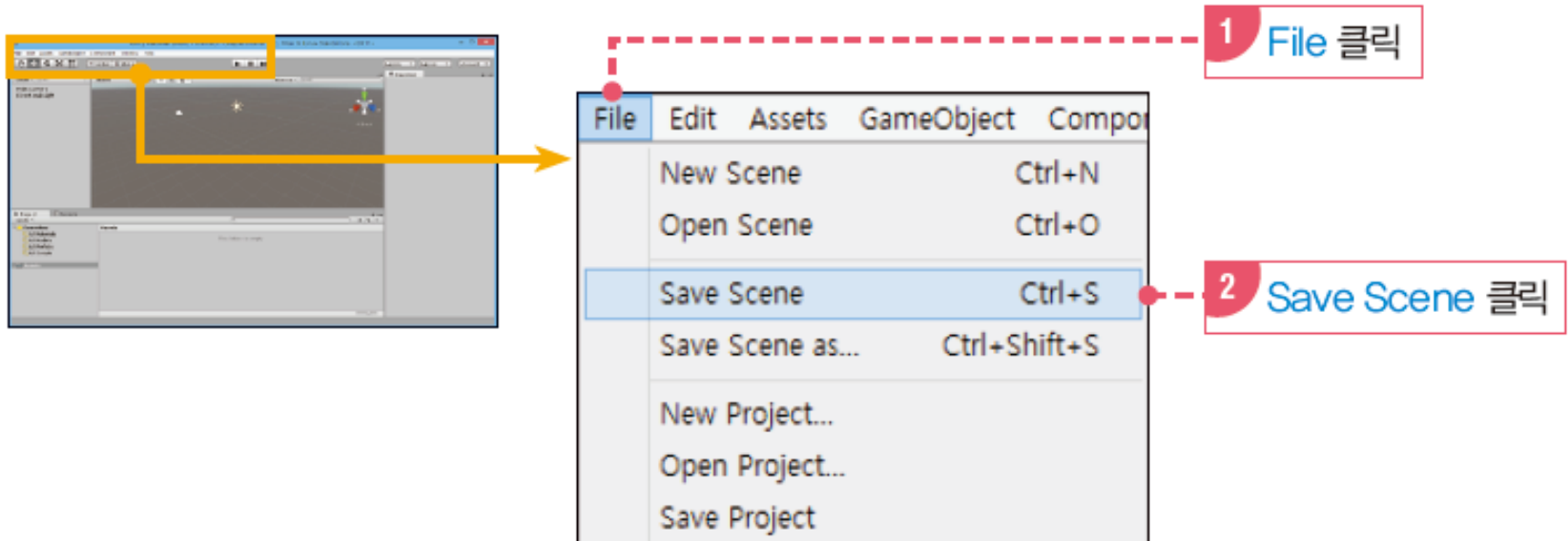
3 Create project 클릭

- 기본 오브젝트 - Hierarchy 뷰와 Scene 뷰에 Main camera와 Directional Light가 한 개씩 배치된 상태

1. 오브젝트를 배치하고 위치 변경하기

2) 씬 추가하기

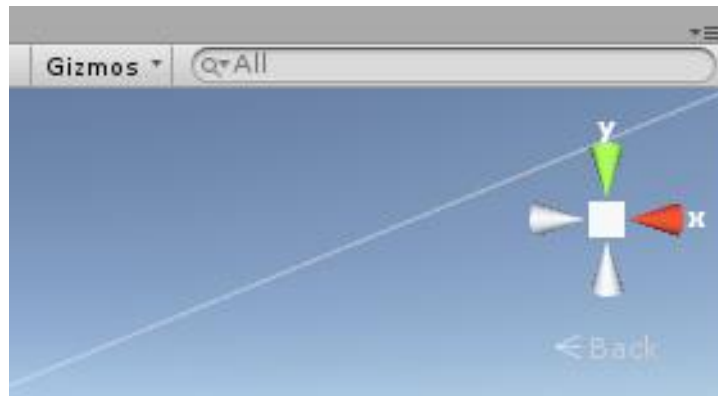
- 유니티의 게임 화면은 씬 단위로 관리
- 씬 이름 입력해 저장
- Project > Asset 에 씬 추가



1. 오브젝트를 배치하고 위치 변경하기

3) 바닥과 벽 만들기

- 시점 확인하기
 - 오브젝트 배치 전 Hierarchy 뷰의 Main Camera 속성값 확인
 - Hierarchy 뷰의 Main Camera 클릭-> Inspector 뷰 확인
 - Transform의 Position이 X:0, Y:0, Z:0으로 되어 있는지 확인
- 씬의 방향 확인과 변경
 - Scene 뷰의 오른쪽 상단에 있는 씬 기즈모의 원뿔 부분 클릭
 - X의 정방향향이 오른쪽, Y의 정방향향이 위쪽에



