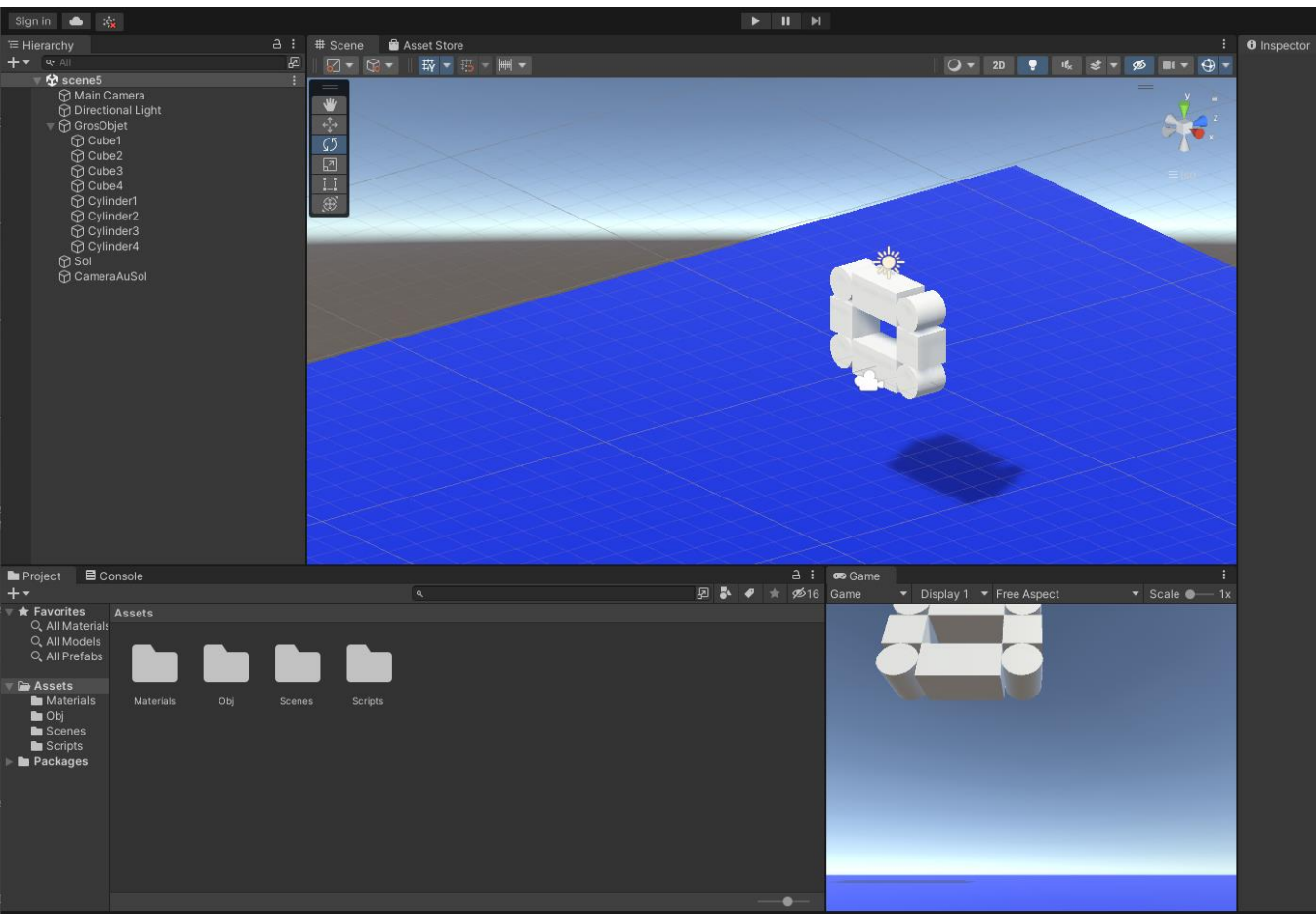


---

# TP UNITY, VENDREDI 21 OCTOBRE 2022



*Reprise de la scène 5, avec un casque VR  
Oculus Quest 1 ou 2*

## *Objectif*

- Rendre compatible la scène 5 du TP avec une utilisation d'un casque VR Oculus Quest 1 ou 2

---

# TP UNITY, VENDREDI 21 OCTOBRE 2022

## *1- Oculus Quest Prise en main du mode Développement*

*L'oculus Quest doit être pris en main via l'application Oculus de votre mobile*

### *Étapes*

#### *Pré-requis:*

- *Avoir un casque Oculus Quest 1 ou 2, ou Rift-S et en connaître les base de fonctionnement*
- *Avoir créé un compte Facebook / Meta*

#### ▪ *Étape 1*

- Allumer le casque Oculus et le mettre sur la tête
- Dans les paramètres du casque, *mettre le Wifi à jour sur la borne qui est utilisée par le mobile*
- Retirer le casque
- Installer de *l'application Oculus* (Play Store ou Apple Store) sur le mobile
- Activer le **Bluetooth** sur le mobile
- Placer le mobile à coté du casque, qui doit être visible sur Bluetooth du mobile.

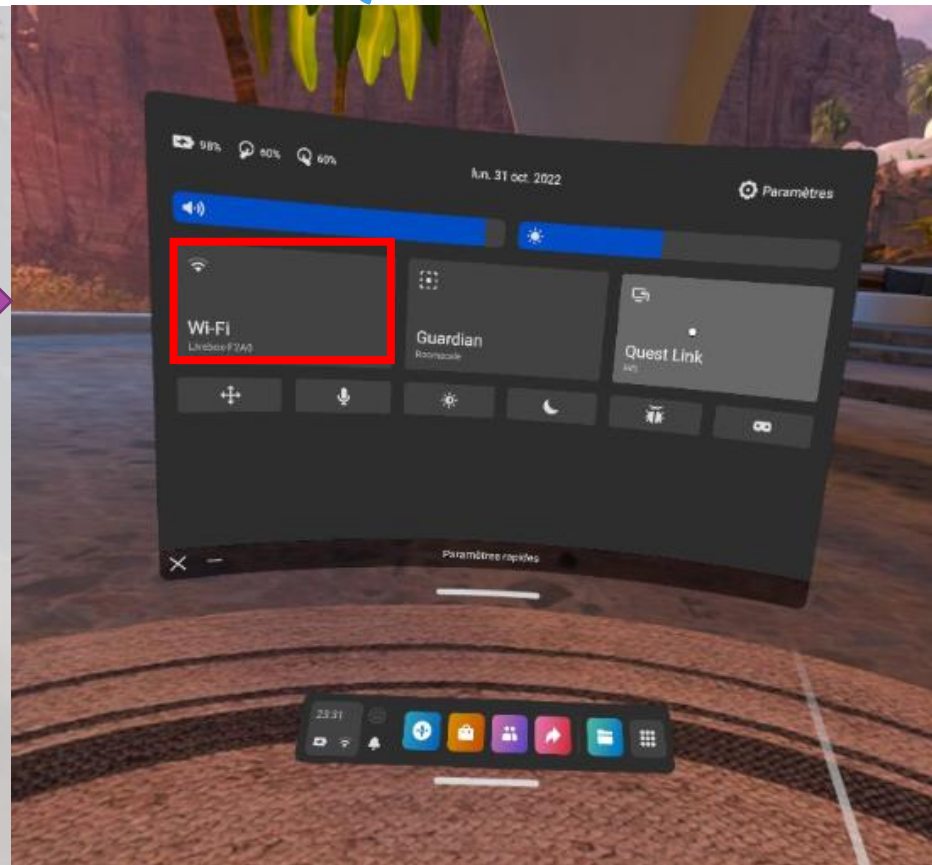
# TP UNITY, VENDREDI 21 OCTOBRE 2022

## 1- Oculus Quest Prise en main du mode Développement

### Mise en œuvre du Wifi sur le casque Oculus Quest 2



Casque sur la tête, appuyer sur le **O** de la manette droite et pointer sur *Paramètres rapides*



Accès au paramétrage du Wifi

---

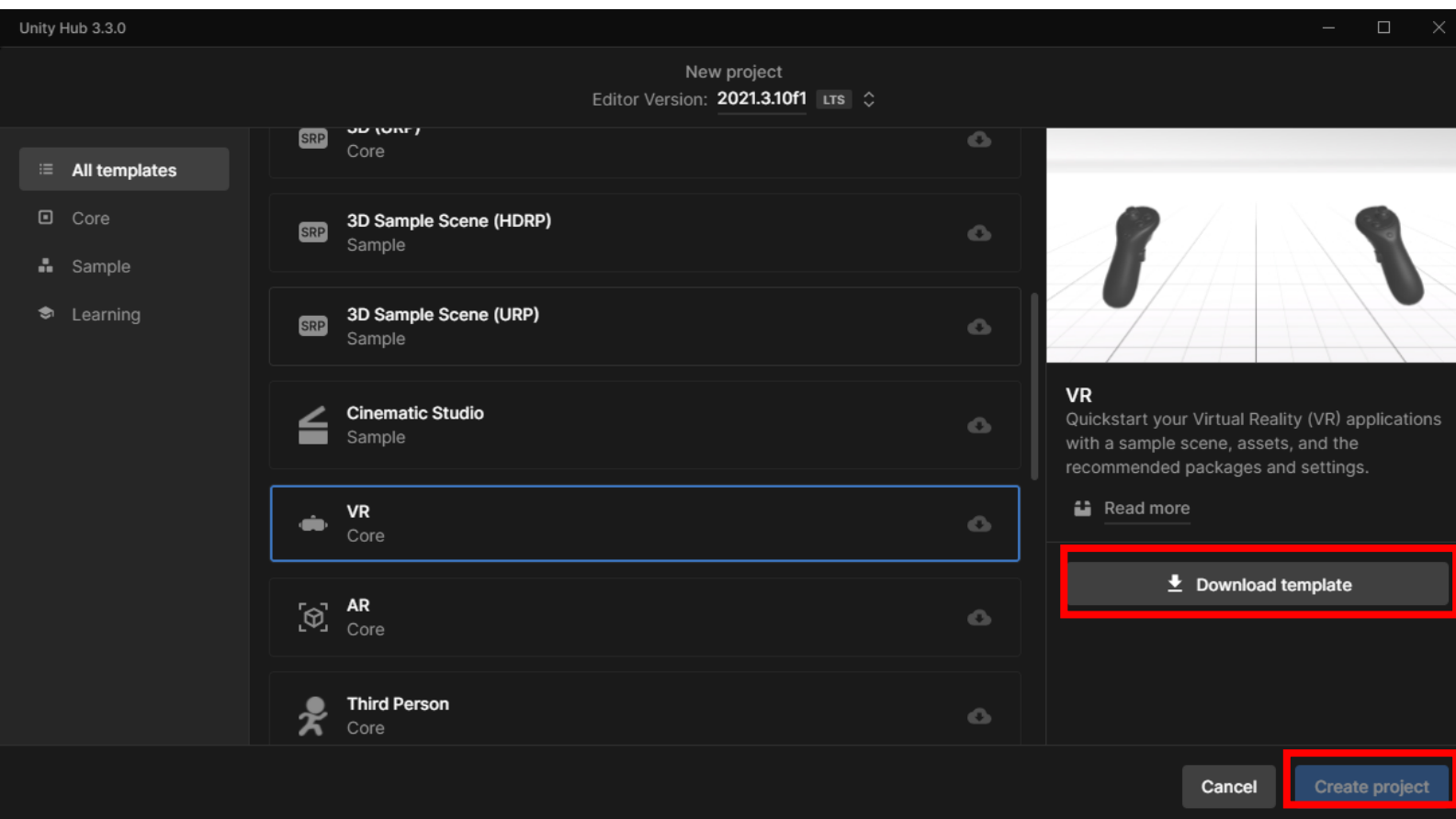
# TP UNITY, VENDREDI 21 OCTOBRE 2022

## **1- Oculus Quest Prise en main du mode Développement**

- *Étape 2: dans l'application Oculus-Meta*
  - Lancer l'application Oculus-Meta sur le mobile, puis **Menu -> Appareils**
  - **-> le casque doit être visible de l'application**
  - Appairer l'Oculus et le mobile via BlueTooth
  - **-> le casque doit à présent être connecté sur l'application**
  - Dans **Paramètres** du Casque
    - Mode développeur, **passer au mode Actif**

# TP UNITY, VENDREDI 21 OCTOBRE 2022

## 2- Installation des éléments nécessaires sur le PC de développement Unity



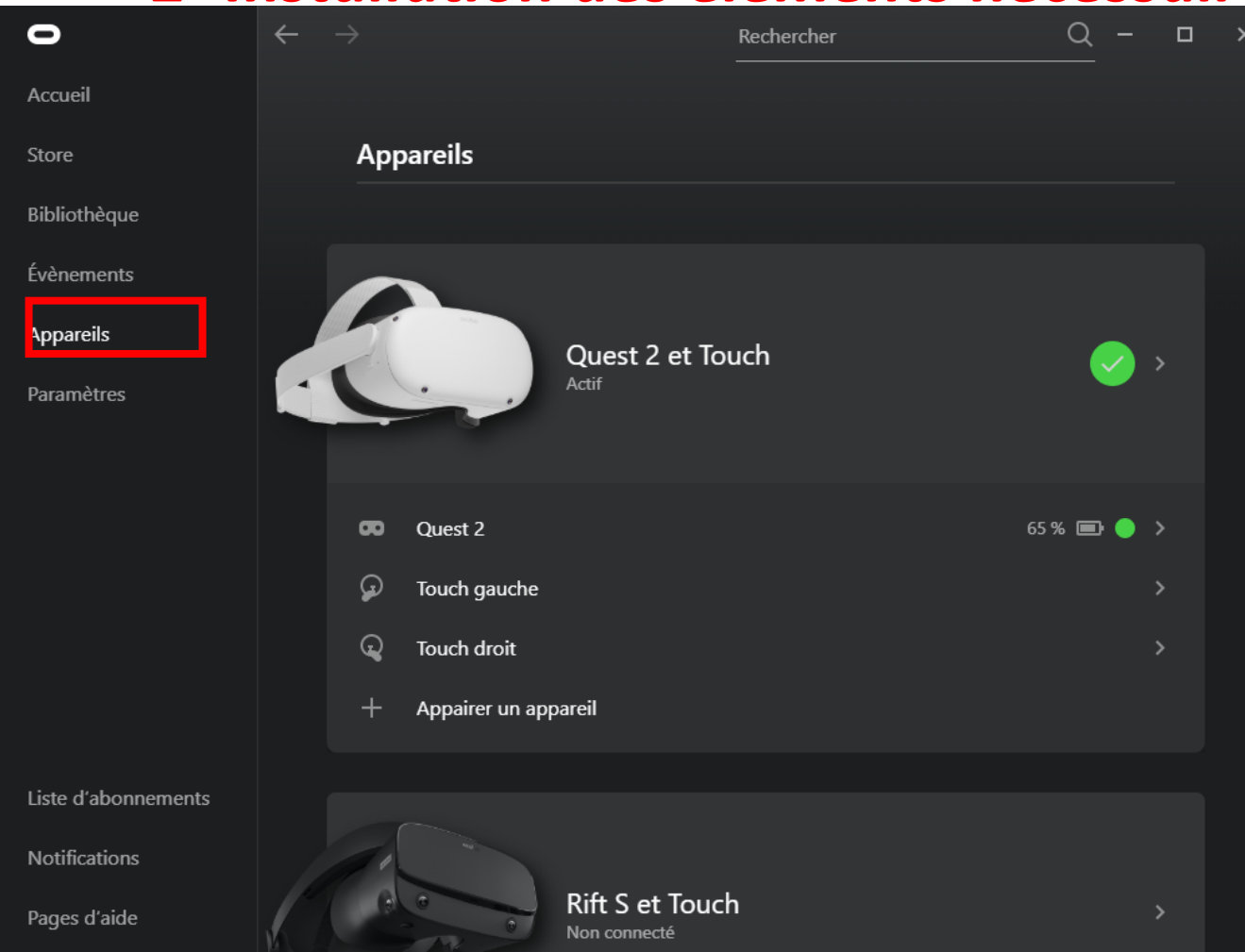
### Étape 3

- Dans Unity Hub, Créez un projet Unity appelé **TPunityVR** dans l'emplacement de votre choix, de type **VR Core**
- Cliquez sur **Download template**
- Puis **Create project**



# TP UNITY, VENDREDI 21 OCTOBRE 2022

## 2- Installation des éléments nécessaires sur le PC de développement Unity



### Étape 3

- Installation de l'application Oculus
- Branchement d'un câble USB-C à l'Oculus vers le PC via **USB-A 3** ou **USB-C** vers le PC qui doit posséder soit entre entrée USB-C ou USB-A 3 (en bleu). **L'USB-A 2 ne fonctionne pas, attention !!!**
- Lancement de l'application Oculus
- Aller dans **Appareils**
- Si tout est Ok, l'appareil est marqué **actif**

