

Week2

B9.

1. 중력 적용하기

Add component 에서 rigidBody 추가. -> 물리효과를 받기 위한 컴포넌트

2. 충돌영역 정하기

Collider : 충돌영역

Radius : 반지름

3. rigidbody 설정

Mass 무게. 중력은 같다. 수치가 높을수록 충돌이 무거워진다.

Use gravity : 중력을 받는지 결정. 보통 체크되어있음

Is kinematic : 외부 물리 효과를 무시. script로만 움직임! -> 움직이는 함정 만들 때 유용

4. 재질 만들기

Material : 재질. 오브젝트의 표면재질을 결정.

만드는 법 : assets에서 new material만들어서 객체에 넣는다.

Albedo : 색, 무늬

Metalic : 금속 재질 수치

Smoothness : 빛 반사 수치

Emission : 텍스처 발광(밝기) 조절

Tiling : 텍스처 반복 타일 개수

4요소 : Mesh, Collider, RigidBody, Material

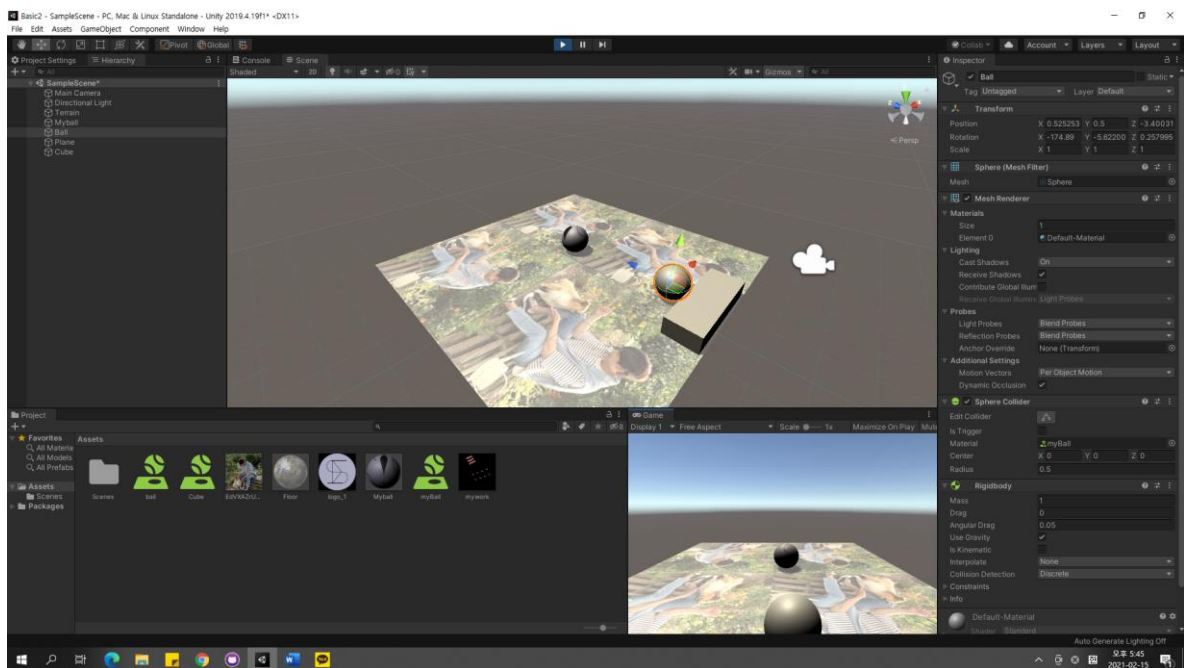
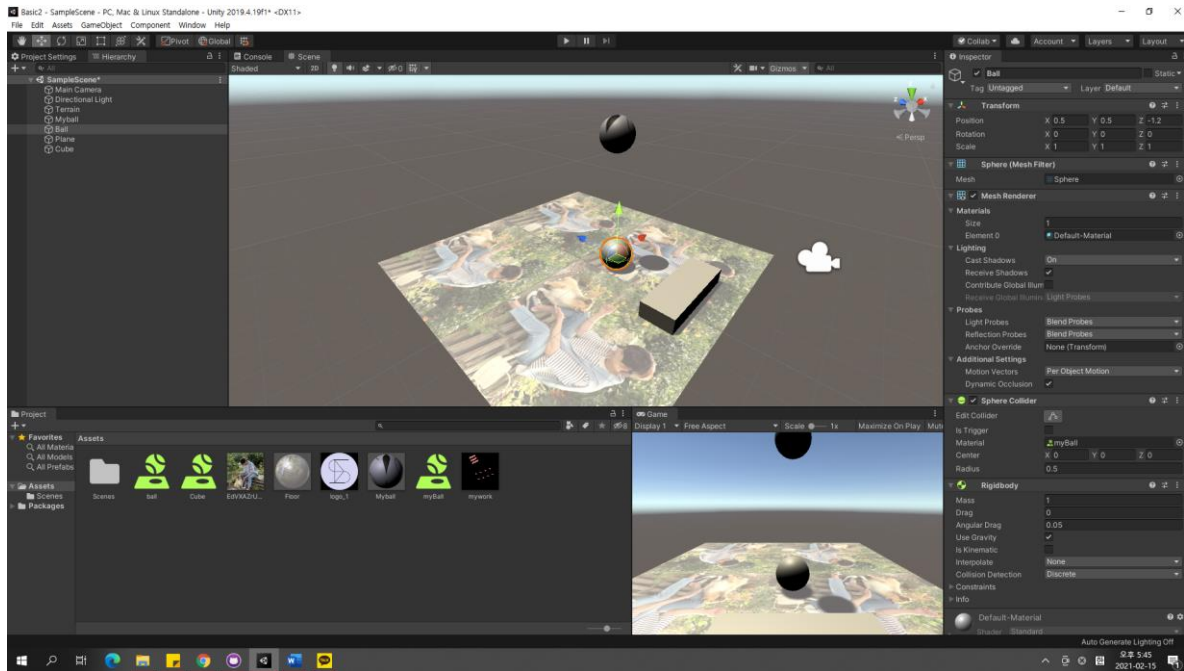
5. 물리재질 : physics material : 탄성과 마찰을 다루는 물리적인 재질 -> collider에 들어간다.

