

Langues des signes  
ESP - EPITECH

# **Softwares et outils**

Pierre Ginisty

# Sommaire

1 – Quel software, présent sur le marché, est le plus adapté à notre projet

a – Unity

b – Unreal Engine

c – Conclusion

2 – Matériels nécessaires

3 – Assets pour l'obtention de mains modélisées

4 – Projets déjà existant

# 1 – Quel software, présent sur le marché, est le plus adapté à notre projet

## a – Unity

Les avantages d'utiliser Unity pour une application AR/VR de hand tracking sont les suivants :

1. **Prise en charge de la technologie hand tracking** : Unity offre une prise en charge native du hand tracking via son plugin XR Interaction Toolkit, ce qui facilite le développement d'expériences de suivi des mains
2. **Optimisation pour le hand tracking** : Le plugin Ultraleap d'Unity est optimisé pour la création d'expériences de hand tracking, offrant un moyen rapide de commencer à développer pour les mains

<https://developer.leapmotion.com/unity>

3. **Facilité de développement** : Unity simplifie le processus de développement en offrant des outils tels que le plugin MediaPipe, qui permet d'incorporer la reconnaissance de gestes et les technologies de suivi des mains pour permettre une interaction utilisateur naturelle  
<https://www.linkedin.com/pulse/exploring-immersion-developing-application-unity-mediapipe-felix/>
4. **Compatibilité multiplateforme** : Unity prend en charge le hand tracking via OpenXR, ce qui permet de développer des applications compatibles avec une variété de plateformes, y compris les casques VR autonomes tels que Meta Quest 2, Pico4, Pico business, ...
5. **Un store contenant multiple assets**
6. **Une très grande communauté** (environ 2 millions) : L'envergure de cette communauté permet d'obtenir un nombre élevé de tutoriel et autres accès à l'information.

En résumé, Unity offre une prise en charge native, des outils optimisés, une facilité de développement et une compatibilité multiplateforme pour le hand tracking, ce qui en fait un choix attrayant pour le développement d'applications AR/VR avec cette fonctionnalité.

# **1 – Quel software, présent sur le marché, est le plus adapté à notre projet**

## **b – Unreal Engine**

Les avantages d'utiliser Unreal Engine pour une application AR/VR de hand tracking sont les suivants :

1. **Prise en charge native du hand tracking** : Unreal Engine offre une prise en charge native du hand tracking à travers son plugin OpenXR, permettant de suivre et d'utiliser les mains et les doigts des utilisateurs comme entrée, accessible à la fois dans Blueprints et C++  
  
<https://learn.microsoft.com/en-us/windows/mixed-reality/develop/unreal/unreal-hand-tracking?tabs=426>
2. **Données détaillées sur la position et la rotation des mains** : Le système de hand tracking d'Unreal Engine fournit des données détaillées sur la position et la rotation de chaque doigt, de la paume entière et des gestes de la main, ce qui permet de créer des expériences de suivi des mains précises
3. **Unity détient plus de projet similaire au notre** : Difficulté de trouver des projets de traduction de langues de signes sur Unreal Engine .
4. **Un store contenant multiple assets**

En résumé, Unreal Engine offre une prise en charge native, des données détaillées sur la position et la rotation des mains, ainsi qu'une compatibilité multiplateforme pour le hand tracking, ce qui en fait un choix attrayant pour le développement d'applications AR/VR avec cette fonctionnalité.

## **1 – Quel software, présent sur le marché, est le plus adapté à notre projet**

### **c – Conclusion**

Les deux softwares précédemment cités ont chacun leurs avantages et leurs inconvénients.

Cependant Unreal Engine se focalise plus sur la réalisation d'applications ou de jeux nécessitant des graphismes photoréalistes.

Pour notre application, les réalismes sont secondaires.

Unity étant en C#, le langage est plus accessible surtout pour les néophytes.

En revanche Unreal Engine, étant en C++, est considéré comme plus « difficile » à maîtriser.

Les deux moteurs détiennent chacun un store détenant tous les éléments qui nous seront nécessaires à la réalisation de notre application.

Pour finir, plusieurs projets de traduction en temps réelle ont déjà été réalisés sur Unity, « EchoSign », « Prototype ASL Sign Language Interpreter », ...

## 2 – Matériels nécessaires

### Pico 4 / Oculus quest 2 (ou 3)

Pico offre en ligne son système « d'hand tracking » pour convaincre les développeurs d'utiliser leurs casques pour leur application.

<https://developer-global.pico-interactive.com/document/unity/hand-tracking/>

<https://developer-global.pico-interactive.com/document/unreal/hand-tracking/>

Oculus offre, lui aussi, la possibilité aux développeurs d'utiliser leur « hand tracking » dans leur projet Unity et Unreal.

Cependant, malgré que leur détection de position de mains soit réputé meilleure que celle de Pico, leur documentation est clairement moins précise et plus difficile à manier :

<https://developer.oculus.com/documentation/unity/unity-handtracking/>

**3 – Assets pour l’obtention de mains modélisées**

## 4 – Projets déjà existant

### - Sign Echo : (Gagnant Pico Dev Jam 2023)

*Sign Echo is an app for the PICO 4 VR headset that enables real-time communication between deaf and hearing users. It translates sign language into spoken words and spoken words into text. Deaf users can sign, and SignEcho will voice those signs out loud. Conversely, it will display spoken language as text in the VR space, making the conversation accessible and inclusive.*

<https://devpost.com/software/signecho>

Made with Unity !

- LeapASL : A platform for design and implementation of real time algorithms for translation of American Sign Language using personal supervised machine learning models

[https://www.youtube.com/watch?v=VW6gtI\\_q0Ls](https://www.youtube.com/watch?v=VW6gtI_q0Ls) (prototype)

<https://www.sciencedirect.com/science/article/pii/S2665963822000434>

Made with Unity !