



가상현실 및 실습



프리팸 관리 및 키보드 입력

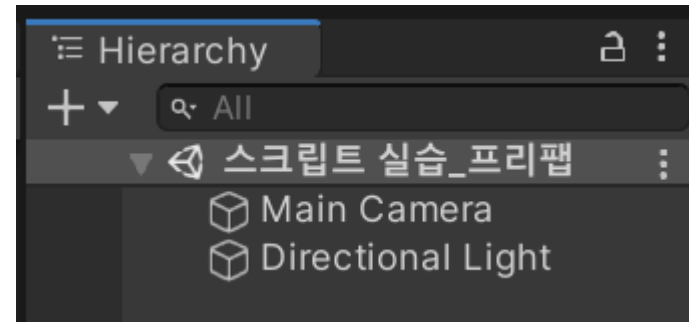
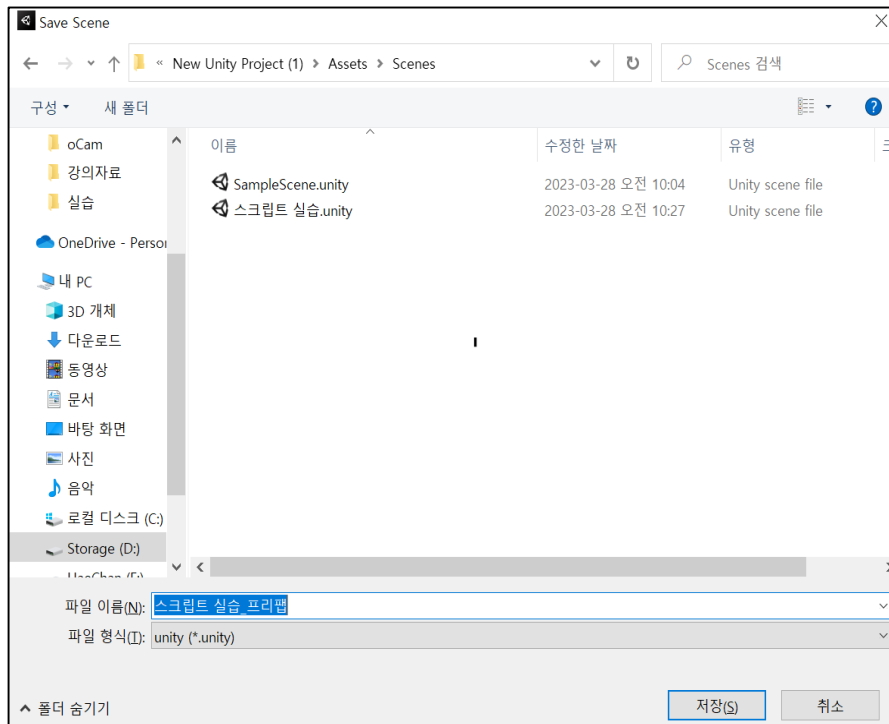