Bounciness : 탄성력, 높을수록 많이 튀어오름

Bounciness combine : 다음 탄성을 계산하는 방식. maximum으로!

Friction : 마찰력. 낮을수록 많이 미끄러짐

Friction combine : 다음 마찰력을 계산하는 방식 -> 마찰력 0하려면 minimum으로!



B10 힘을 이용하여 물체 움직여보기

1. 컴포넌트 가져오기

2. 속도 올리기

Velocity : 현재 이동속도.

RigidBody 관련 코드는 FixedUpdate에 작성!

3. 힘으로 밀기

AddForce(Vec) : vec의 방향과 크기로 힘을 줌

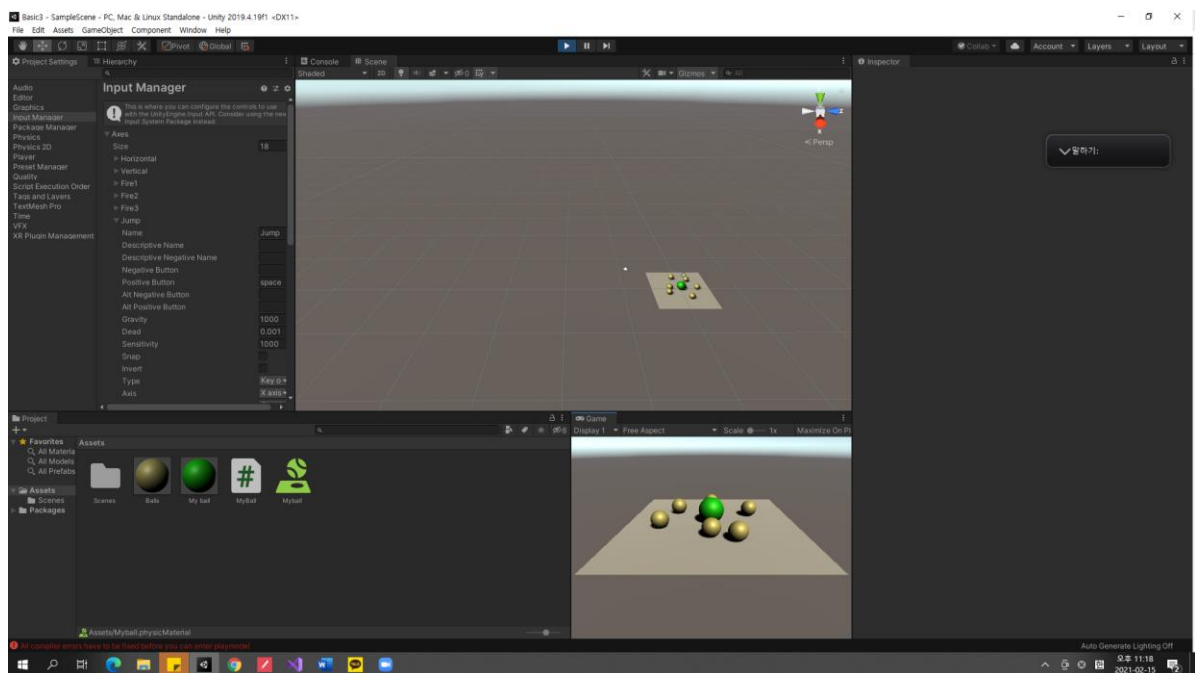
->fixedupdate() => 의 힘 방향으로 계속 velocity 증가

