

## 2,3주차

### Unity 튜토리얼(Roll a ball)

#### 학습 목표

- 커스텀 스크립트에 대한 이해
- Unity의 기본적인 사용법에 대한 이해

#### 기초적인 C# 문법

```
public class Test : MonoBehaviour
{
    // 기본 값
    int a = 10;
    float b = -0.1f;
    bool c = true;
    char d = 'C';
    string e = "KUCC";

    // 유니티에서 쓰는 타입
```

```

Vector2 vector2 = new Vector2(1, 2);
Vector3 vector3 = new Vector3(1, 2, 1);
Quaternion quaternion = Quaternion.identity;

// 기타 등등
enum Week { Mon, Tue, Wed, Thu, Fri, Sat, Sun };
Week week = Week.Mon;

// 자료구조
int[] intArray = new int[10];
int[] intArray2 = new int[] { 1, 2, 3, -1 };
List<int> intList = new List<int>();
Queue<int> intQueue = new Queue<int>();
Stack<int> stackQueue = new Stack<int>();
Dictionary<string, int> dictionary = new Dictionary<string, int>
{
    { "A", 1 },
    { "B", 2 },
    { "C", 3 }
}; // O(1)
Hashtable hashtable = new Hashtable
{
    {"A", 12 },
    {12, "B" },
    {Week.Mon, "월요일" }
};
HashSet<int> hashSet = new HashSet<int>();
// O(1)

private void Awake()
{
    intList.Add(1);
    if (dictionary.ContainsKey("Key"))
    {
        dictionary.Remove("A");
    }
    else
    {
        dictionary.Add("A", 1);
        dictionary["A"] = 3;
    }

    // 조건문 및 논리 연산자
    if (a == 1)
    {

    }
    else if (a != 1)
    {

    }
    else if (a > 1)

```

```

{

}
else if (a < 1)
{

}
else if (a >= 1)
{

}
else if (a <= 1)
{

}
else if (a == 1 || b == -0.1f)
{

}
else if (a == 1 && b == -0.1f)
{

}
else if (c)
{

}
else if (!c)
{

}
else
{

}

switch (a)
{
    case 1:
    case 2:
        if (b == 3)
        {

        }
        break;
    default:

    break;
}

a = b == 0.1f ? -1 : 1;

```

```

        // 반복문
        for(int i = 0; i < intArray.Length; i++)
        {
            int a = intArray[i];
        }

        foreach(int a in intArray)
        {

        }

        int j = 0;
        while (j < 10)
        {
            j++;
        }

        do
        {

        } while (true);
    }

    // 함수
    int Sum(int a, int b)
    {
        return a + b;
    }

    void M()
    {
        //if (!user.isLogin)
        //{
        //    return;
        //}

        //
    }

    MyClassA classA = new MyClassA();
}

// 클래스
public class MyClassA
{
    public static int a = 10;
    protected int b = 2;

    protected virtual void B()
    {

```

```

        // 작업
    }
}

// 클래스 상속
public class MyClassB : MyClassA
{
    protected int c = 10;

    public void A()
    {
        MyClassA.a = 10;
        //this;
        base.b = -1;
    }

    protected override void B()
    {
        base.B();
    }
}

// 구조체
public struct MyStructA
{
}

```

## Roll a ball

Unity에서 공식으로 지원하는 튜토리얼입니다. 이번 주차에서는 “Roll a ball”을 제작을 실습하면서 튜토리얼을 진행하도록 하겠습니다.