1. 오브젝트를 배치하고 위치 변경하기

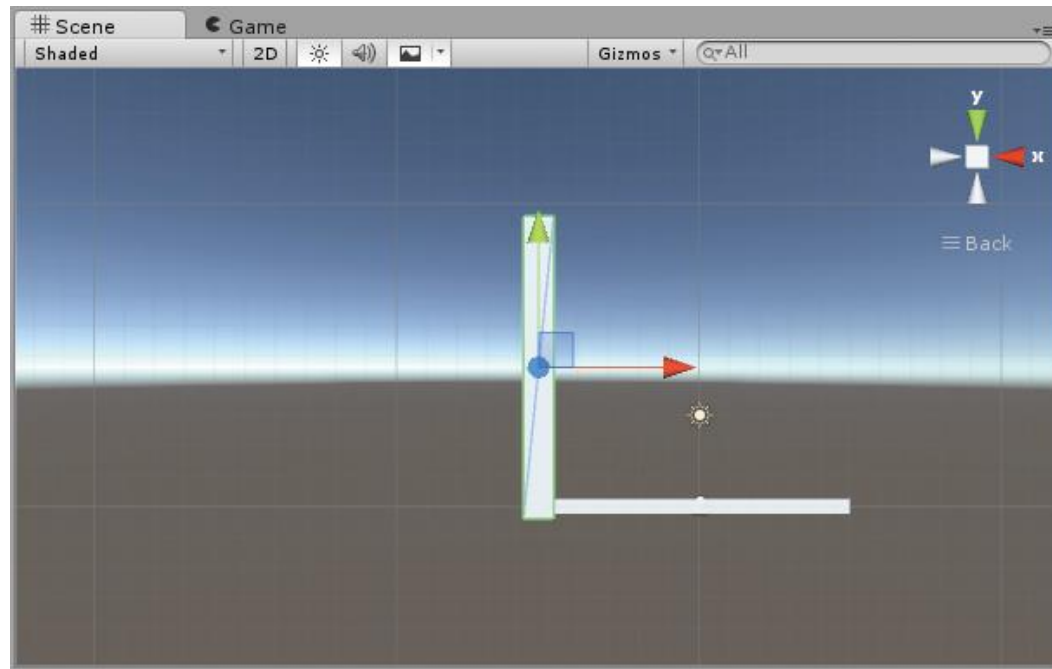
4) 바닥 만들기

- 유니티의 기본 도형 중 Cube 이용
- Create Object > 3D Object > Cube
 - 위치는 X:0, Y:0, Z:0
- 이름은 다른 오브젝트와 구별하기 쉽게 설정 (ex) Floor)
- 크기 (Scale) 을 용도에 맞게 변경
- Scene 뷰 통해 Floor 생성 확인
 - 나타나지 않으면 Hierarchy 뷰 더블 클릭해 수정

1. 오브젝트를 배치하고 위치 변경하기

5) 벽 만들기

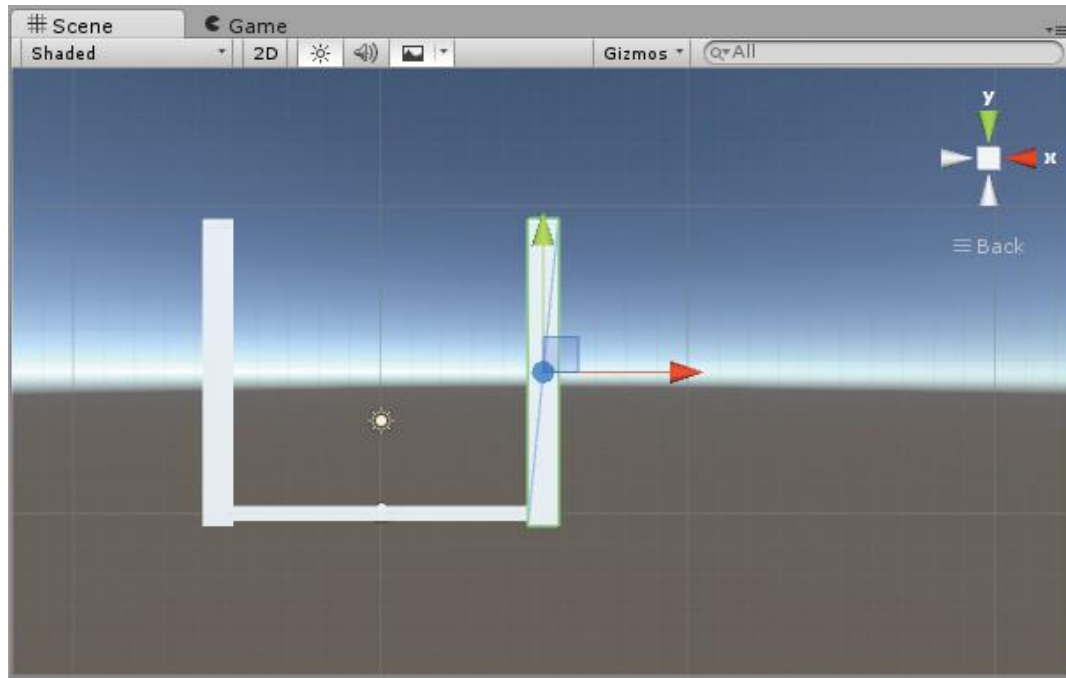
- 벽도 유니티의 기본 도형 중 Cube 이용
- 추가 후 Wall1 이름으로 저장
- 오브젝트의 위치와 크기 변경
- Scene 뷰 확인 시 씬 기즈모로 방향 맞춰줄 것



