

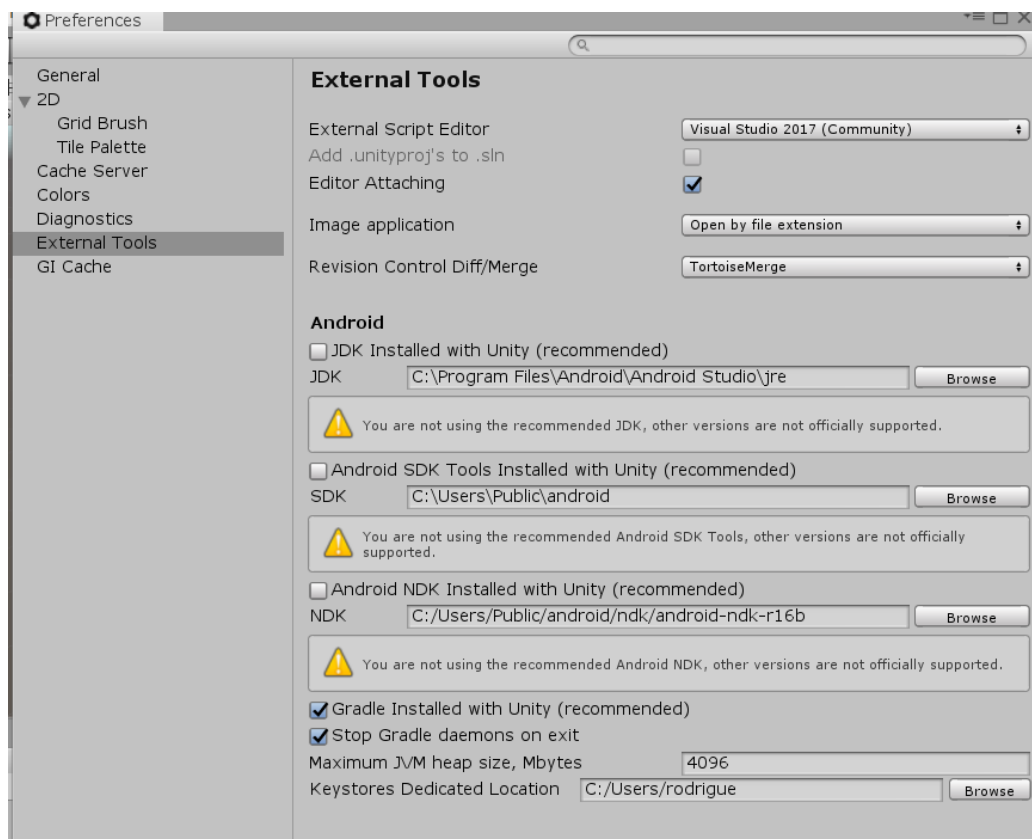
## TP4 - APPLICATION CARDBOARD

Ce TP est le mode d'emploi pour construire des applications pour le Cardboard. Pour cela, il faut faire un saut dans le passé, car le package actuel est encore en voie de développement et donc pas stable.

Téléchargez le Google VR SDK pour Unity(<https://github.com/googlevr/gvr-unity-sdk/releases>). Installez la version 2019.1.4 d'Unity grâce à la page <http://unity3d.com/get-unity/download/archive>. Téléchargez l'Unity Installer pour ajouter à l'éditeur lors de l'installation le support pour Android.

### INSTALLATION ANDROID

Vous avez peut-être déjà dans votre système les outils pour construire des applications Android. Si ce n'est pas le cas, installez le dernier JDK (<http://jdk.java.net>), et puis, je vous conseille d'installer Android Studio. Dans la configuration de celui-ci, cliquez sur le sdkmanager. Vérifiez que le SDK est installé, ainsi que les *command-line tools*. Unity peut vous demander une autre version du NDK, les différentes versions sont disponibles ici : [https://developer.android.com/ndk/downloads/older\\_releases](https://developer.android.com/ndk/downloads/older_releases)



