

Programmation VR

Antoine Cherel



Fondamentaux

Programmation VR

- Un moteur de jeu (+ un SDK VR)
- Un SDK VR au minimum:
 - Camera Rig
 - Contrôleur Droit
 - Contrôleur Gauche
- En option:
 - Un système de locomotion (déplacements continus, téléportation, etc...)
 - Une gestion des Input & Output des contrôleurs (triggers, boutons, joystick, haptique, etc...)
 - Un système d'interaction (préhension d'objets, boutons physiques, leviers, etc...)
 - Un système d'interface (interaction avec des boutons à l'aide de pointeurs lasers, etc...)



Quizz « Historique de la VR »

2017

2018

2020

1960

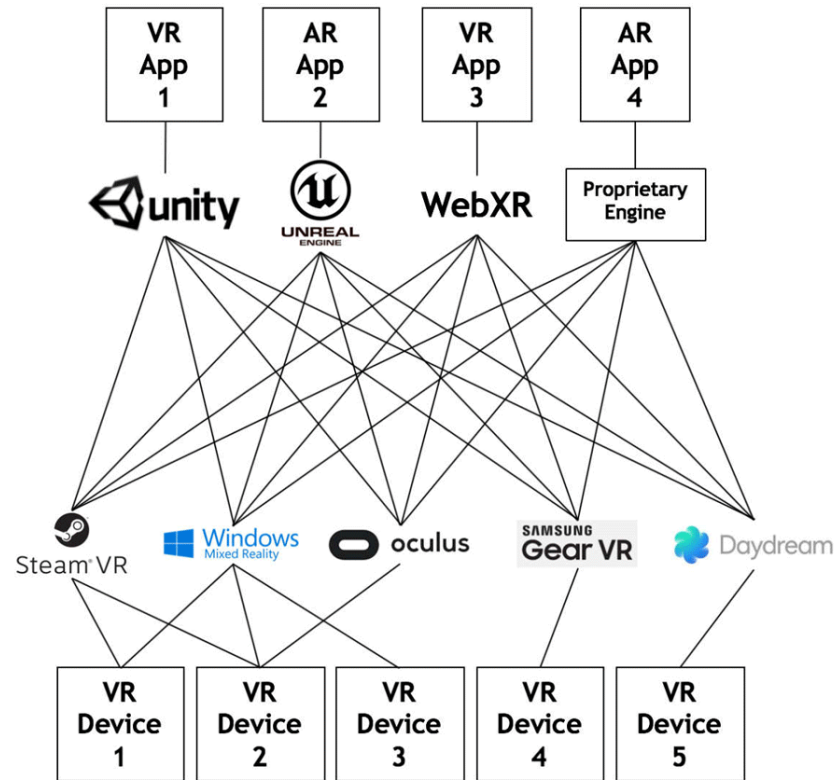
2015

2014

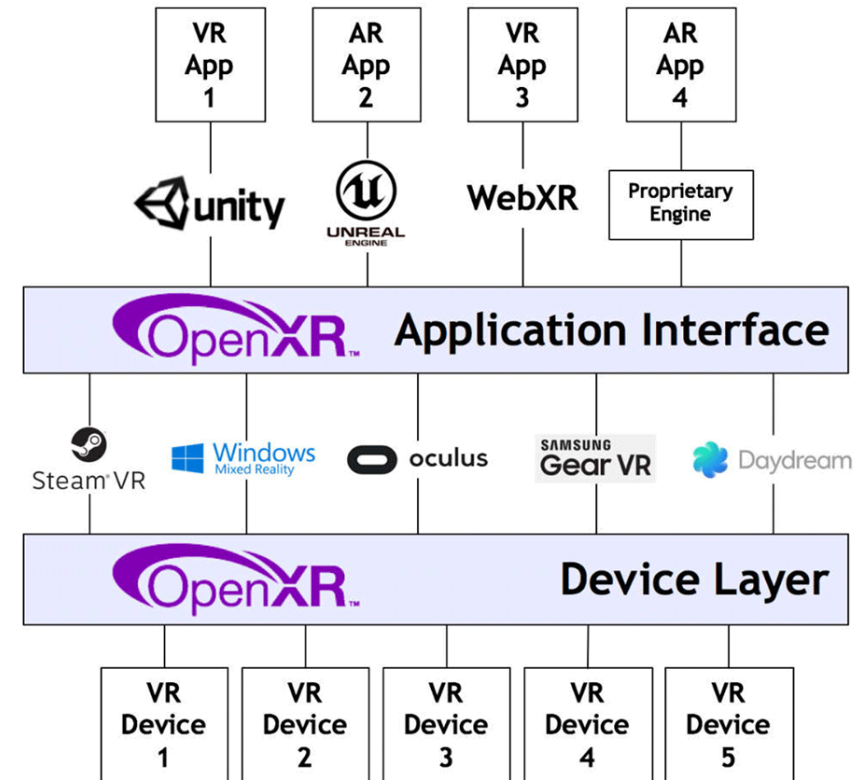
SDK VR avant Open XR

- SteamVR (PC VR)
- Oculus Integration (Casques Oculus)
- Windows Mixed Reality
- Pico SDK (Casques Pico)
- WebXR (VR dans le Web)

SDK VR après Open XR



Before OpenXR



OpenXR Goal



XR Interaction Toolkit

Preview

AR/VR app interactivity without coding

Unity XR Interaction Toolkit

https://gitlab.com/antoine_ch/cours-programmation-vr

Installation

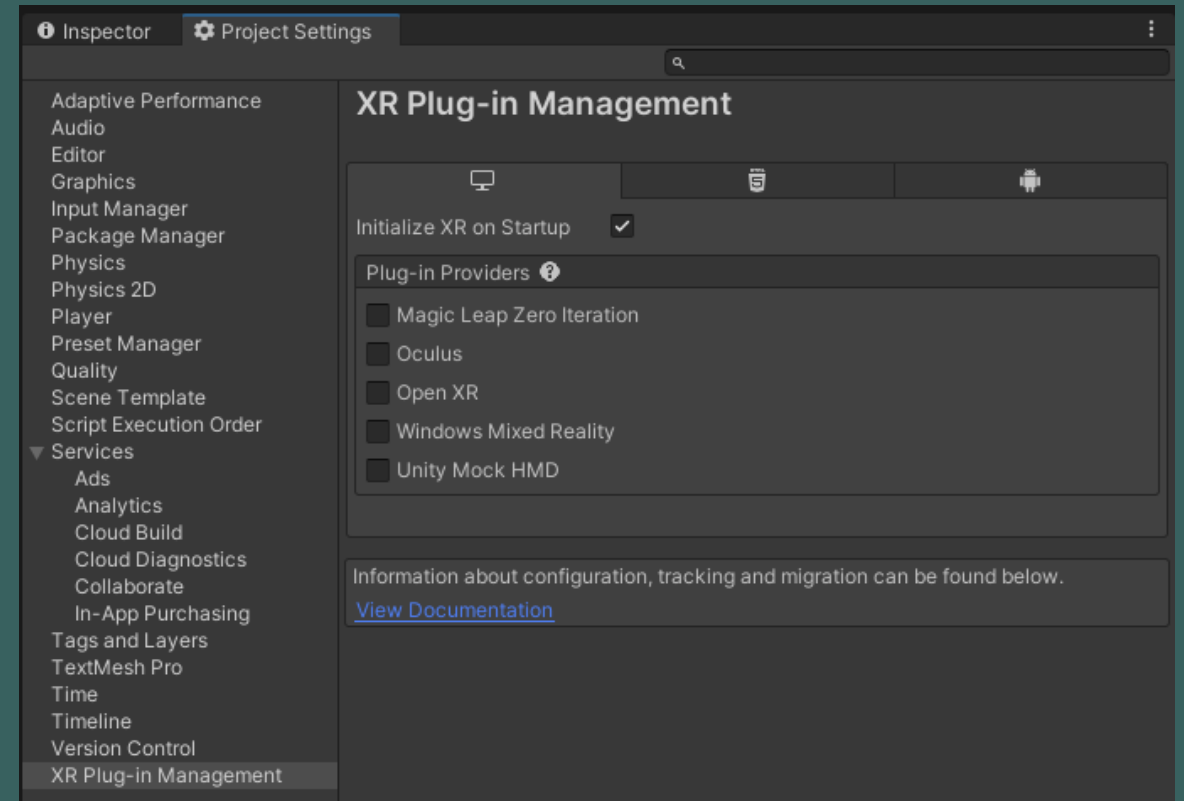
- Installer Oculus
- Installer Unity Hub
 - Installer Unity 2021.3.33