프리팹 실습

1. 프리팹 실습

1) 준비사항

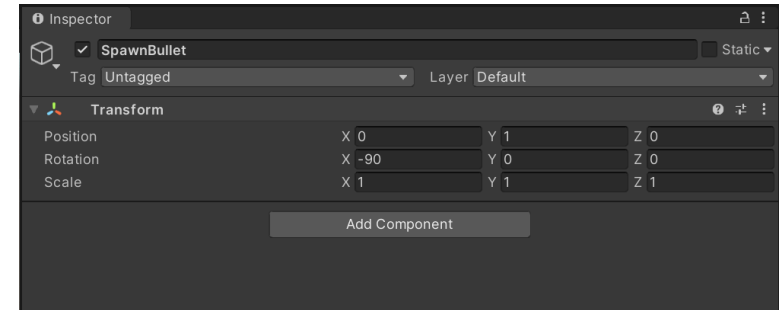
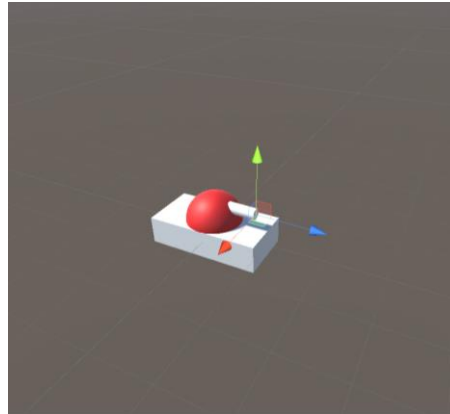
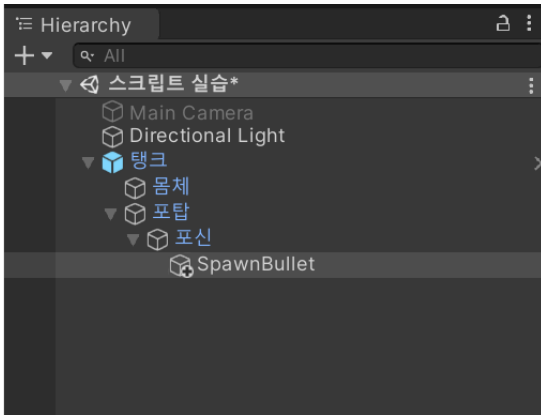
- 새로운 메뉴 > File > New Scene을 선택하여,
새로운 씬 스크립트 실습을 생성



1. 프리팹 실습

2) 오브젝트 배치

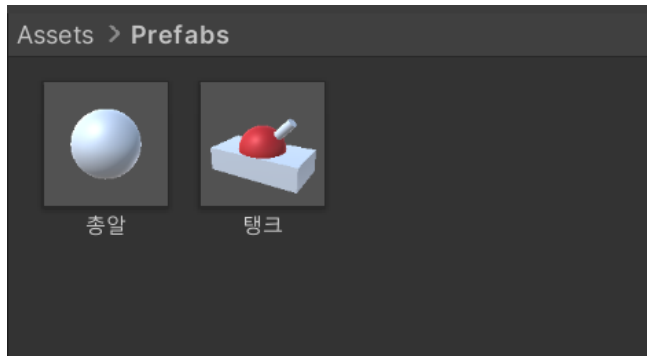
- 빈 게임 오브젝트를 생성
- 이름을 **SpawnBullet** 으로 변경 후 다음과 같이 설정
 - 부모 게임 오브젝트: 포신
 - Transform – 위치: 0, 1, 0
 - Transform - 회전: -90, 0, 0
 - Transform - 스케일: 1, 1, 1



1. 프리팹 실습

2) 오브젝트 배치

- 프리미티브 게임 오브젝트인 Sphere 게임 오브젝트를 생성
- 이름을 **총알**로 변경 후 다음과 같이 설정
 - Transform - 위치: 0, 0, 0
 - Transform - 회전: 0, 0, 0
 - Transform - 스케일: 0.2, 0.2, 0.2
- 프리팹 폴더 내 프리팹으로 저장



1. 프리팹 실습

3) Fire 스크립트

- 모든 게임 오브젝트 및 컴포넌트는 스크립트에서 제어 가능함
- Public은 외부에서 사용할 수 있도록 함
- Instantiate(오브젝트, 위치, 회전) : 오브젝트를 지정한 위치 및 회전에 따라 생성함
- Input.GetKeyDown : 키코드 버튼이 눌렸는지 확인함

```
public class Fire : MonoBehaviour
{
    public GameObject bulletPrefab;
    public Transform spawnBullet;

    // Update is called once per frame
    [Unity 메시지 참조 0개]
    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Space))
        {
            var fire = Instantiate(bulletPrefab, spawnBullet.position, Quaternion.identity);
        }
    }
}
```

