**방향키 함수**if (Input.GetKey(KeyCode.LeftArrow) == true) ->right, up, down  
{transform.Translate(Vector3.left \* speed \* Time.deltaTime); ->right, forward, back }

물리적 힘  
private Rigidbody ballRd; == Rigidbody ballRd = GetComponent<Rigidbody>();  
ballRd.AddForce”(x값, y값, z값)”->”vector3 변수”;

**공 반사**

startPos = new Vector3(0, 0, 0);  
Vector3 currPos = collision.transform.position;

Vector3 incomVec = currPos - startPos;

Vector3 normalVec = collision.contacts[0].normal;

Vector3 reflectVec = Vector3.Reflect(incomVec, normalVec);

reflectVec = reflectVec.normalized;

ballRd.AddForce(reflectVec \* speed);

**충돌 이벤트 처리하기 -> 충돌한 두 오브젝트를 서로 밀어내는 함수**OnCollisionEnter(Collision collision): 충돌한 순간에 호출/OnCollisionStay(Collision collision): 충돌하고 있는 동안 호출/OnCollisionExit(Collision collision): 충돌 후 분리되는 순간에 호출  
collision.gameObject.CompareTag("BLOCK"), collision.gameObject.name == "Small"

**충돌한 두 오브젝트는 서로 통과하는 함수**OnTriggerEnter(Collision collision): 충돌한 순간에 호출/OnTriggerStay(Collision collision): 충돌하고 있는 동안 호출/OnTriggerExit(Collision collision): 충돌 후 분리되는 순간에 호출

마우스 입력->Input.GetMouseButtonDown(0)->0-left, 1-right, 2-middle

**프리팹과 랜덤**private float time = 0.0f;

public GameObject lifeaddPrefab;   
void Update()

{ time += Time.deltaTime;

if(time > span){ time = 0;

GameObject item;

float x = Random.Range(12.48f, 21.4f);}}

**UI스크립트**private Text timerText;  
void Start(){this.timerText = GameObject.Find("Time").GetComponent<Text>();}

public void IncScore(int ChangeScore){score += ChangeScore;

this.scoreText.text = "점수 " + score.ToString();}

**스크립트 불러오기**  
GameObject manager = GameObject.Find("GameManager");

manager.GetComponent<GameManager>().UpdateLife(-1);

Destroy(gameObject, 0.2f);

**Coroutine사용->현재 실행을 잠시 중단**

public float interval;

public float range = 5.0f;

void Start()

{StartCoroutine(CreateWall());}

IEnumerator CreateWall()

{WaitForSeconds wait = new WaitForSeconds(interval);

while(true)

{float wallPosY = Random.Range(-range, range);

transform.position = new Vector3(transform.position.x, wallPosY, transform.position.z);

Instantiate(wallPrefab, transform.position, transform.rotation);

yield return wait;}}