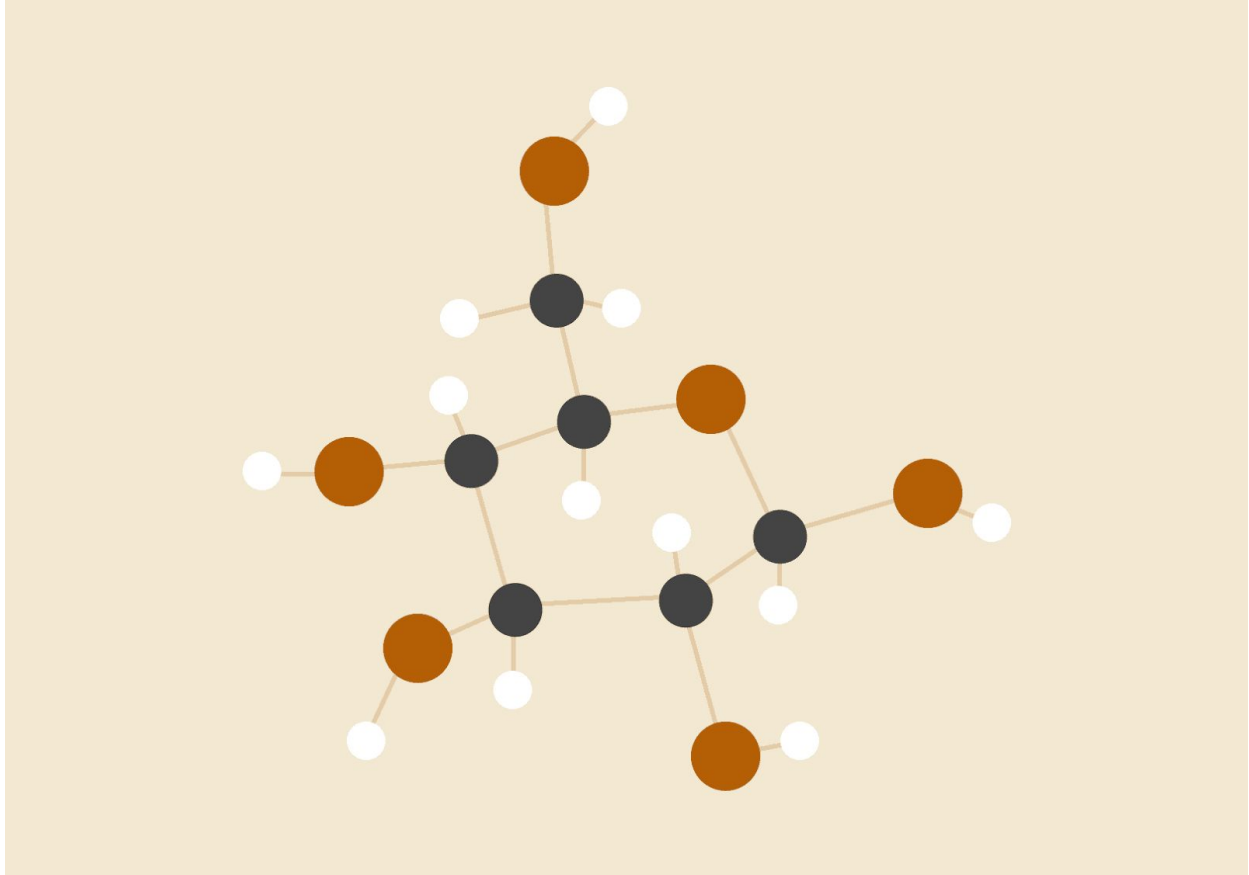


ARCore & Unity

UseInf 2018-2019



Jonathan Michel, Nelson Rocha

26.05.2019

HES-SO Master Lausanne

Art Gallery Project

Objectifs

L'objectif de ce travail pratique est de se familiariser avec des outils permettant de réaliser des applications de réalité augmentée. Ceci s'inscrit dans le contexte du cours d'interfaces utilisateurs avancées (*UseInf*), dispensé dans la formation de Master à la HES-SO.

Certaines contraintes nous ont toutefois été imposées. Il s'agit des outils à utiliser et du temps à disposition pour réaliser notre projet.

Il s'agit donc d'une application de réalité augmentée qui doit fonctionner sur un appareil mobile, et le temps mis à disposition est de deux semaines.

Concept

Notre idée de projet est de faire une application permettant d'accrocher des tableaux virtuels dans des parois réelles. Le but est de simuler une galerie d'art.

L'utilisateur de l'application va pouvoir scanner un objet précis, pré-configuré, qui va permettre à l'appareil mobile de savoir où il se situe. De cette manière, on pourra savoir où sont les parois où l'on peut accrocher des tableaux.

Une fois cette première étape franchie, l'utilisateur pourra alors choisir l'emplacement de son choix sur une des surfaces, et y accrocher virtuellement l'oeuvre de son choix.

C'est cette partie-là qui fait appel à la notion de réalité augmentée. Les différents tableaux disponibles sont pré-enregistrés dans l'appareil mobile, et l'utilisateur, après avoir choisi l'endroit qui lui convient, effectue son choix parmi une liste les recensant.