1. 프리팹 실습

4) Bullet 스크립트

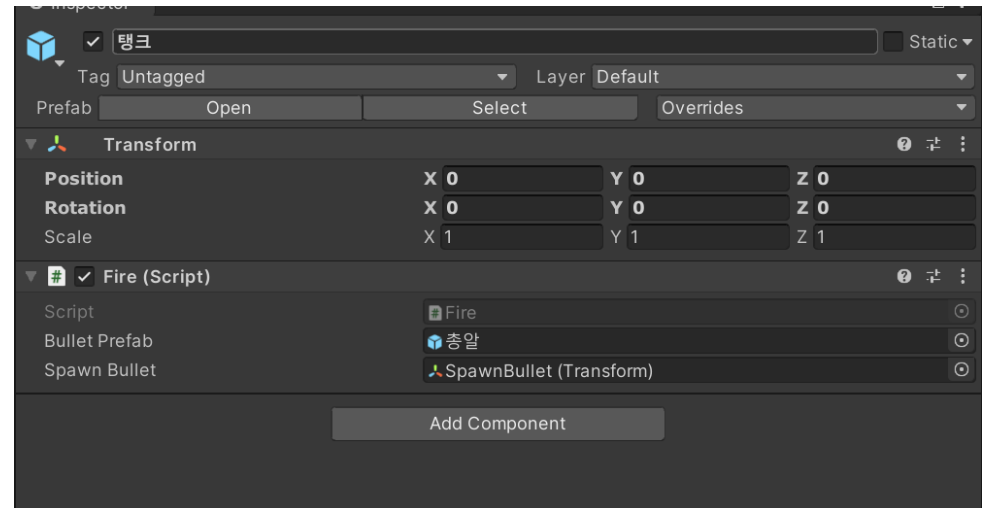
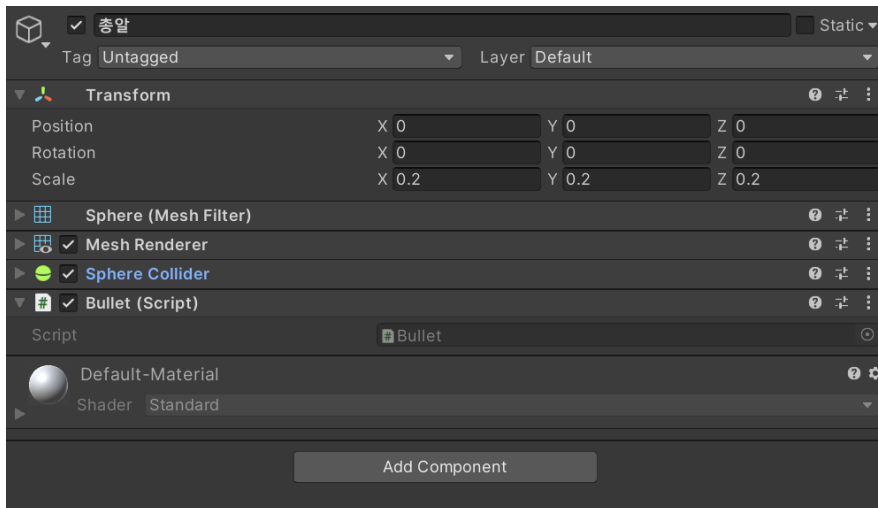
- Translate(Vector3) : Vector3의 방향으로 이동

```
☞ Unity 스크립트(자산 참조 1개) | 참조 0개
public class Bullet : MonoBehaviour
{
    // Update is called once per frame
    ☞ Unity 메시지 | 참조 0개
    void Update()
    {
        transform.Translate(Vector3.forward);
    }
}
```

