

---

# COURS DE REALITE AUGMENTEE

---

Christophe Vestri

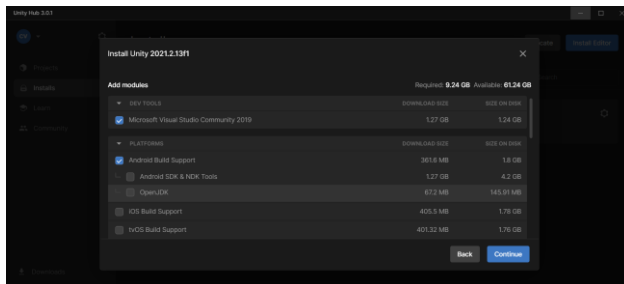
## TD 1

Utiliser son smartphone comme webcam (sera utile pour TD3)

- Si vous avez une webcam, pas forcément utile, mais ca peut être bien pour tester projet sans générer apk+install
- Installer droidCam, sur smartphone et client window : <https://www.dev47apps.com/>
- Lancer droidCam sur smartphone et autoriser usbdebug
- Lancer client via usb (ou autre) <https://www.dev47apps.com/droidcam/connect/>
- Lancer une application caméra pour vérifier que le flux est bien récupéré

Installer [Unity](#) version personnelle :

- Télécharger Unity Hub
- Installer version la dernière version ou une LTS
- Ajouter le système de votre smartphone : Android ou IOS



- Créer un compte Unity ID
- Pour intégrer votre visual studio comme éditeur de script unity:  
<https://learn.microsoft.com/en-us/visualstudio/gamedev/unity/get-started/getting-started-with-visual-studio-tools-for-unity?pivots=windows>

Unity introduction

- Interface : <https://learn.unity.com/tutorial/using-the-unity-interface?language=en>
- Essential concepts : <https://learn.unity.com/tutorial/essential-unity-concepts>
- GameObjects/Scripts/variables/events/prefabs :  
<https://docs.unity3d.com/2022.1/Documentation/Manual/ScriptingSection.html>
- C# for Unity : <https://unity.com/how-to/learning-c-sharp-unity-beginners>

## Exercice Unity : Roll a Ball

- Roll a ball est un petit jeu qui permet d'apprendre qqx fonctionnalités UNITY:
  - gérer les input pour contrôler le mouvement un objet,
  - attacher la caméra à cet objet pour le suivre (ne pas faire cette partie car inutile pour notre utilisation)
  - détecter les collisions et y associer une action
  - Gérer et afficher un score
- Online tutorial : <https://learn.unity.com/project/roll-a-ball>
- Ce qui est intéressant, c'est non pas d'utiliser un keyboard mais l'orientation du smartphone, pour cela, il suffit récupérer l'accélération. Voici un script permettant de gérer le clavier et l'accélération avec smartphone (à vérifier qu'il fonctionne toujours, sinon dites-moi svp)

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

public class PlayerControl : MonoBehaviour
{
    private Rigidbody rb;

    public float speed;

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

    void FixedUpdate()
    {
        if (SystemInfo.deviceType == DeviceType.Desktop)
        {
            float moveHorizontal = Input.GetAxis("Horizontal");
            float moveVertical = Input.GetAxis("Vertical");
            Vector3 movement = new Vector3(moveHorizontal, 0.0f,
moveVertical);
            rb.AddForce(movement * speed);
        }
        else
        {
            float moveHorizontal = Input.acceleration.x;
            float moveVertical = Input.acceleration.y;
            Vector3 movement = new Vector3(moveHorizontal, 0.0f,
moveVertical);
            rb.AddForce(movement * speed);
        }
    }
}
```

## Exercices ARFoundation

<https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.2/manual/index.html>

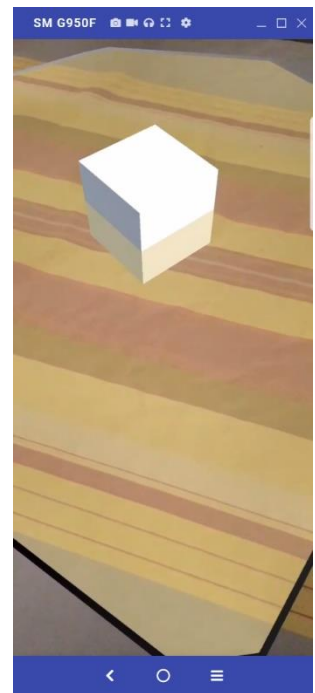
<https://docs.unity3d.com/Packages/com.unity.xr.arsubsystems@4.2/manual/index.html>

Il vous faudra installer ces packages par l'interface unity.

- Créer un nouveau projet
- Build&settings
  - Switch to android
  - Player settings
    - Remove vulkan
      - dans project settings/Player/Auto graphics Api -> false
      - puis graphics Apis cliquer sur -
    - Multithreading rendering on
    - Android version 24 ou 26 mini
    - XR plugin -> ARCore
- Package manager/ unity registry
  - Ajouter ARFoundation
  - Ajouter ARCore XR plugin ou ARKit XR plugin
- Smartphone en mode debug (cliquez 7 à 10 fois sur la version build de votre smartphone)

### Exercice AR Foundation 1 : AR scene

- Créer une scene
  - Supprimer caméra
  - Ajouter AR session et AR Session Origin
  - Ajouter un cube (0.1cm de côté) dans AR Session Origin
- Build settings
  - Ajouter la scene
- Brancher votre smartphone avec cable USB (vérifier USB debug enable)
- Build and Run



### Exercice AR Foundation 2 : Détection de plan

- Utilisation ARPlane
  - <https://learn.unity.com/tutorial/configuring-plane-detection-for-ar-foundation>
- Créer une nouvelle scene
  - Supprimer caméra
  - Ajouter AR session et AR Session Origin
- Ajouter un AR default plane, ajouter une Sphère et créer un Prefab
- Dans AR Session Origin:
  - ajouter AR plane Manager component
  - Drag&Drop le prefab dans PlanePrefab
- Build & run

## Exercice AR Foundation 3 : Détection d'une image

- Utilisation Image Tracking  
<https://www.youtube.com/watch?v=MdeuA0FITS0>
- Créer une nouvelle scene
  - Supprimer caméra
  - Ajouter AR session et AR Session Origin
- dans AR Session Origin: Add component/AR Tracked Image Manager
- Dans prefabs: create/XR/ReferencelImageLibrary
- Dans ReferencelImageLibrary: Ajouter une image .jpg
- Dans AR Tracked Image Manager:
  - D&D ReferencelImageLibrary et un prefab
  - Ajouter nb detect=2
- Build & run

## Exercice Supplémentaire si vous voulez approfondir : ARFoundation Samples

- Récupérer le projet sur Github : <https://github.com/Unity-Technologies/arfoundation-samples>, regardez suivant votre version unity, pour la LTS 2021.3.19f1 ca obligera à tout recompiler mais ca a fonctionné pour moi. Ca permet de regarder les samples, mais il n'y pas beaucoup de fonctionnalités que celle des exercices précédents
- Dans UnityHub on va charger le projet :
  - Add arfoundation-samples directory
- Build & launch
- Si erreurs de compilation
  - Dans Package Manager: Ajout input systems
  - Dans Build settings/player: allows unsafe mode