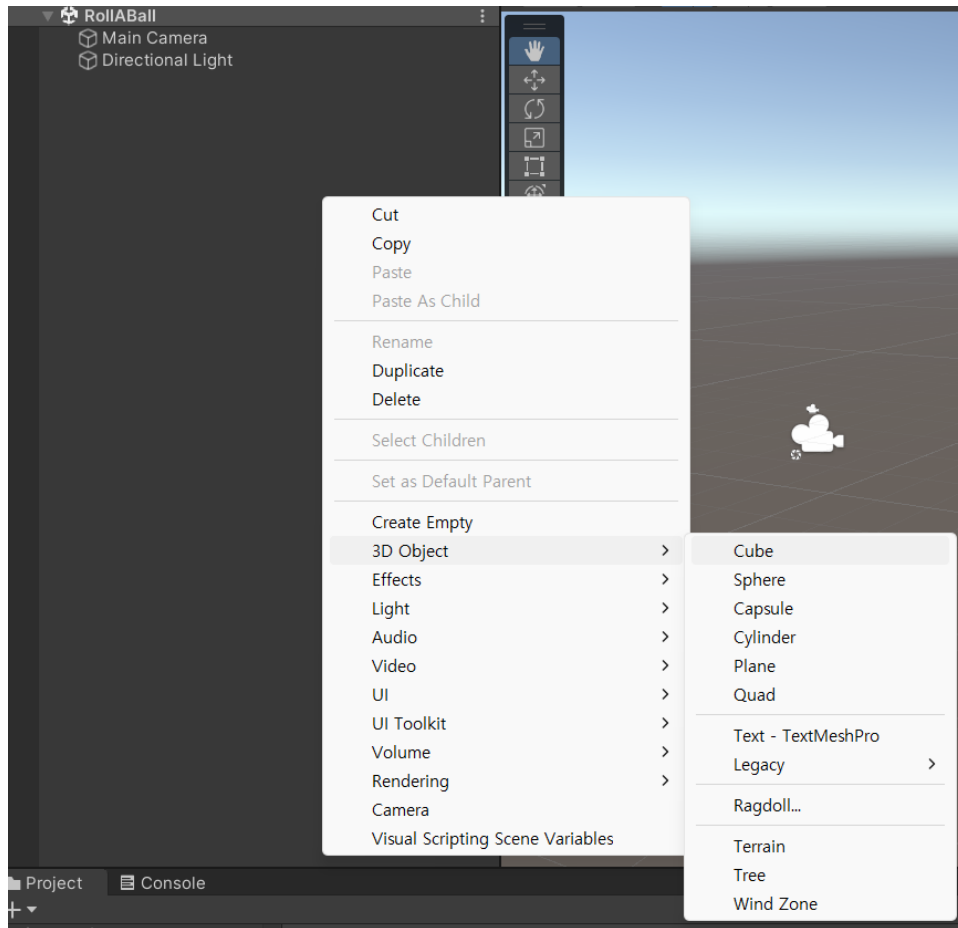
[/https://learn.unity.com/project/roll-a-ball-1](https://learn.unity.com/project/roll-a-ball-1)

[https://www.youtube.com/watch?v=\\_uO5B7bP9jo](https://www.youtube.com/watch?v=_uO5B7bP9jo)