1. 프리팹 실습

5) 스크립트 추가

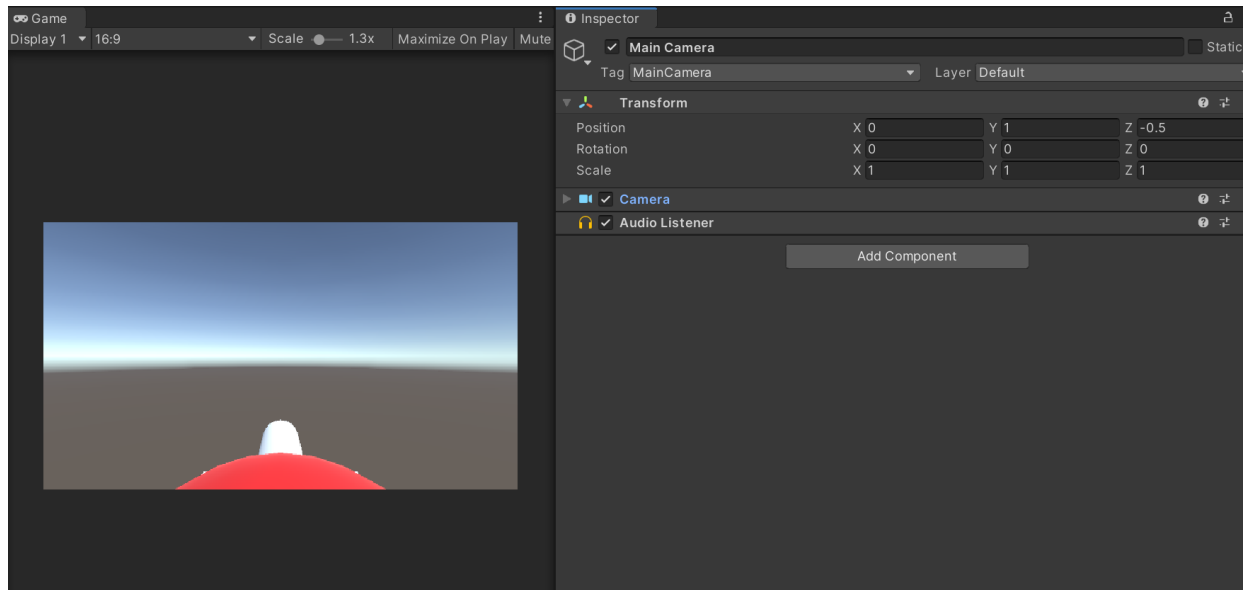
- Bullet Script를 총알 오브젝트에 추가
- Fire Script를 탱크 오브젝트에 추가



1. 프리팹 실습

6) 메인카메라

- Transform – 위치: 0, 1, -0.5
- Transform - 회전: 0, 0, 0
- Transform - 스케일: 1, 1, 1





키보드 입력 기반 상호작용

2. 키보드 입력

1) 오브젝트 정보

- 스크립트 "ObjectControlModule"를 더블 클릭하여, 코드 편집기를 실행함
- 스크립트를 가지고 있는 오브젝트의 Transform 컴포넌트는 "transform" 키워드로 접근할 수 있음

```
void Start()  
{  
    //3차원 값을 저장하는 변수 position에  
    //Transform 컴포넌트의 위치 값(position)을 저장  
    Vector3 position = transform.position;  
    //회전 값을 저장하는 변수 rotation에  
    //Transform 컴포넌트의 회전 값(rotation)을 저장  
    Quaternion rotation = transform.rotation;  
    //3차원 값을 저장하는 변수 scale에  
    //Transform 컴포넌트의 크기 값(localScale)을 저장  
    Vector3 scale = transform.localScale;  
}
```