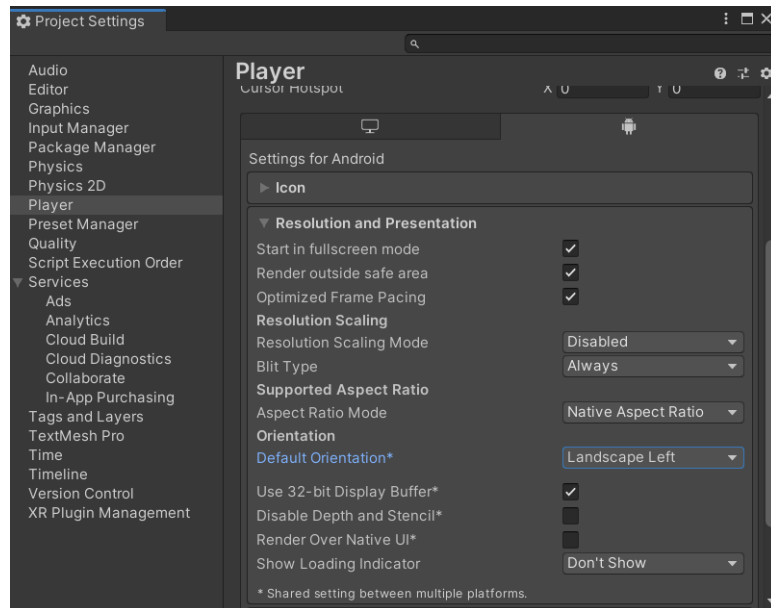
### CONFIGURATION SMARTPHONE

Pour activer votre téléphone en mode développement, cliquez sur *Réglages > A propos de l'appareil* et puis tapez 7 fois (!) sur le numéro de version. Puis, allez dans *Réglages > Options de Développement > Activer débogage USB*.

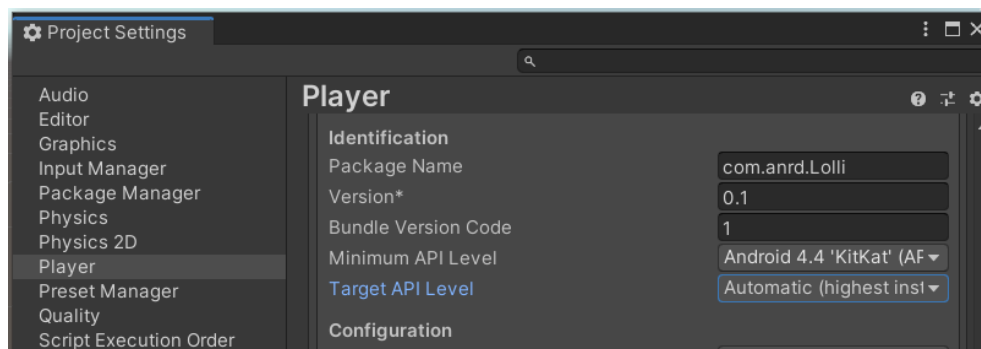
### CONFIGURATION UNITY

Créez un nouveau projet 3D dans Unity. Dans *Edit > Preferences*, allez dans la rubrique *External Tools* et remplissez les champs avec les path des outils Android.

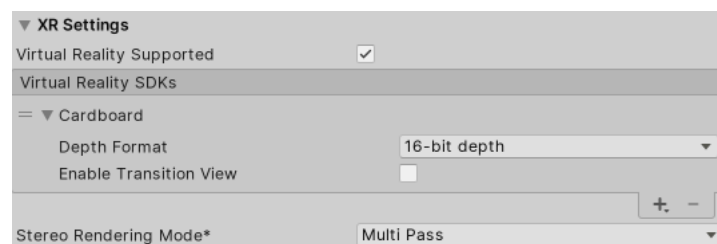
Allez dans *Edit > Project Settings > Player*. Cliquez sur l'icône Android. Modifiez, dans **Resolution et Presentation**, l'orientation à *Landscape Left*.



Dans **Other Settings**, enlevez *Vulkan* de la liste **Graphics API**. Vérifiez que **Color Space** est en mode *Gamma*, et choisissez *Android 4.4 Kit Kat* pour le **Minimum API Level**. Pour finir, modifiez la partie **Package Name** avec la valeur de votre choix en respectant le modèle DNS inversé.



Dans les XR Settings, cochez la case **Virtual Reality Supported**. Cliquez sur le bouton « + » de la liste pour ajouter *Cardboard*. Modifiez le **Stereo Rendering Mode** a *Single Pass*.