ForceMode : 힘을 주는 방식

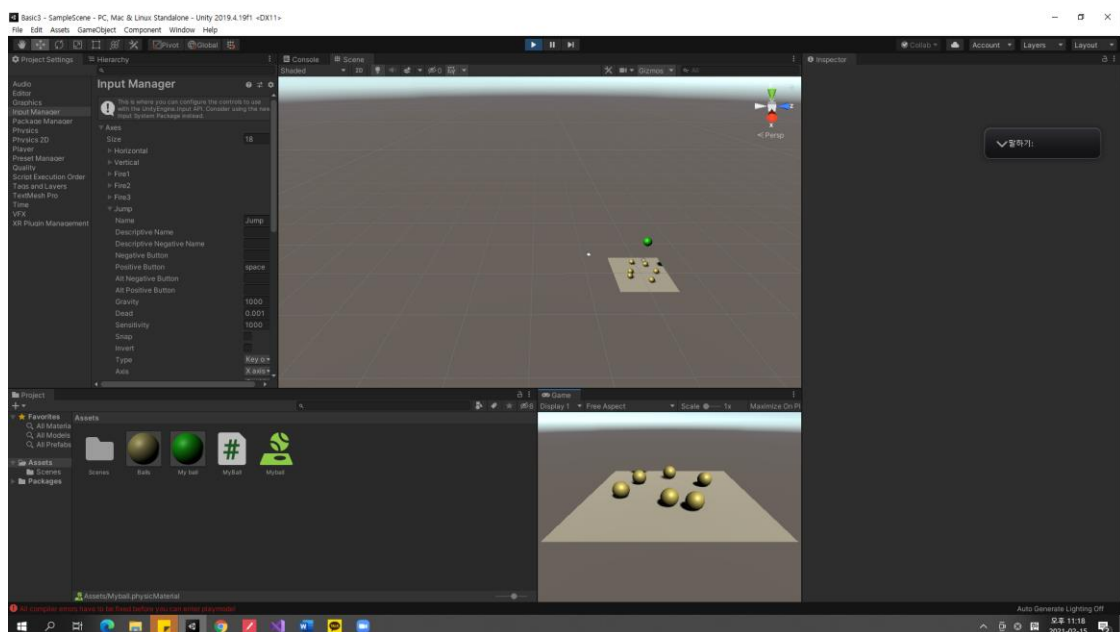
4. 회전력

AddTorque(Vec) : Vec 방향을 축으로 회전력이 생김

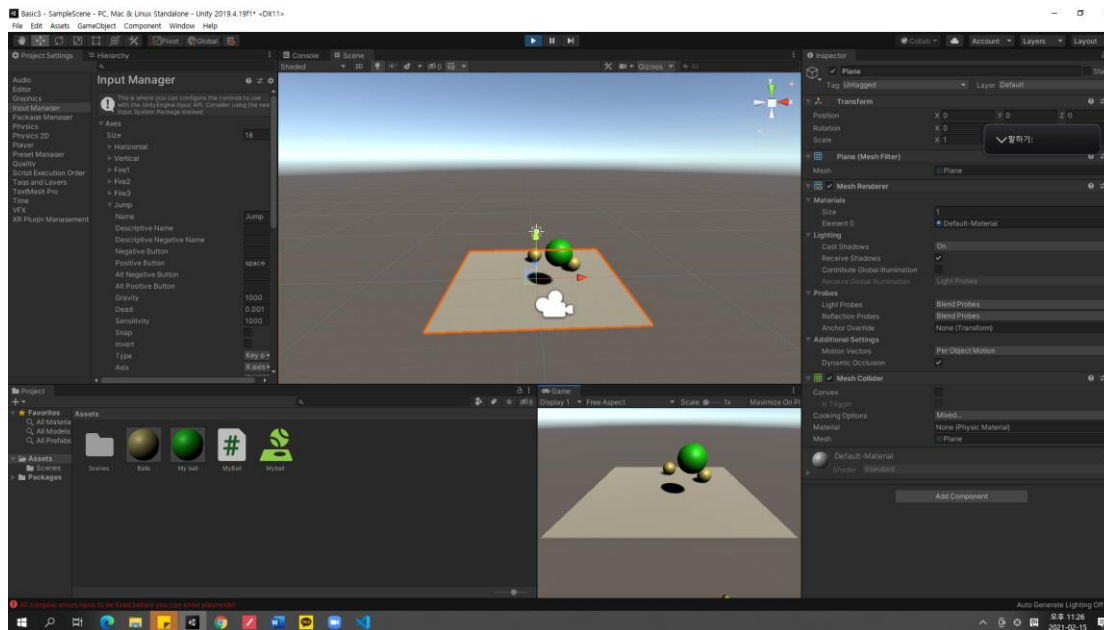
가만히 있는 상태