1. 오브젝트를 배치하고 위치 변경하기

6) 반대쪽 벽 만들기

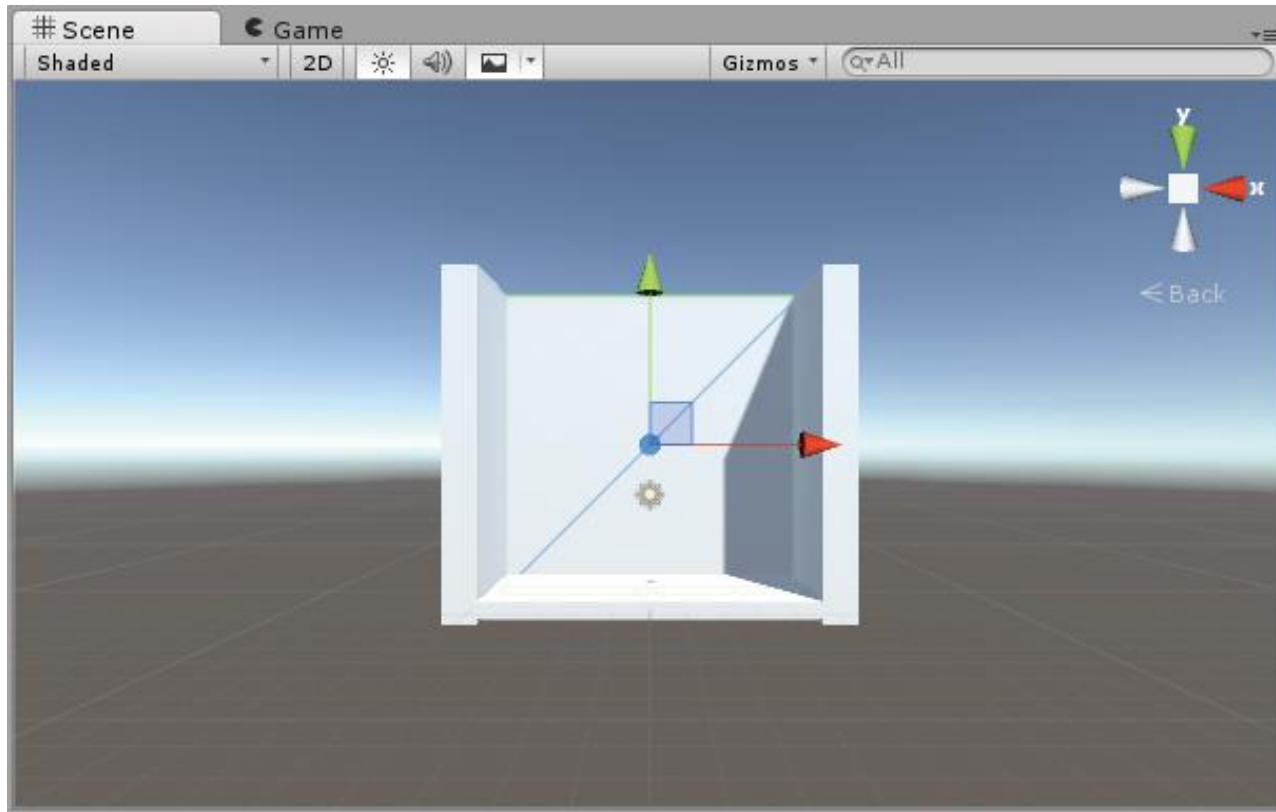
- 크기와 모양이 같은 오브젝트는 Duplicate로 복제
 - Hierarchy 뷰에서 복제할 오브젝트 우클릭해 사용
- 복제 후 Wall2의 위치 변경
 - 좌우벽 완성