### TEST CONFIGURATION

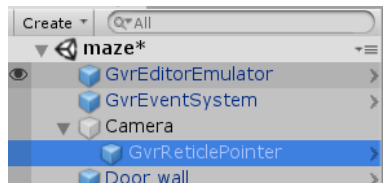
Importez le package GoogleVRforUnity des ressources du TP. Ouvrez la scène **GoogleVR > Demos > Scenes > HelloVR**. Allez dans **File > Build Settings** et ajoutez la scène de test. Puis, sélectionnez Android dans la liste de plateformes et puis cliquez sur **Switch Platform**. Connectez votre téléphone à l'ordinateur en utilisant un câble USB. Cliquez sur **File > Build and Run**. Unity crée une application Android APK à partir de votre projet, l'installe sur votre téléphone et l'exécute. Insérez votre téléphone dans votre Cardboard et testez la démo. Vous verrez un point blanc, le reticle, qui suit votre regard. Une forme géométrique, au centre, se transforme lorsque vous la regardez.

## CREATION DE LA SCENE

Ajoutez une nouvelle scène à votre projet. Vous pouvez réutiliser le labyrinthe des TP précédents. Ajoutez à votre scène le prefab **Google > Prefabs > GvrEditorEmulator**. Lancez votre application. Vous pouvez regarder tout autour !

La caméra est modifiée par le système GoogleVR. Pour éviter de perdre la position, créez un GameObject vide qui contiendra la caméra et nous permettra de modifier sa position de manière dynamique facilement.

Ajoutez le prefab **GvrEventSystem** et puis le prefab **GvrReticle** en tant que fils de la caméra. Ces outils vont vous permettre d'interagir avec les objets. Ajoutez à la caméra le composant **Google VR Physics Ray Caster**.



Créez un cube rouge dans votre scène (ou choisissez un objet déjà présent). Nous allons changer la couleur lorsque l'utilisateur va le regarder. Pour cela, on va lui associer un script très simple :

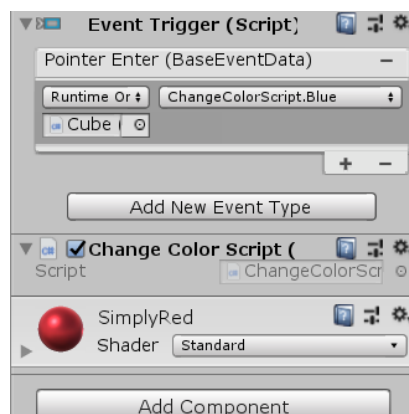
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class ChangeColorScript : MonoBehaviour
{
    (...)
    public void Red() {
        GetComponent<Renderer>().material.color = Color.red;
    }

    public void Blue()
    {
        GetComponent<Renderer>().material.color = Color.blue;
    }

    public void Magenta()
    {
        GetComponent<Renderer>().material.color = Color.magenta;
    }
}
```

Ajoutez au cube le composant **Event > Event Trigger**. Ajoutez à ce trigger les événements PointerEnter pour modifier la couleur à la couleur de votre choix et puis PointerExit pour revenir au rouge :



Pour permettre à l'utilisateur de se déplacer, nous allons créer un script à associer à l'objet contenant la caméra :

```

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

public class PlayerBehaviourScript : MonoBehaviour
{
    private bool walking = true;
    private Vector3 spawnPoint;
    private Camera mainCamera;
    public float speed = 0.5f;

    // Start is called before the first frame update
    void Start()
    {
        mainCamera = Camera.main;
        spawnPoint = transform.position;
    }

    // Update is called once per frame
    void Update()
    {
        if (walking)
        {
            transform.position = transform.position + mainCamera.transform.forward*0.5f* Time.deltaTime;
        }

        if (transform.position.y < -10f)
        {
            transform.position = spawnPoint;
        }
    }
}

```

Il nous permet de modifier la position de l'utilisateur tout en vérifiant qu'il ne va pas « trop » vers le bas. Il pourrait être possible aussi de fixer la composante Y de la transformation. Dans ce mécanisme, l'utilisateur se déplace tout le temps.

Ecrivez une solution permettant à l'utilisateur d'arrêter/démarrer le mouvement (un bouton à ses pieds par exemple ?)

Envoyez votre code et votre compte rendu à [nancy.rodriquez@lirimm.fr](mailto:nancy.rodriquez@lirimm.fr)

### Sources d'inspiration

<https://www.instructables.com/Mobile-Virtual-Reality-Game-IN-UNDER-15-MINUTES/>

<https://www.sgurwinderr.com/2019/11/google-vr-cardboard-tutorial-2019-three.html>