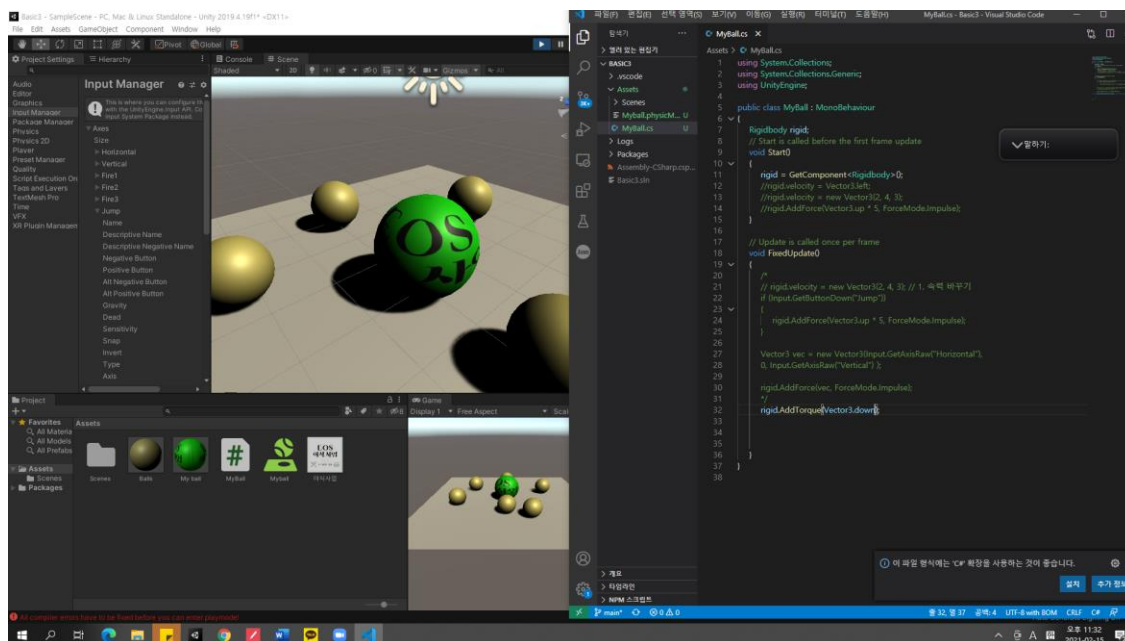
속도 올리기



밀기



회전 & 코드



B11 물리 충돌 이벤트

1. 물리 충돌 이벤트

오브젝트의 재질 접근 : MeshRenderer 통해서

OnCollisionEnter : 물리적 충돌이 시작될 때 호출되는 함수

OnCollisionStay : 충돌중

