

Hololens

Formation pompiers



Mise en place Unity Hololens 2

Auteur	Dorian Tan
Superviseur	Julien Senn
Expert	Lionel Heim
École	He-arc, Neuchâtel
Numéro de projet	282
Date début	20.02.2023
Date de fin	28.07.2023
Soutenance	24.08.2023
GitLab	https://gitlab-etu.ing.he-arc.ch/groups/isc/2022-23/niveau-3/3285-tb-il/282/-/wikis/home

Table des matières

1.	Introduction	3
2.	Logiciels.....	3
2.1.	Unity.....	3
2.2.	Visual Studio.....	4
2.3.	MRTK – Mixed Reality Toolkit	5
3.	Projet pas-à-pas	6
3.1.	Création d'un projet Unity	6
3.2.	Paramètres dans Unity.....	7
3.3.	Intégration MRTK	8
3.4.	Unity.....	10
3.5.	Build dans Visual Studio	12
3.5.1.	Déploiement en WIFI	13
3.5.2.	Déploiement USB	14

1.Introduction

Dans ce document, il est décrit comment mettre en place un projet Unity pour travailler avec les HoloLens 2 de Microsoft.

2.Logiciels

2.1. *Unity*

Il est recommandé d'utiliser la version **2020.3.45f** pour travailler avec les HoloLens 2. Veuillez à vous rendre sur le site officiel de Unity et de télécharger le *Unity Hub*.

<https://unity.com/download>

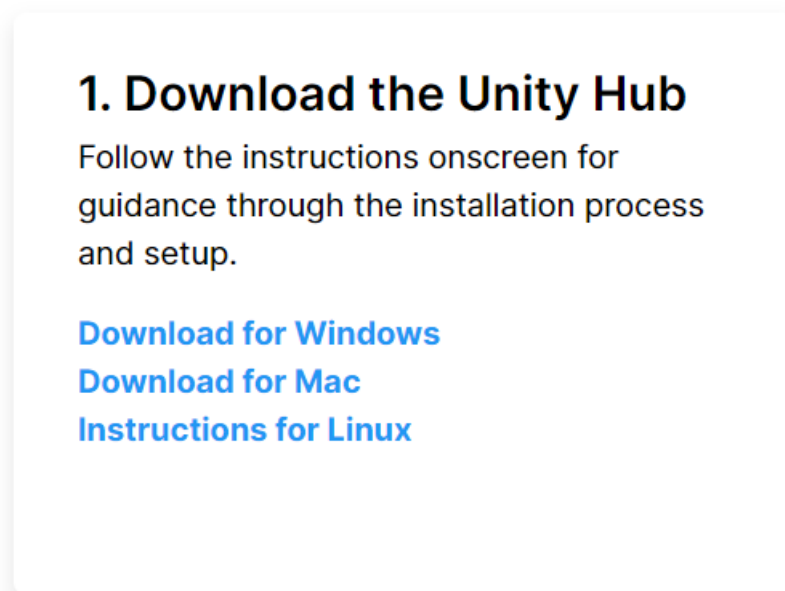


Figure 1: Choix de téléchargement du Unity Hub

Grâce au Unity Hub vous pouvez télécharger toutes les versions de Unity que vous souhaitez. Une fois la fenêtre ouverte vous pouvez sélectionner « Installs » et chercher la version recherchée (figure 2)

Si la version n'apparaît pas, car elle est trop vieille, vous pouvez cliquer sur le bouton « Install editor » (figure 2). Elle vous indiquera un lien pour télécharger des versions antérieures de Unity (figure 3).