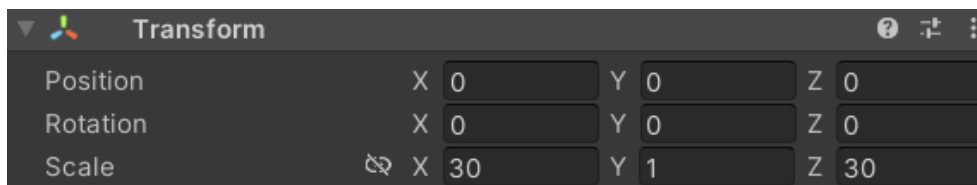
## 큐브 생성

Hierarchy 창에서 우클릭

3D Object > Cube



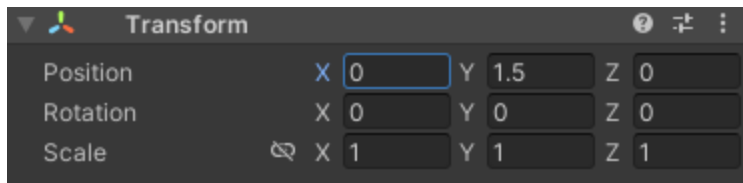
Transform을 아래와 같이 초기화



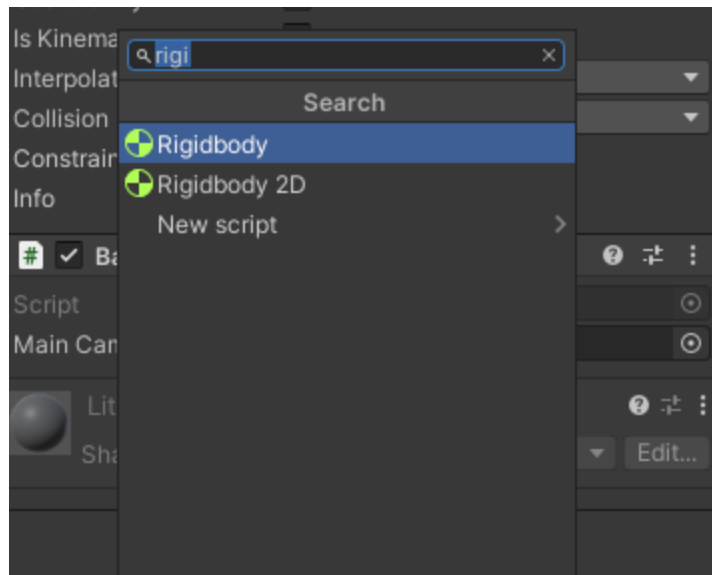
## 공 생성

3D Object > Sphere

아래와 같이 초기화



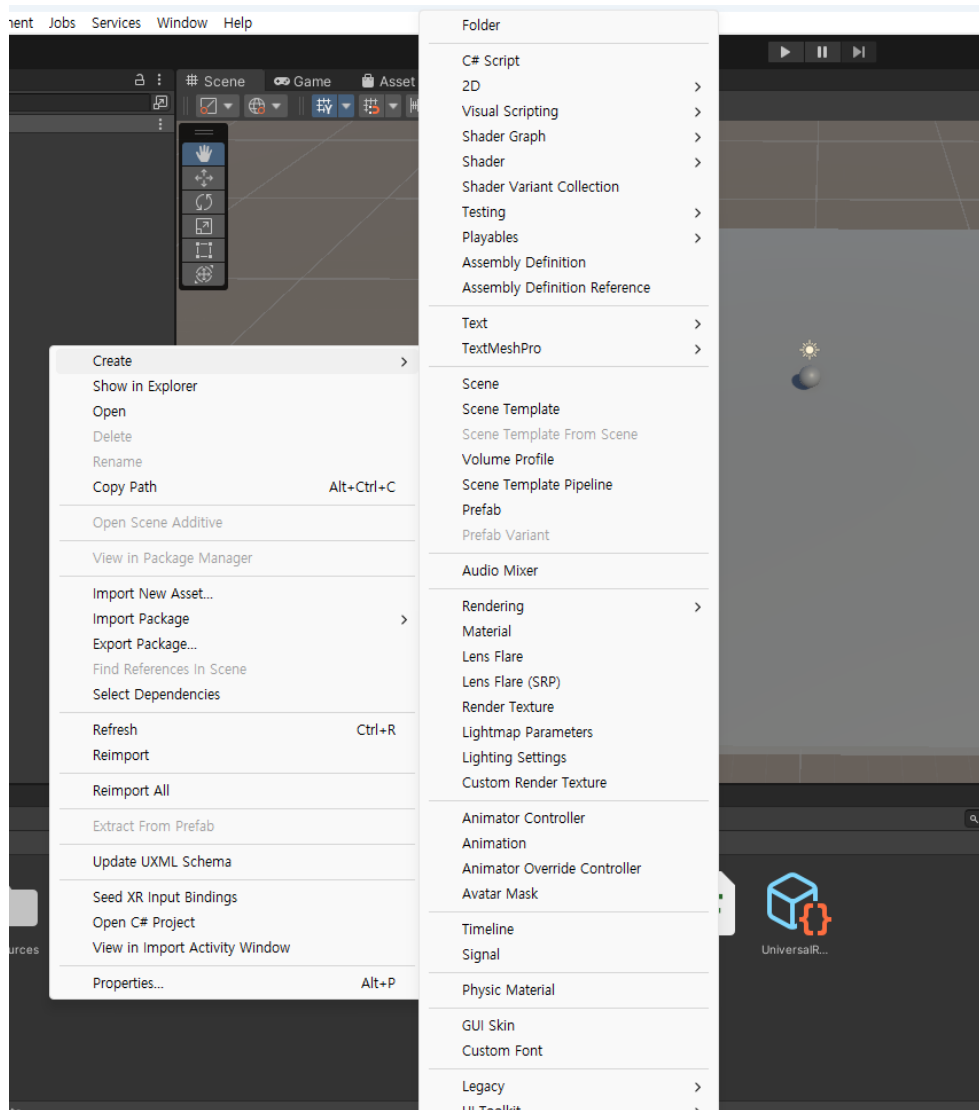
오른쪽 아래 Add Component 클릭후 Rigidbody 추가



밑에 Assets 폴더 창에서 마우스 오른쪽 클릭

Create > C# Script

이름은 BallControl



BallControl 스크립트를 열고 아래 내용을 붙여넣습니다.