1. 오브젝트를 배치하고 위치 변경하기

7) 안쪽 벽 만들기

- Cube 이용해 새 오브젝트 만들기
- Wall3의 위치와 크기 변경하기



1. 오브젝트를 배치하고 위치 변경하기

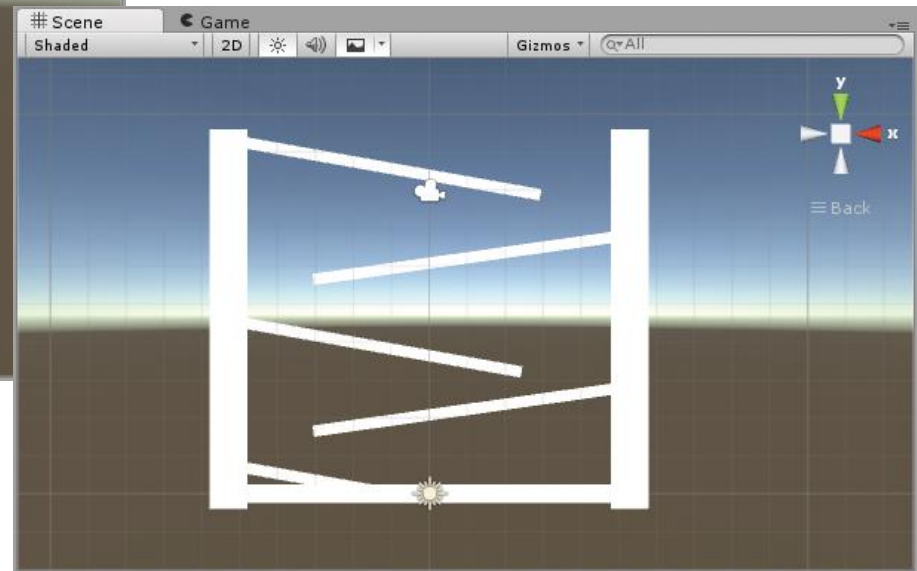
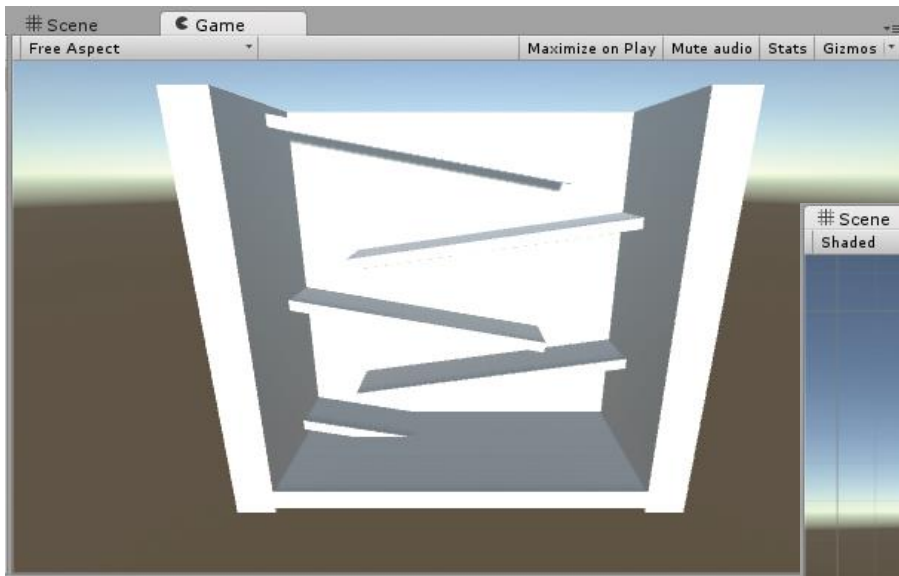
8) 경사면 만들기

- Slope 1 도 Cube 이용해 제작
 - 바닥과 벽을 만드는 과정과 같음
 - Game 뷰 전환해 결과 확인
- Slope 2 는 1을 복제해 사용 (모양과 크기가 같은 특성)
 - 위치를 조정해 게임 화면 구성에 다양함 줄 것
- Slope 3은 오른쪽 벽에 박힌 것처럼 조정
- Slope 추가
 - 같은 방향의 전 단계 오브젝트 복제해 사용

1. 라이트 설정

2) 경사면 만들기

- Game 뷰와 Scene 뷰의 비교



1. 라이트 설정

3) 경사면을 굴러갈 공 만들기

- 기본 도형 중 Sphere 이용
- 화면에 공 배치하기
 - 첫 번째 경사면 보다 조금 위에 위치하도록
 - 게임이 시작되면 튕 떨어지는 연출 위함
- 플레이 버튼을 눌러도 공이 굴러가지 않음
 - 지금의 뷰에는 중력이 존재하지 않기 때문
 - 이외에 마찰, 반발 등도 구현 가능