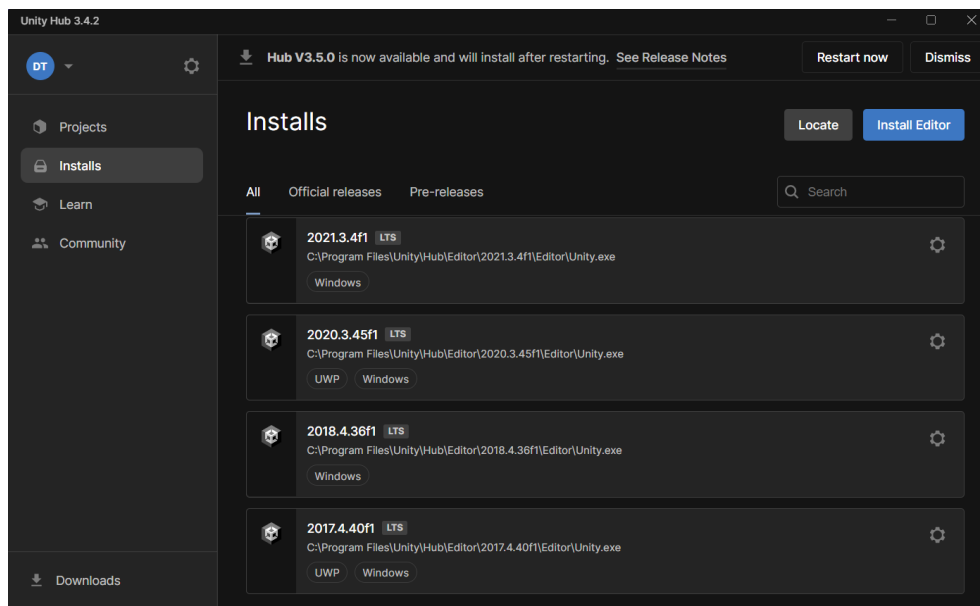


Figure 2: Unity Hub, version de Unity à télécharger

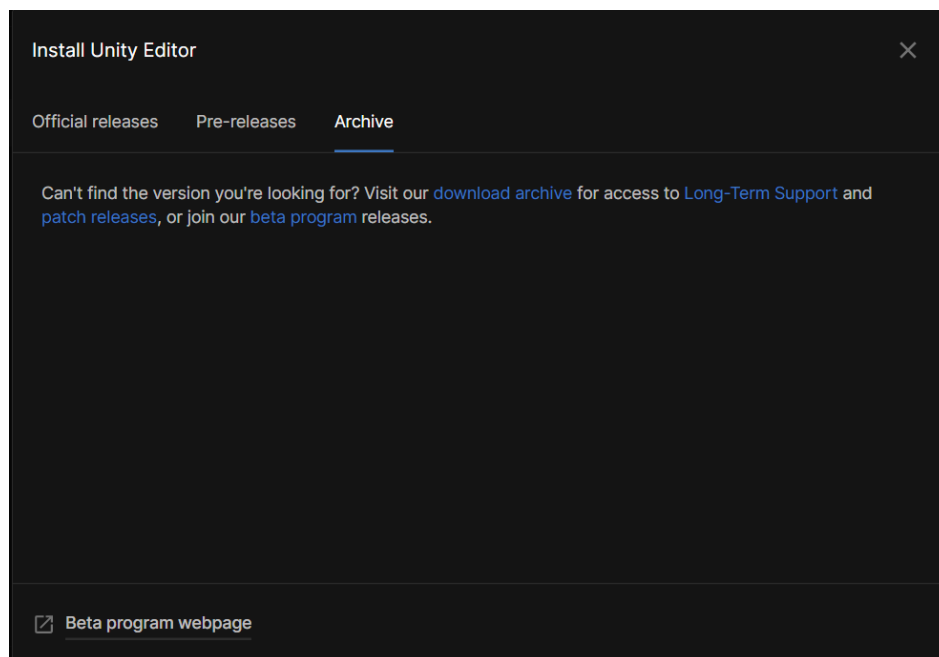


Figure 3: Lien pour télécharger des versions antérieures de Unity

2.2. Visual Studio

La version utilisée actuellement est la **2022** (figure 4).

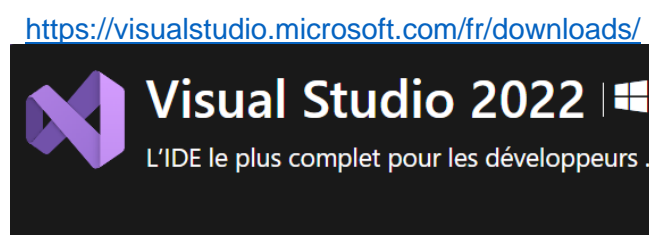


Figure 4: Visual Studio 2022

Lors du téléchargement de la version 2020, il faut ajouter des packages en plus. Le *Universal Windows développement* avec les SDK 11 et 10 comme indiqués dans la figure 5.

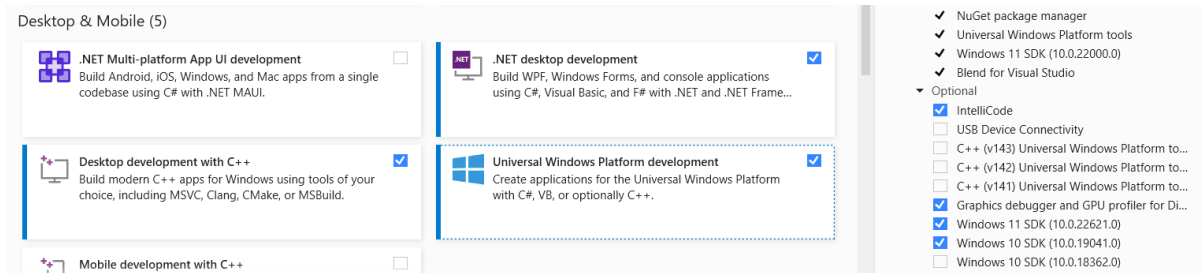


Figure 5: Package universal windos development

De plus les 2 packages *Gaming* pour faire fonctionner avec Unity (figure 6).

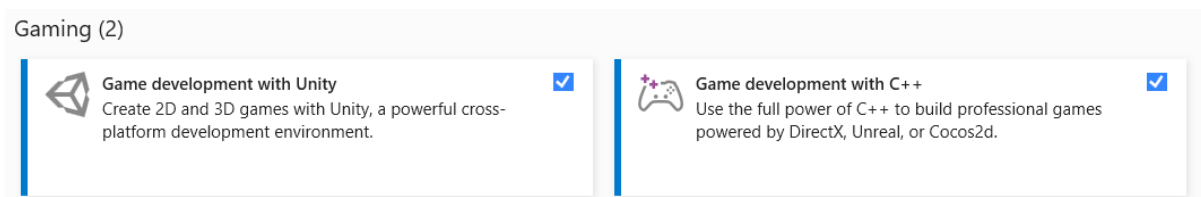


Figure 6: Packages pour Unity

2.3. MRTK – Mixed Reality Toolkit

Pour travailler en réalité, augmenter sur Unity il faut installer le MRTK, c'est ce programme qui contient tous les fichiers pour utiliser des objets en 3D et pour repérer l'espace ainsi que les profondeurs.