---

# TP UNITY, VENDREDI 21 OCTOBRE 2022

## *3- PC de développement - Installation et mise en place de XR Plugin Management*

[Voir Vidéo YouTube [How to Make VR Games in 2022 - Updated Unity VR Tutorial](#) (05/01/2022)]

- Dans **Unity Hub**, Créez un projet Unity appelé **TPunityVR** dans l'emplacement de votre choix, de type **VR Core**
  - Cliquez sur **Download template**
  - Puis **Create project**
- **Aller dans Windows>Package Manager**
- **Les packages nécessaires sont:**
  - **VR (2 packages: Oculus XR Plugin, OpenXR Plugin)**
  - **XR Plugin Management (avec installation des XR Management Implementation)**
  - **XR Interaction Toolkit (recommandé, avec installation de Starter Assets et de XR Device Simulator)**

---

# TP UNITY, VENDREDI 21 OCTOBRE 2022

## *3- PC de développement - Installation et mise en place de XR Plugin Management*

[Voir Vidéo YouTube [How to Make VR Games in 2022 - Updated Unity VR Tutorial \(05/01/2022\)](#)]

- Dans *Windows*->*Package Manager-Unity Registry*, choisir *XR Plugin Management*



Package Manager

⊕ Packages: Unity Registry Sort: Name

- Unity Registry
- In Project
- My Assets
- Built-in

Unity Registry	3.0.0
Unity Technologies	1.6.4 ✓
Unity	1.0.4
Unity	2.2.2 ⓘ
Unity	2.2.5
Unity Profiling Core API	1.0.2
Unity Transport	1.2.0
Unity UI	1.0.0 ✓
Universal RP	12.1.7
Version Control	1.17.2 ⓘ
Visual Effect Graph	12.1.7
Visual Scripting	1.7.8 ✓
Visual Studio Code Editor	1.2.5 ✓
Visual Studio Editor	2.0.16 ✓
WebGL Publisher	4.2.3
Wire	1.1.1
XR Core Utilities	2.0.0
XR Interaction Subsystems	1.0.1
XR Interaction Toolkit	2.0.3
XR Legacy Input Helpers	2.1.10 ✓
XR Plugin Management	4.2.0 ⓘ

## XR Plugin Management Release

Unity Technologies

**Version 4.2.0 - October 21, 2021** ⓘ

Registry Unity

*com.unity.xr.management*

[View documentation](#) · [View changelog](#) · [View licenses](#)

Package that provides simple management of XR plug-ins. Manages and offers help with loading, initialization, settings, and build support for XR plug-ins.

### ▶ Samples

The screenshot shows the Unity Package Manager window. On the left, a list of packages is displayed, with 'XR Plugin Management' selected and highlighted in blue. The main panel on the right shows the details for 'XR Plugin Management', including its version (4.2.0), release date (October 21, 2021), and registry information. A red box highlights the 'XR Plugin Management' title and the 'Release' button. Another red box highlights the 'Import' button next to the 'Example XR Management implementation' sample.

Package Name	Version	Status
Timeline	1.6.4	✓
Tutorial Authoring Tools	1.0.4	
Tutorial Framework	2.2.2	ⓘ
Unity Distribution Portal	2.2.5	
Unity Profiling Core API	1.0.2	
Unity Transport	1.2.0	
Unity UI	1.0.0	✓
Universal RP	12.1.7	
Version Control	1.17.2	ⓘ
Visual Effect Graph	12.1.7	
Visual Scripting	1.7.8	✓
Visual Studio Code Editor	1.2.5	✓
Visual Studio Editor	2.0.16	✓
WebGL Publisher	4.2.3	
Wire	1.1.1	
XR Core Utilities	2.0.0	
XR Interaction Subsystems	1.0.1	
XR Interaction Toolkit	2.0.3	
XR Legacy Input Helpers	2.1.10	✓
<b>XR Plugin Management</b>	<b>4.2.0</b>	<b>ⓘ</b>

**XR Plugin Management** Release

Unity Technologies  
Version 4.2.0 - October 21, 2021 ⓘ  
Registry Unity  
com.unity.xr.management  
[View documentation](#) · [View changelog](#) · [View licenses](#)

Package that provides simple management of XR plug-ins. Manages and offers help with loading, initialization, settings, and build support for XR plug-ins.

**Samples**

Example XR Management implementation Import  
21,44 KB

- Cliquer sur *Import*

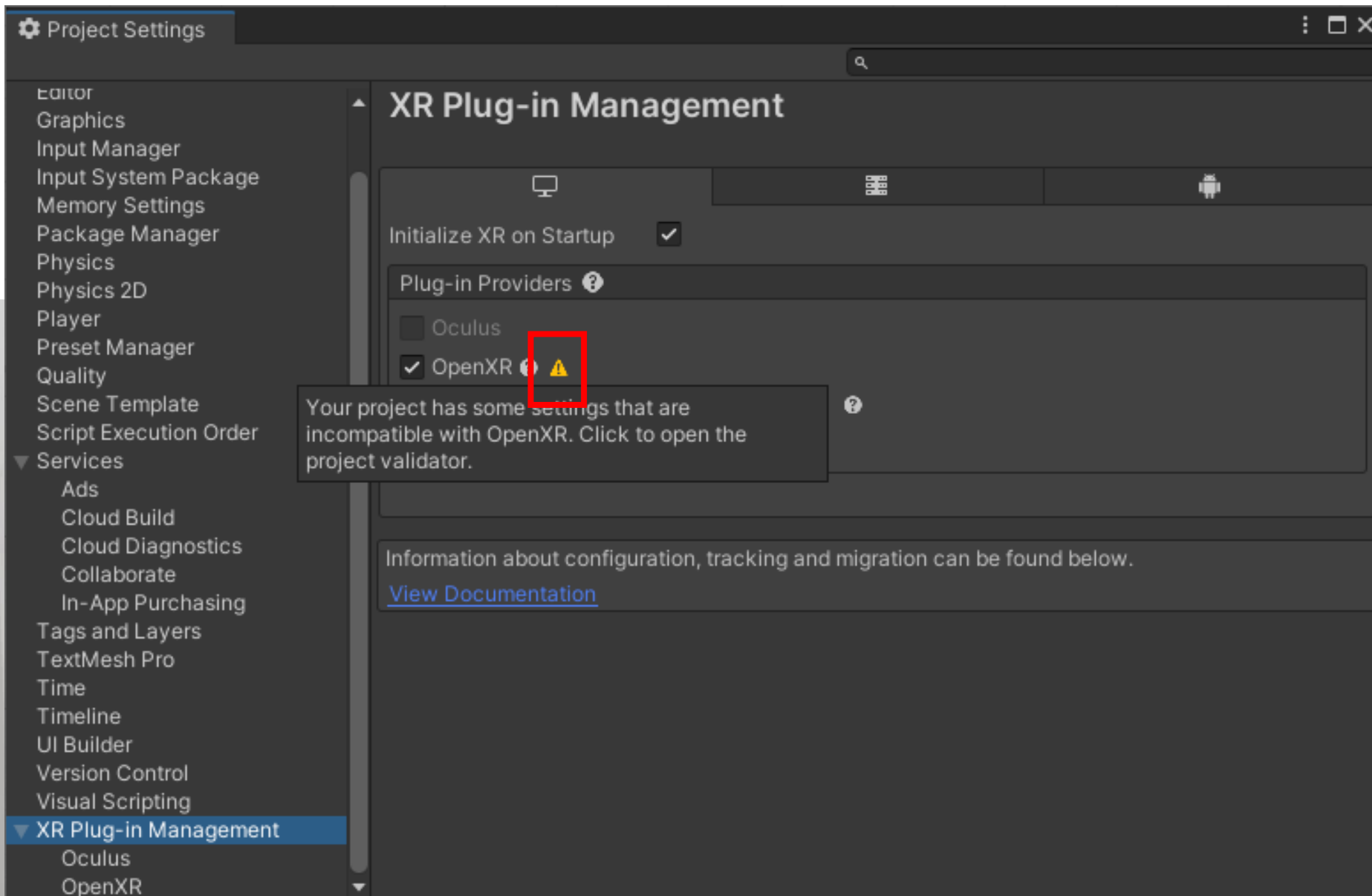
---

# TP UNITY, VENDREDI 21 OCTOBRE 2022

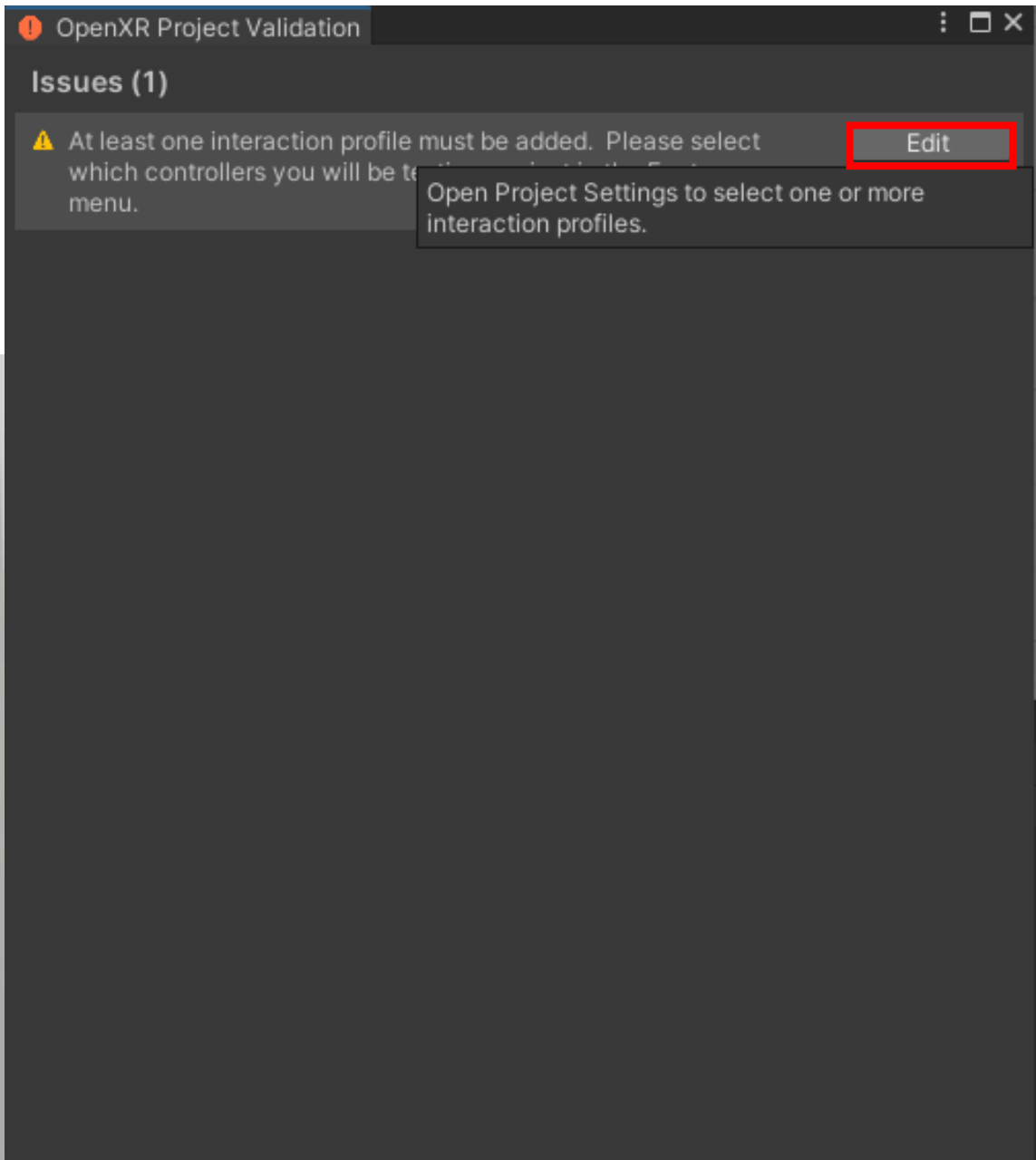
## *3- PC de développement - Installation et mise en place de XR Plugin Management*

[Voir Vidéo YouTube [How to Make VR Games in 2022 - Updated Unity VR Tutorial \(05/01/2022\)](#)]

- Dans *Edit->Project Settings*,
  - Cliquer sur *XR Plugin Management*
  - Dans la partie Windows, Cocher *OpenXR*