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

public class BallControl : MonoBehaviour
{
    public Camera mainCamera;
    private Vector3 offset;

    private Rigidbody rb;
```



```

private void Awake()
{
    rb = GetComponent<Rigidbody>();

    offset = mainCamera.transform.position - this.transform.position;
}

private void Update()
{
    mainCamera.transform.position = this.transform.position + offset;
}

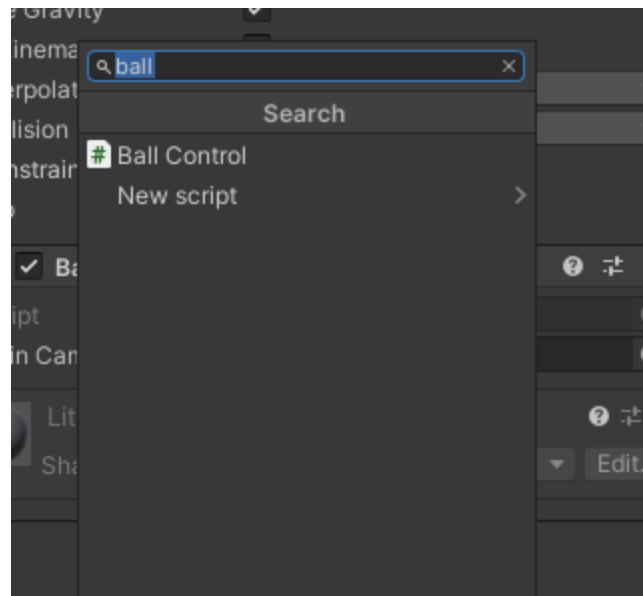
private void FixedUpdate()
{
    float h = Input.GetAxisRaw("Horizontal"); // A, D, <-, ->
    float v = Input.GetAxisRaw("Vertical"); // W, S,

    Vector3 moveDirection = new Vector3(h, 0, v);

