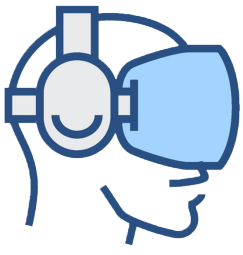
B-INN-000 - Worshop: Statue Game

AR/VR Worshop: Statue Game

It **V**irtually **R**uns!



VR

AR/VR Worshop: Statue Game

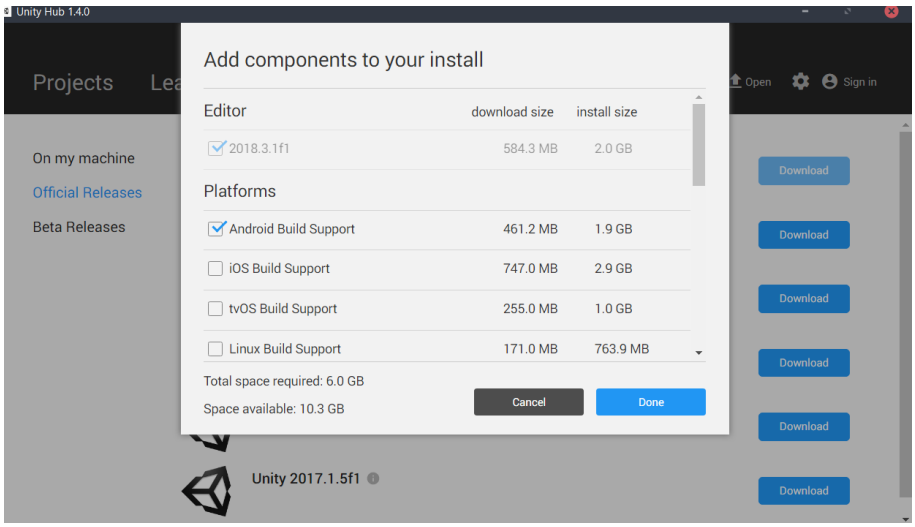
Language: C#

> ⚠ **Your repository must contain all of your source files but NO useless files.**

No, you won't find long and useless pages here. Let's begin exercise #1.

Exercise 00: Installation

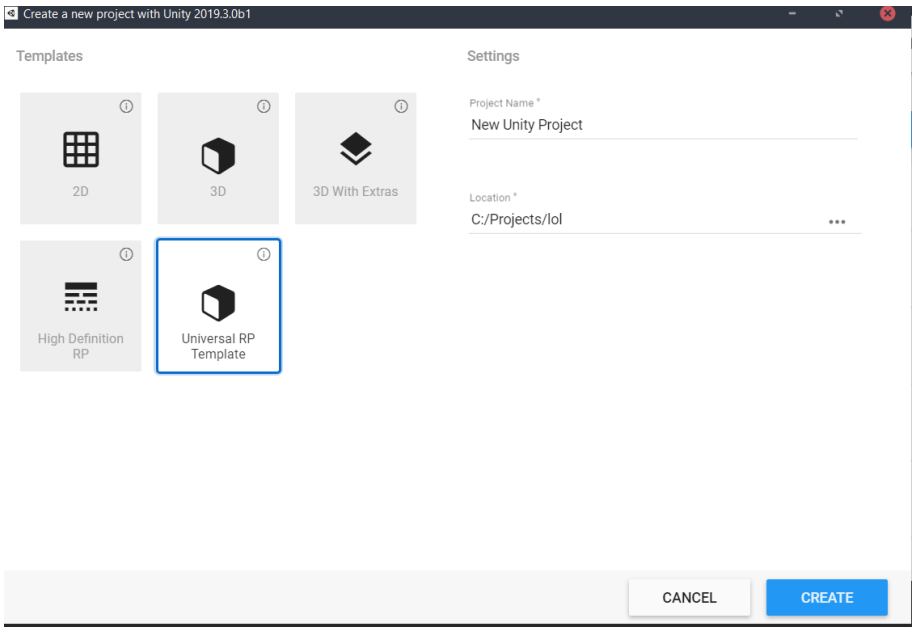
Download Unity Hub : https://store.unity.com/download
Dowload Unity 2019.x or after with android package (on unity hub)