Installation

Dans Unity 2021.3

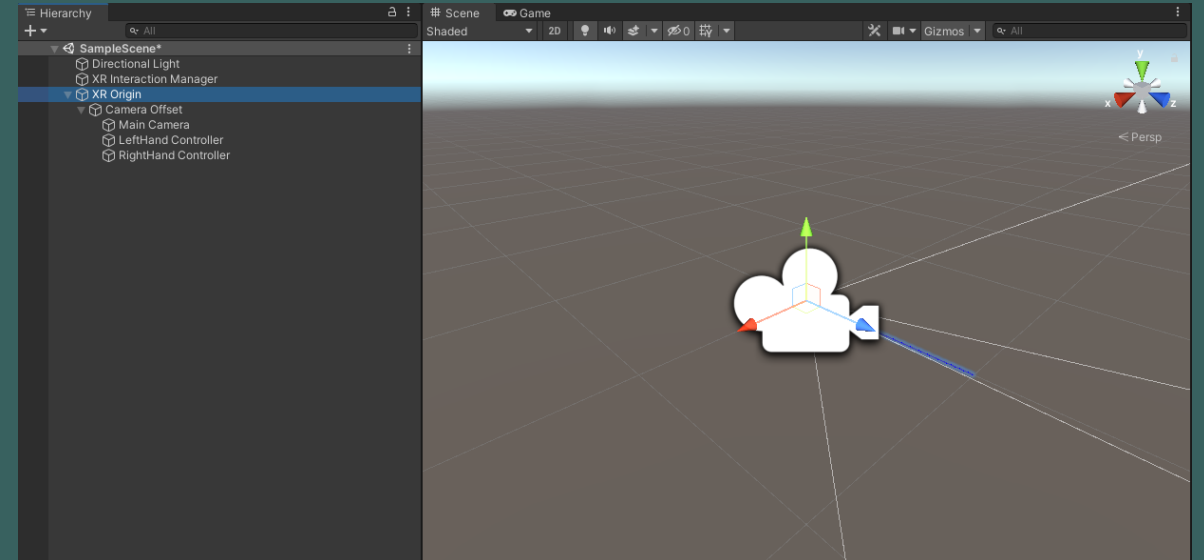
- Project Settings
 - ⇒ XR Plugin Management
 - ⇒ Install
 - ⇒ Cocher *Oculus*
- Package Manager
 - ⇒ XR Interaction Toolkit
 - ⇒ Install



Bases

XR Origin

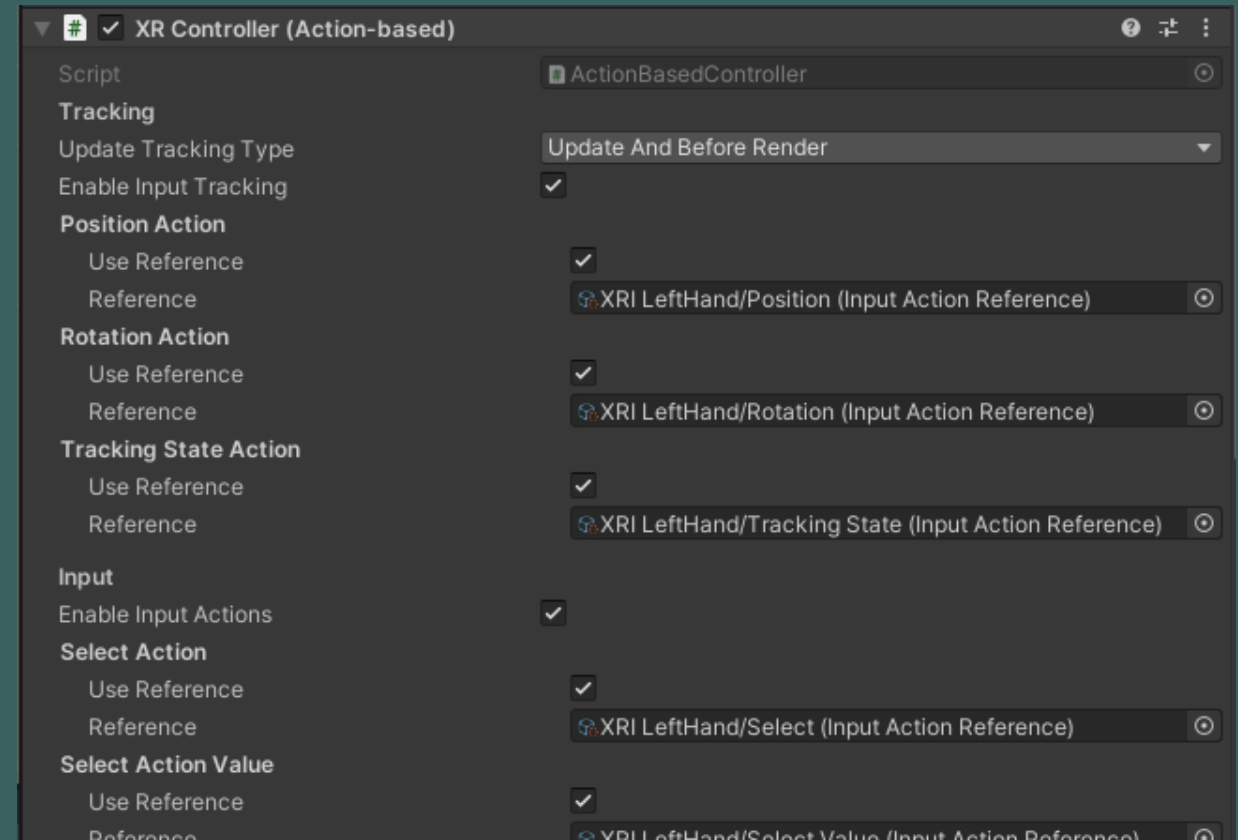
- Camera Rig dans XR Interaction Toolkit
 - Main Camera
 - LeftHand Controller
 - RightHand Controller



- GameObject
 - XR => XR Origin (Action based)

XR Controller

- 2 systèmes différents
 - Action Based (avec le nouvel Input System)
 - Device Based (avec le système d'inputs classique (déprécié))
- Mapping des boutons pour les interactions

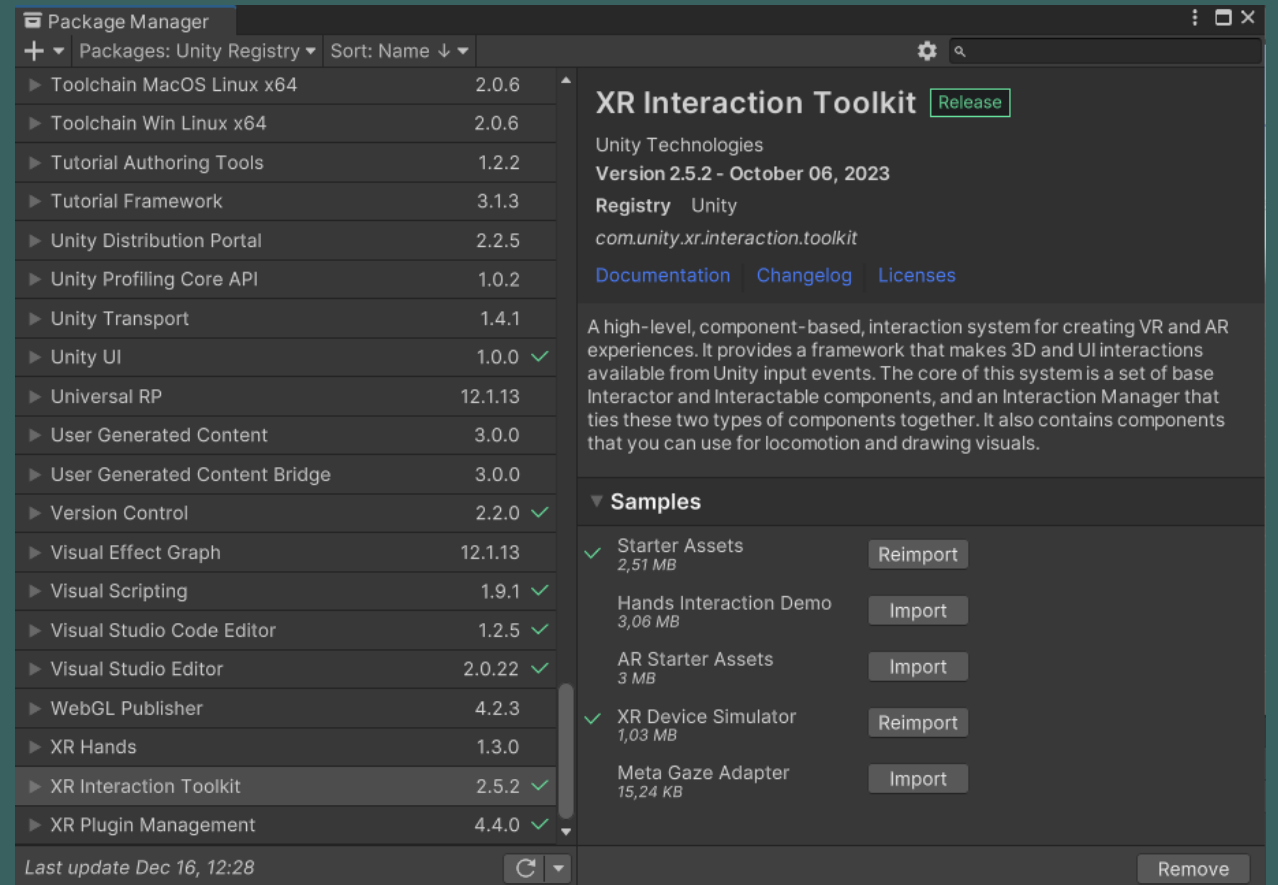


XR Device Simulator

XR Device Simulator

Installation :

