

Unity – Oculus Quest

TP Prise en main

Objectif

Prendre en main le casque Oculus Quest et sa programmation sous Unity

Pré-requis

Être à l'aise avec le moteur Unity, son interface et ses scripts C#

1. Ressources

- Moteur Unity : <https://unity.com/fr>
- Documentation et tutoriels :
 - o <https://learn.unity.com/>
 - o <https://docs.unity3d.com/Manual/index.html>
 - o <https://docs.unity3d.com/Manual/VROverview.html>
 - o <https://docs.unity3d.com/ScriptReference/index.html>
 - o Onglet Learn du Unity Hub
 - o <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@2.5/manual/index.html>
- Développement Oculus/Unity
 - o <https://developer.oculus.com/documentation/unity/book-unity-gsg/?device=QUEST>

2. Guide de démarrage Oculus Quest

<u>Projets S4 ENSIIE</u>	<u>Projets M2RVS!</u>
<ul style="list-style-type: none">- Casques appartiennent à l'ENSIIE- Version 1 de 2019<ul style="list-style-type: none">- Mis à jour en version 50 (fév. 2023)- Version 2 de 2020<ul style="list-style-type: none">- Mis à jour en version 60 (fév. 2024)- Mode développeur activé- Compte "evra-ibisc"	<ul style="list-style-type: none">- Casques appartiennent à l'Univ Evry- Version 1 de 2019<ul style="list-style-type: none">- Mis à jour en version 50 (fév. 2023)- Mode développeur activé- Compte « intervention-iup »

1ère prise en main

- Regarder la vidéo sur la santé et la sécurité. **Prenez garde notamment aux collisions avec les personnes et les objets environnants.**
- https://www.oculus.com/safety-center/quest/?locale=fr_FR
- Ajuster le casque sur votre tête en modifiant le DIO, le serrage et la position. **S'assurer de voir correctement.**
- Paramétrer le **périmètre de sécurité** (*guardian*) : sol puis surface de jeu soit stationnaire soit *roomscale*
- Se familiariser avec les **interactions** et les menus.
- Si besoin consulter l'aide dans Paramètres -> Aide

Matériel associé

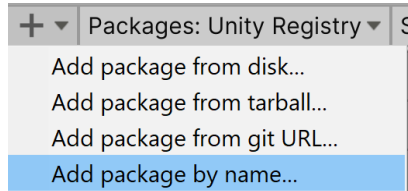
- Adaptateur et câble USB 2 type C
 - o Permet la charge et le transfert de votre build depuis le PC
 - o Si votre PC le permet, permet l'utilisation du mode PC VR (« Quest Link »)
 - Un câble USB 3 sera plus optimal pour ce mode
 - o Eteignez le casque quand vous ne l'utilisez pas, pour économiser la batterie.
- Manettes :
 - o Droite/gauche, se repèrent à l'aveugle avec la position de la « gâchette de paume »
 - o Fournies avec 2 piles, fonctionnent avec piles rechargeable si vous en avez
 - o Éviter de laisser les piles à l'intérieur si vous n'utilisez pas pendant plusieurs jours
 - o Si non reconnues au démarrage, appuyer sur les boutons, ou enlever et remettre les piles

Préparation du développement

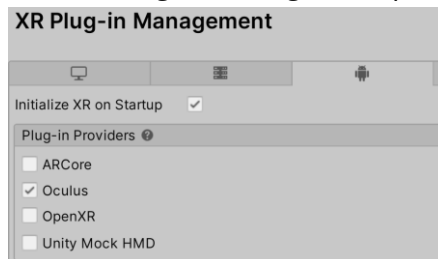
- Installer Unity (ce TP est vérifié sur 2021.3.11)
 - o Modules : Android Build Support, Android SDK & NDK tools, OpenJDK, Windows Build Support
- Installer Sidequest
 - o <https://sidequestvr.com/setup-howto>
 - o Permet de déployer des applications non officielles en mode développeur
 - o Permet de streamer le casque vers le pc
 - o Pour déployer votre build : menu -> "install apk from folder"
 - o Votre application apparait alors dans le casque dans Applications -> sources inconnues
- Brancher l'Oculus Quest au PC
 - o Autoriser la 1^{ère} connexion depuis le home virtuel
- Lancer Sidequest et vérifier que le casque est reconnu
- Si votre PC est compatible VR
 - o Télécharger et installer le logiciel Oculus :
https://www.oculus.com/Setup/?locale=fr_FR
 - o Brancher le casque et suivre les instructions