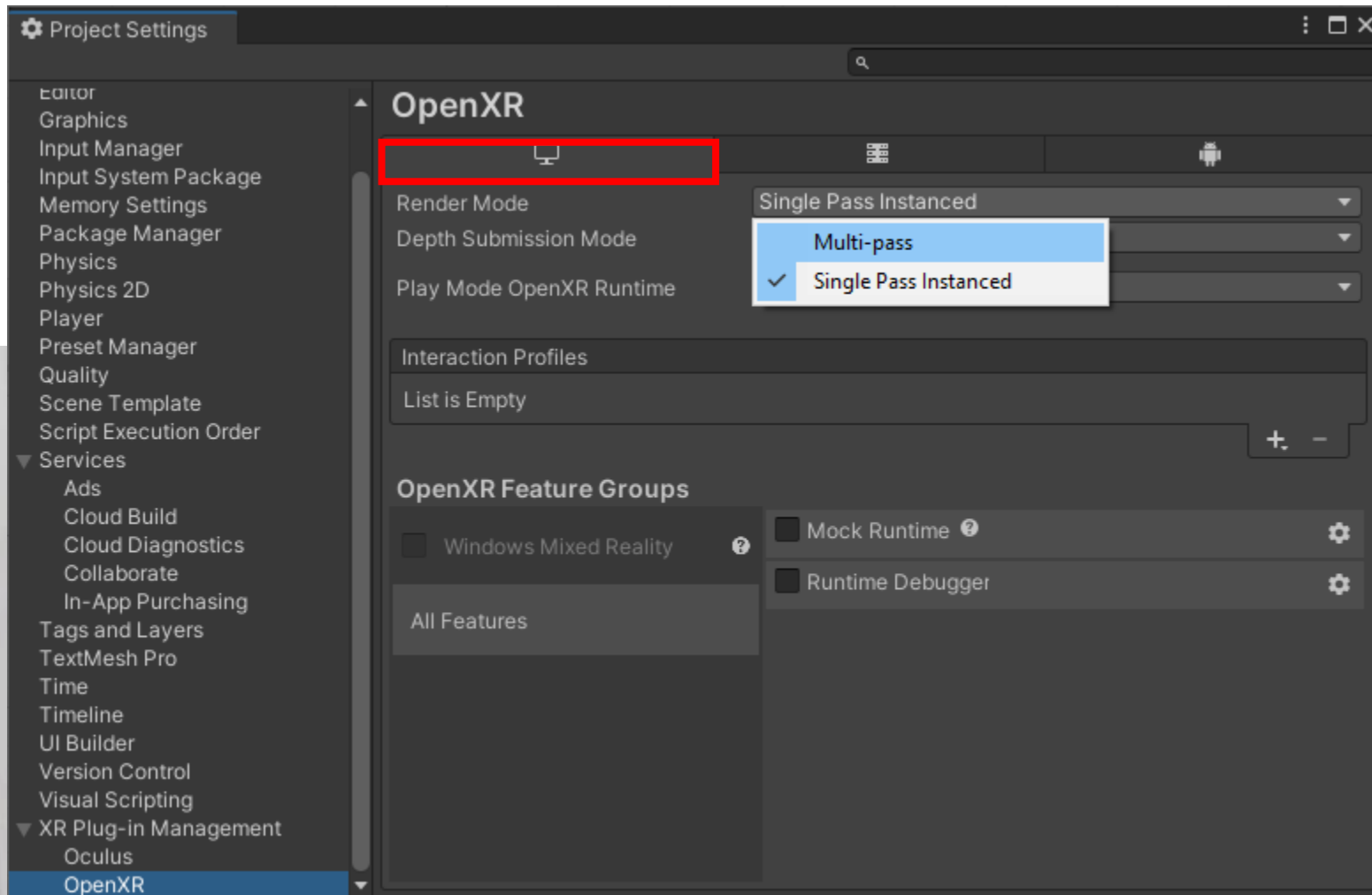


- Cliquer sur l'avertissement

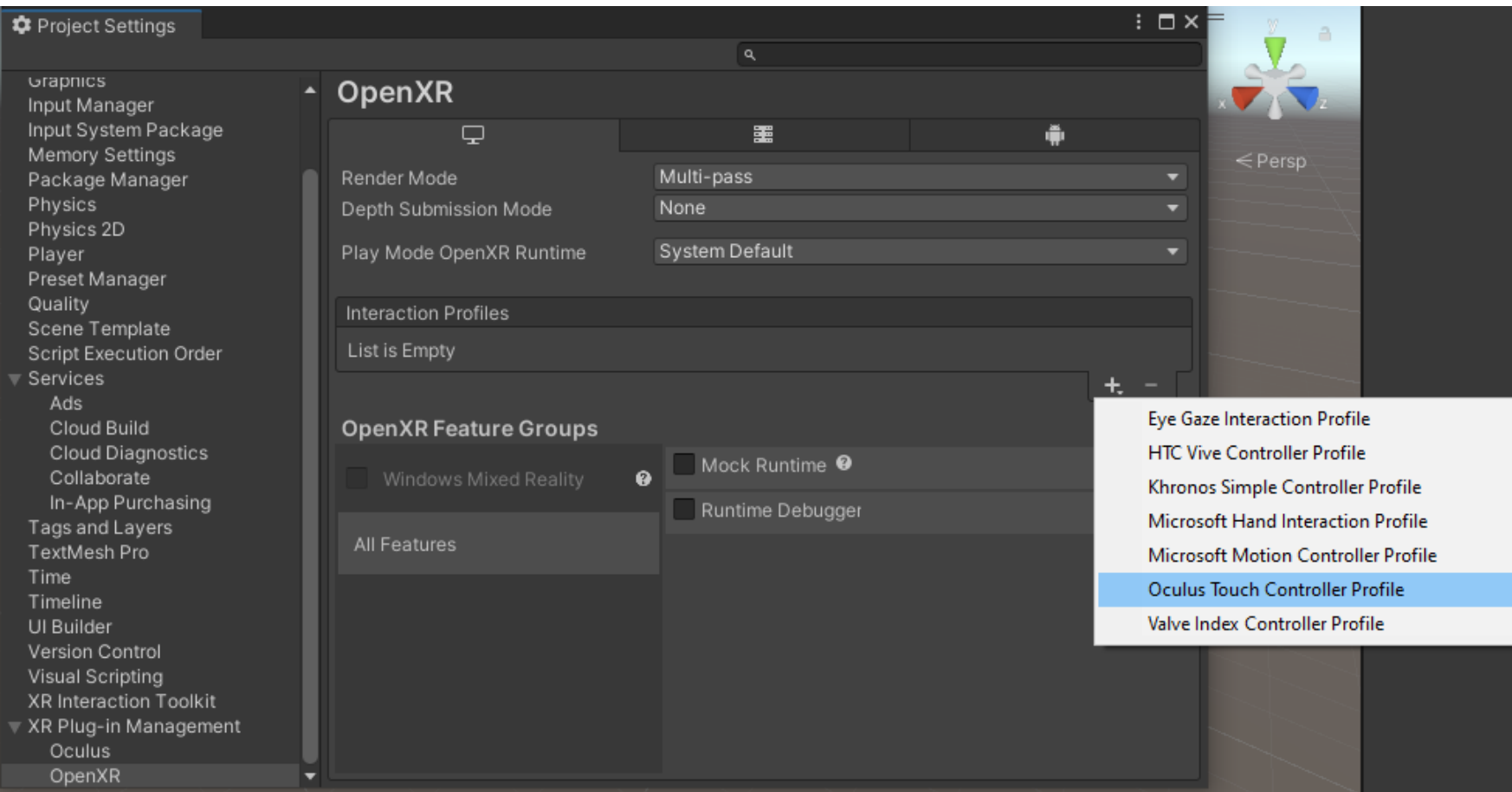


- Cliquer sur *Edit*

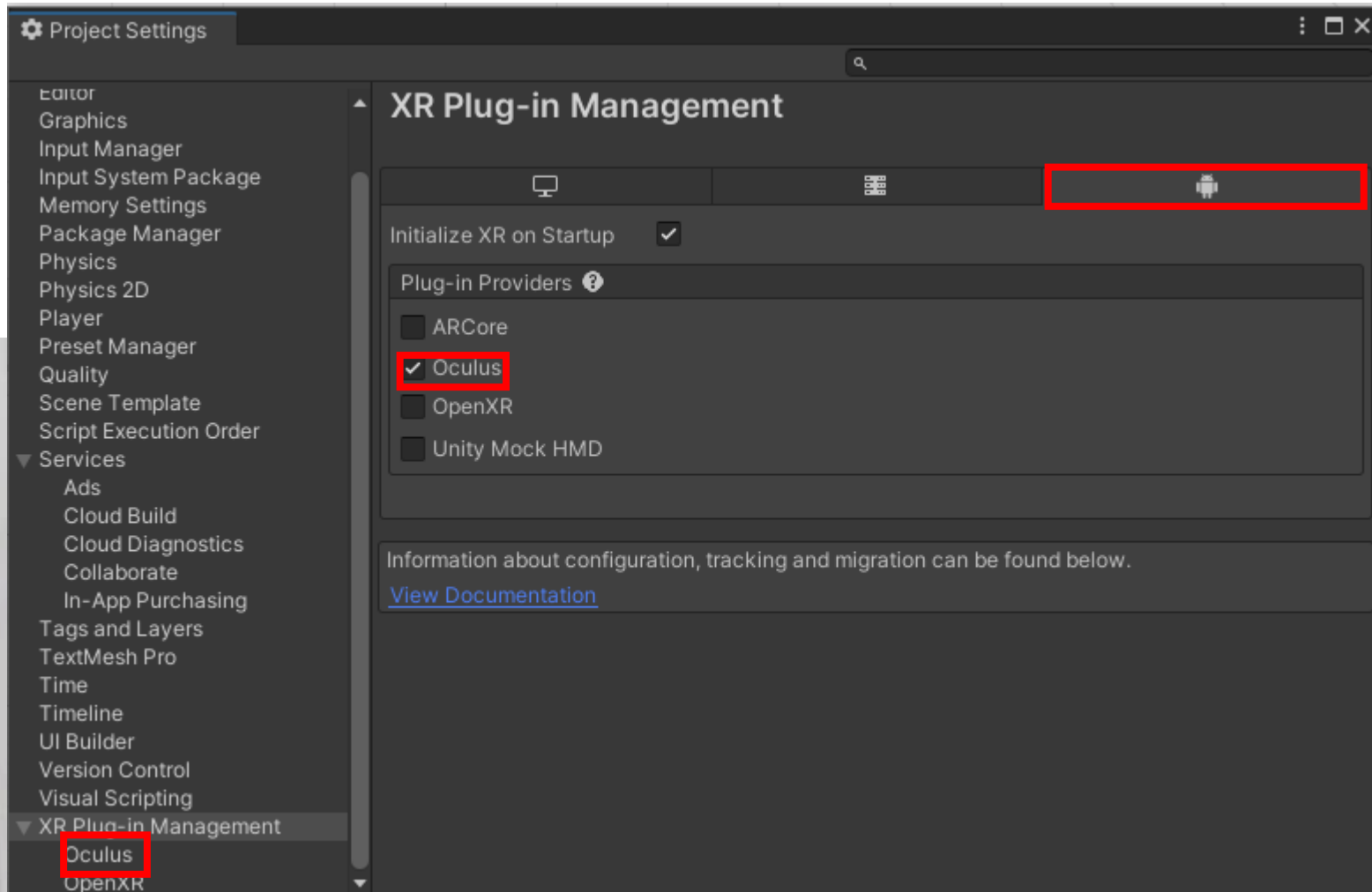




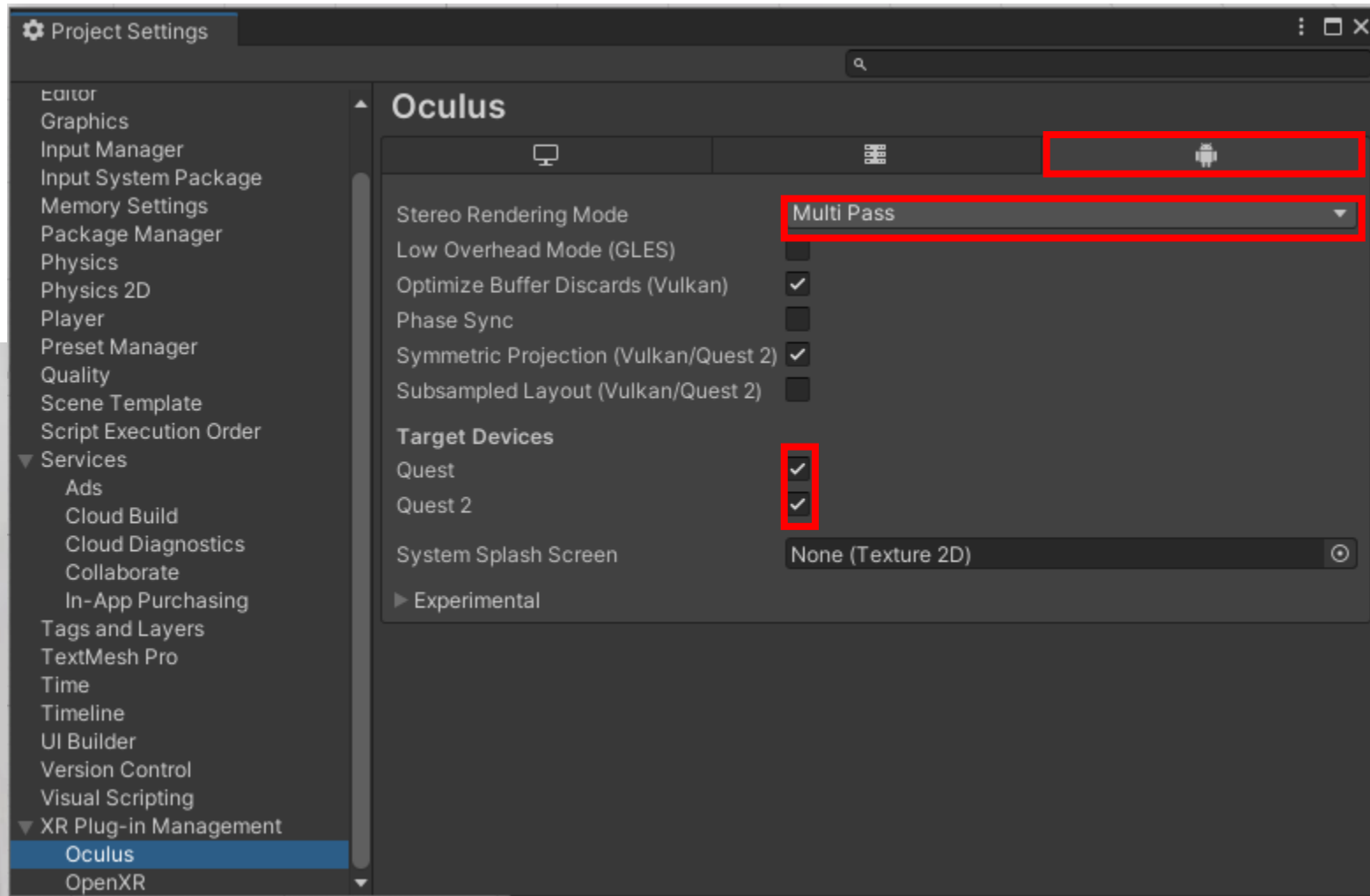
- XR Plug-in Management – OpenXR, *Windows*
  - Render Mode
    - *Cliquer sur Multi-pass*



- XR Plug-in Management – OpenXR, *Windows*
  - Interaction Profiles
    - *Cliquer sur + et choisir Touch Controller Profile*



- XR Plug-in Management –, ***Android***
  - Plug-in Providers
    - ***Oculus***  
***-> Oculus***  
***apparaît dans***  
***XR Plug-in***  
***Management***



- XR Plug-in Management –, *Oculus*
- Dans *Android*
  - Cocher *Multi Pass* dans *Stereo Rendering Mode*
  - Cocher *Quest* et *Quest2* dans *Target Devices*

---

# TP UNITY, VENDREDI 21 OCTOBRE 2022

## *4- PC de développement - Installation et mise en place de XR Interaction Toolkit*

[Voir Vidéo YouTube [How to Make VR Games in 2022 - Updated Unity VR Tutorial](#) (05/01/2022)]

- Dans *Windows*->*Package Manager-Unity Registry*, choisir *XR Interaction Toolkit*



Unity Package Manager interface showing the XR Interaction Toolkit package details.

Package Name	Version
Timeline	1.6.4 ✓
Tutorial Authoring Tools	1.0.4
Tutorial Framework	2.2.2 ⓘ
Unity Distribution Portal	2.2.5
Unity Profiling Core API	1.0.2
Unity Transport	1.2.0
Unity UI	1.0.0 ✓
Universal RP	12.1.7
Version Control	1.17.2 ⓘ
Visual Effect Graph	12.1.7
Visual Scripting	1.7.8 ✓
Visual Studio Code Editor	1.2.5 ✓
Visual Studio Editor	2.0.16 ✓
WebGL Publisher	4.2.3
Wire	1.1.1
XR Core Utilities	2.0.0
XR Interaction Subsystems	1.0.1
<b>XR Interaction Toolkit</b>	<b>2.0.3</b>
XR Legacy Input Helpers	2.1.10 ✓
XR Plugin Management	4.2.0 ⓘ

**XR Interaction Toolkit** Release

Unity Technologies  
**Version 2.0.3 - August 15, 2022**  
Registry Unity  
*com.unity.xr.interaction.toolkit*  
[View documentation](#) · [View changelog](#) · [View licenses](#)

A high-level, component-based, interaction system for creating VR and AR experiences. It provides a framework that makes 3D and UI interactions available from Unity input events. The core of this system is a set of base Interactor and Interactable components, and an Interaction Manager that ties these two types of components together. It also contains components that you can use for locomotion and drawing visuals.

Last update Oct 31, 13:03 ↻

**Install**

## XR Interaction Toolkit

- Cliquer sur *Install*

The screenshot shows the Unity Package Manager interface. On the left, a list of packages is shown, with 'XR Interaction Toolkit' selected at version 2.0.3. The main panel displays the details for this package, including its name, version (2.0.3), release date (August 15, 2022), and registry (Unity). Below the description, there is a section for 'Samples' which is highlighted with a red box. This section lists two samples: 'Starter Assets' (93,49 KB) and 'XR Device Simulator' (41,13 KB), each with an 'Import' button also highlighted with a red box. At the bottom right, there is a 'Remove' button.

Package Name	Version
Test Framework	1.1.31
TextMeshPro	3.0.6
Timeline	1.6.4
Tutorial Authoring Tools	1.0.4
Tutorial Framework	2.2.2
Unity Distribution Portal	2.2.5
Unity Profiling Core API	1.0.2
Unity Transport	1.2.0
Unity UI	1.0.0
Universal RP	12.1.7
Version Control	1.17.2
Visual Effect Graph	12.1.7
Visual Scripting	1.7.8
Visual Studio Code Editor	1.2.5
Visual Studio Editor	2.0.16
WebGL Publisher	4.2.3
Wire	1.1.1
XR Core Utilities	2.0.0
XR Interaction Subsystems	1.0.1
<b>XR Interaction Toolkit</b>	<b>2.0.3</b>

**XR Interaction Toolkit** Release

Unity Technologies  
Version 2.0.3 - August 15, 2022  
Registry Unity  
`com.unity.xr.interaction.toolkit`  
[View documentation](#) · [View changelog](#) · [View licenses](#)

A high-level, component-based, interaction system for creating VR and AR experiences. It provides a framework that makes 3D and UI interactions available from Unity input events. The core of this system is a set of base Interactor and Interactable components, and an Interaction Manager that ties these two types of components together. It also contains components that you can use for locomotion and drawing visuals.