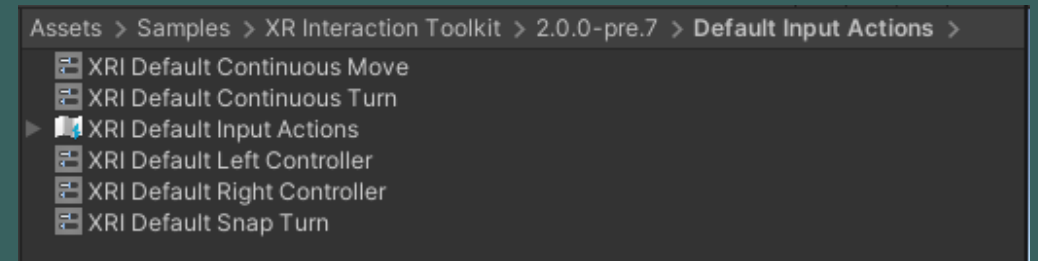
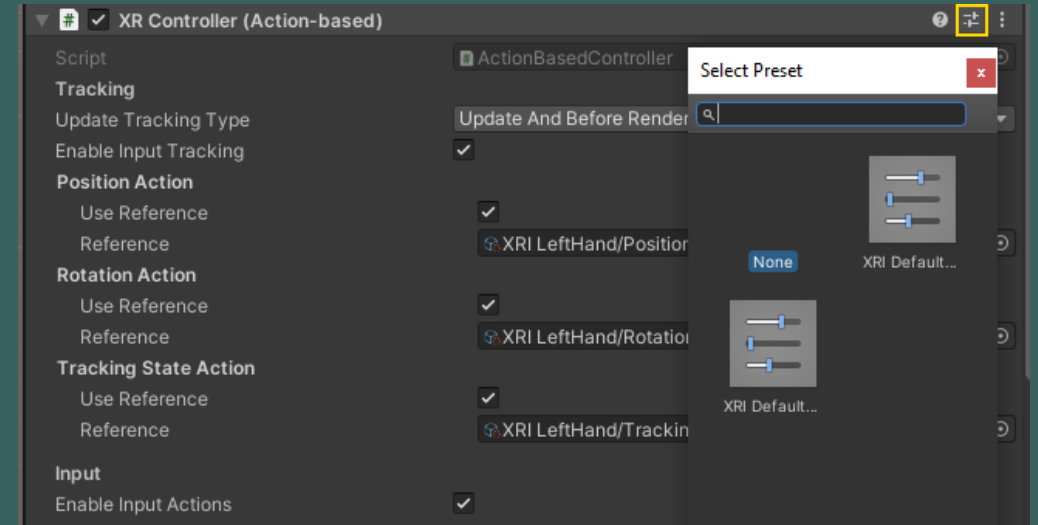
- Package Manager
 - ⇒ XR Interaction Toolkit
 - ⇒ Importer
 - ⇒ Start Assets
 - ⇒ XR Device Simulator



XR Device Simulator

Ajustements :

- Ajouter les presets sur les XR Controller
- Ajouter un Input Action Manager sur le XR Origin
 - Puis y rajouter *XRI Default Input Action*
- Ajouter le prefab XR Device Simulator à la scene



XR Device Simulator

Touches :

- Click droit : déplacement tête
- Click droit + click molette : rotation tête
- Click droit + scroll : déplacement rig (avant/arrière)

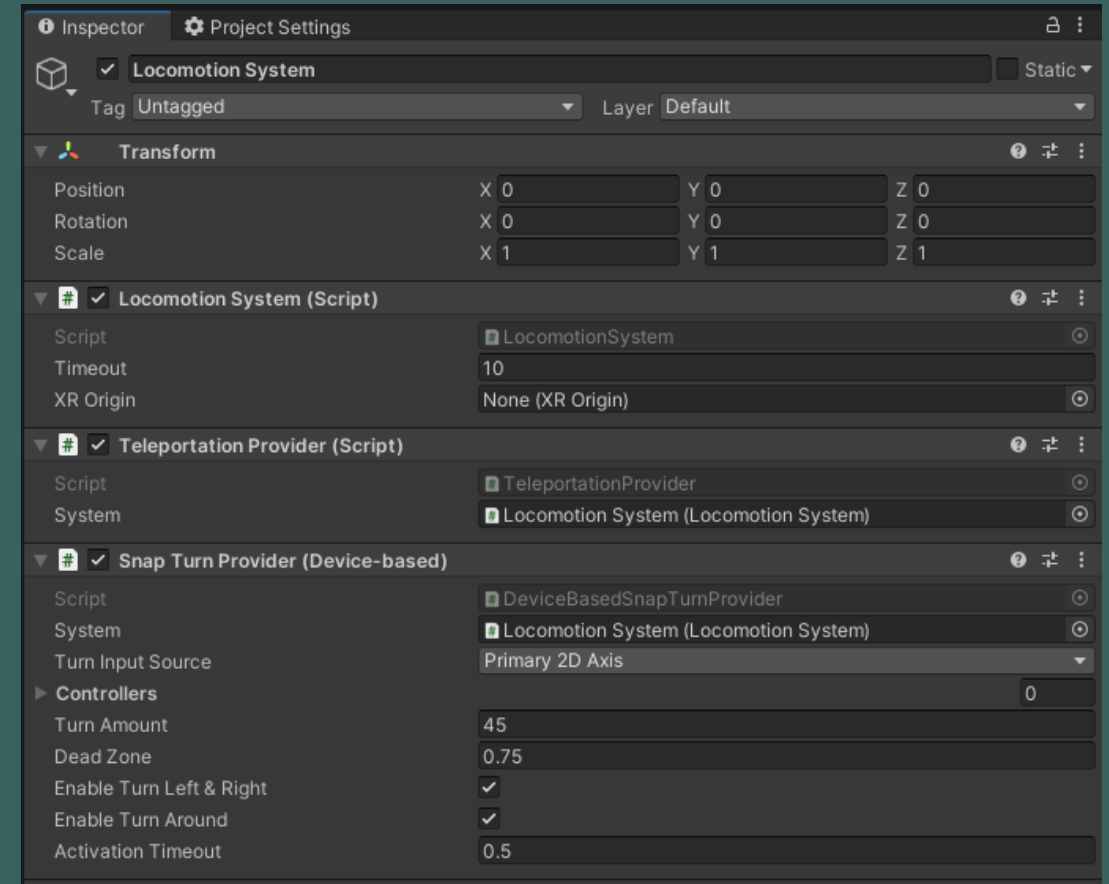
Controller (shift ou space)

- + souris : déplacement controller
- + click molette : rotation controller
- + touche G : interaction objets / téléporter
- + click gauche : interaction UI

Déplacements

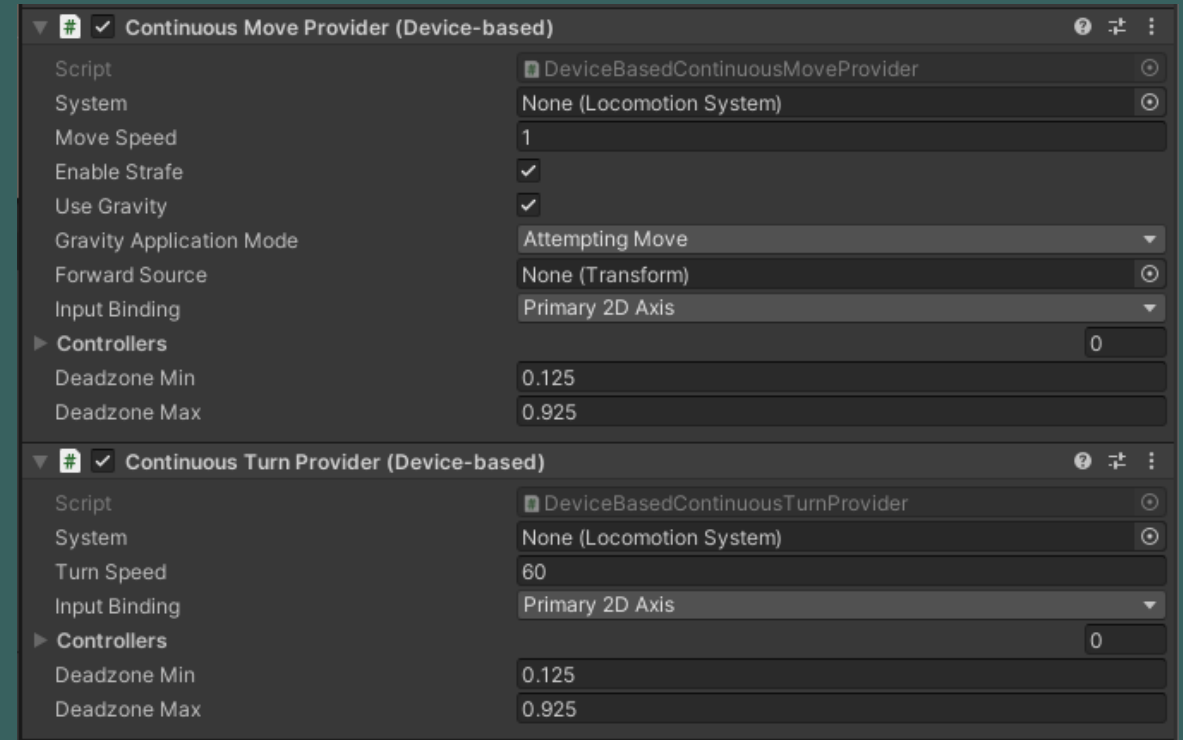
Locomotion System

- Gestion des déplacements en VR
- Possibilité de rajouter:
 - Téléportation Provider
 - support des déplacements en Téléportation
 - Snap Turn Provider
 - support des rotations de 45° avec le joystick