OnCollisionExit : 충돌후

Color : 기본 색상 클래스, Color32 : 255 색상 클래스

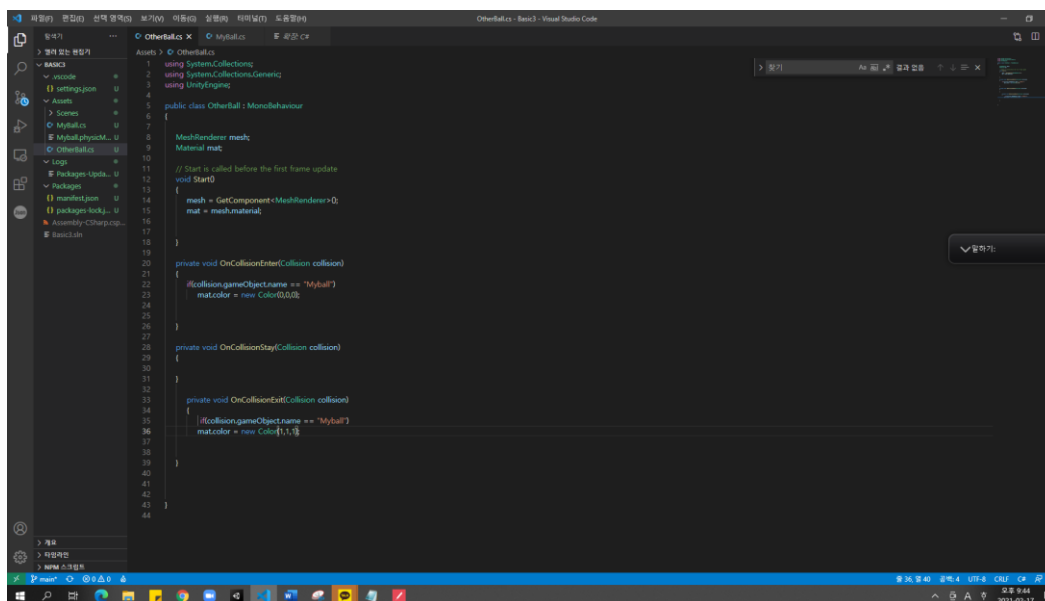
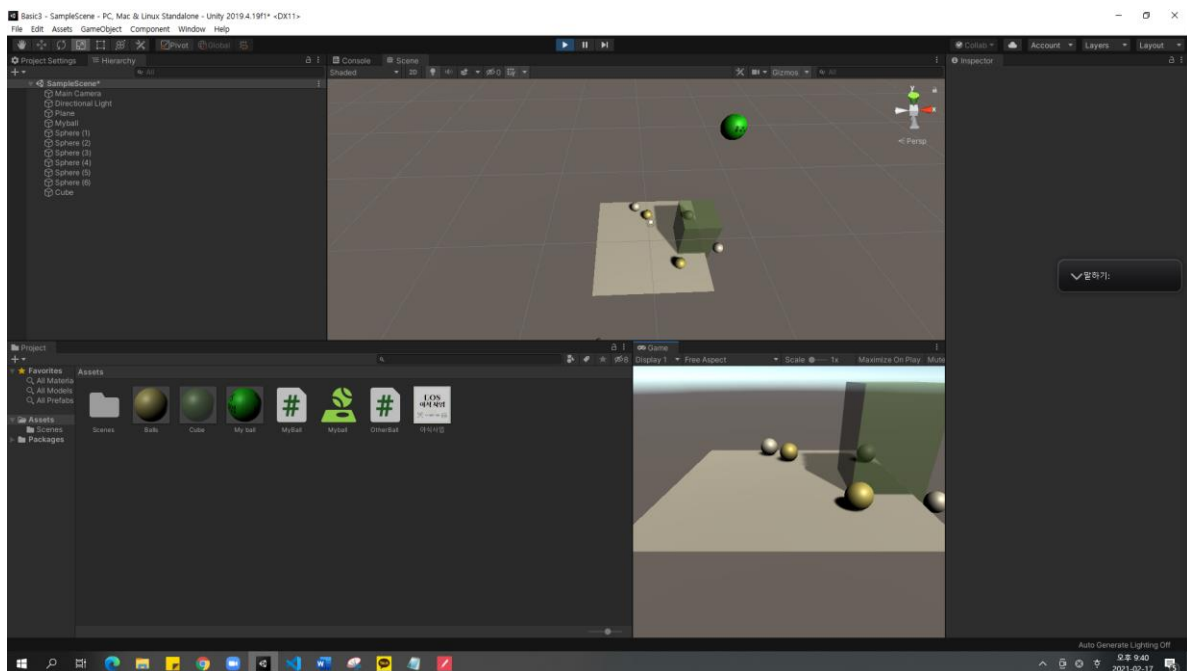
Collision은 충돌 정보