**▼ Samples**

Starter Assets 93,49 KB	<b>Import</b>
XR Device Simulator 41,13 KB	<b>Import</b>

Last update Oct 31, 13:03 Refresh Remove



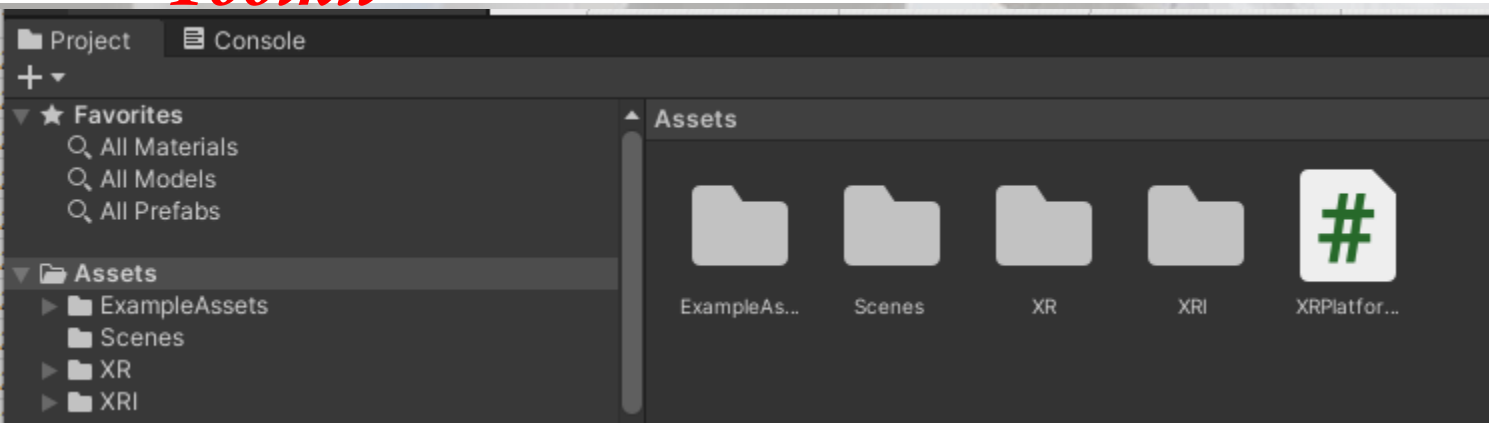
### Dans Samples

- Starter Assets *Import*
- XR Device Simulator *Import*

---

# TP UNITY, VENDREDI 21 OCTOBRE 2022

## *4- PC de développement - Installation et mise en place de XR Interaction Toolkit*



Dans *Project*

- Nous avons à présent plusieurs répertoires.

---

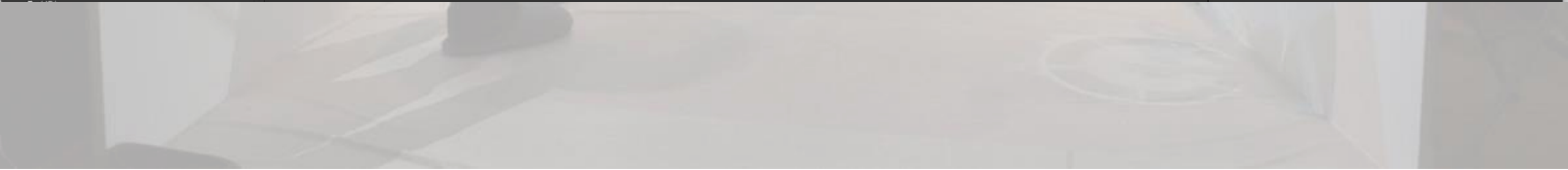
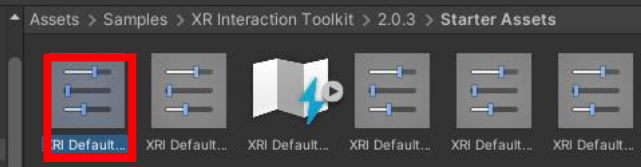
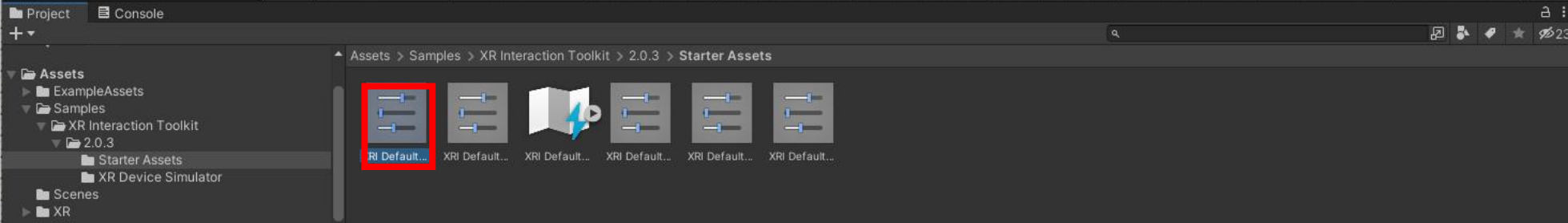
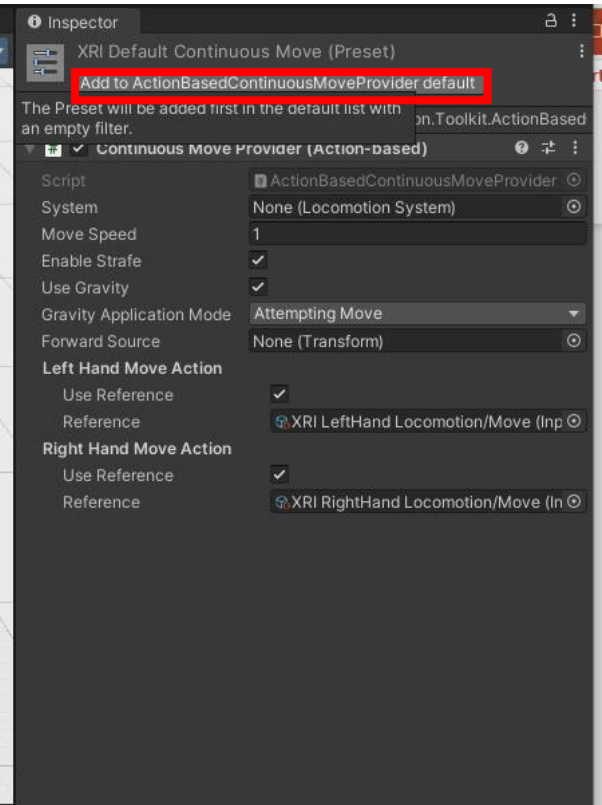
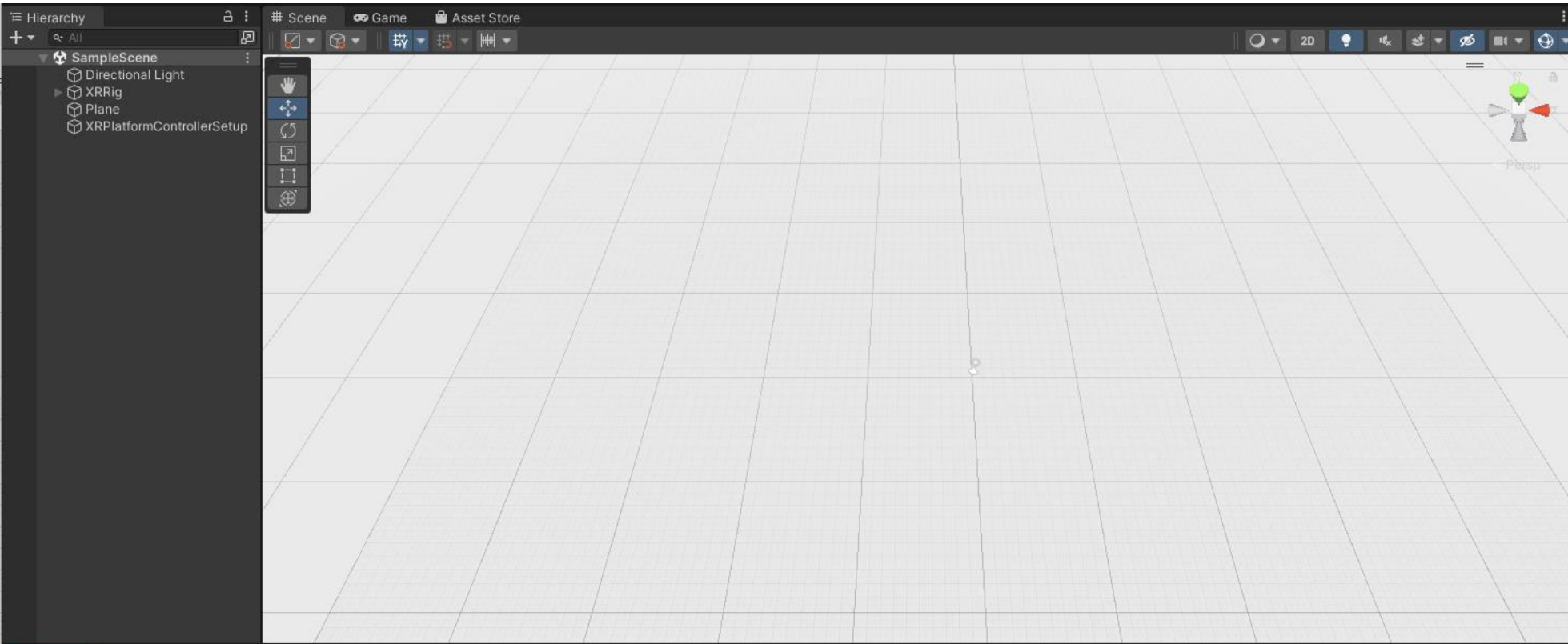
# TP UNITY, VENDREDI 21 OCTOBRE 2022

## *4- PC de développement - Installation et mise en place de XR Interaction Toolkit*

*Mise à jour des presets*

[Voir Vidéo YouTube [How to Make VR Games in 2022 - Updated Unity VR Tutorial \(05/01/2022\)](#)]

- Dans *Assets->Samples->XR Interaction Toolkit-><version>->Starter Assets*,
  - Cliquer sur *XR Default Continuous Move.preset*
  - Dans la fenêtre Inspector, cliquer sur *Add to ActionBaseContinuousMoveProvider default*





---

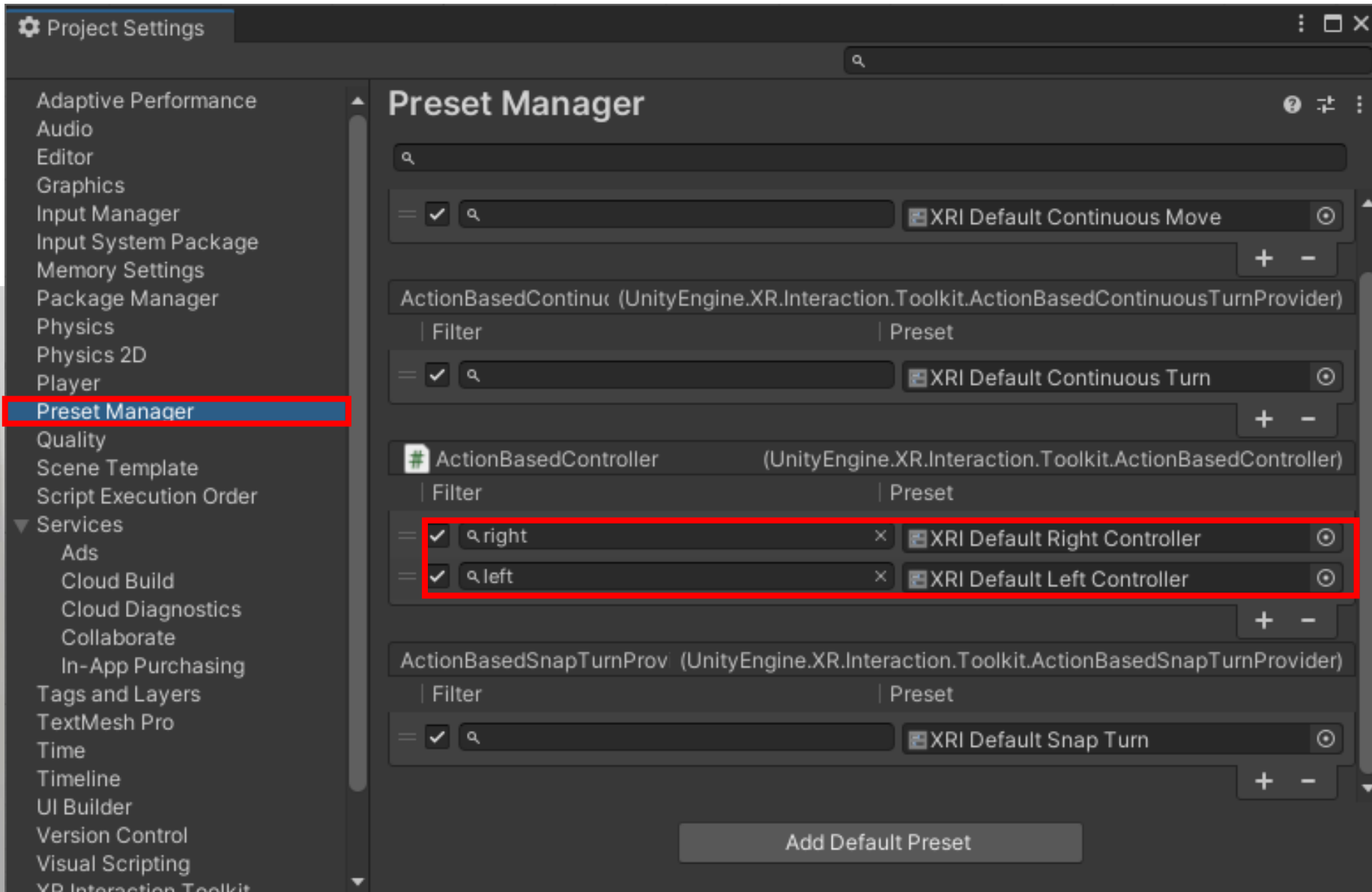
# TP UNITY, VENDREDI 21 OCTOBRE 2022

## *4- PC de développement - Installation et mise en place de XR Interaction Toolkit*

*Mise à jour des presets*

[Voir Vidéo YouTube [How to Make VR Games in 2022 - Updated Unity VR Tutorial](#) (05/01/2022)]

- Dans *Assets->Samples->XR Interaction Toolkit-><version>->Starter Assets*,
  - Faire les deux mêmes opérations pour:
    - *XR Default Continuous Move.preset*
    - *XR Default Continuous Turn.preset*
    - *XR Default Left Controller.preset*
    - *XR Default Right Controller.preset*



Dans *Project Setting* ->  
*Preset Manager*

- Mettre **right** dans Filter XRI Default Right Controller
- Mettre **left** dans Filter XRI Default Left Controller