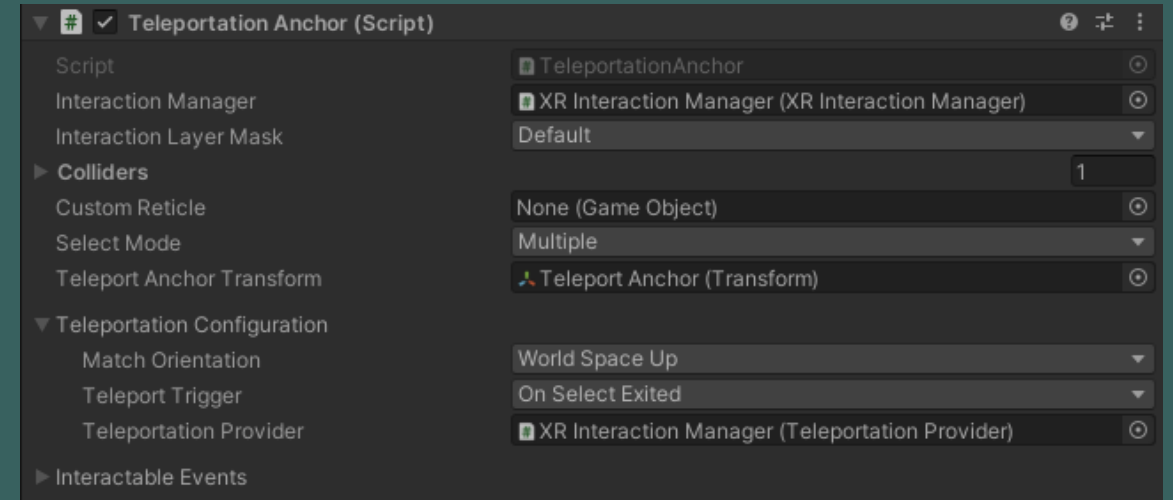
Locomotion System

- Gestion des déplacements en VR
- Possibilité de rajouter:
 - Continuous Move Provider
 - support translations continues
 - Continuous Turn Provider
 - support des rotations continues
- Attention, les déplacements continus peuvent provoquer plus de Motion Sickness...



Teleportation Anchor

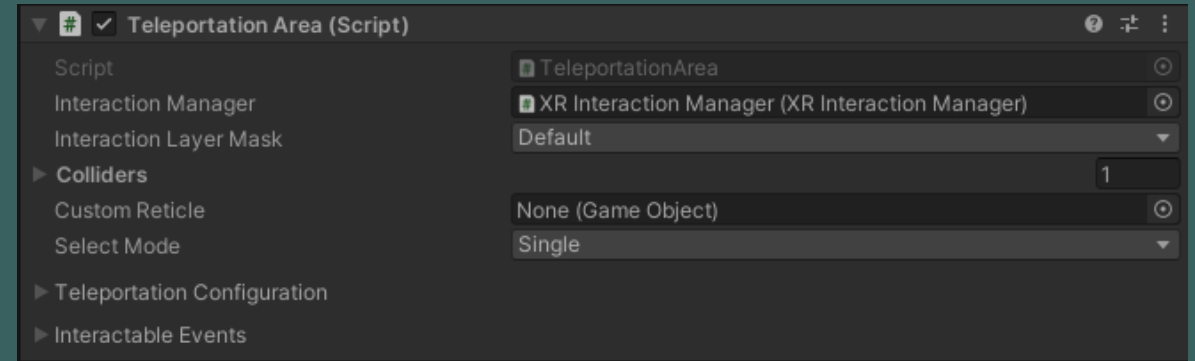
- Emplacement fixe ou l'on peut se téléporter
- Nécessite un collider



- Possibilité d'utiliser les Interactable Events pour appeler vos fonctions

Teleportation Area

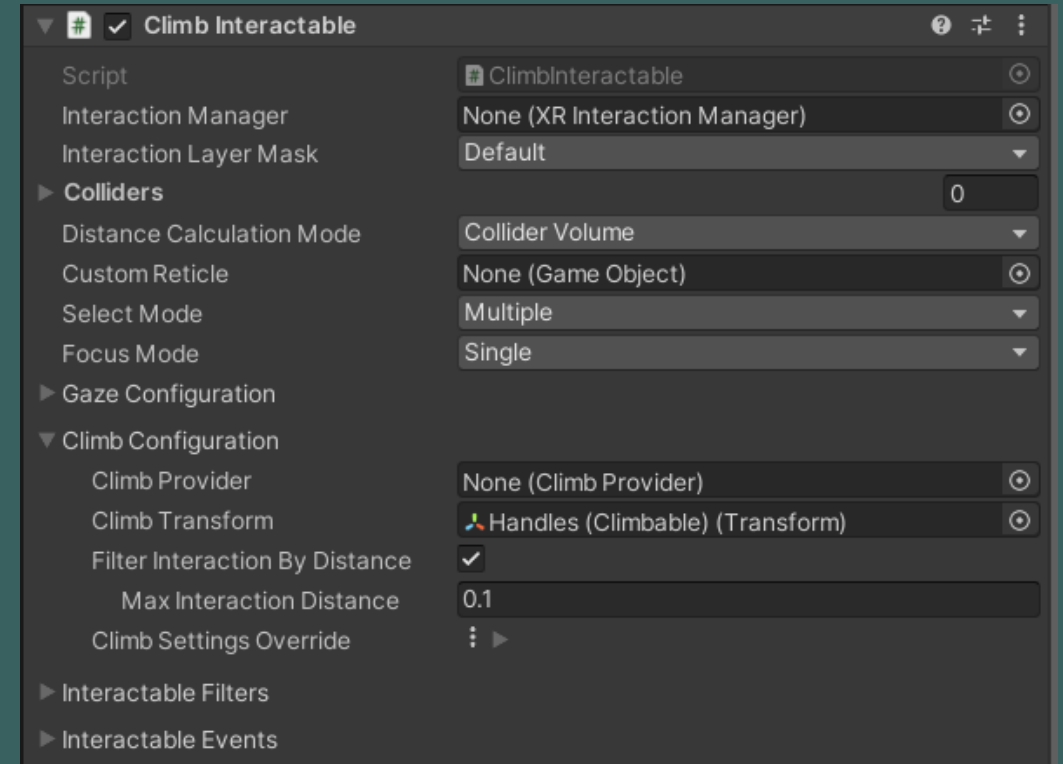
- Zone dans laquelle on peut se téléporter
- Nécessite un collider



- Possibilité d'utiliser les Interactable Events pour appeler vos fonctions

Climb Interactable

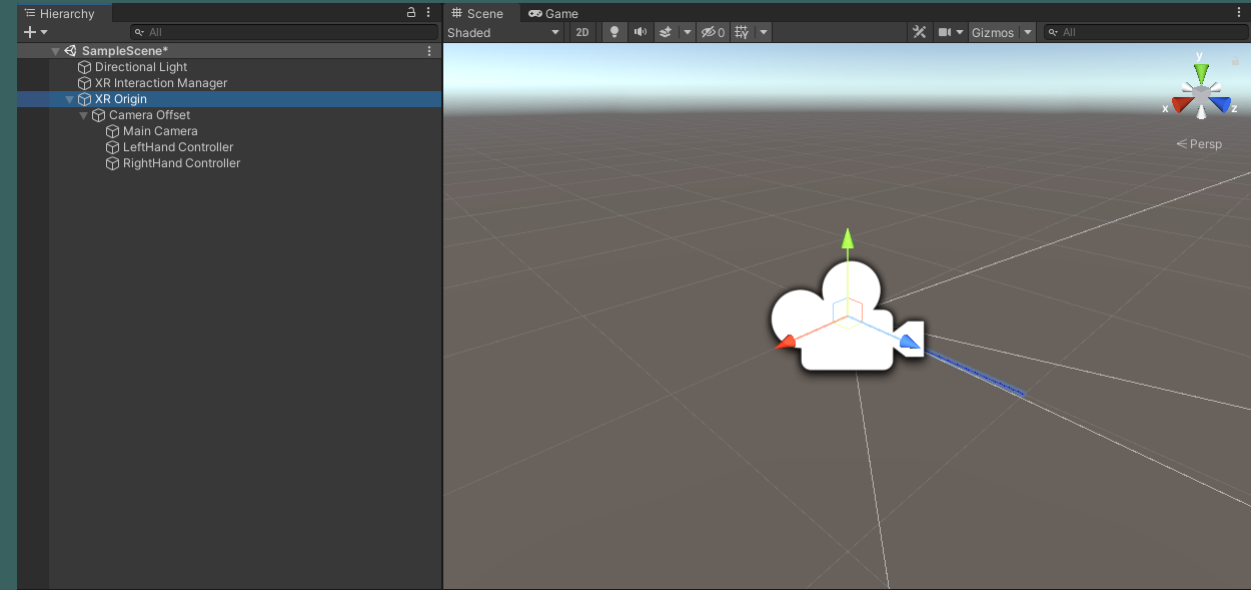
- Objets sur lesquels on peut grimper
 - Echelle
 - Mur d'escalade
 - Nécessite un (ou plusieurs) collider(s)
 - Nécessite un Rigidbody
-
- Possibilité d'utiliser les Interactable Events pour appeler vos fonctions