2. 키보드 입력

1) 오브젝트 이동

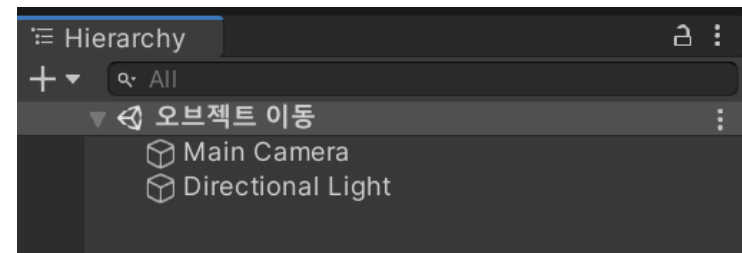
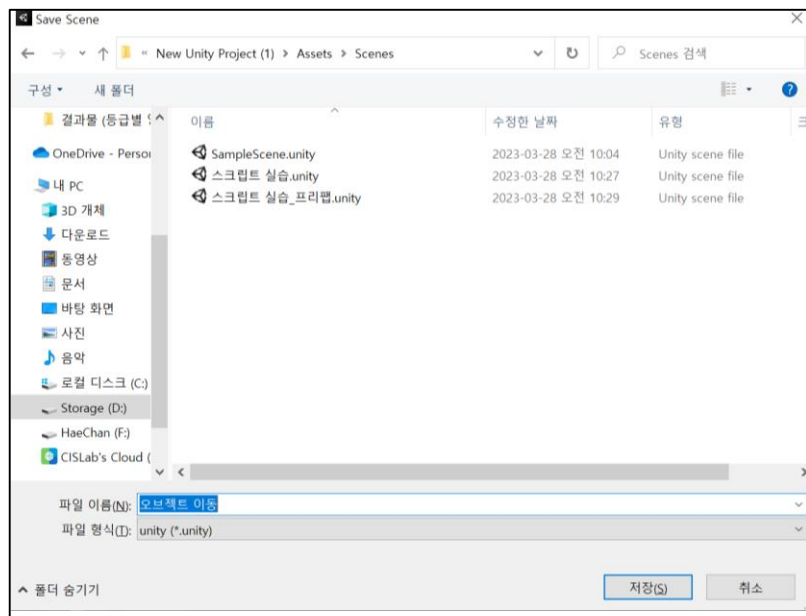
- 오브젝트의 위치 변형은 주로 transform.position 값을 직접 변형하거나, 오브젝트 이동 명령을 내리는 transform.Translate(단위 벡터) 메소드를 통해 수행됨
- 예. 앞 방향으로 3만큼 이동
 - `transform.position = transform.position + transform.forward * 3`
 - `transform.Translate(Vector3.forward * 3);`

단위 벡터	
Vector3.forward	Vector3(0, 0, 1)
Vector3.back	Vector3(0, 0, -1)
Vector3.left	Vector3(-1, 0, 0)
Vector3.right	Vector3(1, 0, 0)
Vector3.up	Vector3(0, 1, 0)
Vector3.down	Vector3(0, -1, 0)
Vector3.one	Vector3(1, 1, 1)
Vector3.zero	Vector3(0, 0, 0)

3. 키보드 입력 실습

1) 준비사항

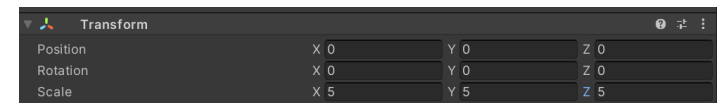
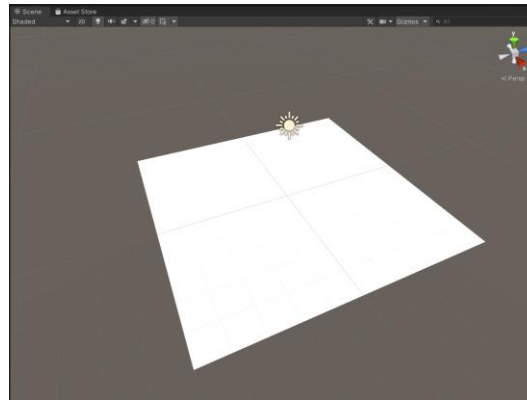
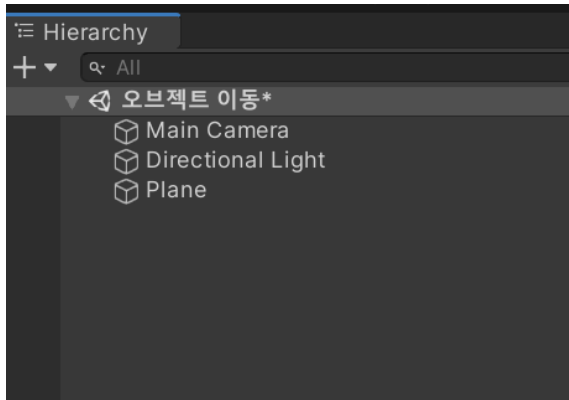
- 새로운 메뉴 > File > New Scene을 선택하여,
새로운 씬 스크립트 실습을 생성