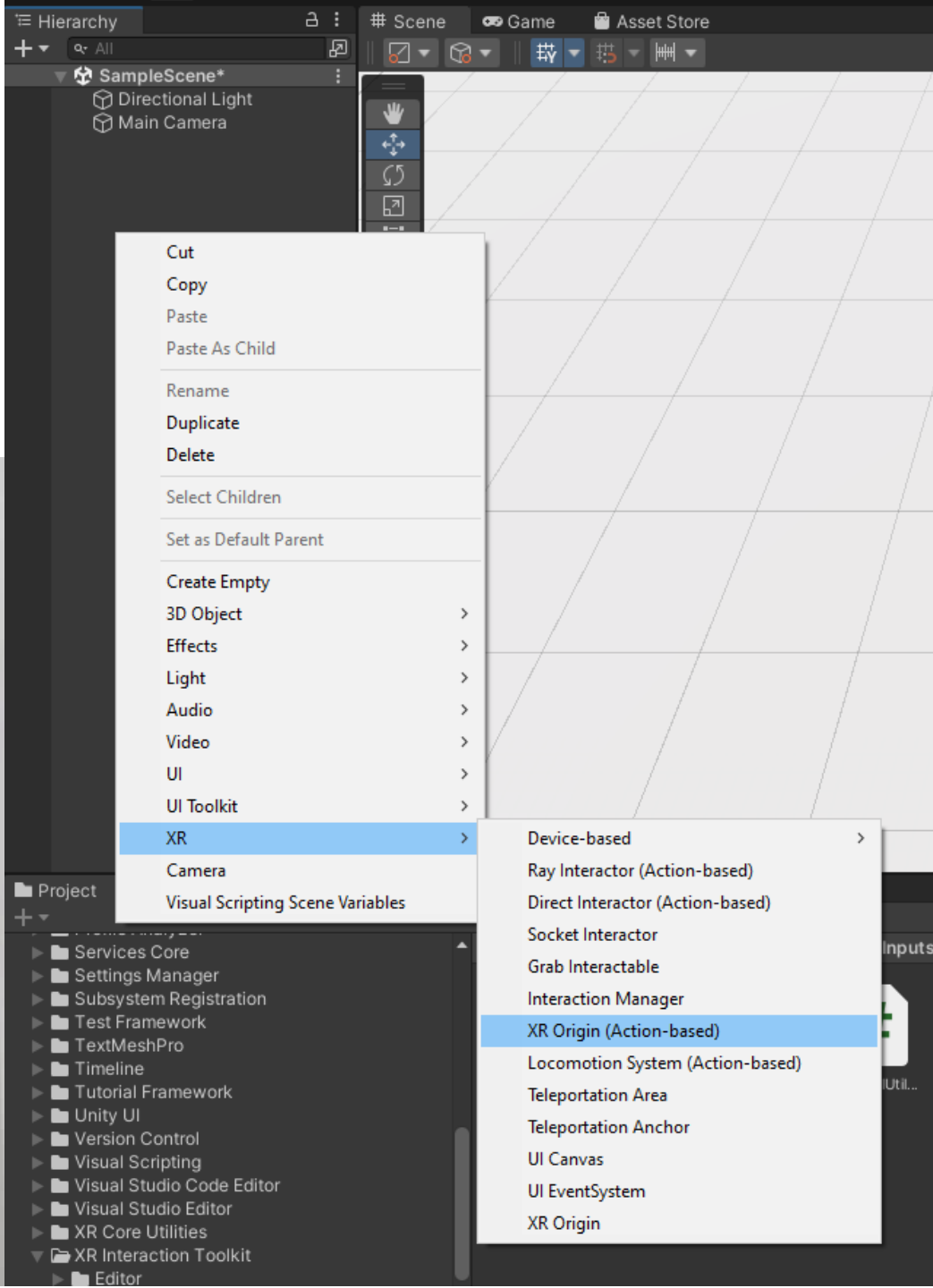
---

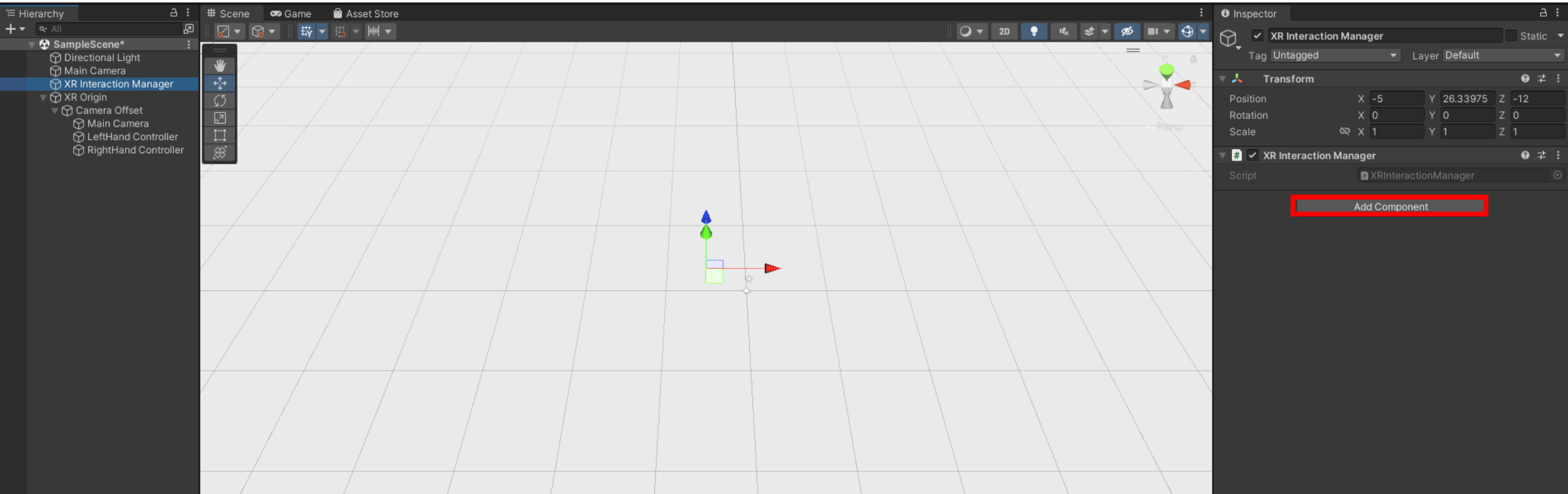
# TP UNITY, VENDREDI 21 OCTOBRE 2022

*5- PC de développement – Scène de test – Caméra liée au Ray Casting lié aux contrôleurs*

[Voir Vidéo YouTube [How to Make VR Games in 2022 - Updated Unity VR Tutorial](#) (05/01/2022)]

- Dans *Hierarchy*
  - Clic droit -> XR -> XR Origin (Action based)





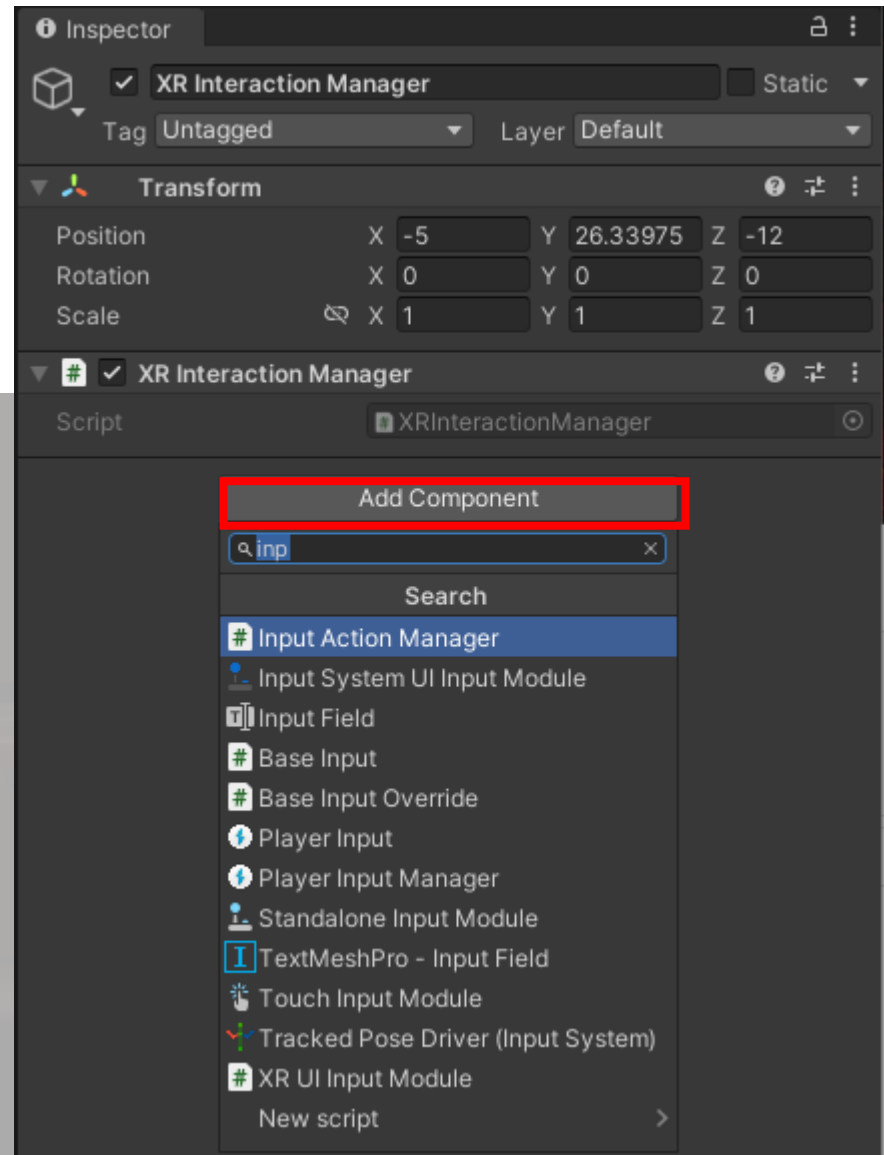
[Voir Vidéo YouTube [How to Make VR Games in 2022 - Updated Unity VR Tutorial \(05/01/2022\)](#)]

- Dans *Inspector* de *XR Interaction Manager*
  - **Add Component**



[Voir Vidéo YouTube [How to Make VR Games in 2022 - Updated Unity VR Tutorial](#) (05/01/2022)]

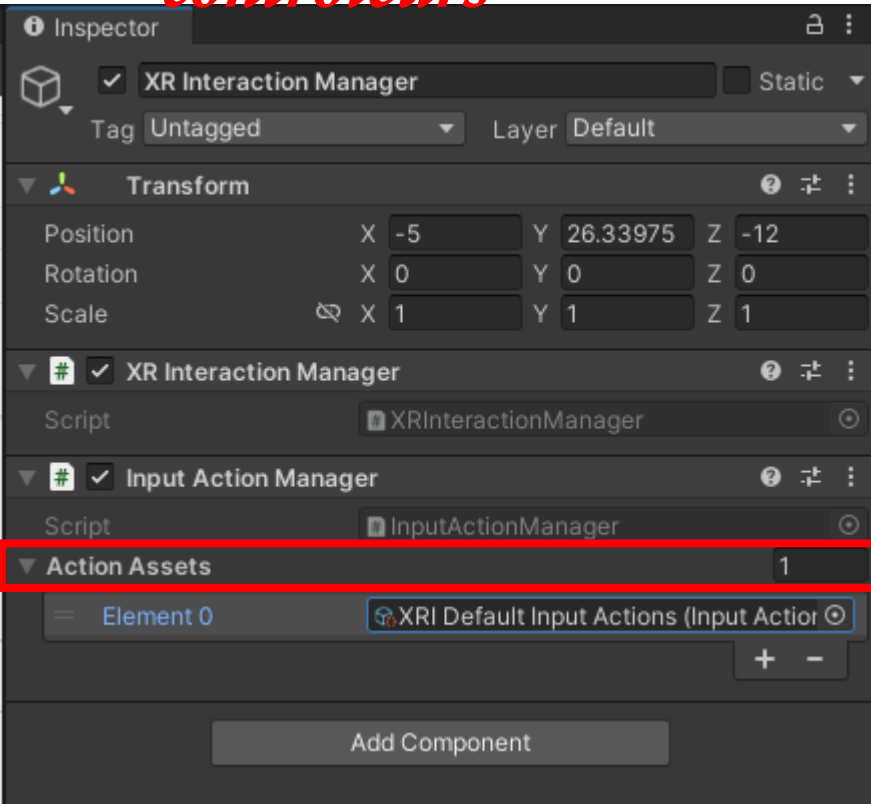
- Dans *Add Component* de *Inspector* de *XR Interaction Manager*
- Rajouter *Input Action Manager*



---

# TP UNITY, VENDREDI 21 OCTOBRE 2022

*5- PC de développement – Scène de test – Caméra liée au Ray Casting lié aux contrôleurs*



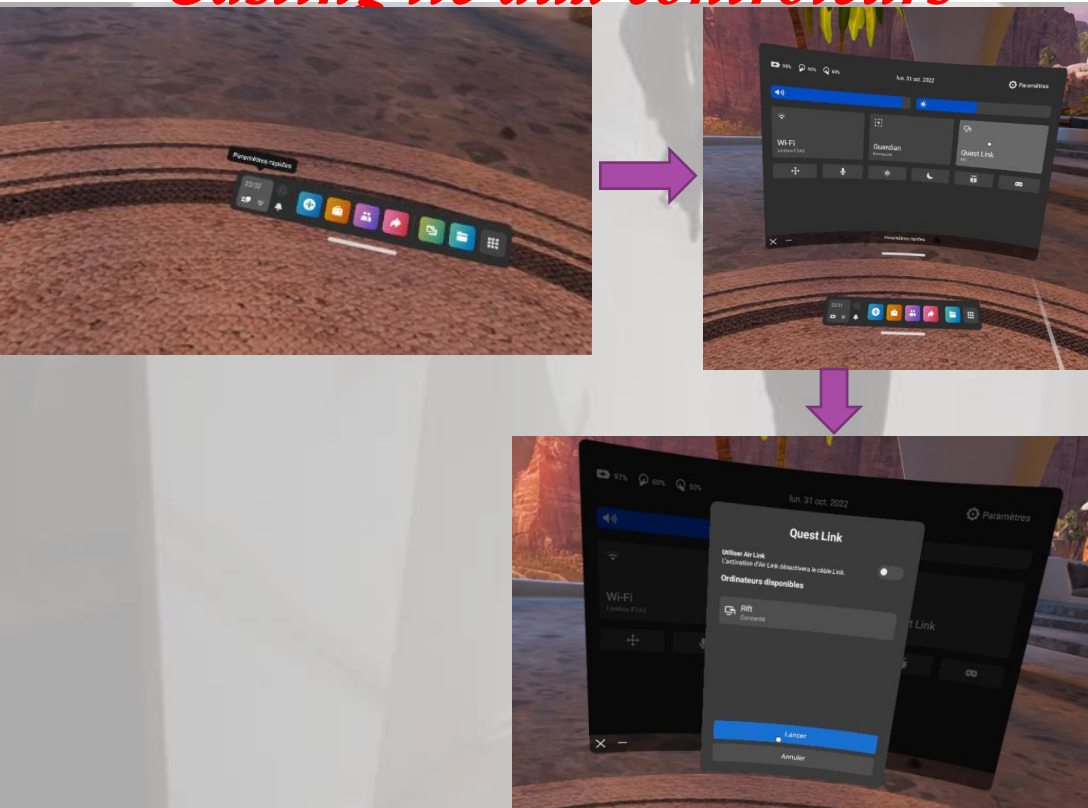
[Voir Vidéo YouTube [How to Make VR Games in 2022 - Updated Unity VR Tutorial](#) (05/01/2022)]

- Dans *Action Assets* de *Inspector* de *XR Interaction Manager*
  - Rajouter *XRI Default Input Actions*
- Sauver la scène en **ScèneTest**

---

# TP UNITY, VENDREDI 21 OCTOBRE 2022

## *6- PC de développement – Run de la Scène de test – Caméra liée au Ray Casting lié aux contrôleurs*

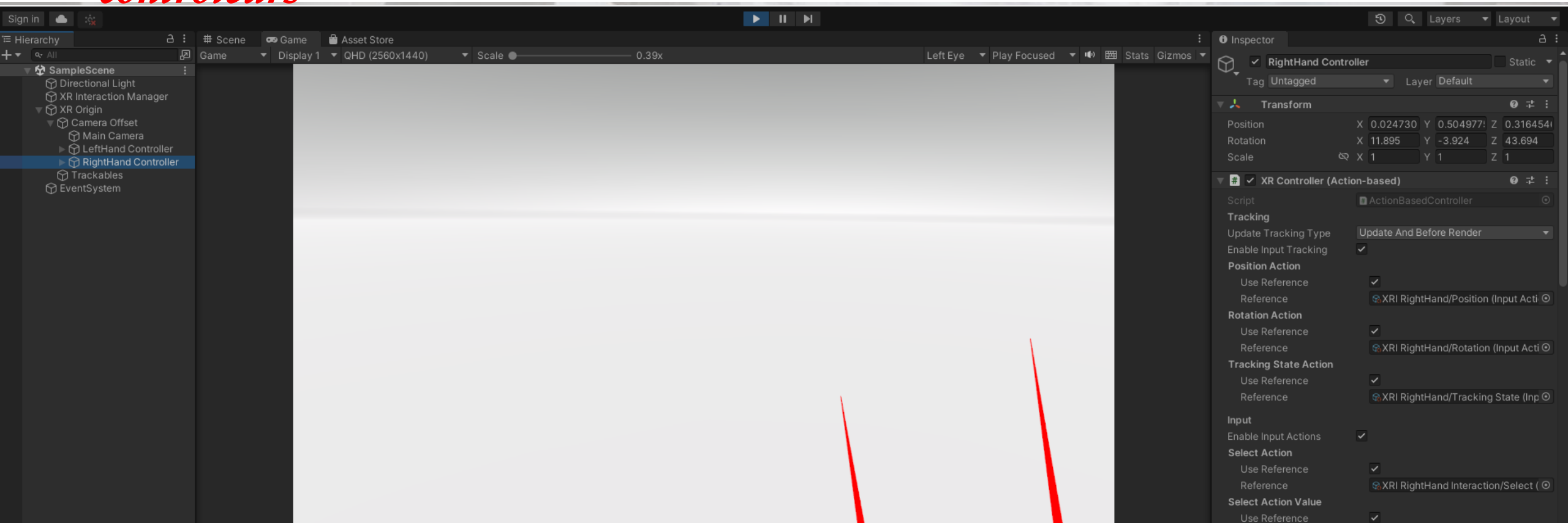


- Activer le Quest Link à partir du Casque
- Le contrôleur pilote le curseur de la souris du PC et un clic de la gâchette du contrôleur peut lancer le programme Unity

---

# TP UNITY, VENDREDI 21 OCTOBRE 2022

*5- PC de développement – Scène de test – Caméra liée au Ray Casting lié aux contrôleurs*