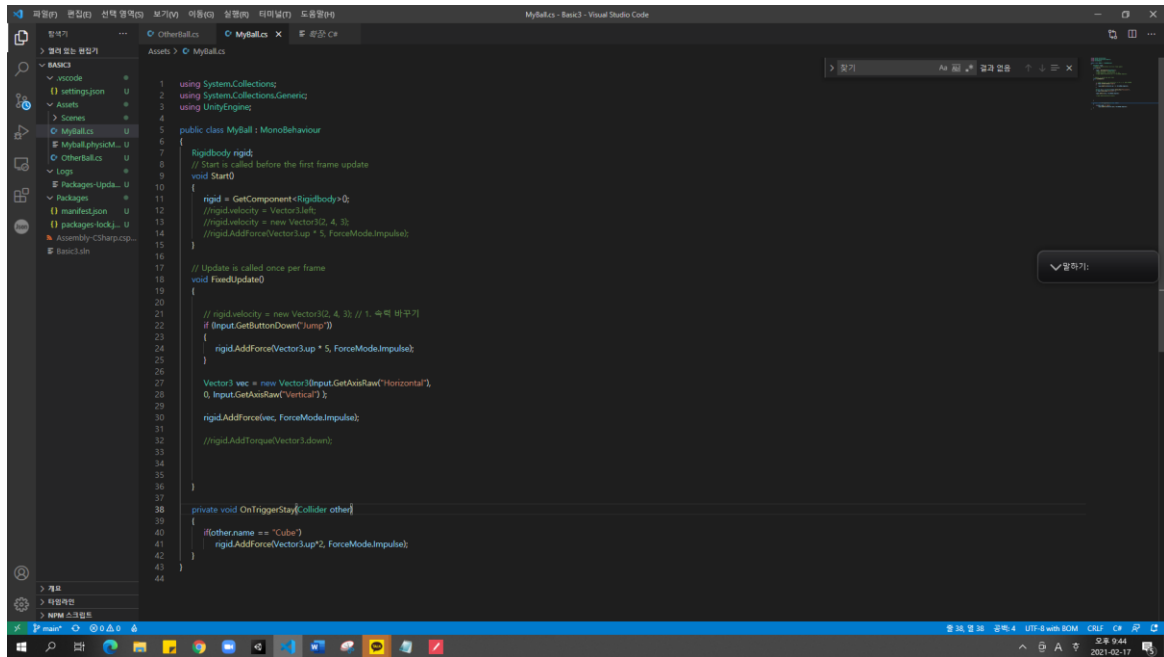
2. 트리거 이벤트

TriggerEnter : 콜라이더가 충돌하기 시작했을 때

TriggerStay : 콜라이더가 계속 충돌하고 있을 때

TriggerExit : 충돌이 끝났을 때





B11. UGUI 기초

Canvas 캔버스 : ui가 그려지는 도화지 역할.

스크린 : 게임이 표시되는 화면, 해상도로 크기 결정

World(카메라 o), 스크린(카메라 x)

Text : 문자열

Image : 그림파일을 sprite로 설정해야 ui 적용 가능

비율 맞추기 : preserve aspect, set native size 실제 크기.

Image type : simple 가로세로 그냥 늘어나. Sliced 이미지를 잘라서 모서리? 가운데를 채운다?, tiled : 타일놓는거. 이미지 크기 고정. Transform 크기에 맞춰서 여러 개 넣어.

Filled : 어떻게 채워지나? Radial 360. 180 80 90 가로 세로 사진 두개 색 변화로 쿨타임 효과 구현 가능!

UI는 Hierarchy의 순서에 따라 위 아래 결정

Sliced 기능으로 자유자재 크기로 이미지 구현 가능

버튼 : Interactable 반응을 한다. Transition : 반응을 어떻게 할것인가.

Color tint : highlighted 마우스를 올렸을 때 pressed 눌렀을 때 disabled 비활성화

Color multiplier 색 섞는 감도

Navigation : 탭을 눌렀을 때 어떻게 갈건지.