De cette manière, il pourra facilement simuler une galerie d'art, où simplement enrichir les murs d'une pièce de son domicile d'oeuvres qui ne pourrait pas s'y trouver par d'autres moyens.

Finalement, pour que celui-ci n'ait pas à recommencer l'opération de placement de tableaux à chaque fois qu'il souhaite les voir, l'application sauvegardera son travail sur l'appareil mobile.

De cette manière, il sera libre de s'y replonger ultérieurement simplement en scannant l'objet qui identifie la pièce.

Outils

Pour la réalisation de notre projet, nous avons les contraintes matérielles et logicielles suivantes:

- L'application doit s'exécuter sur un mobile Android
- Celle-ci doit faire usage de la librairie ARCore
- Le développement à proprement parler se fait sur Unity et Visual Studio

ARCore est un kit de développement logiciel développé par Google, spécifiquement dédié aux applications de réalité augmentée. C'est un produit très jeune (sa première release date du 1er Mars 2018), mais compte tenu de la taille de la communauté de développeurs Android, il est très utilisé.

Quand à Unity, c'est l'environnement de développement de jeux amateurs par excellence. Il jouit d'une forte popularité, après son lancement en 2005, grâce à son but initial: démocratiser le développement de jeux par des privés. En effet, avant Unity, cette tâche était extrêmement complexe et coûteuse. En offrant au grand public un moteur de jeu avec une interface de contrôle, et en s'alliant à Microsoft pour proposer un environnement de développement complet et intégré, Unity a réussi son pari.

A noter qu'il permet de déployer le fruit du développement sur plus d'une vingtaine de plateformes, à l'heure actuelle.

Tous ces produits étant libres d'utilisation, ou proposant une version gratuite, cela en fait les candidats idéaux pour notre expérience

Implémentation

L'implémentation s'est révélée ardue. Afin de proposer à l'utilisateur de pouvoir déplacer, redimensionner et faire tourner l'oeuvre qu'il ajoute, nous avons récupéré les fichiers de manipulation et de reconnaissance de gestes fournis dans les codes exemples. Le fonctionnement est le suivant :

Le script *Manipulator* reçoit des callbacks du *ManipulationSystem* lorsque les différents gestes de l'utilisateur sont détectés (Tap, drag, pinch, twist, ...) . En fonction des gestes, le script décide ce qu'il faut faire. Le script *PaintingPlacementManipulator* hérite de ce-dernier et est destiné à gérer les peintures. Il place une oeuvre (*PaintingPrefab*) lorsque l'utilisateur touche l'écran sur une zone qui ne contient pas déjà un objet.

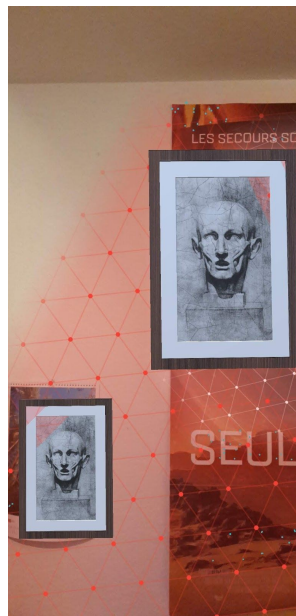
Le *PaintingPrefab* est instancié comme enfant du *ManipulatorPrefab*, il contient plusieurs scripts de manipulation (implémentant la classe de base *Manipulator*) pour la sélection,

la translation, la rotation, la mise à l'échelle, le déplacement en hauteur.

Afin d'aider l'utilisateur à savoir quand il peut déposer une oeuvre, nous avons intégré un guide *PlaneDiscovery* qui affiche des messages quand l'AR Core n'arrive pas à se repérer et projette une grille sur les surfaces qu'il a détecté. Bien que moyennement agréable à l'oeil, celle-ci est fondamentale pour savoir si le mur a bien été détecté.

Concernant la peinture nous nous sommes servis d'un assets existant et gratuit dans le store de unity.

Le résultat obtenu est le suivant :



Exemple d'intégration de tableaux

Pour ce qui est de la reconnaissance de salle, nous pensions nous servir de QR codes. Il semblerait que la fonctionnalité ne soit pas supporté de base dans ARCore. Il est par contre possible de reconnaître un motif précis/une image. Cependant, par manque de temps nous n'avons pas pu de coder cette partie. L'utilisateur ne peut donc pas sauver les peintures qu'il ajoute pour les reconsulter dans un second temps.

Pendant l'implémentation, nous avons régulièrement eu des soucis avec Unity qui ne reconnaissait plus les scripts des différents objets. Pour résoudre ces problèmes, il fallait réimporter les scripts voire les recréer et à chaque fois les reconfigurer. Il semblerait qu'il s'agisse d'un problème connu dans Unity.

Conclusion

- La reconnaissance sur les murs est difficile étant donné que les textures sont uniformes. Pour de meilleurs résultats il faut se servir de murs ou sont accolés des posters par exemple.
- N'ayant tous les deux jamais ni de Unity ni d'AR Core, il a été chronophage et difficile de prendre en mains les différents outil. Heureusement la documentation est bien fournie et les forums d'aide bien remplis.
- En conclusion, nous avions peu de temps à disposition compte tenu de l'ampleur du projet. Ces quelques semaines nous ont permis de prendre en main les outils et d'entrevoir le potentiel de l'AR.