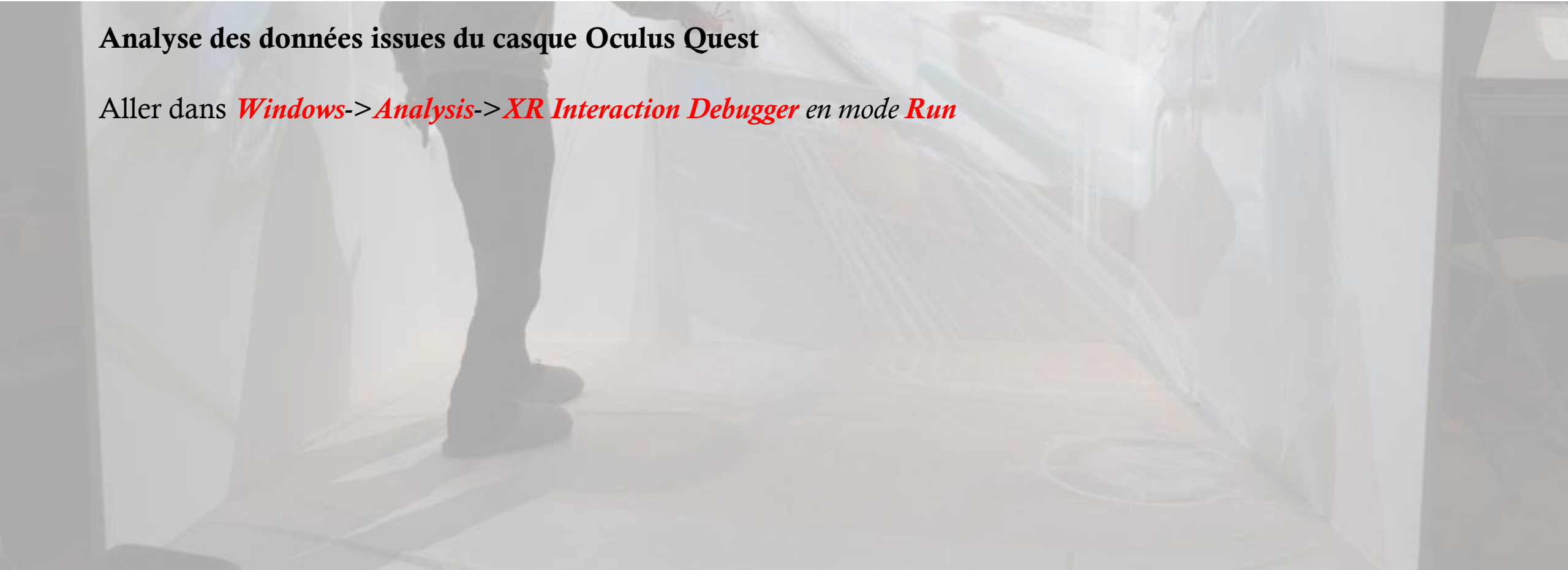
---

# TP UNITY, VENDREDI 21 OCTOBRE 2022

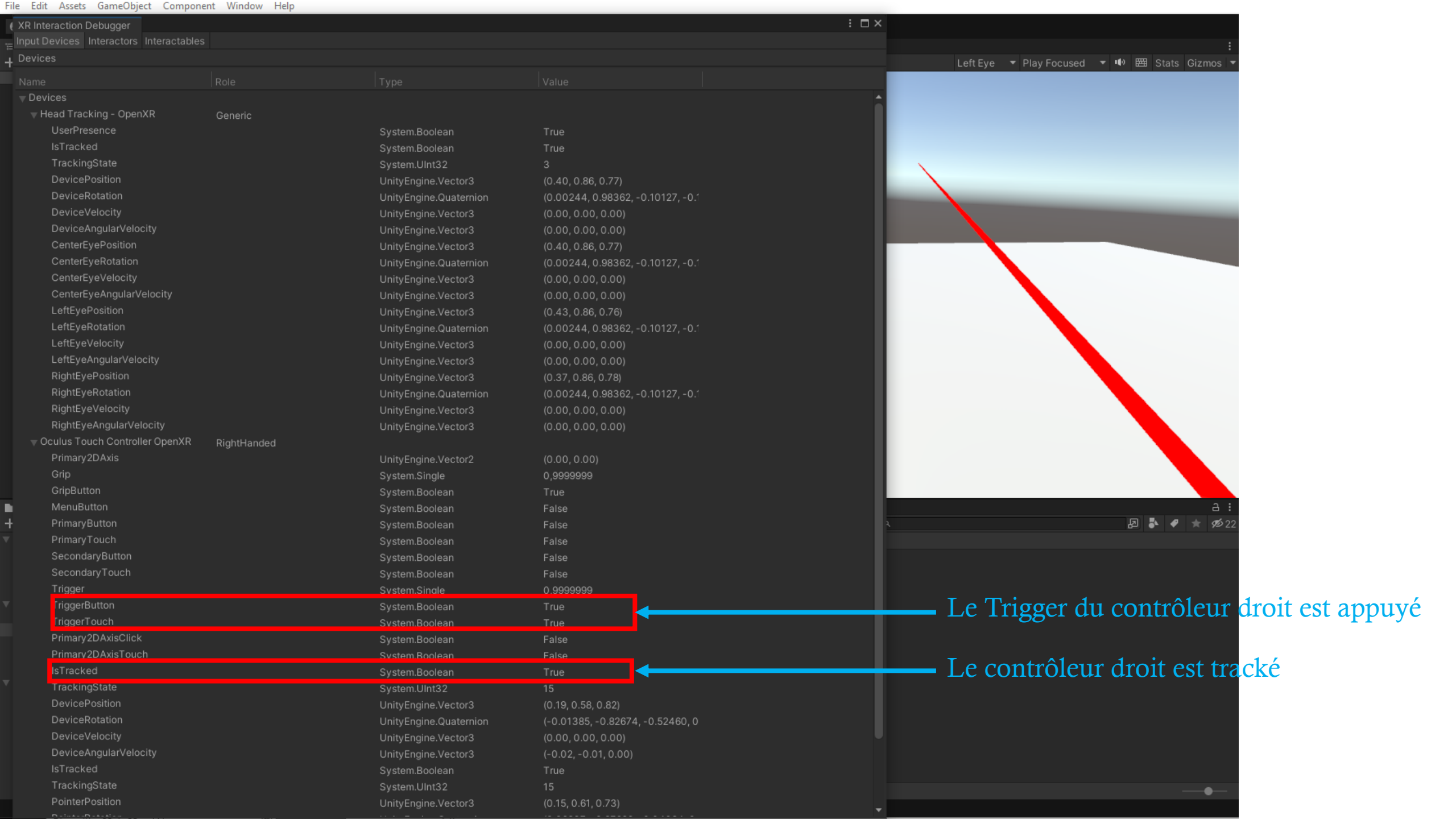
*5- PC de développement – Scène de test – Caméra liée au Ray Casting lié aux contrôleurs*

Analyse des données issues du casque Oculus Quest

Aller dans *Windows->Analysis->XR Interaction Debugger* en mode *Run*









---

# TP UNITY, VENDREDI 21 OCTOBRE 2022

## *6- PC de développement – Scène de test – Caméra liée au Ray Casting lié aux contrôleurs*

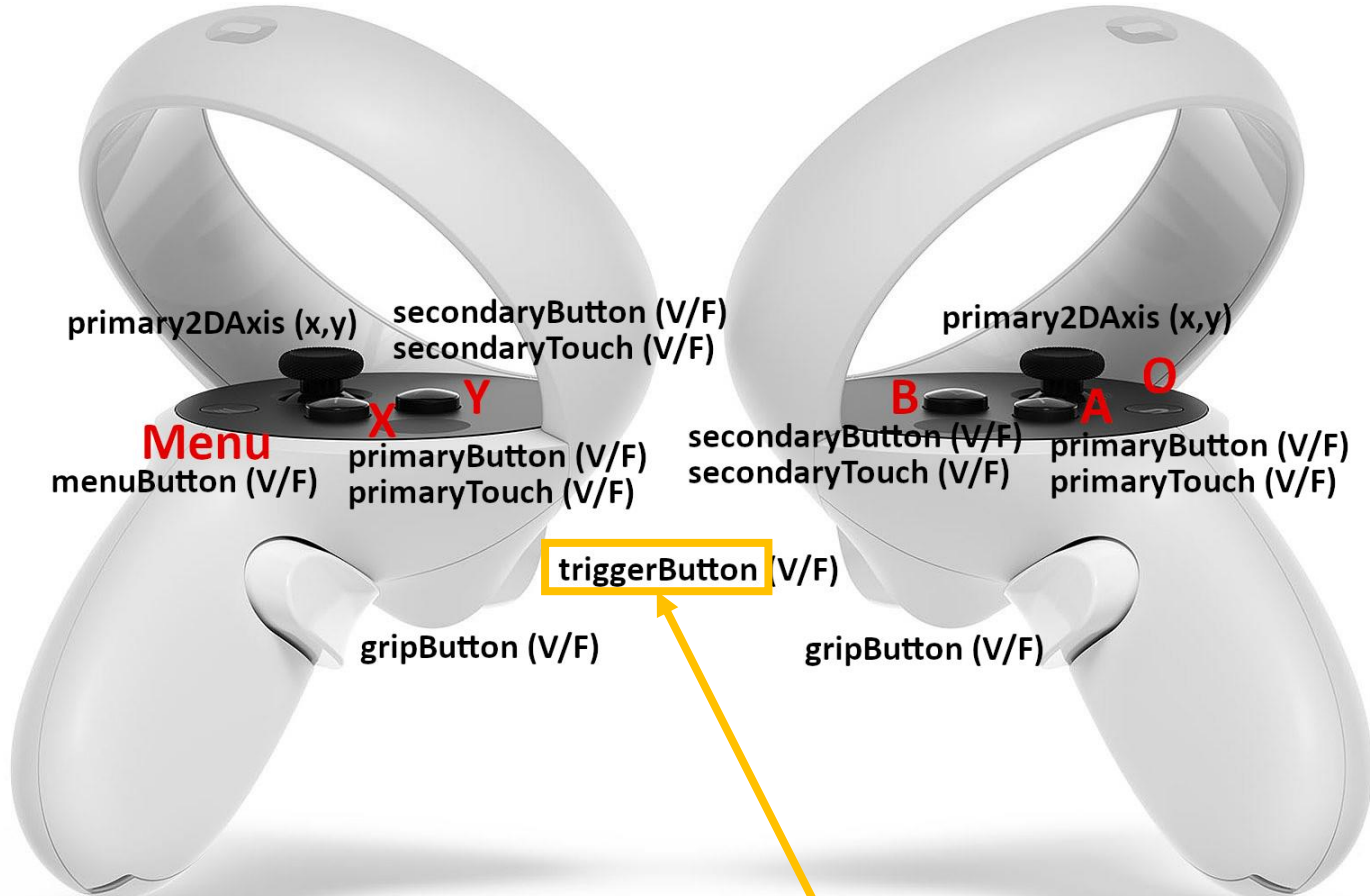
Récupérer des données du bouton Trigger issues du contrôleur gauche de l'Oculus Quest dans un script C#

```
using UnityEngine.XR; // Besoin de la classe XR
private InputDevice targetDevice;
InputDeviceCharacteristics leftControllerCharacteristics = InputDeviceCharacteristics.Left
| InputDeviceCharacteristics.Controller; // Ne considère les données que du contrôleur
gauche

List<InputDevice> devices = new List<InputDevice>();
InputDevices.GetDevicesWithCharacteristics(leftControllerCharacteristics, devices);
if (devices.Count > 0)
{
    targetDevice = devices[index: 0];
    targetDevice.TryGetFeatureValue(CommonUsages.triggerButton, out bool
ltriggerButtonValue); // Contrôleur gauche: Appuie du bouton trigger (true, false)
```

## Contrôleur gauche

## Contrôleur droit



```
TryGetFeatureValue(CommonUsages.<objet>, out  
<type> <variable affectée>)
```

Ex:

```
TryGetFeatureValue(CommonUsages.triggerButton,  
out bool triggerButtonValue)
```

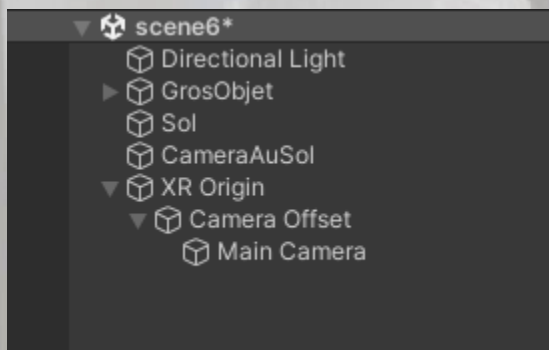


---

# TP UNITY, VENDREDI 21 OCTOBRE 2022

## *7- Reprise de la scène 5*

Étape 1 – Remplacement de Main Camera par *XR Origin* dans *Hierarchy*, à placer aux mêmes coordonnées position et orientation. Sauvegarder la scène en *Scène6*.



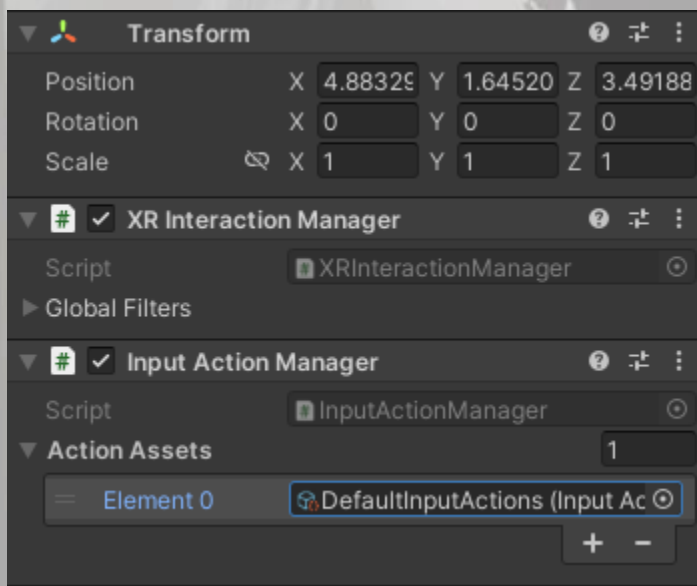
---

# TP UNITY, VENDREDI 21 OCTOBRE 2022

## 7- Reprise de la scène 5

Étape 2 – Ajout de l'interactivité

- Dans *Hierarchy*, ajouter un *XR Interaction Manager*, dans lequel on va ajouter un *Input Action Manager*, pour lequel le *Action Assets* est complété par *DefaultInputActions*



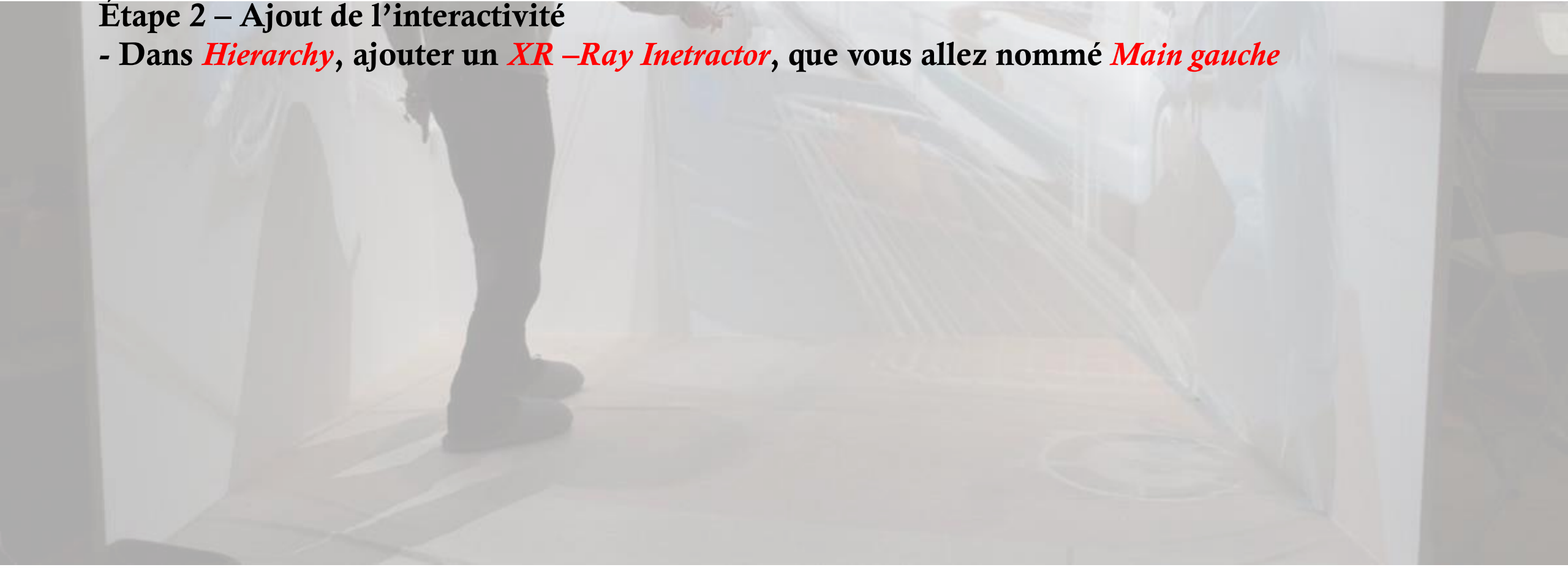
---

# TP UNITY, VENDREDI 21 OCTOBRE 2022

## *7- Reprise de la scène 5*

Étape 2 – Ajout de l'interactivité

- Dans *Hierarchy*, ajouter un *XR –Ray Inetractor*, que vous allez nommé *Main gauche*