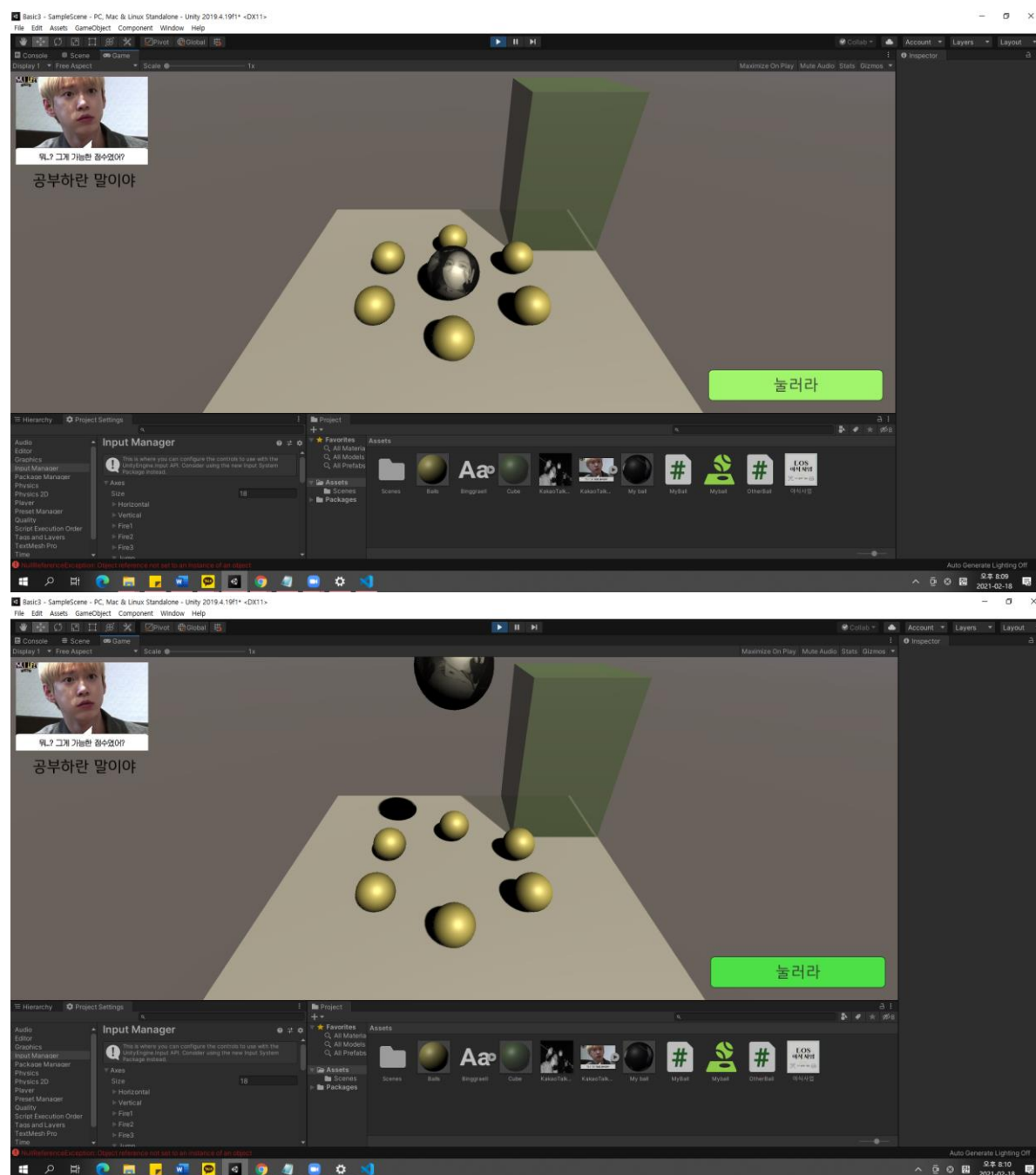
OnClick 버튼 클릭시 호출되는 이벤트 함수

버튼 클릭은 누리기 떼기 한 세트 동작

앵커 : 어느 한곳을 잡아서 거리 유지

앵커-파란점 : 컴포넌트에서의 기준점 shift

앵커-중앙네보 : 컴포넌트 위치(alt)



BE1 3D게임 만들기

1. 게임 개발은 계획부터

2. 만드는 중

Roate(Vector3) 매개변수 기준으로 회전

아이템 먹히는거 SetActive(bool) 오브젝트 활성화 함수

audioSource: 사운드 재생 컴포넌트, AudioClip필요

Tag : 오브젝트 구분

카메라 조절 : 공따라가기! LateUpdate()

```
Transform playerTransform;  
Vector3 Offset;  
  
// Start is called before the first frame update  
void Awake()  
{  
    playerTransform = GameObject.FindGameObjectWithTag("Player").transform  
;  
    Offset = transform.position - playerTransform.position;  
}  
  
// Update is called once per frame  
void LateUpdate()  
{  
    transform.position = playerTransform.position + Offset;  
}
```

결승선

형태가 없고 전반적인 로직을 가진 오브젝트를 매니저로 설정

Find 계열 함수는 부하를 초래할 수 있음

게임 restart : SceneManager.LoadScene("Example1_0");

스테이지 추가

LoadScence의 매개변수는 int도 가능

유저 인터페이스