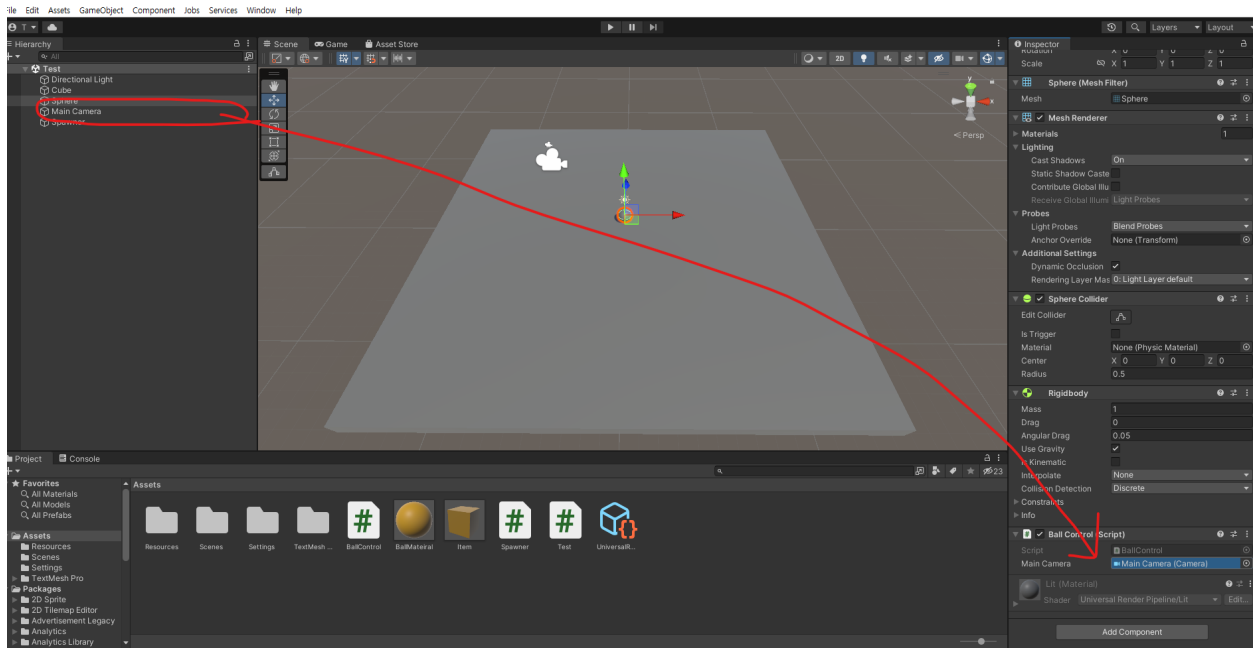
    rb.AddForce(moveDirection * 10f);
}
}

```

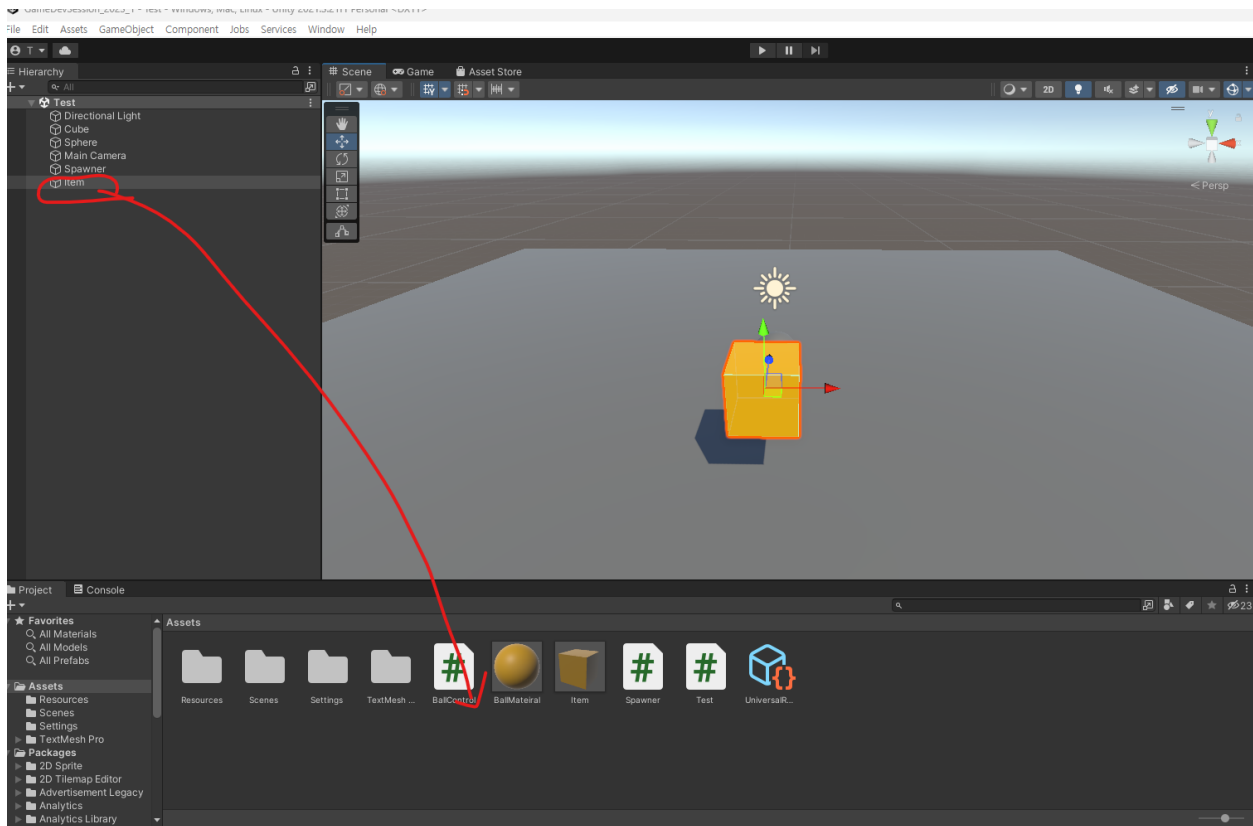
공 오브젝트를 클릭하고 Add Component > Ball Control



Hierarchy 창의 Main Camera를 잡고 끌어서, Ball Control의 Main Camera 공간에 집어넣습니다.



Hierarchy에서 우클릭, 큐브 생성  
F2를 누르면 이름 변경이 가능합니다. Item으로 변경  
해당 오브젝트를 잡고 끌어서 Assets에 집어 넣습니다.



Assets 폴더에서 우클릭후 C# Script 생성  
이름은, Spawner, 아래 내용을 붙여넣습니다.

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

public class Spawner : MonoBehaviour
{
    public GameObject itemPrefab;
    public int count = 10;