Interactions

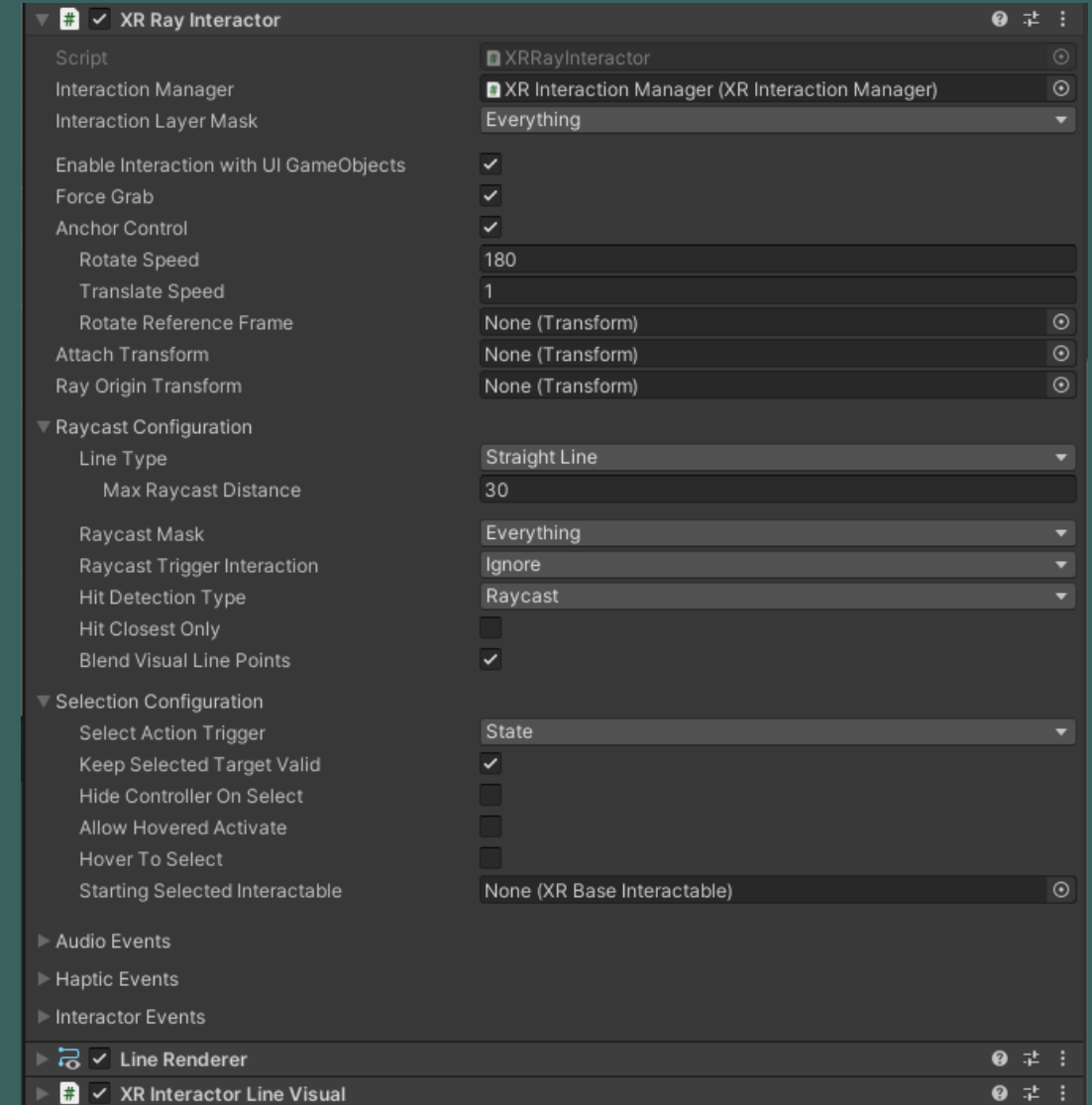
XR Interaction Manager

- Manager des interactions en VR
 - Grab, Canvas, etc...



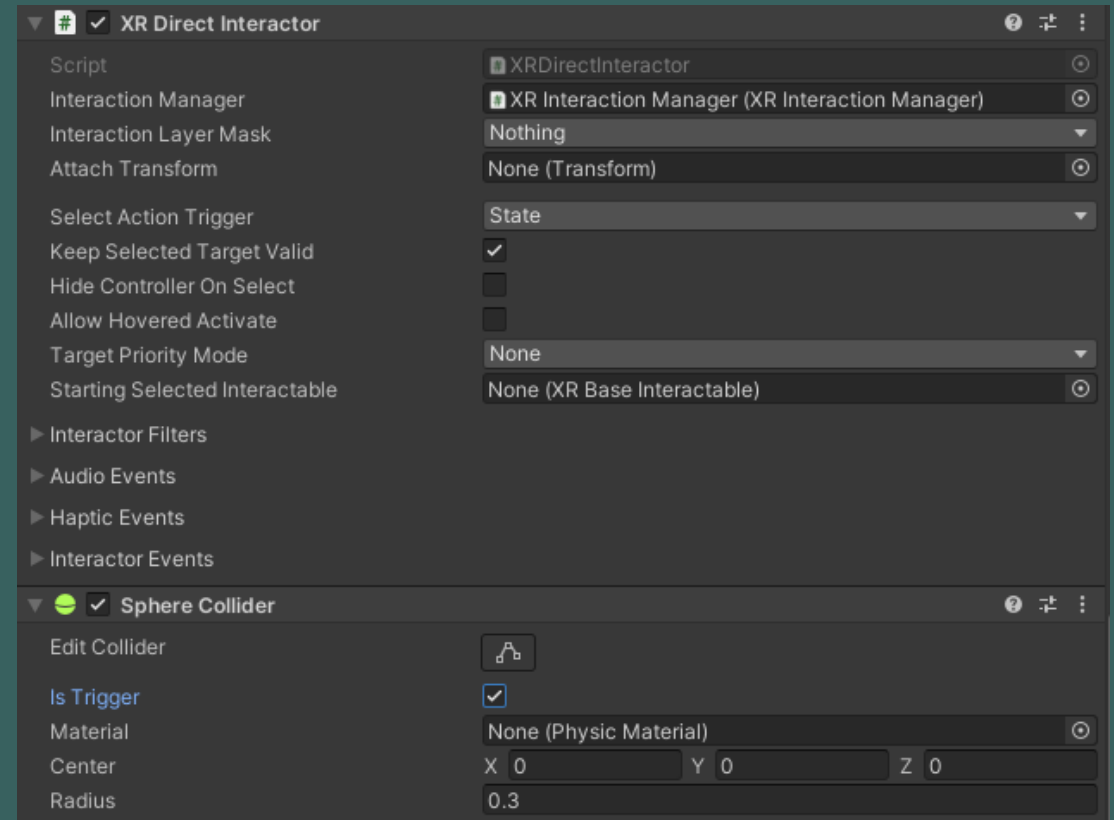
XR Ray Interactor

- Pour interagir à distance, avec des rayons
- Différents Event pour y appeler vos fonctions
 - *Comme les évènements OnClick des boutons*
- XR Interactor Line Visual
 - Pour régler l'apparence du Ray Interactor



