



LOG8704 – Développement logiciel en réalité étendue

Automne 2024

TP No. 1 - Documentation Technique

2050481 – Samuel Laporte

2073093 - Elizabeth Michaud

2025011 - Gabrielle Côté

2083465 - Alexandre Gélinas

Soumis à : Maïa-Soileil Chartray

24 Septembre 2024

Introduction

L'application *Pokemon Shower* permet aux utilisateurs de découvrir leurs Pokémon préférés en réalité augmentée. Grâce à des marqueurs basés sur les cartes Pokémon, l'application détecte ces cartes et affiche le Pokémon en superposition sur l'écran du téléphone.

De plus, même sans posséder de cartes, les utilisateurs peuvent faire apparaître leurs Pokémon dans leur environnement. Ils ont la possibilité d'ajuster la position, la rotation et la taille des personnages pour les intégrer parfaitement à leur décor.

Choix techniques

Nous utilisons Unity 6 pour le développement de notre application. Ce choix nous offre de nombreuses technologies qui facilitent grandement la création d'une application mobile de réalité augmentée. Plusieurs concepts ont été intégrés pour assurer le bon fonctionnement de l'application, notamment :

- **AR Plane Manager** : Permet de comprendre l'environnement et de déterminer où placer les éléments virtuels.
- **AR Occlusion Manager** : Masque les éléments obstrués par des objets physiques réels.
- **AR Ray Interactor** : Facilite les interactions de l'utilisateur sur l'écran avec le monde virtuel de la scène de rendu. Utile pour positionner et déplacer les Pokémon.
- **XR Grab Interactable** : Permet d'interagir avec les Pokémon. Dans notre cas, cela inclut la possibilité de les déplacer, de les faire tourner et de les agrandir. Cette composante est intégrée à tous nos prefabs de Pokémon.
- **AR Tracked Image Manager** : Permet d'ajouter à la scène AR un *prefab* d'objet lorsque l'image qui lui est associée est détectée par l'application. Dans notre cas, cela nous permet de faire apparaître un Pikachu lorsque sa carte de jeu est reconnue.

Tests effectués

Nous avons testé l'application sur plusieurs appareils Android, notamment le Samsung S10+, le Samsung Galaxy A54, le Samsung S24 et le Samsung Galaxy A51. Cela couvre une large gamme de modèles, allant des plus anciens aux plus récents.

Les tests ont été réalisés dans un environnement intérieur, éclairé par une lumière standard. Il était essentiel que le sol présente des motifs visuels variés, car un sol uniformément uni complique la détection et le suivi de la surface par le AR Plane Manager.

Nous avons également identifié un bogue connu concernant l'interaction avec les objets instanciés à partir de marqueurs. Bien que nous puissions déplacer ou agrandir ces objets, une

fois que nous relâchons notre doigt, l'objet retourne automatiquement à la position du marqueur. Notre objectif est de permettre la création d'objets à partir du marqueur, puis de les manipuler librement avec les interactions supportées.

Pistes d'amélioration

Nous avons eu beaucoup de problèmes à utiliser les Input des touches pour instancier un objet. L'API pour ce faire n'est pas très intuitive et cela nous a causé beaucoup de difficultés.

Nous avons rencontré un autre blocage concernant l'utilisation du Depth API. Pour une raison encore inconnue, nous ne sommes pas parvenus à récupérer le *framebuffer* de profondeur, ce qui limite notre capacité à obtenir des informations détaillées sur l'environnement. Nous avons effectué plusieurs tests pour vérifier que nos appareils prennent bien en charge le Depth API, ce qui a été confirmé. Cependant, nous ne comprenons pas pourquoi la fonction de l'Occlusion Manager, *TryAcquireEnvironmentDepthCpuImage*, ne nous retourne rien. Nous souhaitons afficher la distance entre un objet sélectionné et l'utilisateur, mais ce problème nous en a empêché.

L'application que nous avons développée n'est qu'un prototype destiné à tester les capacités de l'API pour la réalité augmentée sur mobile. Une fonctionnalité particulièrement intéressante serait de pouvoir jouer au jeu de cartes Pokémon en réalité augmentée, permettant de visualiser les interactions (attaques, défenses, etc.) entre les Pokémon. Cela donnerait vie à ce jeu de société et rendrait l'expérience encore plus amusante.

Marqueur

Voici le marqueur utilisé dans notre application.



Image 1: Carte de Pikachu

Lien

GitHub: <https://github.com/LEX0RE/LOG8704>

Vidéo démonstrative:

https://drive.google.com/file/d/1H1ACmYKPofMvzX_PhBvN3sKNMXFAxamZ/view?usp=sharing

Sources images

Pikachu Icon : <https://www.pokemon.com/fr/pokedex/pikachu>

Charmander Icon : <https://www.pokemon.com/us/pokedex/charmander>

Bulbasaur Icon : <https://www.pokemon.com/us/pokedex/bulbasaur>

Squirtle Icon : <https://www.pokemon.com/us/pokedex/squirtle>

Pokeball Icon : https://www.freeiconspng.com/img/45334#google_vignette

Pikachu Card: <https://www.millenium.org/news/380835.html>

3D Model: <https://www.models-resource.com/3ds/pokemonxy>