3. 키보드 입력 실습

2) 오브젝트 배치

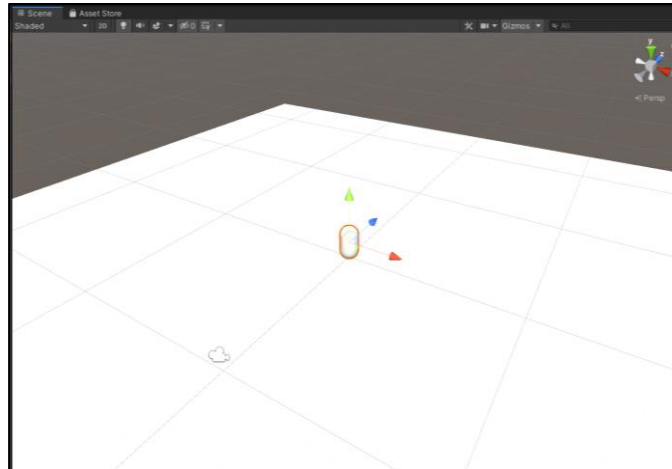
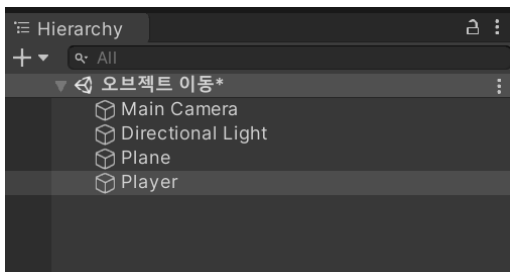
- Plane 오브젝트를 생성
 - Transform – 위치: 0, 0, 0
 - Transform - 회전: 0, 0, 0
 - Transform - 스케일: 5, 5, 5



3. 키보드 입력 실습

2) 오브젝트 배치

- Capsule 오브젝트를 생성
- Capsule 오브젝트의 이름을 Player로 변경
 - Transform - 위치: 0, 1, 0
 - Transform - 회전: 0, 0, 0
 - Transform - 스케일: 1, 1, 1

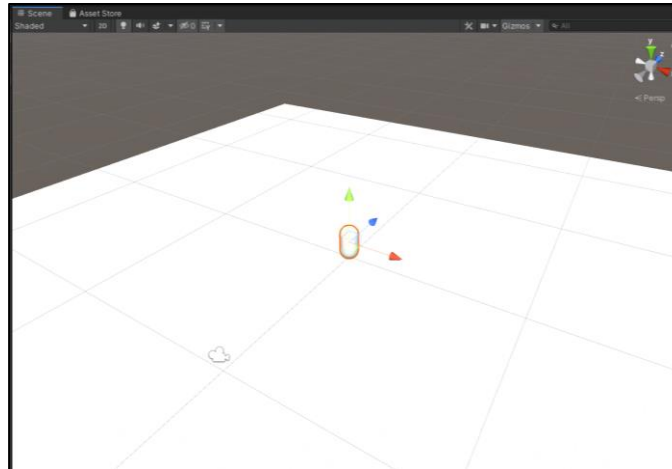
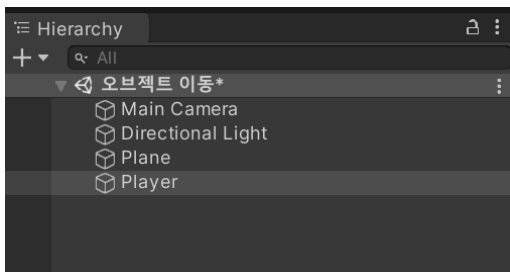


Transform			
Position	X 0	Y 1	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

3. 키보드 입력 실습

2) 오브젝트 배치

- Capsule 오브젝트를 생성
- Capsule 오브젝트의 이름을 Player로 변경
 - Transform - 위치: 0, 1, 0
 - Transform - 회전: 0, 0, 0
 - Transform - 스케일: 1, 1, 1