3. Exercice 1 : Construction du projet pour Oculus Quest

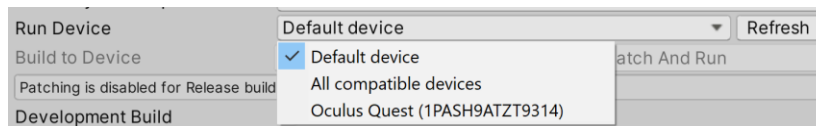
- Créer un nouveau projet de type VR
- Ouvrir le *Package Manager* (Menu *Window*)
 - o Vérifier que *XR Plugin Management 4.2.1* et *Oculus XR Plugin 3.0.2* sont installés
 - *com.unity.xr.management*
 - *com.unity.xr.oculus*
 - o Installer *XR Interaction Toolkit 2.5.1* et parmi les *Samples* importer les *Starters Assets*
 - *com.unity.xr.interaction.package*



- o Mettre à jour les autres packages si nécessaire
- Ouvrir les *Project Settings* (Menu *Edit*)
 - o Paramétrer *XR Plug-in Management* pour fonctionner avec Oculus



- o Vérifier les éventuels problèmes dans *Project Validation*
- Ouvrir et vérifier les *Build Settings* (Menu *File*)
 - o Choisir la plateforme cible (Android ou Windows) et valider avec *Switch platform*
 - o Pour Android, vérifier dans la liste *Run Device* que votre oculus est bien détecté



- Ouvrir et vérifier les *Player Settings* (depuis *Build Settings* ou *Project Settings*)
 - o *Company name, product name, version*
 - o Pour Android, *Other settings*
 - *Graphics API : OpenGL ES*
 - *Identification*
 - *Package name*
 - *Version...*
 - *Minimum API Level : 19**
 - *Target API Level : auto*
 - *Minimum API Level*

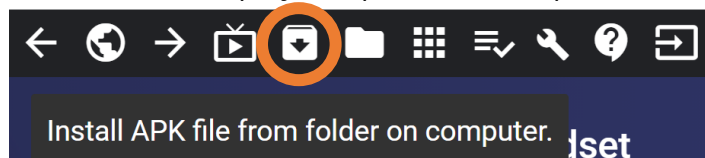
*Parfois impossible à choisir car grisé, c'est souvent un problème réseau => en changer, ex. partage 4G

4. Exercice 2 : Build et test de la démo

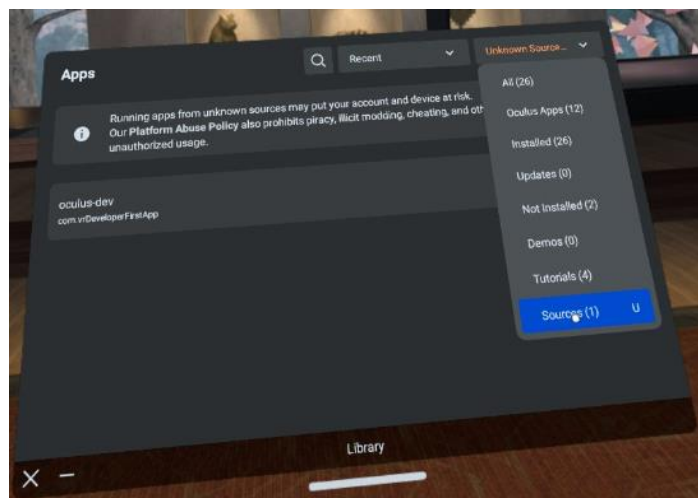
- Changer le nom du produit et celui du package par *com.votreformation.votrenomdeprojet*. Ces 2 noms permettront d'identifier l'application dans le casque
- Ouvrir la scène *Assets/Samples/XR Interaction Toolkit/2.5.1/Starter Assets/Demo Scene*
- Brancher le casque

Cas Android

- Depuis les *Build Settings*, générer votre application pour le casque
 - o Choisir la scène cible
 - o Option 1 : *Build and run*
 - Va générer l'apk, l'enregistrer sur le disque du PC, déployer l'apk sur le casque, lancer l'application dans le casque
 - o Option 2 : *Build*
 - Va générer l'apk et l'enregistrer sur le disque du PC
 - Il faut ensuite déployer l'apk soi-même, par ex. avec Sidequest



- Lancer l'application depuis le casque (*Apps -> Sources inconnues -> votre Product Name*)



- NB : le temps de build and run est variable mais en général extrêmement long surtout la 1^{ère} fois (optimisations ensuite)
- Chaque déploiement écrase le précédent à moins de changer à chaque fois le nom du package (Player settings). Veiller à conserver vos apk fonctionnels intermédiaires sur votre PC.