**Transform**

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

**XR Controller (Device-based)**

Script XRController

**Tracking**

Update Tracking Type Update And Before Render  
Enable Input Tracking

**Input**

Enable Input Actions

Pose Provider None (Base Pose Provider)

**Controller Node** Left Hand

**Select Usage** Grip

Activate Usage Trigger

UI Press Usage Trigger

Axis To Press Threshold 0.1

Rotate Object Left Primary Axis 2D Left  
Rotate Object Right Primary Axis 2D Right  
Move Object In Primary Axis 2D Up  
Move Object Out Primary Axis 2D Down  
Directional Anchor Rotation Primary Axis 2D

**Model**

Model Prefab None (Transform)  
Model Parent None (Transform)  
Model None (Transform)  
Animate Model

**XR Ray Interactor**

Script XRRayInteractor

**Interaction Manager** XR Interaction Manager (XR)

Interaction Layer Mask Everything

Enable Interaction with

Force Grab

Anchor Control

Translate Speed 1

Rotate Reference Frame None (Transform)

Rotation Mode Rotate Over Time

Rotate Speed 180

Attach Transform None (Transform)

Ray Origin Transform None (Transform)

**Raycast Configuration**

Line Type Straight Line

← Contrôleur gauche

← Sélection d'un objet avec le *GripButton* (voir le contrôleur Oculus)

← Component créé précédemment dans *Hierarchy* →

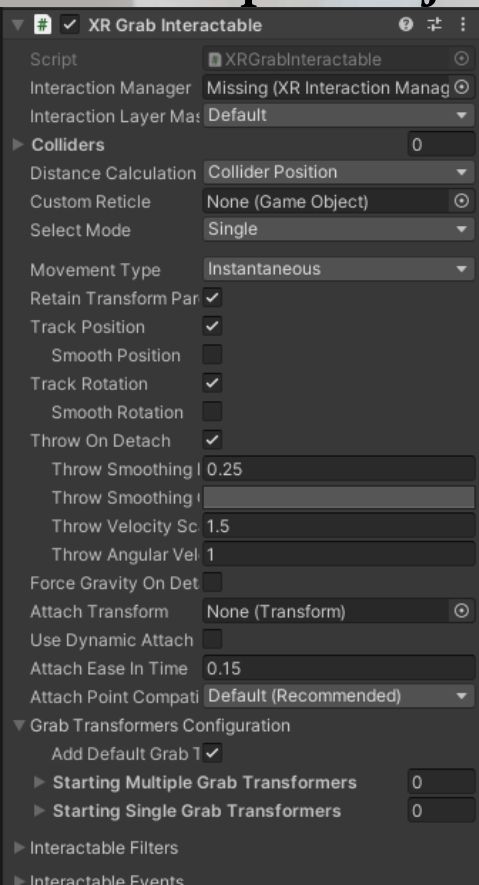
- Directional Light
- GrosObjet
  - Cube1
  - Cube2
  - Cube3
  - Cube4
  - Cylinder1
  - Cylinder2
  - Cylinder3
  - Cylinder4
- Sol
- CameraAuSol
- XR Interaction Manager**
- XR Origin
  - Camera Offset
    - Main Camera
    - Main gauche

---

# TP UNITY, VENDREDI 21 OCTOBRE 2022

## 7- Reprise de la scène 5

Étape 3 - Ajout du composant *XR Grab Interactable* à l'ensemble des objets sous GrosObjet





---

# TP UNITY, VENDREDI 21 OCTOBRE 2022

## *7- Reprise de la scène 5*

Étape 4 – Modification de *SelectionObjet.cs* en *SelectionObjetXR.cs*

```
using UnityEngine.XR.Interaction.Toolkit; // Interactivité
XRBaseInteractable m_Interactable; // Permet de gérer l'interaction de l'objet avec un
RayCasting ou Main Virtuelle Simple
static Color s_SelectionnableColor = new Color(0.929f, 0.094f, 0.278f); // Couleur de
l'objet sélectionnable
    static Color s_SelectionneColor = new Color(0.019f, 0.733f, 0.827f); // Couleur de
l'objet sélectionné
    static Color s_EnChuteColor = new Color(0.019f, 0.13f, 0.827f); // Couleur de
l'objet En chute
    static Color s_TombeAuSolColor = new Color(0.019f, 0.827f, 0.13f); // Couleur de
l'objet Tombé au sol
```

---

```
// Création des liaisons entre le component Interactable et la classe SelectionObjetXR à la fin
de la création de l'objet courant qui supporte le script
protected void OnEnable()
{
    renderer = this.transform.GetComponent<Renderer>();
    oldColor = renderer.material.color;
    Affiche("*** " + transform.name + " ENABLED");
    etatObjets = EtatsObjets.Idle;
    cameraAuSol.enabled = false;
    m_Interactable = GetComponent<XRBaseInteractable>(); // Récupère le Component
Interactable qui doit se trouver sur le GameObject courant
    m_Interactable.firstHoverEntered.AddListener(OnFirstHoverEntered); // Ajoute un événement
d'entrée dans l'état Sélectionnable de l'I3D
    m_Interactable.lastHoverExited.AddListener(OnLastHoverExited); // Ajoute un événement de
sortie de l'état Sélectionnable de l'I3D
    m_Interactable.selectEntered.AddListener(OnSelectEntered); // Ajoute un événement lié
l'état Sélectionné de l'I3D
    m_Interactable.firstSelectEntered.AddListener(OnFirstSelectEntered); // Ajoute un
événement d'entrée dans l'état Sélectionné de l'I3D
    m_Interactable.lastSelectExited.AddListener(OnLastSelectExited); // Ajoute un événement
de sortie de l'état Sélectionné de l'I3D
}
```

---

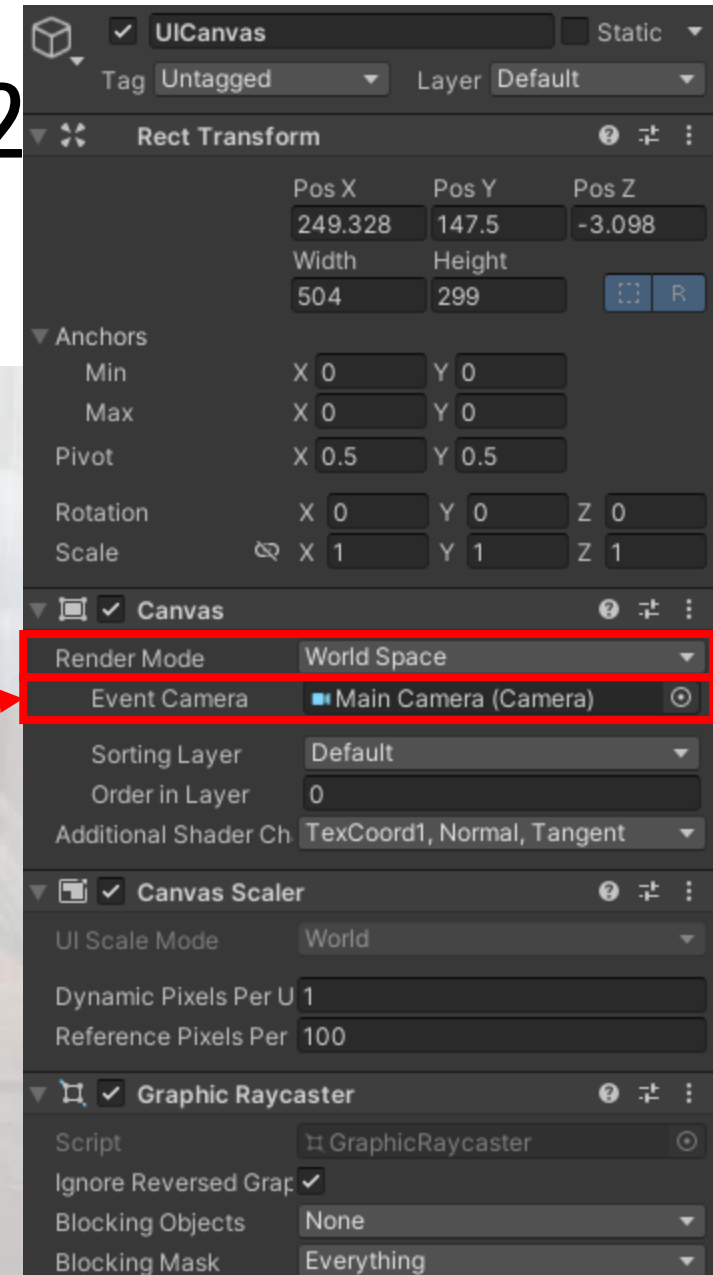
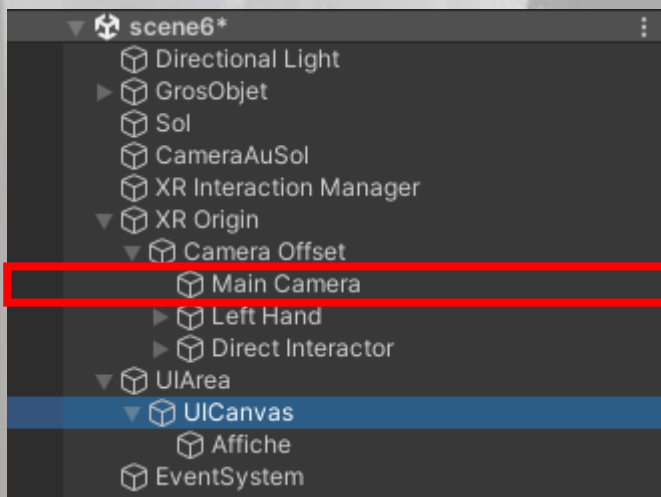
```
// Gestion des événements par des fonctions de la classe SelectionObjetXR
protected virtual void OnFirstHoverEntered(HoverEnterEventArgs args) =>
SetSelectionnable();
protected virtual void OnLastHoverExited(HoverExitEventArgs args) =>
ResetSelectionnable();
protected virtual void OnSelectEntered(SelectEnterEventArgs args) => SetManipule();
protected virtual void OnFirstSelectEntered(SelectEnterEventArgs args) =>
SetSelectionne();
protected virtual void OnLastSelectExited(SelectExitEventArgs args) => SetEnChute();
// dans l'état Selectionne
public void SetSelectionne()
{
    if (etatObjets == EtatsObjets.Sélectionnable)
    {
        renderer.material.color = s_SelectionneColor;
        etatObjets = EtatsObjets.Sélectionné;
        Affiche(this.transform.name + " est sélectionné.");
        transform.GetComponent<Rigidbody>().isKinematic = true; //
On arrête la physicalisation de l'objet
    }
}
```

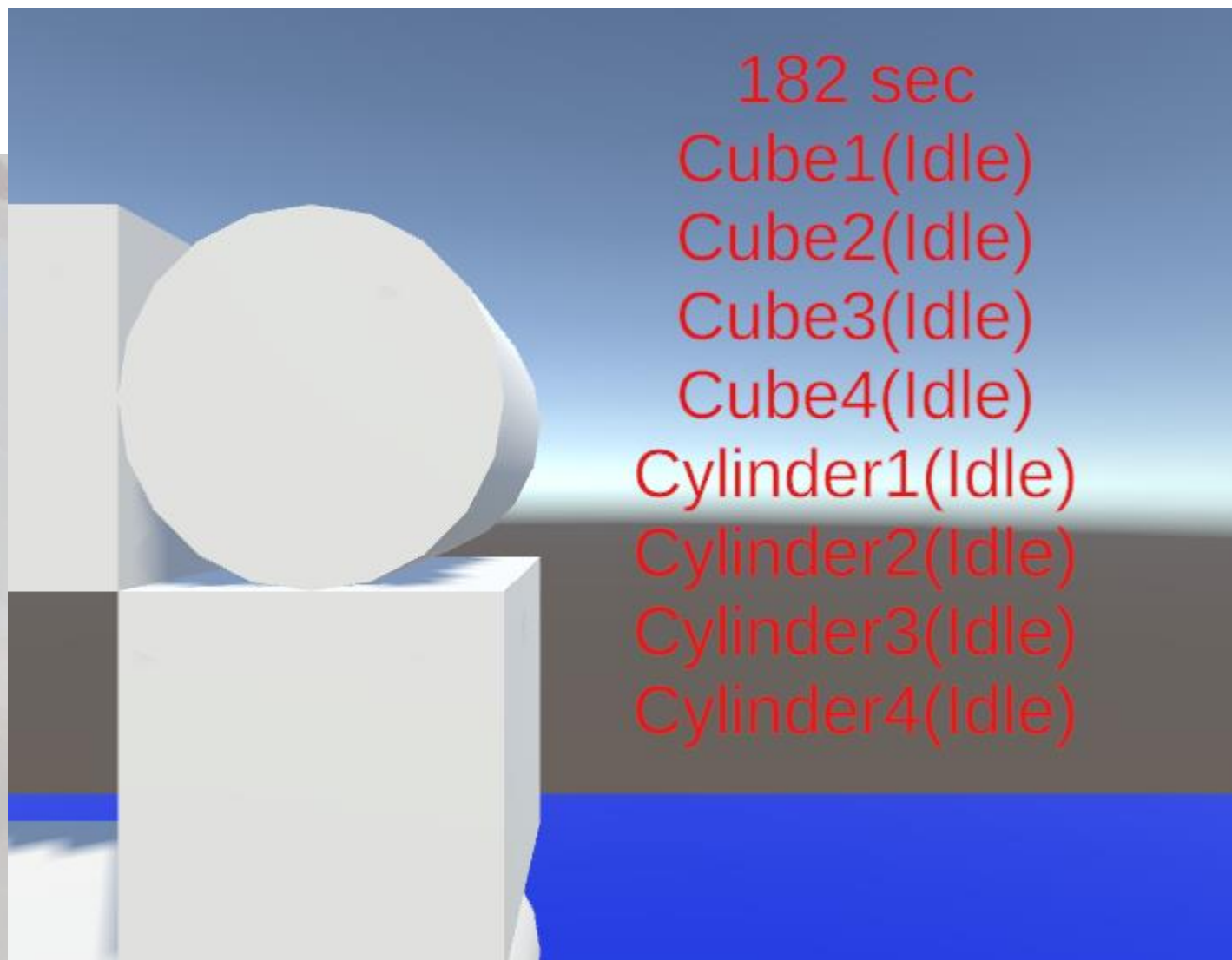
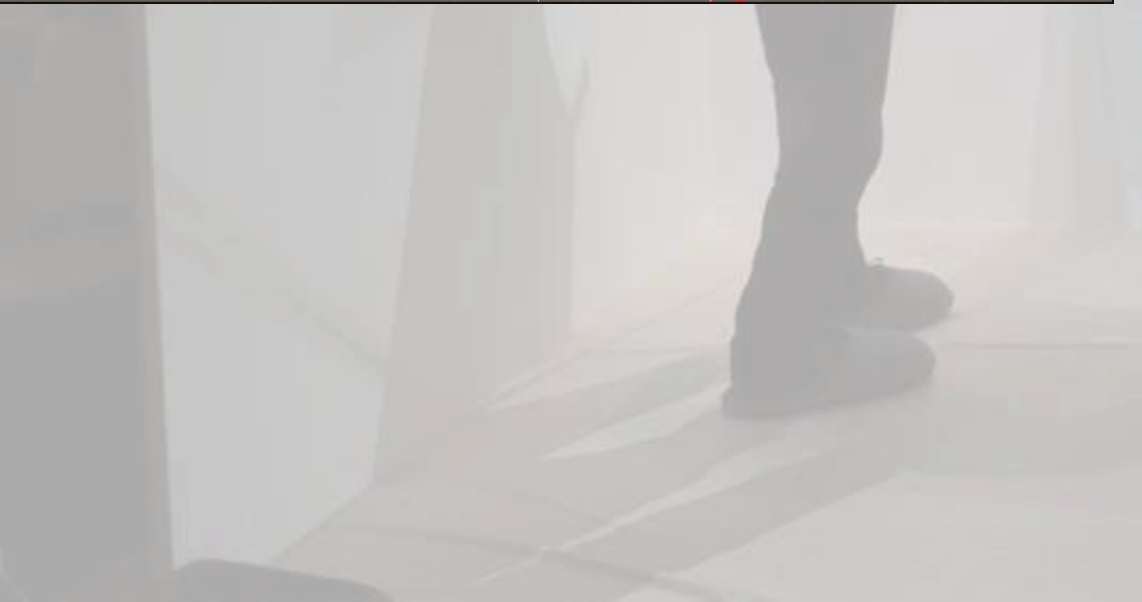
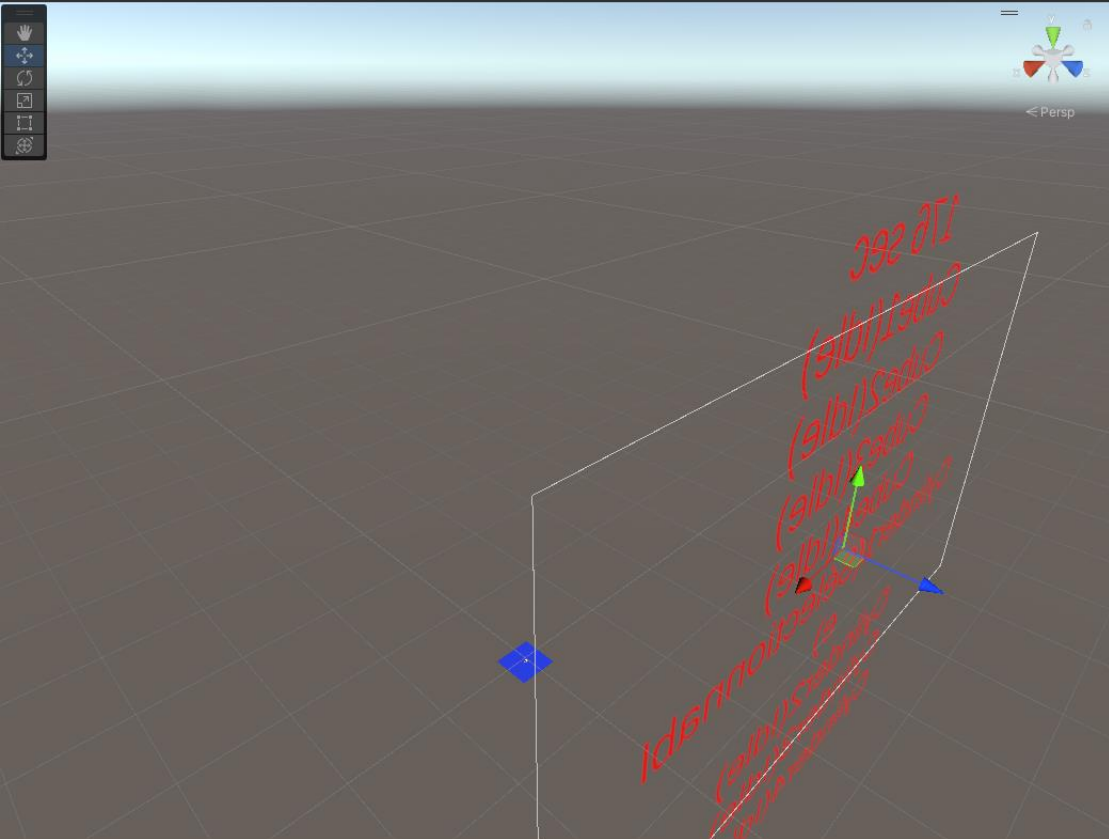
# TP UNITY, VENDREDI 21 OCTOBRE 2022