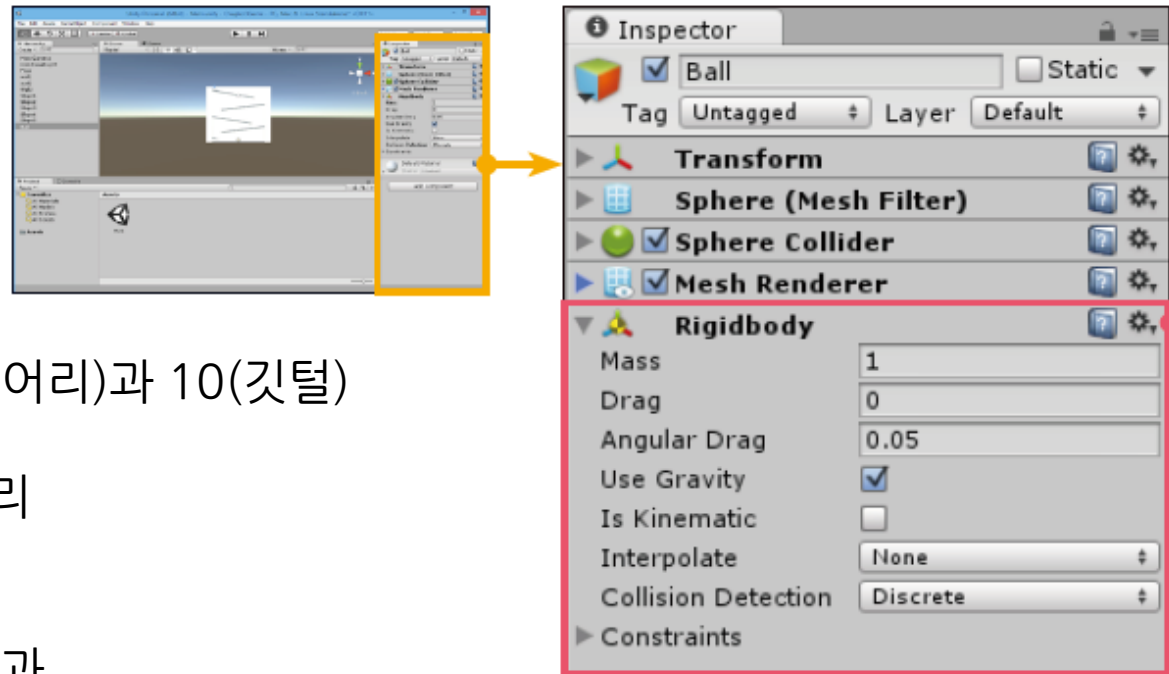
2. 카메라 시점 변경하기

- Scene 뷰에서 작업한 결과물이 실제 게임에서 어떻게 보이는 지 확인
- Game 뷰 전환 - 제대로 보이지 않으면 카메라 위치 변경
- Camera Preview
 - Hierarchy 뷰에서 카메라 오브젝트 선택하면 볼 수 있음
 - 기즈모의 카메라에 체크 된 경우
 - 카메라 위치 변경할 때 유용
- **Ctrl + Shift + F** 를 통해 카메라 뷰를 쉽게 변경할 수 있음.

3. 물리동작 설정하기

1) 공에 중력 적용하기

- Hierarchy 뷰에 추가한 오브젝트에 리지드 바디 설정
 - 오브젝트 선택 후 Component > Physics > Rigidbody
- Inspector 뷰에서도 추가 가능

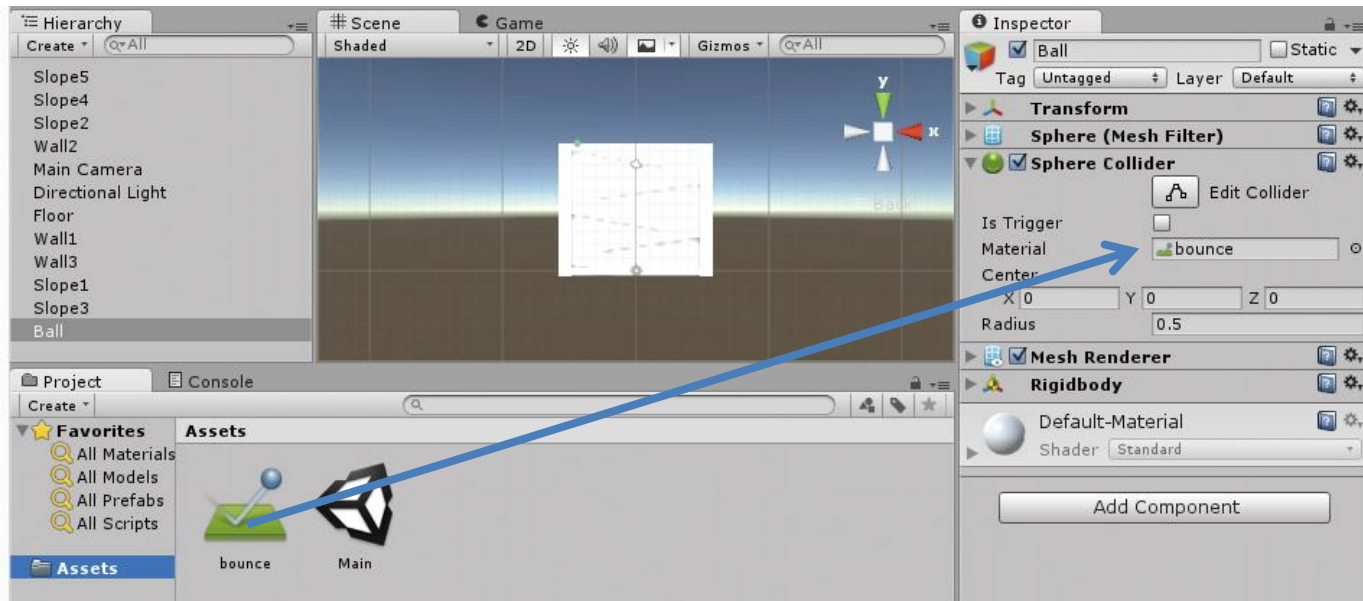


- 질량
- 저항력
 - .001(금속 덩어리)과 10(깃털)
- 회전 저항력
- 스크립트로 처리
- 부드럽게
- 충돌감지
- XYZ축 멈춤 효과

