

Le visage dans toute sa splendeur (LVDTSS)

Introduction

LVDTSS a été créé dans le cadre d'un projet universitaire pour mettre en application et approfondir nos connaissances en réalité augmentée. Au cours du projet, nous avons dû faire des concessions dues au matériel à notre disposition. Cependant, nous avons également fait des découvertes qui nous ont permis d'aller plus loin que ce que nous imaginions.

Description

À l'heure actuelle, cette application est capable de positionner des accessoires, bien positionnés sur le visage d'une personne, comme :

- Des chapeaux
- Des lunettes
- Des boucles d'oreilles

En passant sur la caméra arrière du téléphone vous pourrez vous servir d'un catalogue d'accessoires préconçus afin de faire défiler le menu sur l'accessoire voulu. Pour sélectionner l'objet, il y a deux choix :

1. Sélectionner l'objet dans le menu situé en bas de l'écran. Comme la sélection avec la caméra frontale.
2. Toucher l'objet qui apparaît en 3D dans votre écran. Il se mettra à faire une rotation puis viendra se positionner sur vous.

Les objets sont cumulables. Cela peut rendre la visualisation absurde ou merveilleuse, à vous de choisir.

Travail réalisé

La montée en compétence

Afin de réaliser le projet, il était nécessaire de monter en compétence que ce soit dans le logiciel Unity et son fonctionnement de base ou encore avec les différents outils nous permettant de faire de la réalité augmentée. Effectuant mon alternance au centre de réalité virtuelle du groupe Stellantis (anciennement PSA) j'avais déjà quelques notions en réalité virtuelle et dans l'utilisation de Unity mais il me manquait beaucoup d'élément pour la réalité augmentée. Ainsi, j'ai suivi les tutoriels suivant :

1. <https://learn.unity.com/tutorial/tracking-faces-inar?uv=2020.2&projectId=5fc7bd28edb2a001fc8b017> : J'ai fait ce tutoriel en compagnie de mon groupe de projet pour comprendre le fonctionnement du tracking de la tête avec ARFoundation et ARFaceFoundation.

2. <https://www.youtube.com/watch?v=RgomLumqwCk> : Ce tutoriel a été suivi en compagnie de Nicolas Bourneuf, nous avons pu apprendre à gérer les canvas pour pouvoir faire par la suite l'interface graphique du choix des accessoires à tester.
3. <https://youtu.be/I9j3MD7gS5Y> : Dernier tutoriel qui nous a permis d'apprendre à gérer le tracking d'image.

L'application

Elle a été principalement réalisée en groupe. Le tracking du visage de l'utilisateur a été la première étape puisqu'il est indispensable à notre application. Nous nous sommes inspirés de ce que nous avons appris avec le premier tutoriel avec l'ARFoundation et le face tracking. Nous avons donc commencé par travailler avec des masques comme le tutoriel avant de nous concentrer sur des accessoires tels que des chapeaux, des boucles d'oreilles ou encore des lunettes. Cette dernière partie a été effectuée par Chloé Xintray et Tony Lim qui ont travaillé en pair programming, comme Nicolas Bourneuf et moi. Ils ont donc utilisé l'ARFace pour pouvoir positionner au mieux les accessoires sur le visage, le nez, les oreilles ou la tête. Cette méthode était intéressante mais pas 100% efficace, en effet, les accessoires correspondaient avec un type de visage mais pas la majorité d'entre eux. Nous avons donc ajouté un " Mesh Filter " pour corriger ce problème. Ainsi, les accessoires se sont mieux adaptés à la morphologie de chacun, ils étaient également mieux orientés et dimensionnés en fonction de la position du visage.

Une fois la base bien ficelée, Tony et Nicolas ont ajouté de l'occlusion sur les accessoires. En effet, ils se positionnaient certes correctement sur les personnes mais si on levait la tête, un chapeau, par exemple, s'orientait en conséquence mais sans donner l'impression qu'il était autour de notre tête, c'est-à-dire que l'on voyait le fond du chapeau sur notre front. Or ça retire tout le sentiment de réalisme de l'outil, c'est pourquoi il nous fallait le corriger.