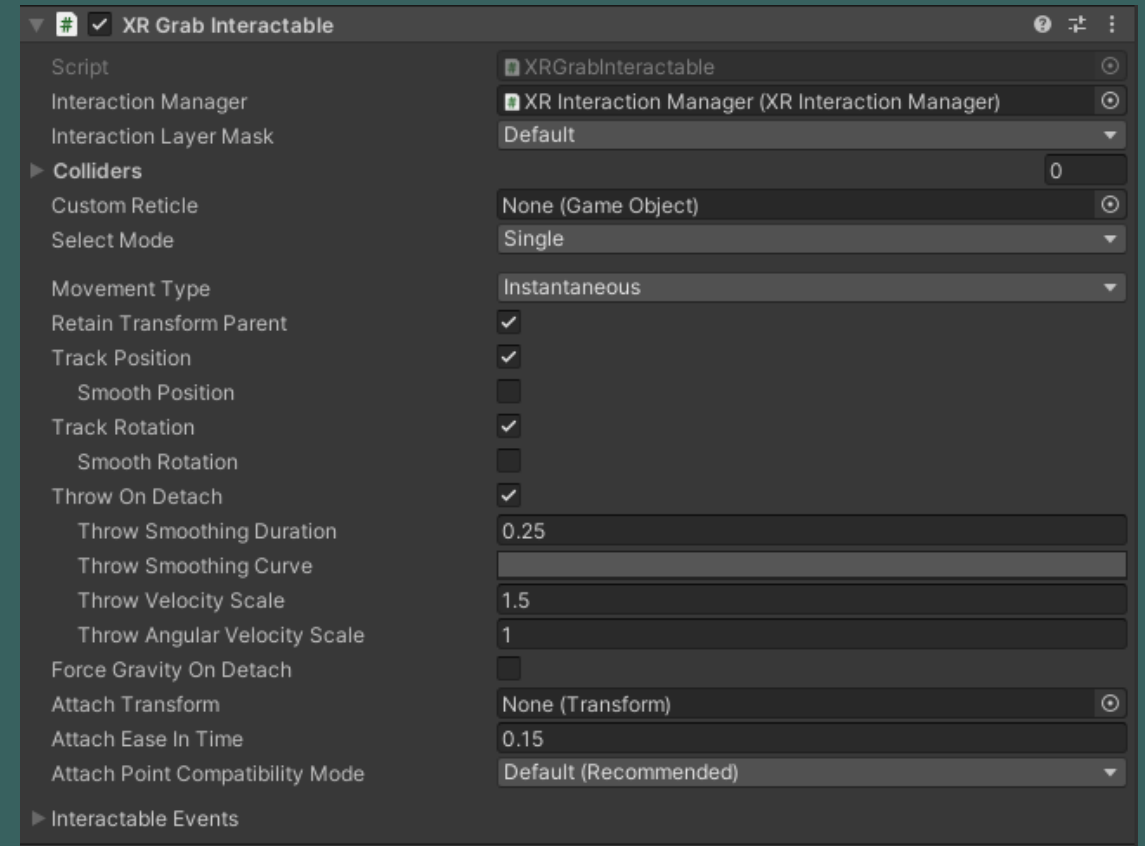
XR Direct Interactor

- Pour interagir en touchant les objets
 - Nécessite un Collider en Trigger sur chaque Hand Controller
- Choix des Layers pour les interactions
- Attention, par défaut, le XRI ne supporte qu'un Interactor (entre le Ray et le Direct) en même temps (tutoriel page 33)



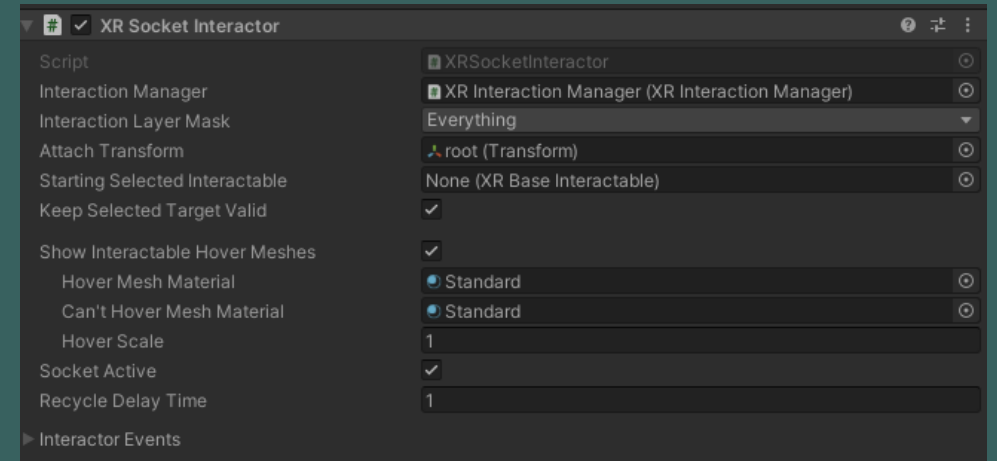
XR Grab Interactable

- Permet de rajouter la préhension à un objet
- Nécessite un Rigidbody & un Collider
- Possibilité d'utiliser les Interactable Events pour appeler vos fonctions



XR Socket Interactor

- Emplacement ou déposer des Interactables
- Nécessite un Collider en mode trigger



- Possibilité d'utiliser les Interactor Events pour appeler vos fonctions

XR Socket Interactor

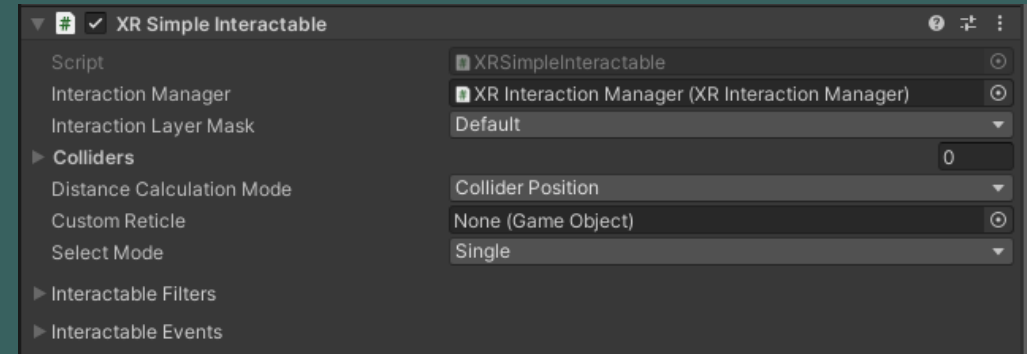
Exemple de fonction de callback

```
using UnityEngine;
using UnityEngine.XR.Interaction.Toolkit;

public class CustomSocketCode : MonoBehaviour
{
    public void OnSelect(SelectEnterEventArgs args)
    {
        Debug.Log(args.interactableObject.transform.name);
    }
}
```


XR Simple Interactable

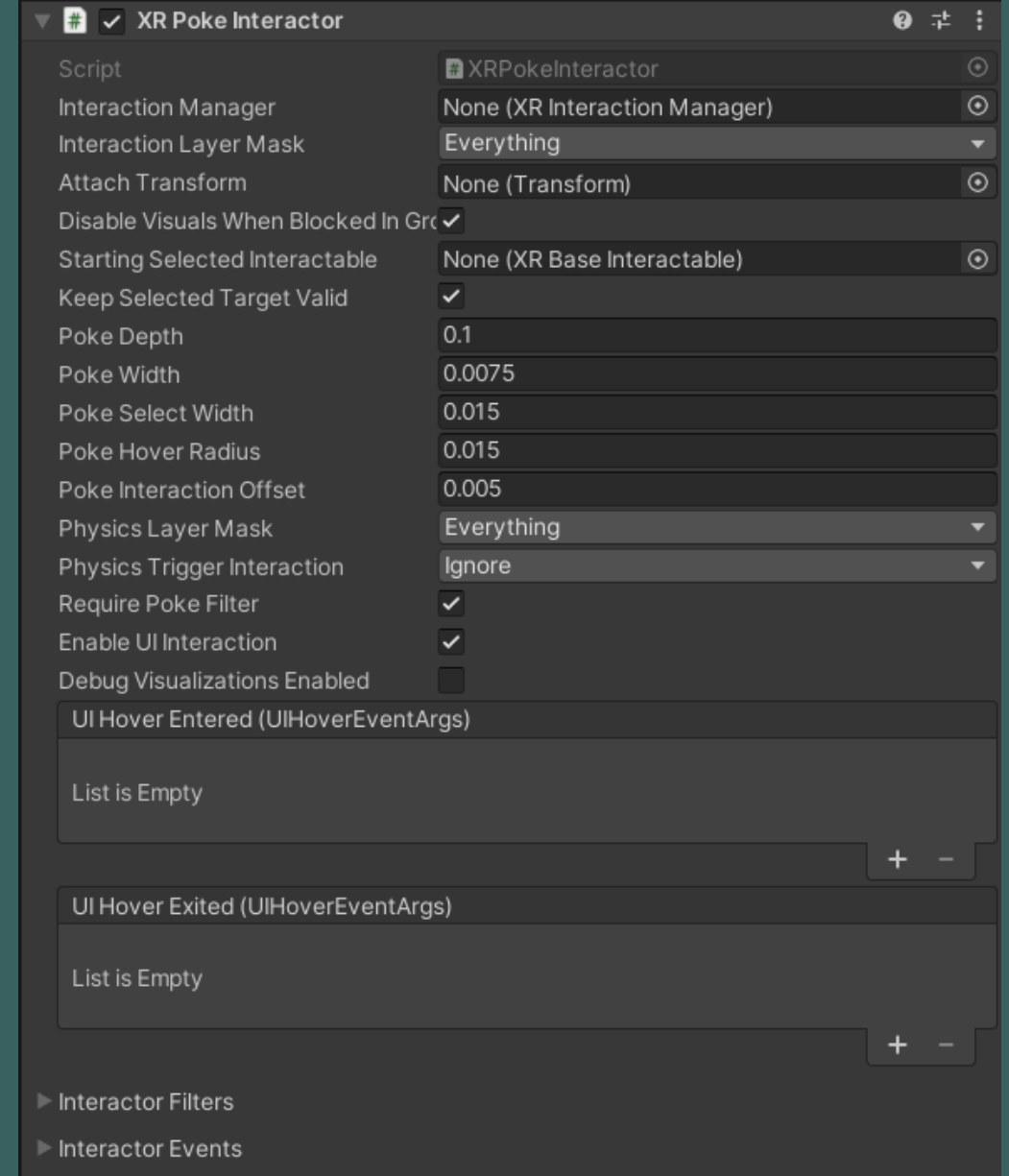
- Permet de créer des objets sélectionnables et actionnables (tel un bouton physique)
- Nécessite un Collider