Pour installer le MRTK, il faut se rendre sur le site de Microsoft :

<https://www.microsoft.com/en-us/download/confirmation.aspx?id=102778>

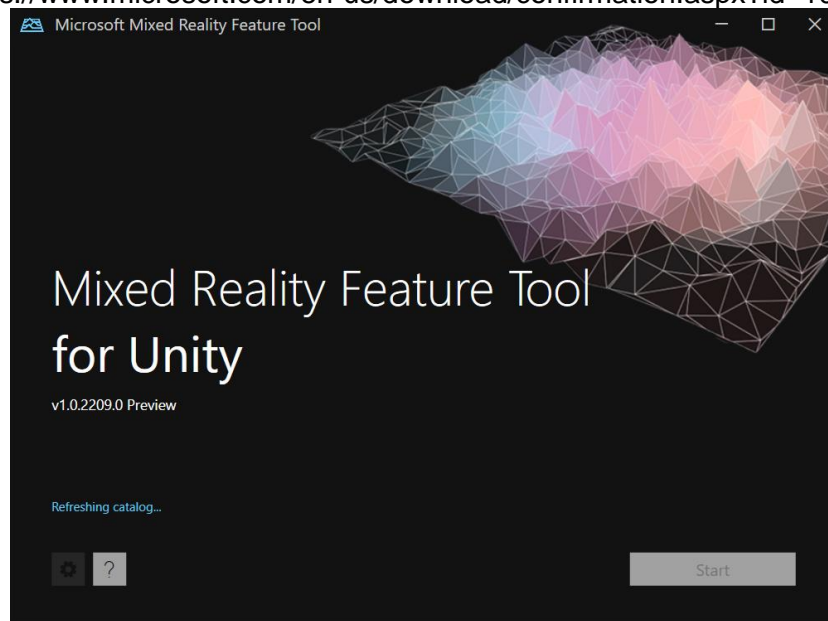


Figure 7: MRTK interface

3. Projet pas-à-pas

Ici est décrit comment créer un projet Unity pour utiliser les HoloLens, de l'éditeur en passant par le build et le déploiement dans le casque.

3.1. *Création d'un projet Unity*

Dans le Unity Hub, appuyer sur « New Project » (figure 8).

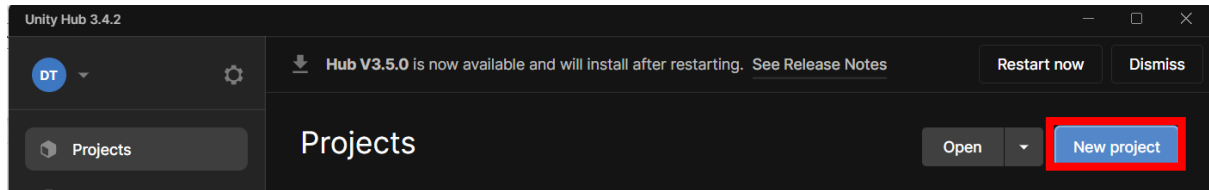


Figure 8: Unity Hub interface

En premier lieu, changer la version de l'éditeur pour celui que vous souhaitez (figure 9).

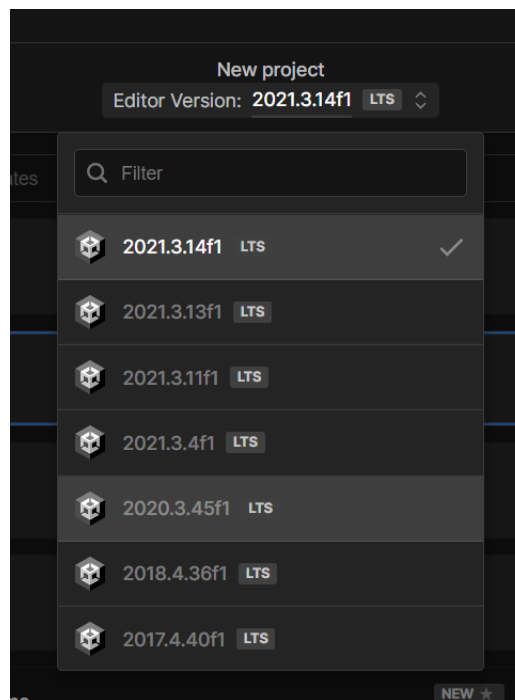


Figure 9: Changement de version de l'éditeur

Sélectionner projet 3D (figure 10).

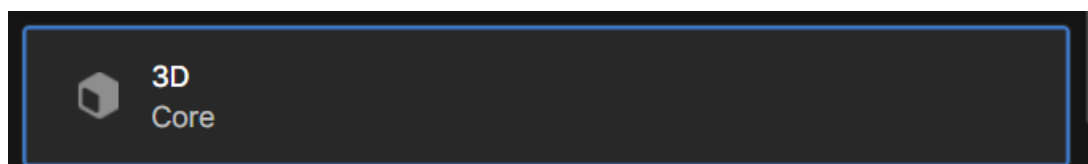


Figure 10: Projet 3D

Renommer le projet et changer l'emplacement comme désiré puis appuyer sur « Create project » (figure 11).

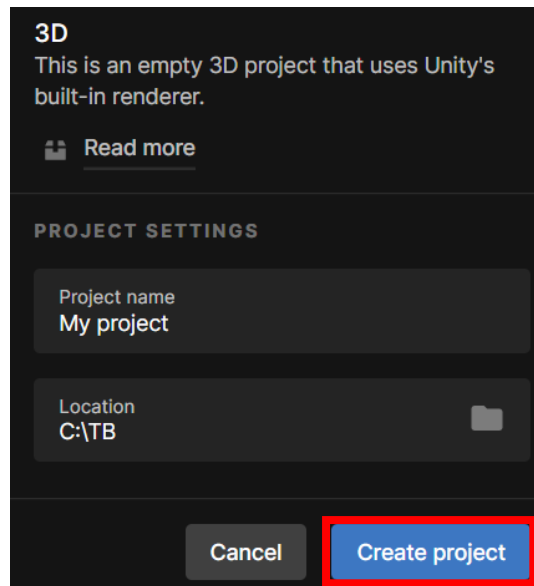


Figure 11: bouton création de projets

3.2. Paramètres dans Unity

Une fois dans l'éditeur Unity, commencer par changer de plateforme pour le build. Pour ce faire il faut aller dans les « build settings » situés dans « File » en haut à gauche (voir figure 12).