3. 물리동작 설정하기

2) 공의 움직임 변경하기

- 공이 힘차게 굴러가게 보이도록 해야
 - 공과 바닥 사이의 마찰을 줄이면 OK
- Project 뷰에서 Physic Material 추가 후 오브젝트에 적용
 - 드래그 앤 드롭으로 편리하게 사용



3. 물리동작 설정하기

Physic Material 바운스 설정

Dynamic Friction	움직일 때 사용되는 마찰. (동적 마찰력) 0 ~ 1까지의 실수를 넣을 수 있고, 0에 가까울수록 얼음과 같고 1에 가까울 수록 지면과의 마찰력이 높아 짐.
Static Friction	물체가 표면위에 놓여 있을 때 사용되는 마찰. (정적 마찰력) 0 ~ 1까지의 실수를 넣을 수 있고, 0의 값은 얼음과 같고, 1의 값은 물체를 움직이지 못하게 함.
Bouncyness	물체의 탄력성을 적용. 0 ~ 1까지의 실수를 넣을 수 있고, 0은 탄력성이 없고, 1에 가까울 수록 탄력성이 높아짐. 즉 1로 셋팅할 시 물체는 영원히 튕기게 됨.
Friction Combine Mode	두 개의 충돌체에 마찰력을 어떻게 결합시킬 것인지 여부 셋팅 Average : 두 충돌체의 마찰력을 동등하게 부여 Min : 두 충돌체의 마찰력 값중 작은 값을 사용. Max : 두 충돌체의 마찰력 값중 높은 값을 사용. Multiply : 두 충돌체의 마찰력이 서로 곱해짐.
Bounce Combine	두 충돌체에 탄력을 어떻게 결합시킬 것인지 여부 셋팅 Average : 두 충돌체의 마찰력을 동등하게 부여 Min : 두 충돌체의 마찰력 값중 작은 값을 사용. Max : 두 충돌체의 마찰력 값중 높은 값을 사용. Multiply : 두 충돌체의 마찰력이 서로 곱해짐.
Friction Direction 2	특정 방향으로 마찰력을 다르게 지정하게 싶을 때 방향 지정
Dynamic Friction 2	추가 된 방향에 대해 별도의 운동 마찰력을 지정
Static Friction 2	추가 된 방향에 대해 별도의 정지 마찰력을 지정

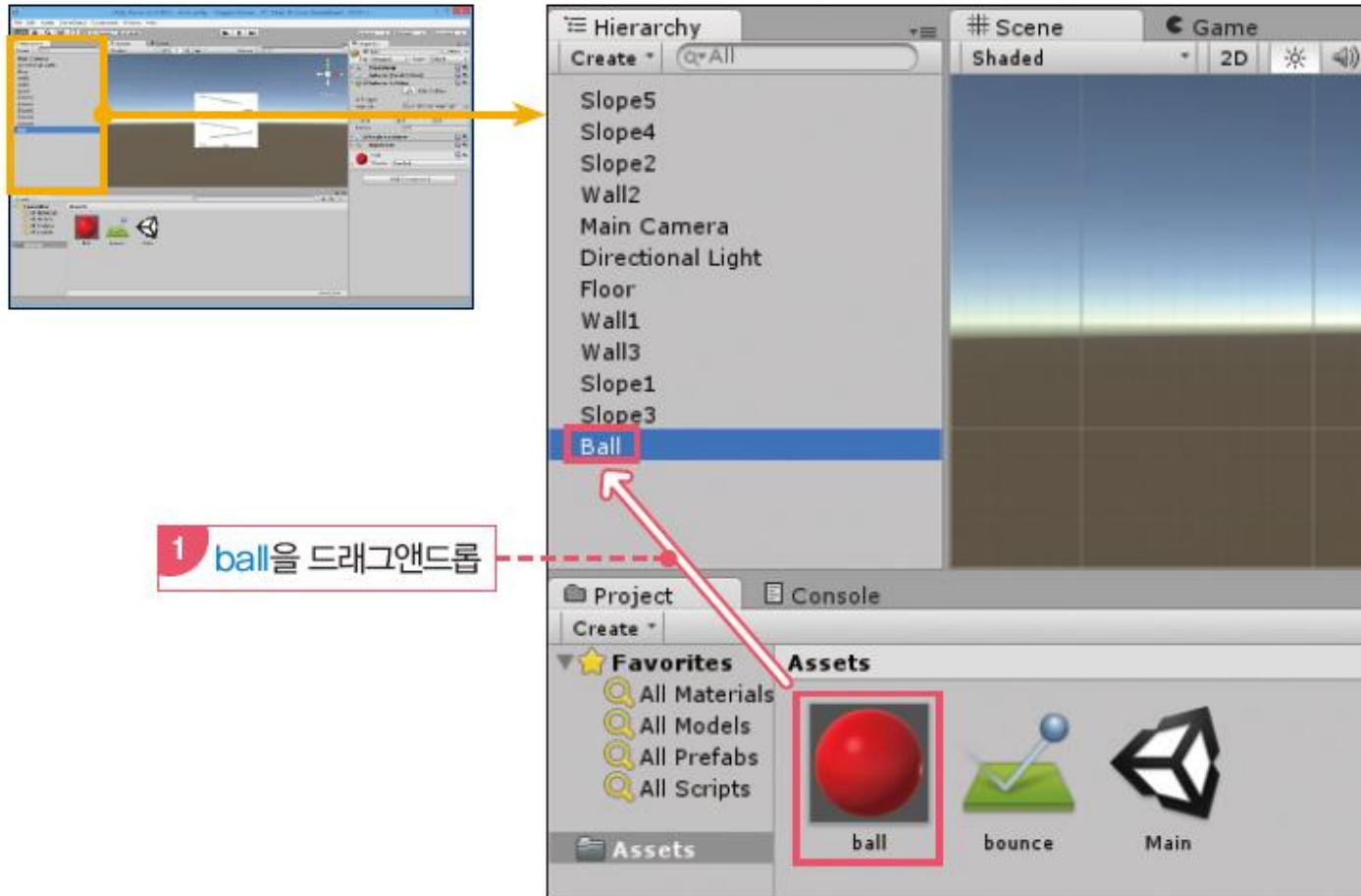
4. 공 색상 변경 (매터리얼 생성 및 응용)