- Possibilité d'utiliser les Interactable Events pour appeler vos fonctions

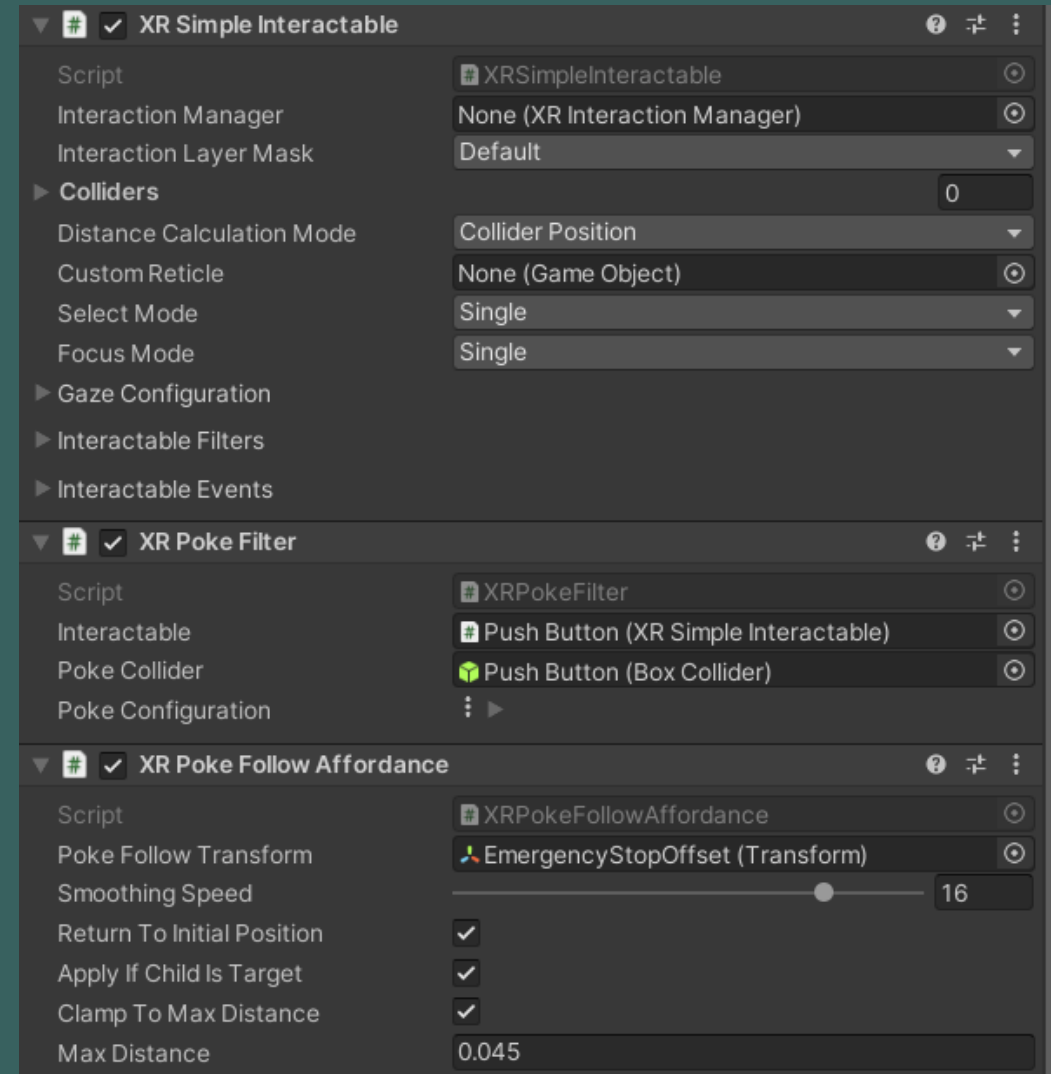
XR Poke Interactor

- Pour interagir en touchant les objets
 - Nécessite un Collider en Trigger sur chaque Hand Controller
 - Comme un « Direct Interactor » mais avec des feedback visuels
- Choix des Layers pour les interactions
- Attention, par défaut, le XRI ne supporte qu'un Interactor (entre le Ray et le Direct) en même temps (tutoriel page 33)



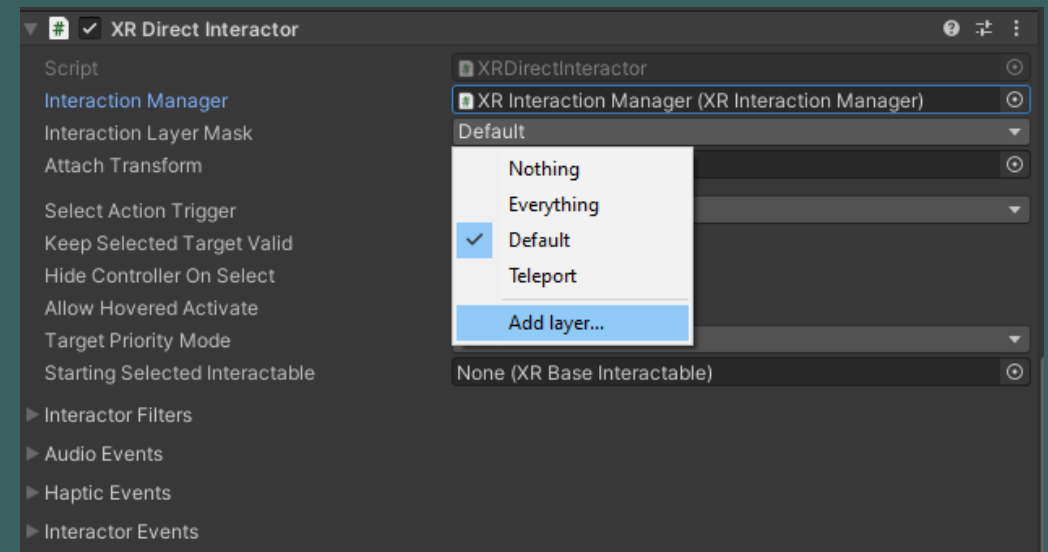
XR Poke Interactions

- Amélioration des Simple Interactors, pour avoir plus de retours visuels
- Possibilité d'utiliser les Interactable Events pour appeler vos fonctions



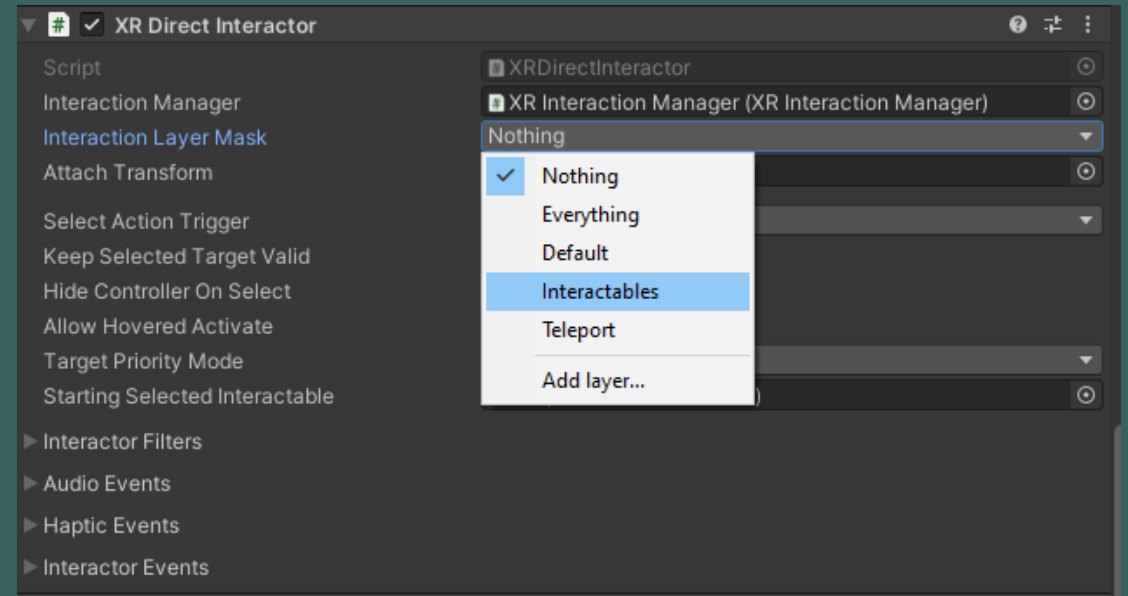
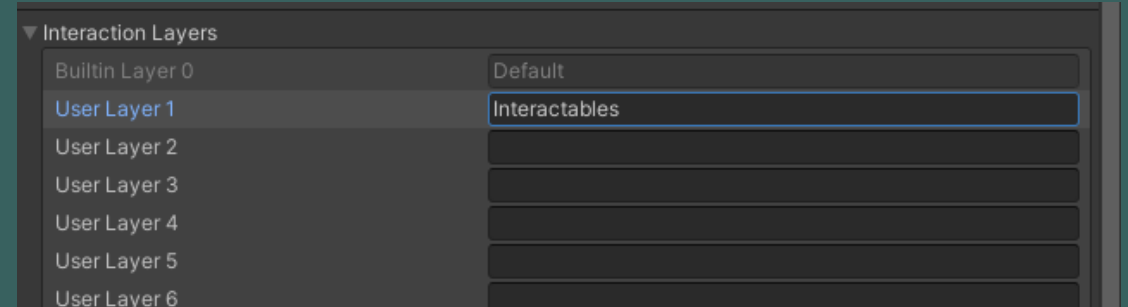
Avoir les 2 XR Interactors