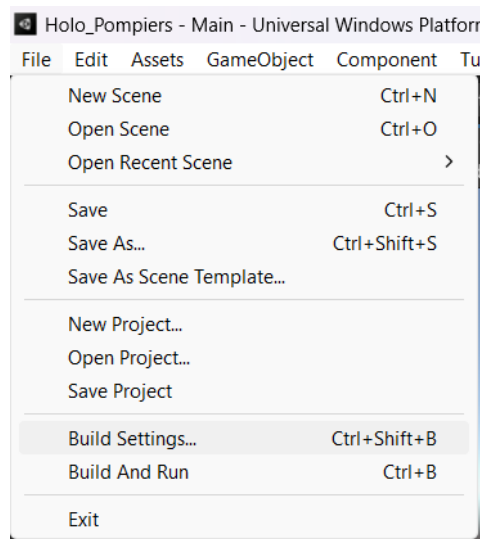


Figure 12: emplacement du Build Settings

Choisir « Universal Windows Platform » dans l'onglet à gauche (figure 13).

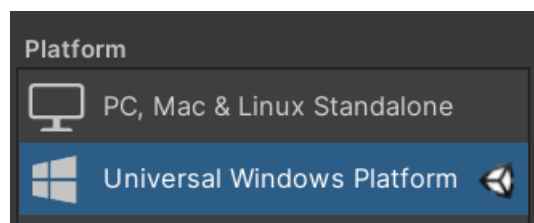


Figure 13: Choix de plateforme

Après avoir changé de plateforme, changer les paramètres comme le montre la figure 14 ci-dessous. Certains de ces paramètres changent si vous utilisez un HoloLens 1 ou si vous voulez déployer en USB et non par WIFI. Si tel est le cas changé « Build and Run on » en « USB Device » (figure 15).

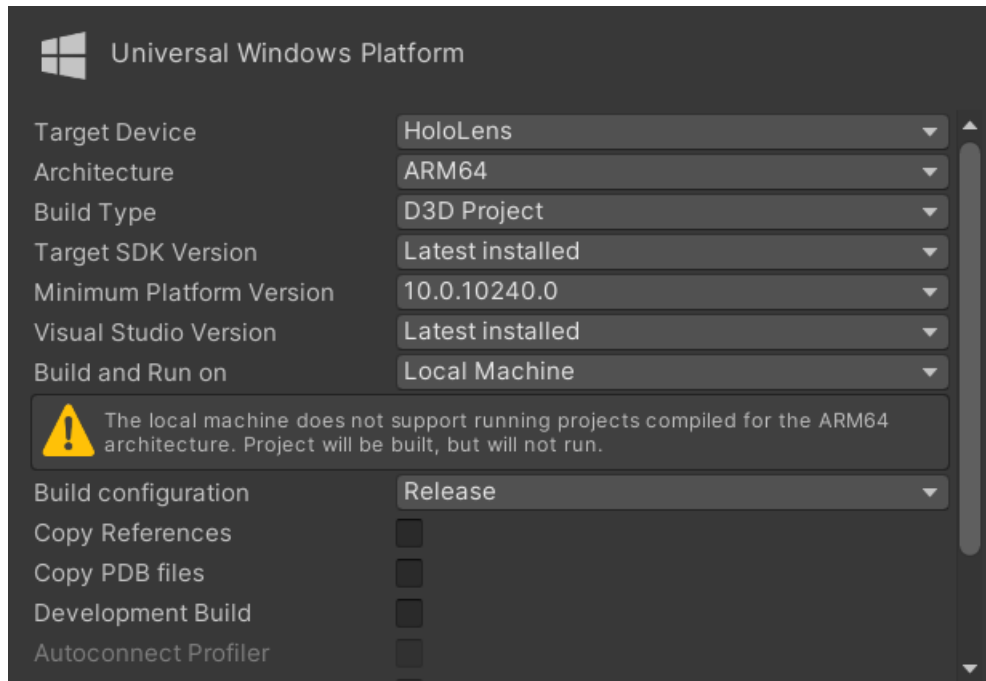


Figure 14: paramètres de build

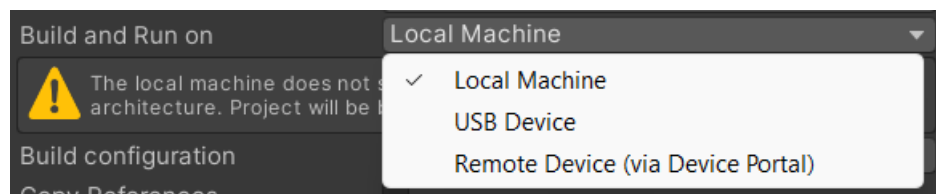


Figure 15: Changement pour le déploiement

3.3. *Intégration MRTK*

Maintenant que le projet est créé, vous pouvez intégrer le MRTK dedans. Une fois le logiciel (MRTK) ouvert il faut attendre quelques secondes avant que le bouton « Start » ne soit activé (figure 7).

Ensuite, rechercher votre projet à l'aide des 3 points (figure 16) puis appuyer sur « Discover Features ».