Transform			
Position	X 0	Y 1	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

3. 키보드 입력 실습

3) PlayerController 스크립트

- Script 폴더 내 스크립트를 생성 후 PlayerController로 이름 변경
- Input.GetKey() 메소드를 사용하여 오브젝트 이동 정의

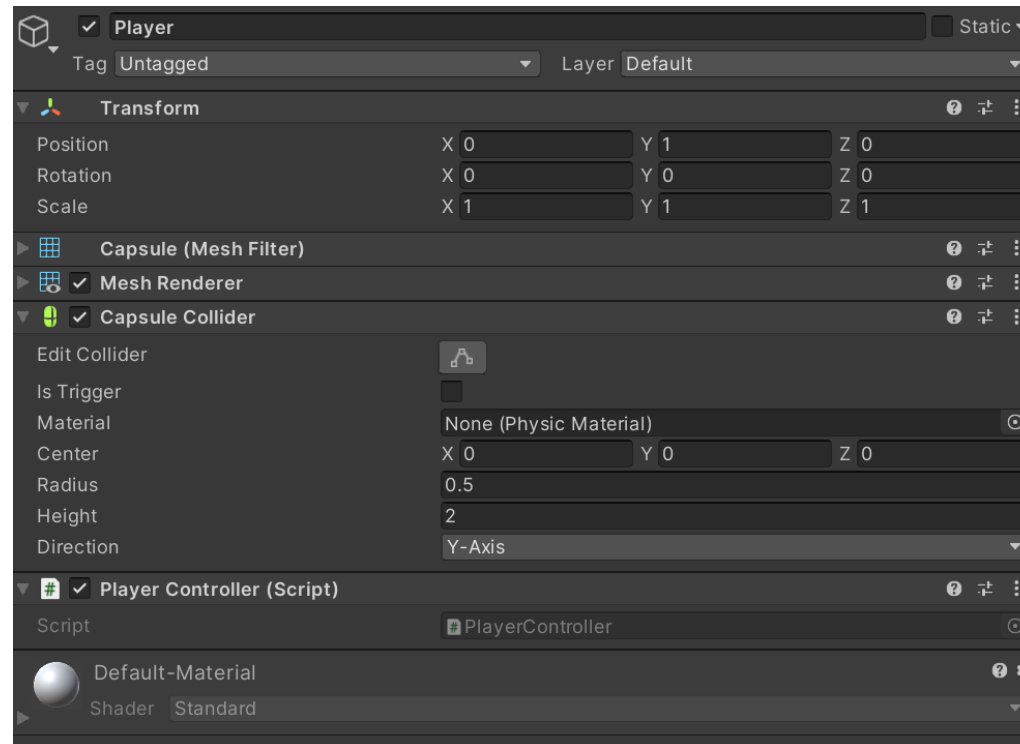
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