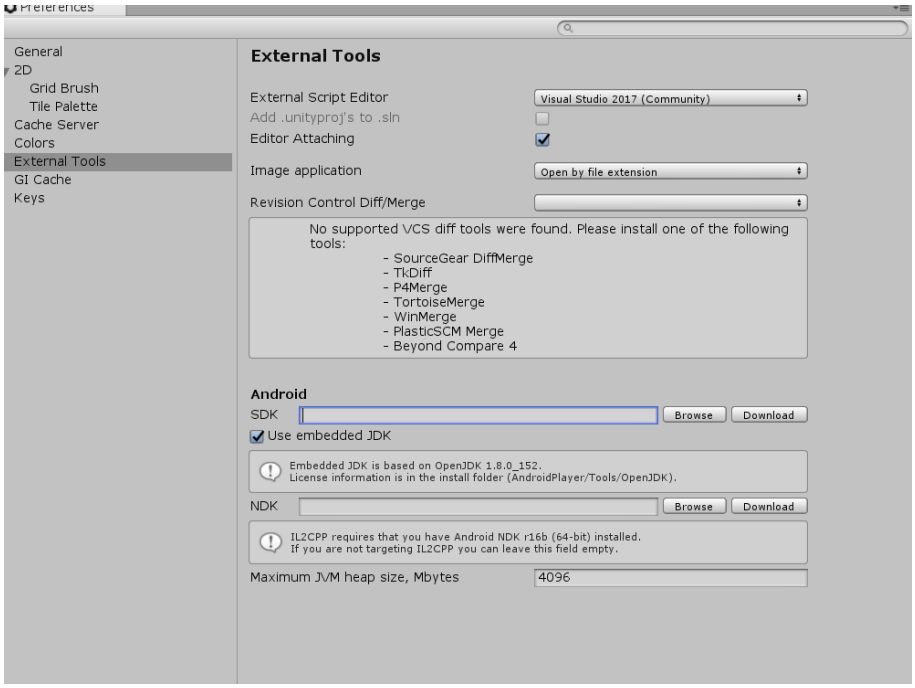
Dowload android studio if it's not installed

> ⚠ **Watch out where you install Android SDK**

Create a unity project in unity hub with a ”Universal RP Template” template

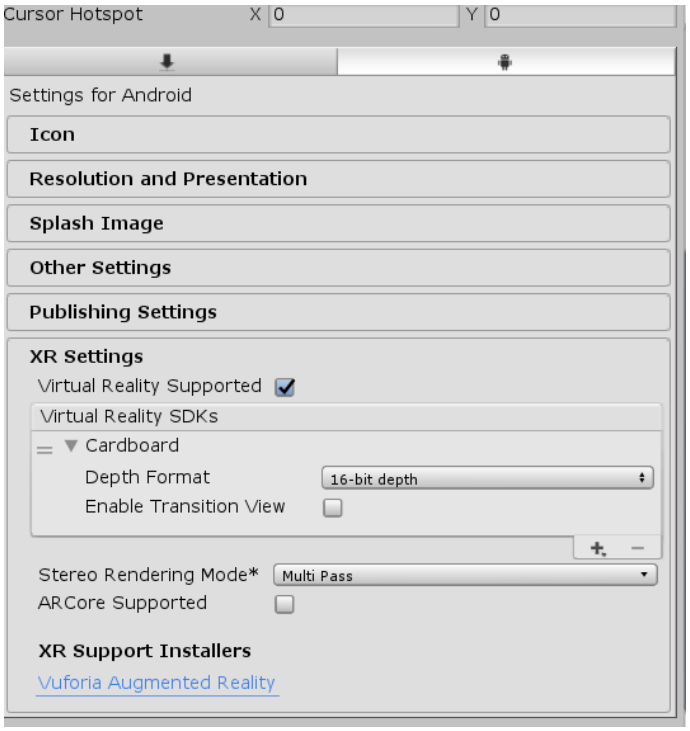


Link Android sdk with unity: Go to edit->preference->external tool and set android sdk path