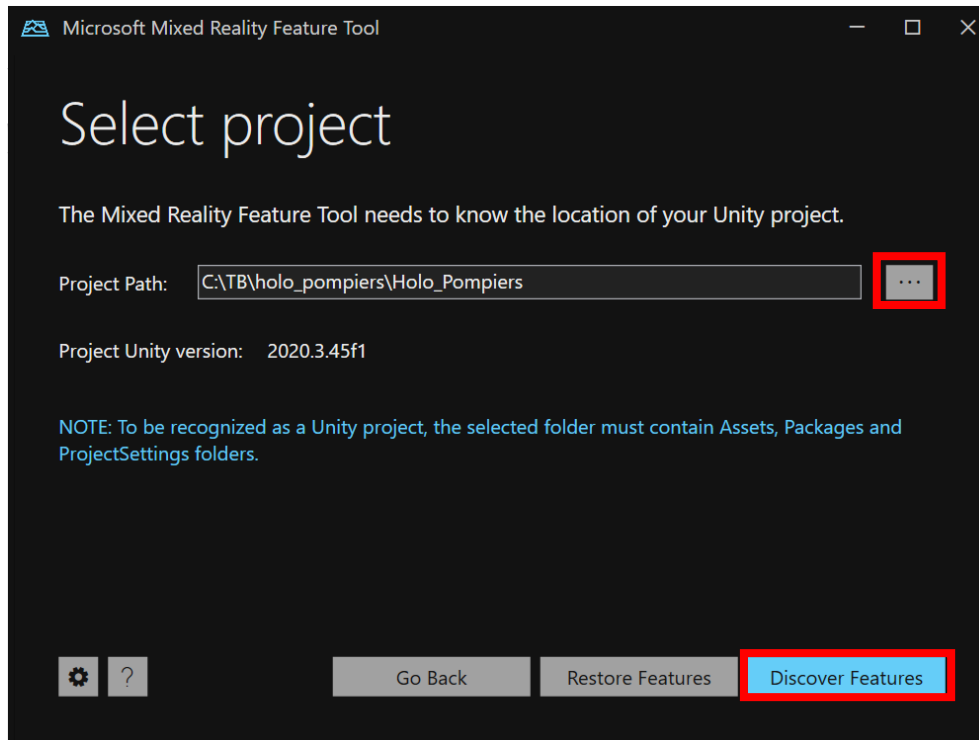


Figure 16: sélection de projets

Plusieurs packages peuvent être ajoutés à votre projet selon vos différents besoins, cependant les indispensables sont ceux sélectionnés dans les figures 17 et 18.

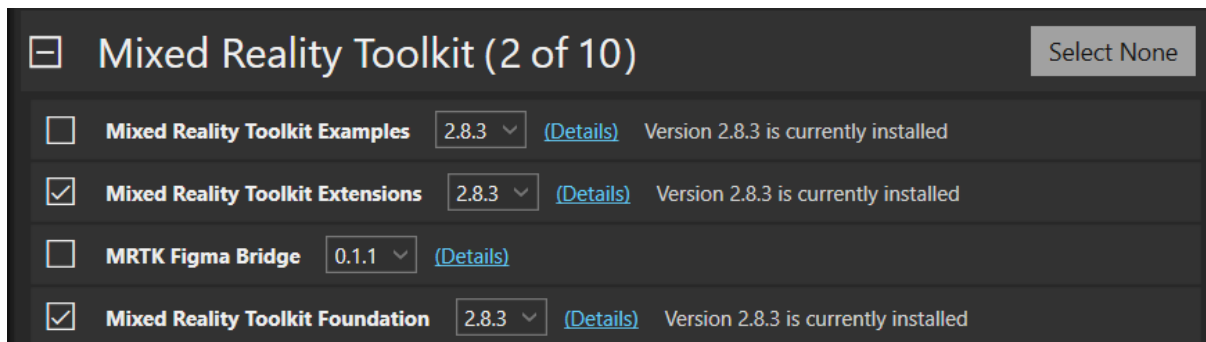


Figure 17: packages MRTK

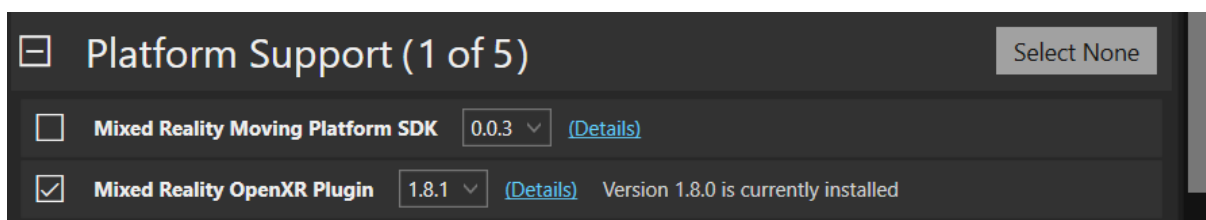


Figure 18: package MRTK

Une fois les packages sélectionnés appuyer sur « Get features » en bas à droite ». Dans l'onglet de gauche devraient se trouver les packages que vous souhaitez (figure 19). Puis appuyer sur « Import » en bas à droite.

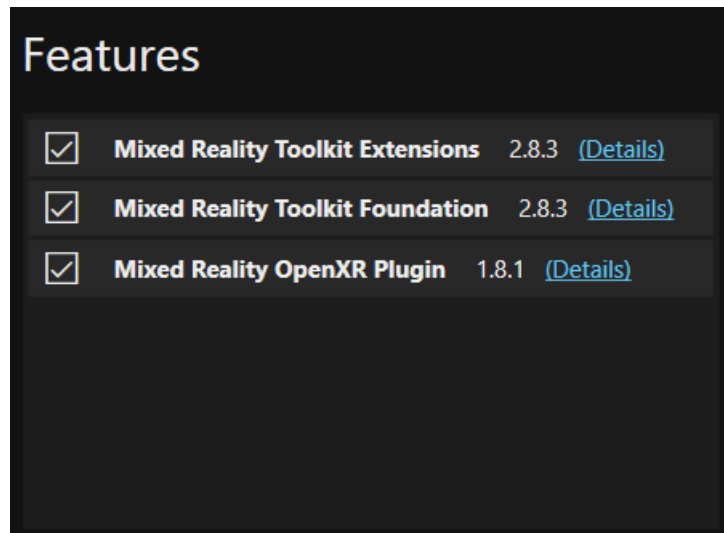


Figure 19: Features souhaité

3.4. Unity

Maintenant l'onglet « Mixed Reality » est apparu dans Unity (figure 20).