    private void Awake()
    {
        for(int i = 0; i < count; i++)
        {
            float angleBetween = 2 * Mathf.PI / count * i;
            Vector3 spawnPosition = new Vector3(15f * Mathf.Cos(angleBetween), 1.5f,
                15f * Mathf.Sin(angleBetween));
```

```

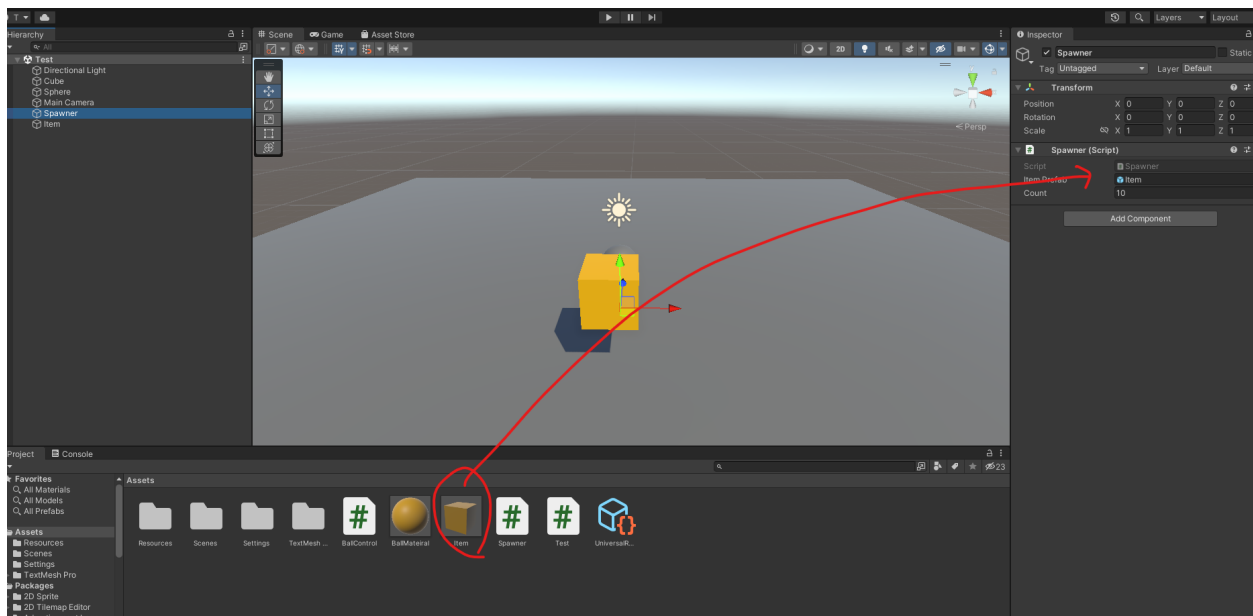
        Instantiate(itemPrefab, spawnPosition, Quaternion.identity);
    }
}
}

```

Hierarchy 창에서 우클릭후 Create Empty

이름은 Spanwer, 오브젝트 클릭후 Add Component > Spawner

전에 밑에 Assets에 추가한 Item을 끌어다 Spawner 컴포넌트의 Item Prefab에 붙여넣습니다.



게임 실행 화면