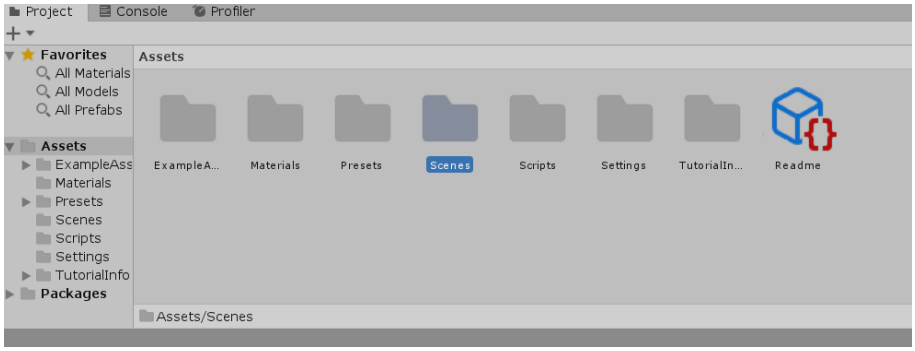
Set up the project setting for build an android project Switch the build setting to Android Set the player setting to :

- Other setting :
  - Set color to 32 bits
  - Package name to com.poc.ARVRWorkshop
  - Minimum API level to your phone
- Xr setting :
  - Activate virtual reality
  - Add card board to VR SDK



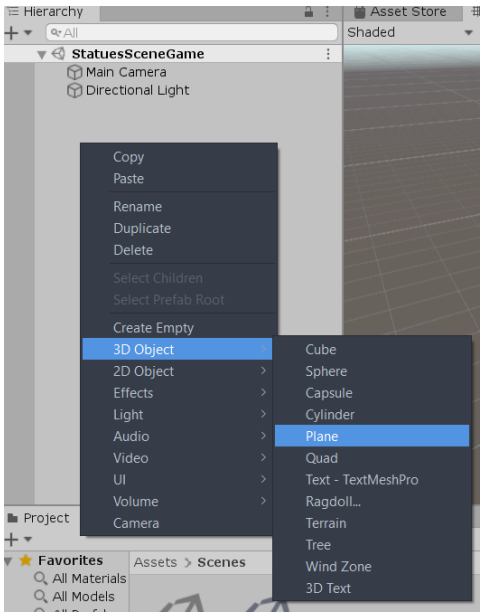
Exercise 01: Set Up the scene

Go to the Asset Explorer in the editor and create a new scene in the ”Scenes” folder name it ”StatuesSceneGame”



> ⚠ **When you create something in the project explorer it’s : right click -> create**

Create a 3d object plane name it ”Wall” and rotate it to -90° on the x axis. Go to the Asset Explorer in the editor and create a new scene in the ”Scenes” folder name it ”StatuesSceneGame”



Create a 3d object capsule name the capsule ”Player” place the player at the front of the Wall set the main camera as child of the player object in the hierarchy and place the camera at coordonate (0,0,0).