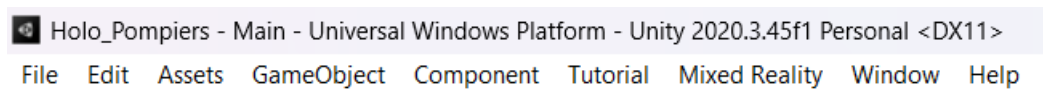


Figure 20: Unity onlget

Une page apparaît pour configurer votre projet avec les paramètres recommandés. Accepter les paramètres proposés (figure 21).

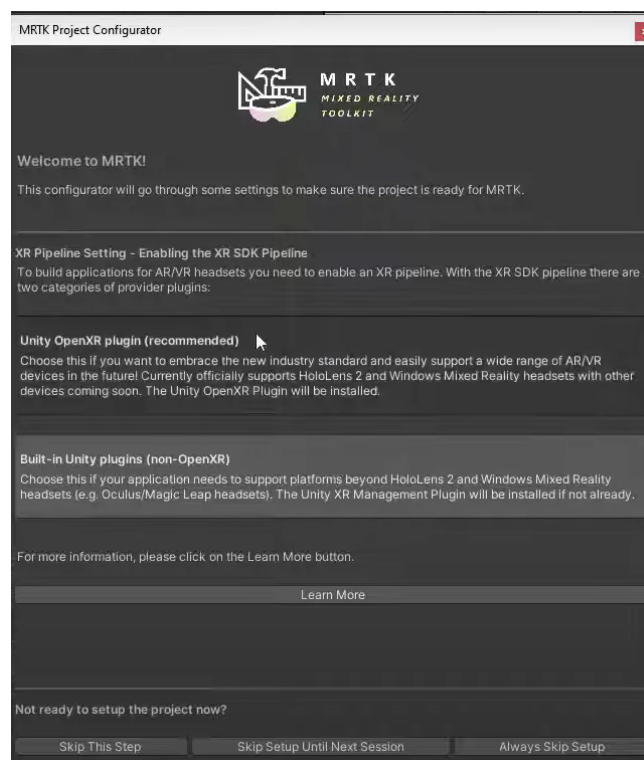


Figure 21: Paramètre recommandé

Une fois activer une fenêtre s'affiche, il faut ajouter les paramètres comme ci-dessous (figure 22). Et ensuite, appuyer sur « Next ».

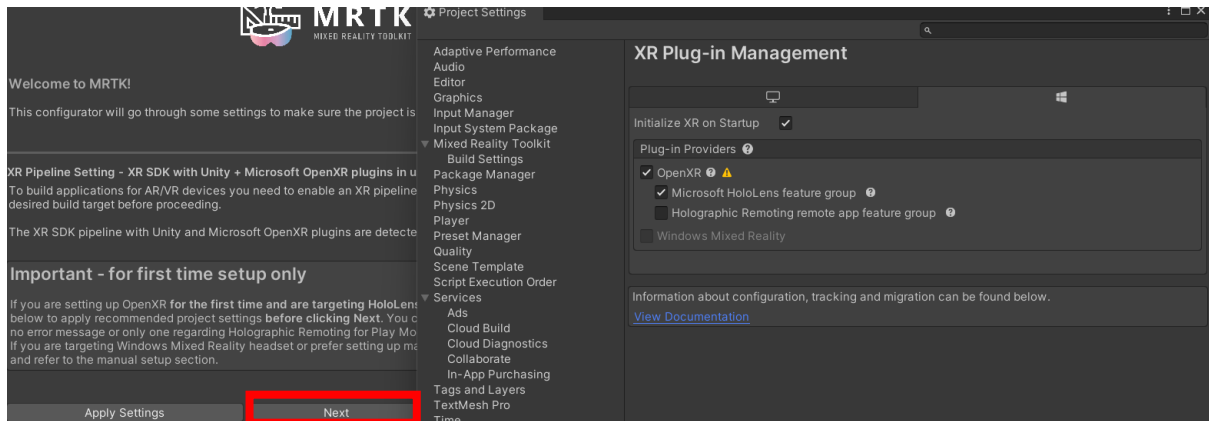


Figure 22: Paramètres XR Plug-In

Puis dans « OpenXR », changer la *Depth SubmissionMode* en « Depth 16 Bit » (figure 23). Il faut aussi ajouter « Microsoft Hand Interaction Profile ».