## 7- Reprise de la scène 5

Étape 5 – Ajout d'une zone de texte de le monde 3D

-Création d'un *EmptyObject UIArea* et d'un *UI->Canvas* sous *UIArea*







---

# TP UNITY, VENDREDI 21 OCTOBRE 2022

## *7- Reprise de la scène 5*

Étape 6 – Modification de *GereSelection.cs* en *GereSelectionXR.cs*

```
using TMPro;

public Canvas canvas;
public TextMeshProUGUI affiche; // Zone d'affichage 2D dans l'espace 3D
// Affiche le texte 'text' dans la zone d'affichage 2D référencée par 'canvas' et
// 'affiche'
private void Affiche(string text)
{
    Debug.Log(text);
    affiche.text = text;
}
```

```
// Mise à jour du temps (sec) et de l'états des objets sous le GameObject courant
```

```
void Update()
```

```
{
```

```
    if (PartieTerminee() == false)
```

```
    {
```

```
        compteur_temps = Time.time.ToString("0.") + " sec" + "\n";
```

```
        affichage = compteur_temps; // affichage est de type string
```

```
        for (int i = 0; i < nfils; i++)
```

```
        {
```

```
            string text = tfils[i].name + "(" +
```

```
tfils[i].GetComponent<SelectionObjetXR>().etatObjets + ")\n";
```

```
            affichage += text;
```

```
        }
```

```
        Affiche(affichage);
```

```
    }
```

```
else
```

```
{
```

```
    if (partieTerminee == false)
```

```
    {
```

```
        affichage = "PARTIE TERMINEE EN " + compteur_temps;
```

```
        Affiche(affichage);
```

```
        partieTerminee = true;
```

```
    }
```

```
}
```

```
}
```



---

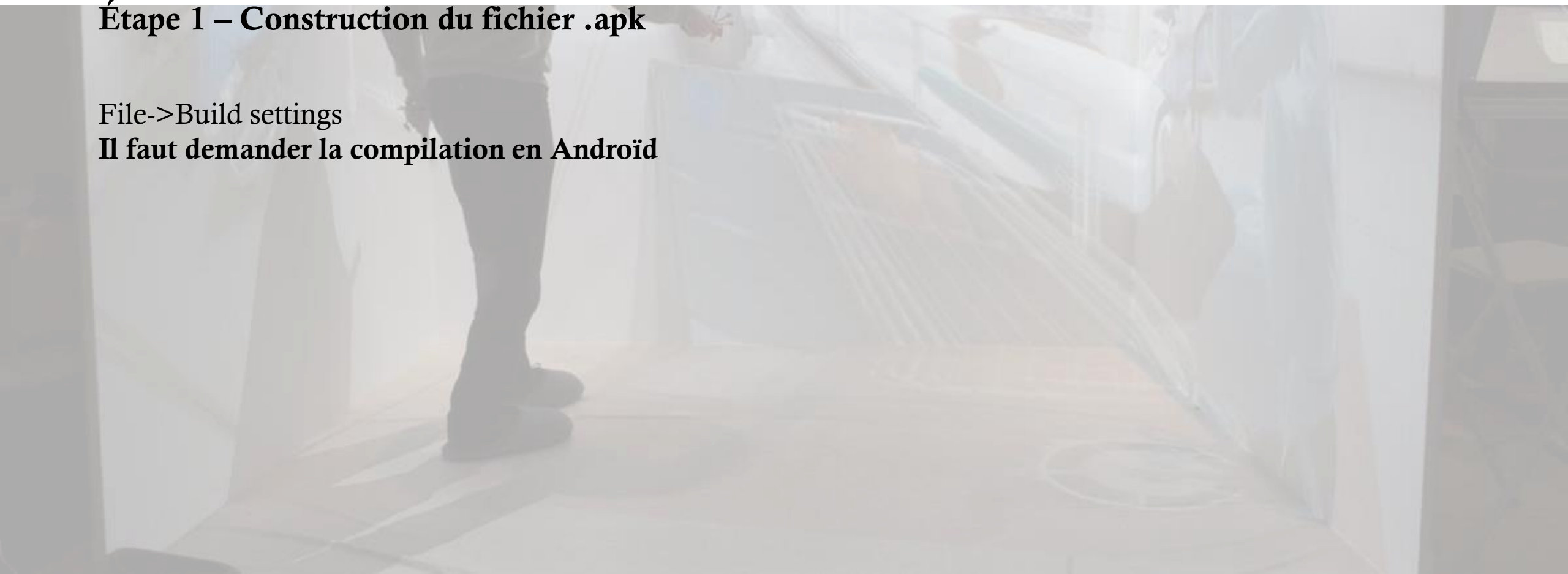
# TP UNITY, VENDREDI 21 OCTOBRE 2022

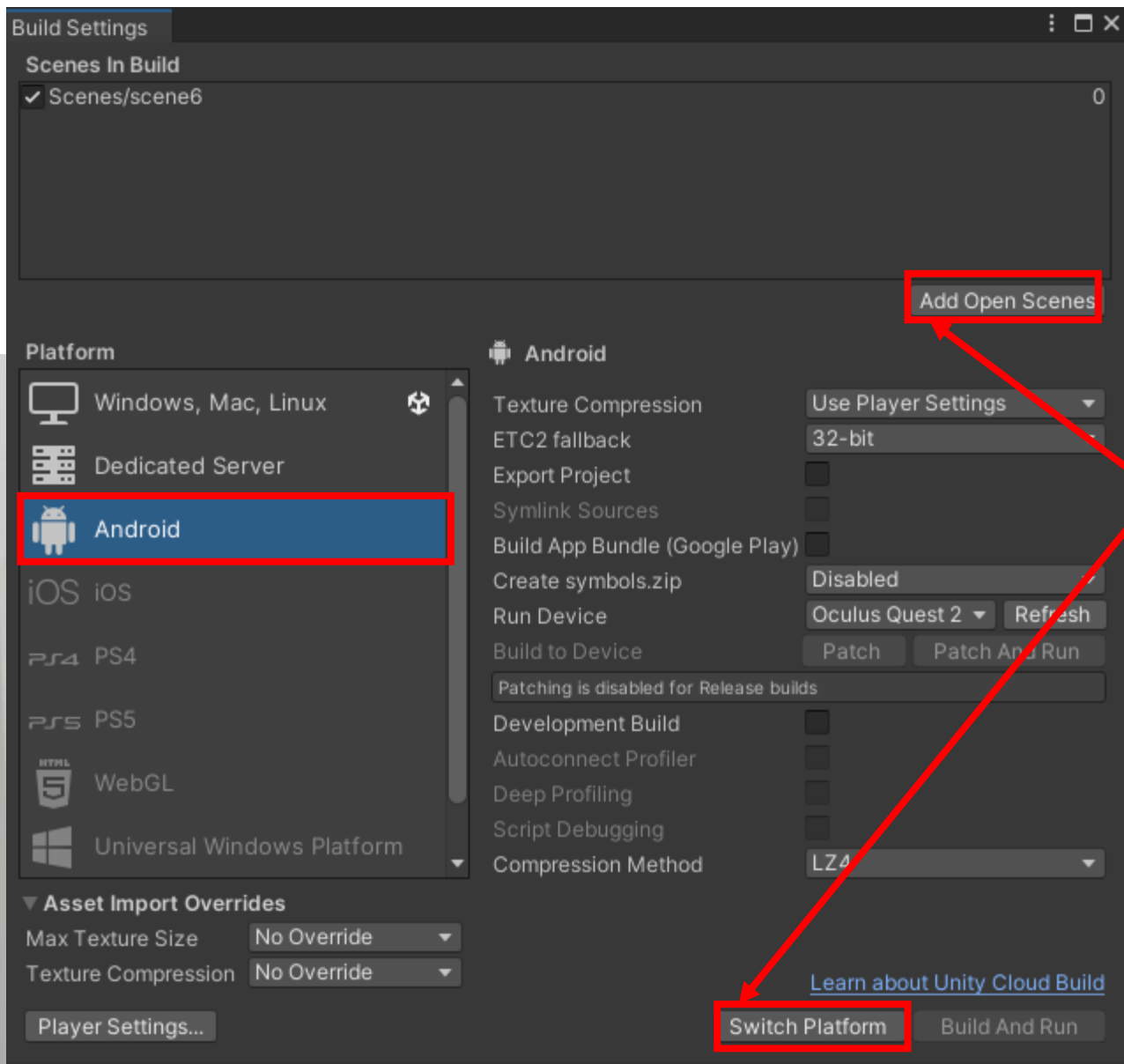
## *8- Mise de TPunityVR sur le casque Oculus Quest*

Étape 1 – Construction du fichier .apk

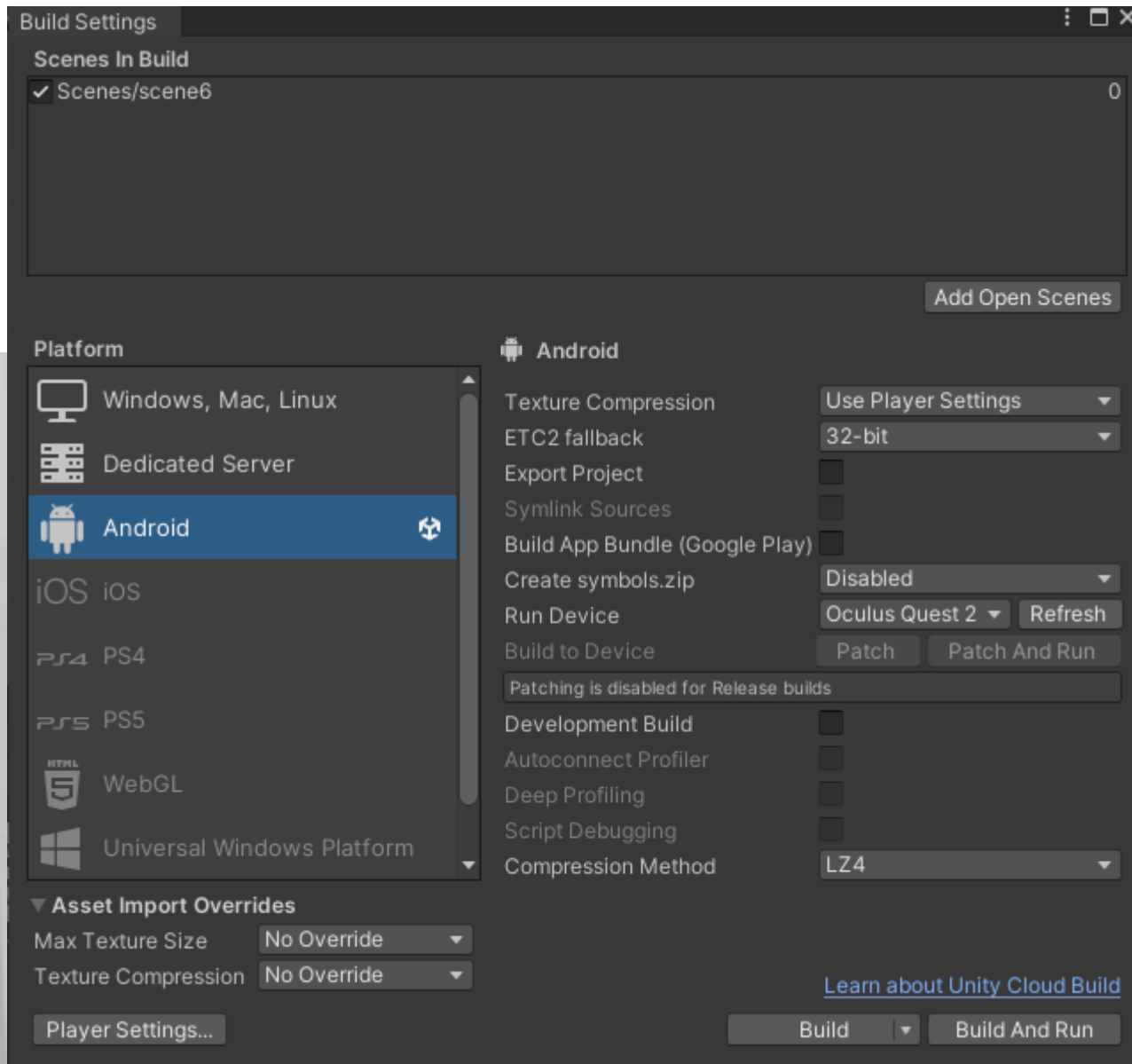
File->Build settings

**Il faut demander la compilation en Android**





Aller dans *Android* -> *Switch Platform*  
Ajouter la scène *scene6*



Puis **Build** et donner le nom du fichier *.apk* à construire

---

# TP UNITY, VENDREDI 21 OCTOBRE 2022

## *8- Mise de TPunityVR sur le casque Oculus Quest*

Étape 2 – Installation du fichier .apk sur le casque Oculus Quest

Utilisation du logiciel SideQuest (à télécharger sur <https://sidequestvr.com/setup-howto> )