Exercise 02: Save and Test the App

Save your scene by Control + S on the scene, plug your phone to your computer. Next you'll need to build and run your application by clicking on : File -> Build and Run

> ⚠ **Your phone need to be in developper mode, with the USB debugging mode activate**

when the build will finish if no error occured your phone will automatically start the builded application

Exercise 03: Create Statues

First you need to create an empty game object name "StatuesSpawner" and create a script : "Spawn" this script will Spawn all the statue with a given prefab

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

public class Spawn : MonoBehaviour
{
    public int maxStatues;
    public GameObject prefab;
    private List<GameObject> satues = new List<GameObject>();
    // Update is called once per frame
    void Update()
    {

    }
}
```

the Spawn script will instantiate one statue per frame if the number of statue is less or equal than the number max of statues

> ⚠ **add the statues to the satues list**
> Don't forget to add the script as component to "StatuesSpawner"

Now create a new Capsule game object name ”Statue” and add a rigid body to the statue Create a script : ”StatueController” which make the statues come to the player

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

public class StatueController : MonoBehaviour
{
    private Rigidbody rigidBody;
    public GameObject player;

    // Start is called before the first frame update
    void Start()
    {
        rigidBody = GetComponent<Rigidbody>();
    }

    // Update is called once per frame
    void Update()
    {

    }
}
```

⚠ don’t forget to add the script to the statue’s game object
save the statue as prefab and set Spawn script of StatuesSpawner to statue’s prefab

Exercise 04: Add Statue destruction

Add a public method name : ”DestroyStatue” to the ”Spawn” script.

```
public void DestroyStatue(GameObject statue) {}
```

DestroyStatue destroy and remove from the list the given Statue gameObject

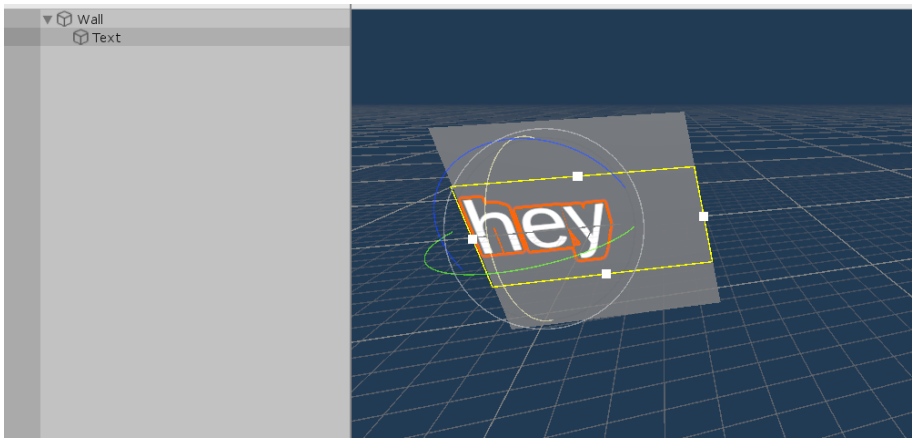
Create a script : ”StatueDestroyer”, StatueDestroyer destroy all the statue in a given field of view

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

public class StatueDestroyer : MonoBehaviour
{
    public Spawn spawner;
    public uint fealdOfViewAngle = 90;

    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {

    }
}
```

⚠ Angle function could be usefull ;)

Exercise 05: Add a score board

Add a son GameObject to ”wall” name ”text” and add ”TexMeshPro - Text” component to ”text”



Add TexMeshPro public varibale to StatueDestroyer and a method who set the score of the player

```
public TextMeshPro textMeshPro;
```

```
void SetScore(uint score){}
```

⚠ don't forget : **using TMPro;**

Exercise 06: Create a Rest System

Create Script name : "RestController" the rest controller will turn the sky red if your not looking at the wall and turn green when your looking at the wall

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

public class RestController : MonoBehaviour
{
    public Camera cameraView;
    public uint delay;
    public LayerMask layer;

    // Update is called once per frame
    void Update()
    {

    }
}
```

⚠ you could use :

- **"RayCast"** function for detecting the wall
- **"Lerp"** function for the color skybox

Exercise 07: Create loose

Add an OnTriggerEnter function who detect if a statue touch the player

```
private void OnTriggerEnter(Collider other){}
```

If the player is touched destroy all the statue, desactivate the spawner and display you loose on the wall