- De base, nous devons choisir entre des Ray ou des Direct Interactors pour les mains
- Nous pouvons modifier le Rig et se téléporter avec le Joystick
- Modifiez votre Rig de cette manière:
 - une copie du Hand Controller en enfant de chaque main
 - Un Direct Interactor dans chaque main, un Ray Interactor dans chaque Controller enfant
- Puis faites « Add layer... », dans Interaction Layer Mask



Avoir les 2 XR Interactors

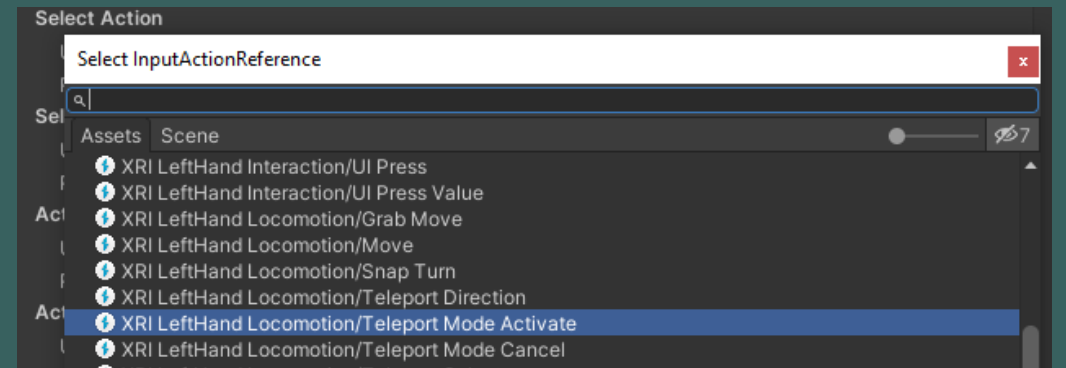
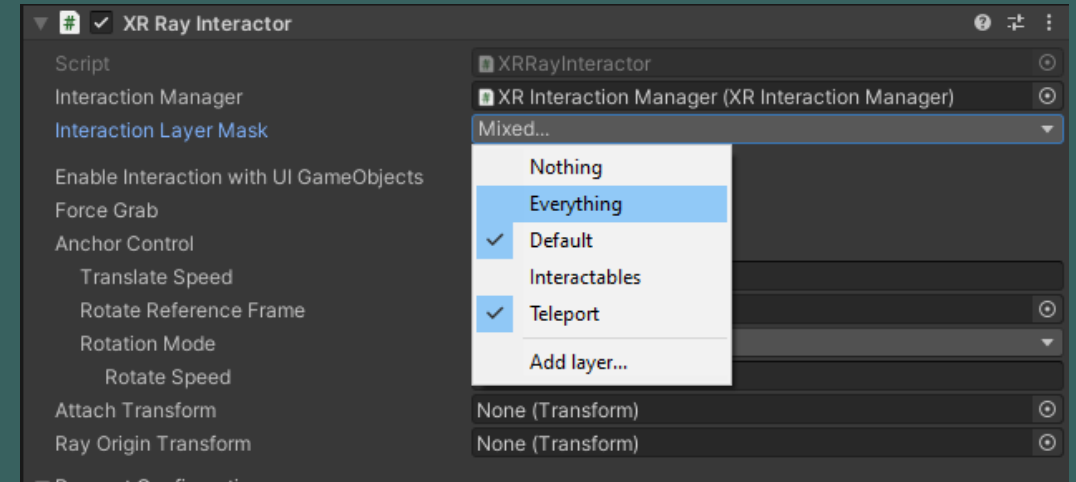
- Ajoutez un Layer « Interactables »
 - Vous modifierez vos **Grab Interactables & Simple Interactables** pour être dans cet Interaction Layer
- N'autorisez que ce Layer pour vos Direct Interactors de vos Hand Controllers



Avoir les 2 XR Interactors

Dans les 2 Hand Controller (with Ray Interactor)

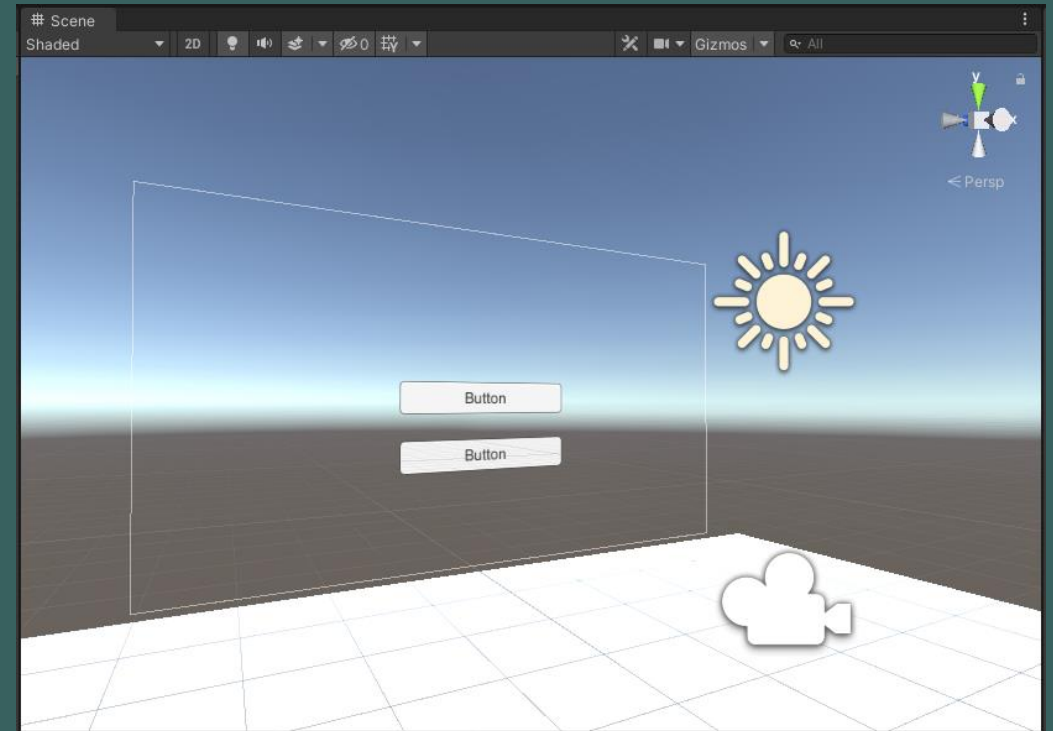
- Dans le XR Ray Interactor:
 - Désélectionnez « Interactables » dans l'Interaction Layer Mask des Ray Interactors
- Dans le XR Controller
 - Décochez:
 - Enable Input Tracking
 - Rotate Action (use Reference)
 - Directional Anchor Rotation Action (use Reference)
 - Translate Anchor Action (use Reference)
 - Modifiez Select Action & Select Action Value:
 - XRILeftHandLocomotion / Teleport Mode Activate
 - XRIRightHandLocomotion / Teleport Mode Activate



Canvas

World Space Canvas

- Utilisation de World Space Canvas
- Utilisation des composants Unity UI habituels
- GameObject
 - ⇒ XR
 - ⇒ UI Canvas



Pour aller plus loin

Ressources supplémentaires

Plugins

- [Google Resonance](#) Gestion du son spatialisé
- [Netcode for GameObjects](#) Gestion du réseau (jeux VR multi-joueurs)

Ressources

- [VR with Andrew](#) Chaine YouTube (en Anglais) spécialisée dans le Unity XR Interaction Toolkit

Des questions ?



TP Shooter