공의 색 변경하기

- 공이 눈에 잘 띄도록 색깔 변경
 - Project 뷰에서 Material 추가
 - 알아보기 쉽도록 이름 변경과 색깔 선택
- Project 뷰에 추가한 매트리어얼
 - Hierarchy 뷰의 오브젝트에 드래그 앤 드롭해 사용
 - 매티리얼을 변경하면 매티리얼을 적용한 오브젝트도 변경

4. 공 색상 변경

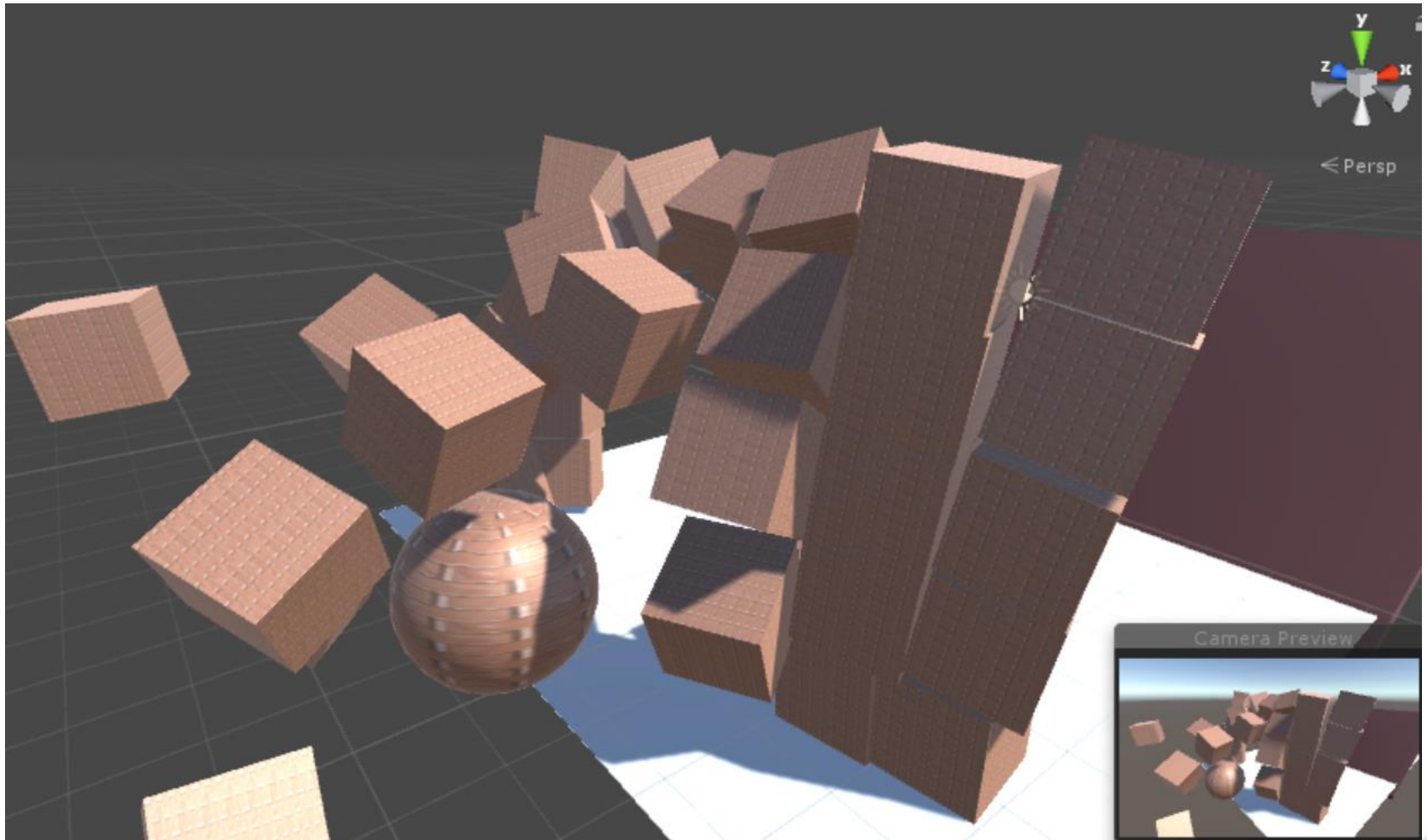
공의 색 변경하기



5. 유니티 활용 팁

- 1) 카메라 각도 쉽게 셋팅하기
- 2) 플레이중에는 객체의 속성값을 변경하여도 반영되지 않는다.
- 3) 단축키가 작동하지 않을 때에는 한/영키 상태를 확인한다.
- 4) 유니티 클래스 이름은 파일 이름과 동일하게 주어야 한다.