// Unity 스크립트 | 참조 0개
public class PlayerController : MonoBehaviour
{
    // Update is called once per frame
    // Unity 메시지 | 참조 0개
    void Update()
    {
        if (Input.GetKey(KeyCode.W))
        {
            transform.Translate(Vector3.forward * Time.deltaTime * 10);
        }
        if (Input.GetKey(KeyCode.S))
        {
            transform.Translate(Vector3.back * Time.deltaTime * 10);
        }
        if (Input.GetKey(KeyCode.A))
        {
            transform.Translate(Vector3.left * Time.deltaTime * 10);
        }
        if (Input.GetKey(KeyCode.D))
        {
            transform.Translate(Vector3.right * Time.deltaTime * 10);
        }
    }
}
```

3. 키보드 입력 실습

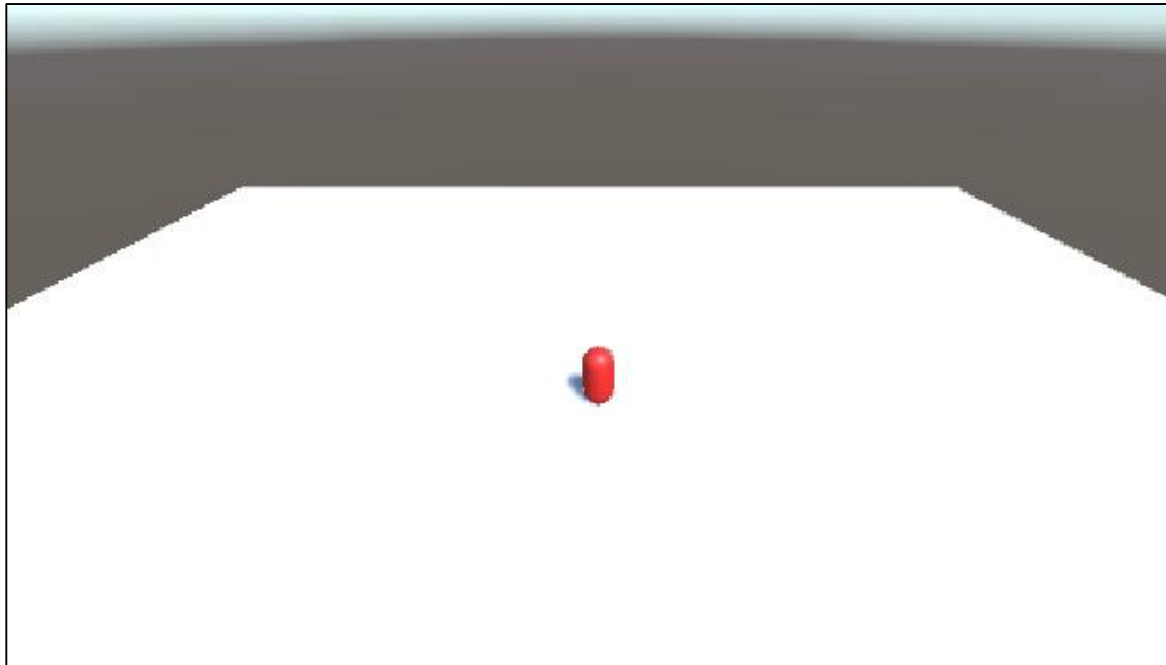
3) PlayerController 스크립트

- Player 오브젝트에 AddComponent로 PlayerController 스크립트 추가



3. 키보드 입력 실습

4) 씰 재생



3. 키보드 입력

1) Input.GetAxis() 메소드 사용 키보드 입력

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

☞ Unity 스크립트 | 참조 0개
public class PlayerController1 : MonoBehaviour
{
    public float speed = 10f;

    // Update is called once per frame
    ☞ Unity 메시지 | 참조 0개
    void Update()
    {
        //수평방향의 키보드 입력 시, -1 ~ 1
        float inputX = Input.GetAxis("Horizontal");
        //수직방향의 키보드 입력 시, -1 ~ 1
        float inputZ = Input.GetAxis("Vertical");

        Vector3 player = new Vector3(inputX * speed, 0, inputZ * speed);

        //velocity 변수(vector3) -> 속도를 지정
        transform.GetComponent<Rigidbody>().velocity = player;
    }
}
```

4. 실습과제

- 제출 기한: ~ 4/7 23:59까지
- 제출 방법: EL 과제 카테고리를 통해 제출
(파일 명: 가상현실및실습_학번_이름)

과제 내용	배점
새로운 씬 "Assignment_1"을 생성하여 작업할 것	1
콘텐츠를 재생했을 때, 콘솔 창에 자신의 학번과 이름이 1회 출력될 것	2
1인칭 시점의 카메라일 것 (맵의 자식 개체일 필요는 없음)	3
오브젝트 "Tank"는 WASD 키를 통해 이동할 것, Space 키를 통해 포탄이 나갈 것	4

4. 실습과제

- 제출할 프로젝트 형태

➤ 다음의 "Assets", "Packages", "ProjectSettings" 3가지 폴더를 압축하여 제출