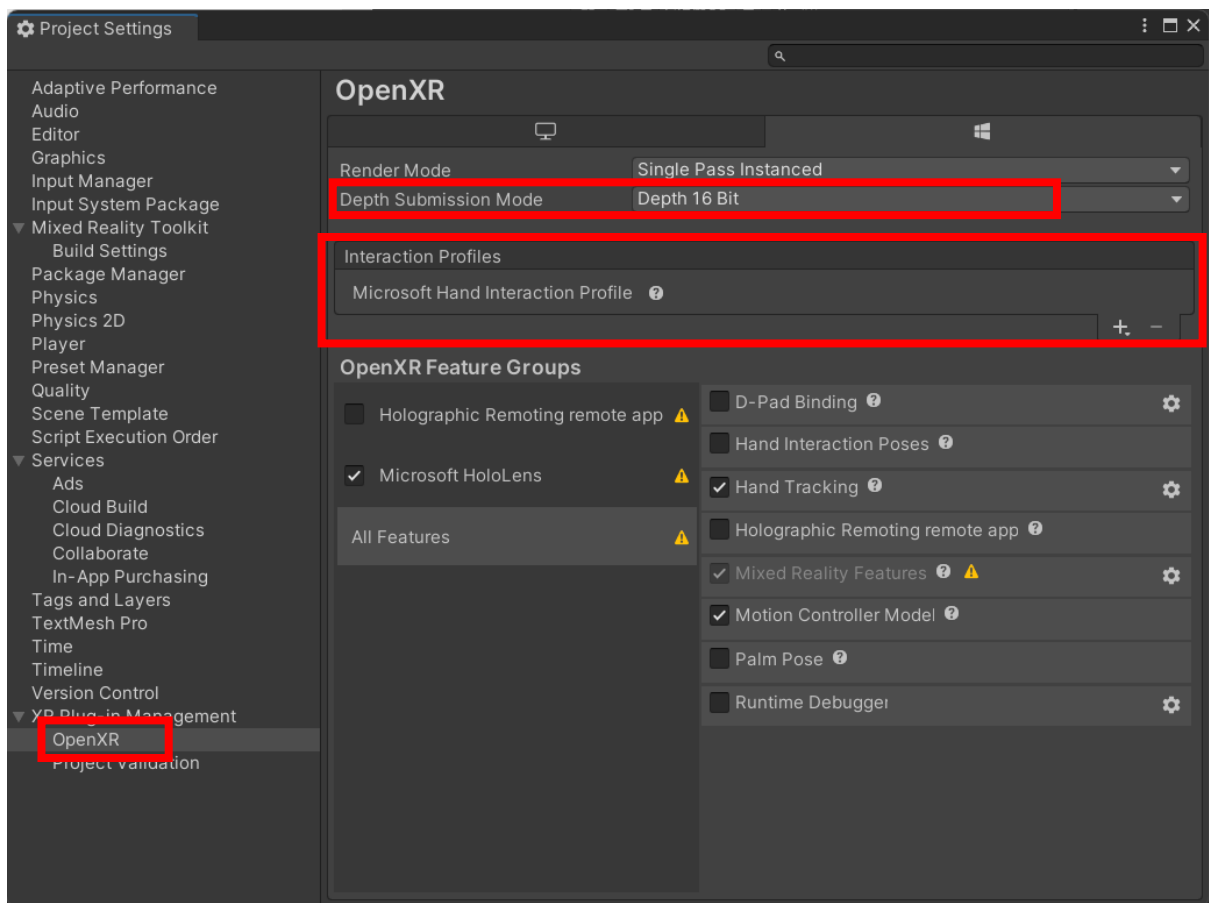


Figure 23: OpenXR Depth Submission

Une fois ces paramètres ajoutés, vous pouvez « Apply » le tout. Ajouter de plus le TMP Essentials qui est proposé une fois que vous avez « Apply » (figure 24).

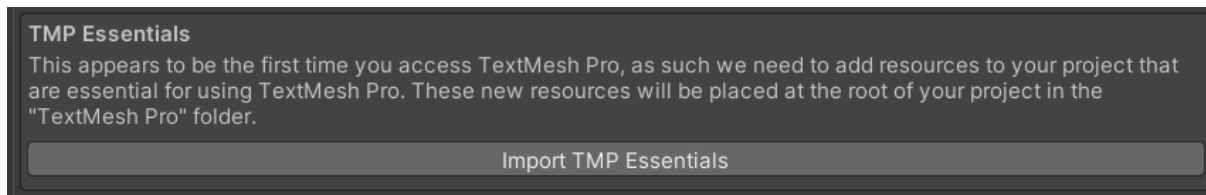


Figure 24: TMP essentiel

3.5. *Build dans Visual Studio*

Retournez dans les « Build Settings » (figure 12). Assurez-vous d'être dans « Universal Windows Platform ». Cliquez sur Build (figure 25).

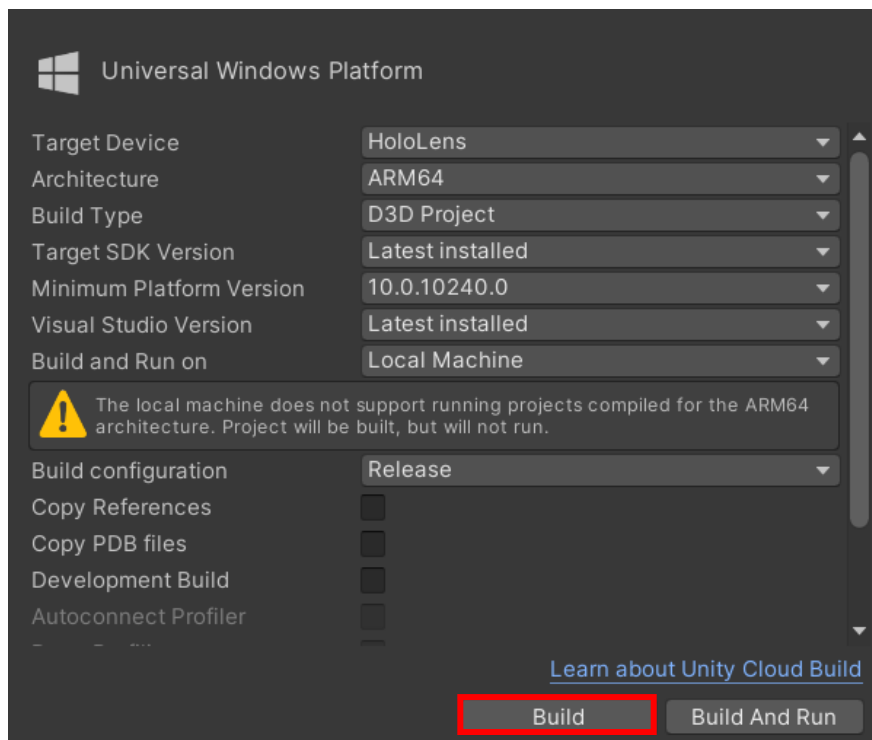


Figure 25: Build settings, créer un build

Veillez à créer un build dans un dossier vide.