6. 실습



3D콘텐츠 이론 및 활용

유길상

3주. 유니티 컴포넌트

1,2교시 : 컴포넌트 활용

3,4교시 : 스크립트의 구성

학습 목표

- 스크립트의 역할에 대하여 이해한다.
- 스크립트를 작성하기 위한 컴포넌트 구성을 이해하고 사용할 수 있다.
- 스크립트의 구성(클래스, 함수, 메서드)에 대하여 이해한다.

학습 내용

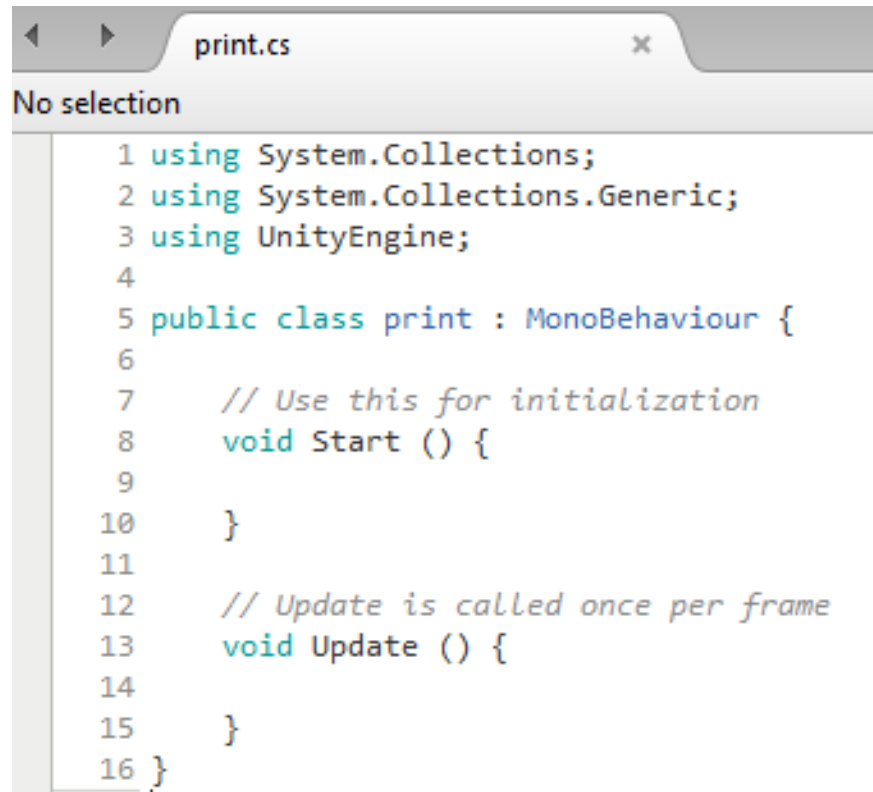
- 스크립트 컴포넌트 이해
- 스크립트 클래스의 구성
- 모노디벨롭 편집기
- 스크립트로 간단한 오브젝트 움직이기 실습

1. 스크립트 컴포넌트

1) 의미

- 월드에 있는 오브젝트를 조작하거나,
- 오브젝트에 동작(행위)을 주기 위한 언어

2) 구성



```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class print : 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 }
  
```

1. 스크립트 컴포넌트

3) 컴파일(Compile)

- F8 (Build)
- Ctrl + F8 (ReBuild)

4) 용어

- 클래스(class)
- 함수(function), 메서드(method), 속성(Properties, Attribute)
- 모노디벨롭(Monodevelop)
- 모노비헤이비어(Monobehaviour)
- 디버그(Debug), 릴리스(Release)
- 콘솔(Console)
- 객체지향 프로그래밍(Object Oriented Programming)

2. 스크립트 작성 기초

1) 결과 확인하기

- Print 함수
- 키보드 입력 시 처리 예

```
void Update () {  
    if (Input.GetKeyDown (KeyCode.LeftArrow)) {  
        transform.Translate (-1, 0, 0);  
    }  
}
```

3주 종합 실습

1) 포탑 만들기

참고: <http://shop.youngjin.com/goods/view?no=100006656>

- 매트리어얼 제작 및 활용
- 프리펩 제작 및 활용
- 도구활용
 - 오브젝트 이동
 - 복제
 - 회전(Rotation)
 - 빈 오브젝트 활용 계층화



부분별 프리펩 제작