Cas PC

- Brancher le casque, lancer l'application Oculus sur le PC et lancer Quest Link
- Exécuter la scène dans l'éditeur comme habituellement
- Vérifier que le *head tracking*, le *hand tracking* et toutes les interactions sont fonctionnelles.

- Etudier attentivement les objets, *prefabs* et composants.
- Consulter la documentation du package :
<https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@2.5/manual/index.html>

5. Exercice 3 : Construction d'une nouvelle scène

En réutilisant des composants de la scène exemple, créez une nouvelle scène selon les indications suivantes :

- L'utilisateur apparait sur un côté d'une grande pièce fermée de 10m x 10m
- Au milieu de la pièce, une « table » de 2m de long, 1m de large et 80cm de haut
- Sur la table, on trouve 3 objets : un cube, une pyramide et une sphère, de petites tailles et avec une couleur unie
- L'utilisateur peut naviguer
 - o En se téléportant par *Raycast* où il souhaite dans la pièce, comme dans l'exemple
 - o Par le stick, avec une vitesse 2 fois plus lente que dans l'exemple
 - o Sans pouvoir traverser les murs
- Il peut manipuler les 3 objets par *Main virtuelle simple*
 - o Les mains sont visibles sous forme de contrôleurs virtuels
 - o Chaque objet change d'apparence lorsqu'il est sélectionnable (bleu) et quand il est sélectionné (jaune), et reste à sa couleur d'origine sinon
 - o La sphère n'est pas soumise à la gravité
 - o Il n'y a pas de sélection par *Raycast*
- Les ombres sont de bonne qualité
- Les objets font un son lors de leur collision avec l'environnement
- Une UI au mur nous indique ce que l'on peut faire

6. Travail en groupe

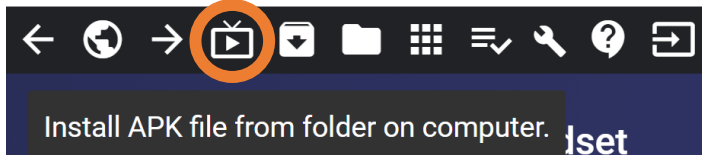
- Paramétrer un dépôt et un client git
- Télécharger un exemple de *.gitignore* pour Unity sur la page du cours
- Faire un test de push/pull

7. Capture et stream vidéo

- Pour capturer une vidéo de votre application après son lancement, appuyer sur le bouton Oculus, et sélectionner l'option « Enregistrer une vidéo »



- Faire de même pour arrêter la capture
- La vidéo .mp4 peut-être visionnée dans le Quest dans l'application *Camera*
- Pour la récupérer sur le PC, aller dans le répertoire `sdcard/Oculus/VideoShots` (via Sidequest par ex.)
- Tester également le stream depuis Sidequest



8. Outils de debug Android (sous réserve)

- Avec votre casque branché, localiser et lancer `adb.exe` (a priori dans un dossier du type `C:\Program Files\Unity\Hub\Editor\2021.3.Xf1\Editor\Data\PlaybackEngines\AndroidPlayer\SDK\platform-tools`)
- Ajouter dans votre projet un script qui affiche des messages dans la console lorsque les boutons A, B, et la gâchette sont appuyés
- Déployer
- Lancer la commande `adb logcat -s Unity` dans un terminal
- Tester l'application depuis le casque en observant les logs sur le PC

9. Tutoriels et liens

- Youtube
 - o [Frontend Fanatics](#)
 - o [Valem](#)
 - o [Dilmer Valecillos](#)
- [Sidequest](#)
- Vous pourrez télécharger et tester les tutoriels et jeux disponibles en démo ou sur le compte du casque