Une fois le build créé, double cliquer sur `.sln` (figure 26).

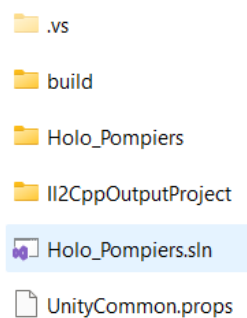


Figure 26: fichier de build

3.5.1. *Déploiement en WIFI*

Pour la façon en wifi, il faut ajouter l'IP du casque dans les paramètres de Visual Studio. Pour l'ajouter, il faut aller dans les Projets -> Propriétés (figure 27).

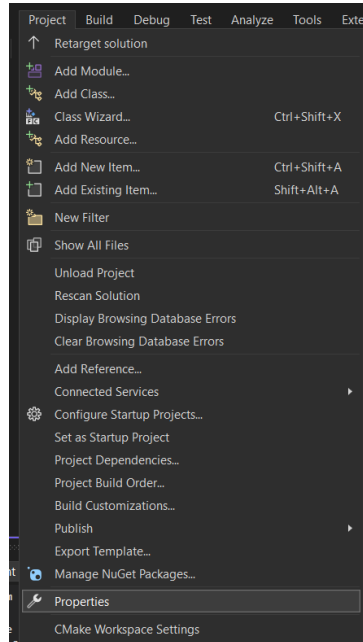


Figure 27: Project properties

Puis dans Debugging, entrer l'IP dans « Machine Name » (figure 28).

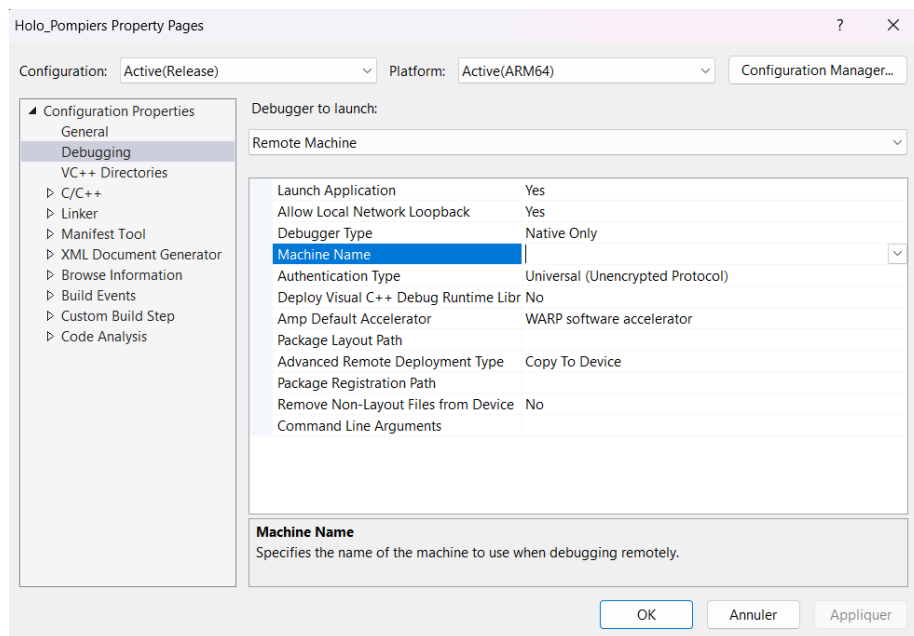


Figure 28: Emplacement adresse IP

Une fois l'IP rentré, il faut lancer en mode sans debug (figure 29).

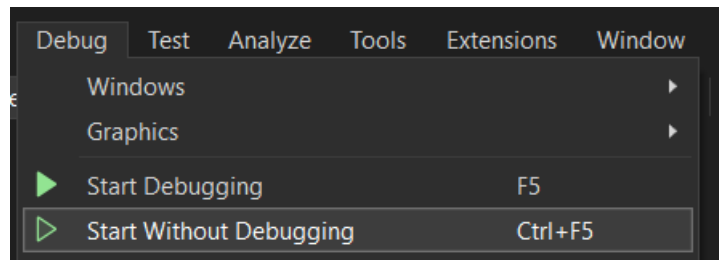


Figure 29: Lancer le déploiement

Pendant le déploiement il faut s'assurer que le HoloLens est en mode développeur et qu'il dans le même réseau wifi que l'ordinateur. De plus il est préférable que le casque ne tombe pas en veille lors du déploiement, cela peut créer des problèmes.

3.5.2. *Déploiement USB*

Pour la méthode avec USB il suffit de mettre en « Device » et de lancer sans debug (figure 30).

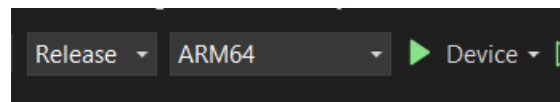


Figure 30: Déploiement USB