Pour terminer, les prefabs que nous avons utilisés pour représenter les accessoires n'étaient pas réalistes. En effet, ce sont des objets 3D et non pas des images 3D scannées. Elles ne sont donc pas très jolies et font un peu fausses. Nicolas a donc manipulé les textures, les matériaux et surtout les lumières pour pouvoir donner un peu plus l'illusion que nous les portons pour de vrais.

Le menu interactif

J'ai réalisé ce menu en pair programming avec Nicolas Bourneuf. Ce menu est celui permettant de faire défiler les différents accessoires à tester et de les sélectionner. Voici les différentes étapes de développement de ce menu :

1. Création d'un Canvas : C'est un objet qui nous permet de tracer, dans Unity, un cadre correspondant à celui du téléphone. Il permet également d'y fixer les différents éléments dont nous allons avoir besoin (i.e. les boutons de défilements et de sélections).
2. Nous avons besoin de trois boutons. Un bouton central, pour sélectionner ou désélectionner l'accessoire à essayer (par exemple : on clique sur les lunettes pour qu'elles apparaissent sur notre visage et on clique à nouveau dessus pour qu'elles disparaissent). Ce bouton change de couleur quand un accessoire est sélectionné : il devient vert le cas échéant. Nous avons développé cet élément pour que

l'utilisateur sache à tout moment s'il va sélectionner ou sélectionner l'accessoire. Et deux autres boutons, une flèche à droite et une flèche à gauche pour faire défiler les différents accessoires disponibles.

3. Le script SwitchAccessoire : ça a été une des étapes les plus importantes. Ce script permet les interactions avec les trois boutons décrits au-dessus. Cela va nous permettre de positionner un accessoire, le supprimer ou les changer sur l'ARFace en fonction de l'état du bouton.

Le menu interactif en caméra arrière

L'objectif de ce menu était de scanner des QR Code représentant des accessoires et de pouvoir les essayer comme avec la caméra avant. Par exemple : nous avons un QR Code qui représente un boucle d'oreille. Nous le scannons avec la caméra arrière du téléphone. Elle apparaît alors à l'écran et dans le menu interactif au pied de l'écran et deux choix s'offrent alors à nous :

1. Cliquer dans le menu sur l'accessoire pour pouvoir le porter.
2. Cliquer sur l'accessoire à l'écran, ce qui le fera tourner sur lui-même pour pouvoir l'observer sous tous ses angles.

Quel que soit le choix de l'utilisateur, la caméra repassera automatiquement en mode frontale pour pouvoir essayer ou enlever l'accessoire.

Cette partie a également été réalisée en pair programming avec Nicolas Bourneuf. Nous avons tout d'abord, basé notre outil sur l'ImageTracking de l'ARFoundation et OpenCV, mais le problème est que ce dernier bloquait l'utilisation normale de la caméra. Ensuite, nous avons fait une tentative avec le "facetracking" mais le problème était que selon l'outil que nous utilisions (iOS ou Android), nous n'avions pas le même résultat et c'était un souci. Nous avons donc finalement opté pour un modules déjà existant dans la librairies ARFoundation pour la reconnaissance d'image. Le script qui gère ces interactions est "followImage" (il se situe dans le composant d'objet ARSession).

Pour pouvoir utiliser les deux technologies (face tracking et image tracking), nous avons divisé le code en deux partie :

1. La première gère la caméra frontale et le face tracking.
2. La seconde gère la caméra arrière et l'image tracking.

Nous avons également ajouté un bouton en haut à droite de l'écran pour pouvoir choisir quelle caméra on veut utiliser (script associé : "switchCamera", il se situe au même endroit que le script "followImage").

Les problèmes rencontrés et leurs solutions

L'occlusion des accessoires :

Nous l'avons vu plus haut, au départ nous avons eu des soucis avec les accessoires qui ne se "cachaient" pas derrière la tête des utilisateurs (pour les chapeaux ou les branches des lunettes par exemple).

Comme solution, nous avons opté pour l'ajout de matériaux d'occlusions permettant de masquer les parties que nous ne devons pas voir. Nicolas a même modifié certains prefabs en enlevant certaines parties des accessoires.

L'ARFace qui ne charge pas tout correctement :

A certain moment, l'ARFace ne chargeait pas correctement tous les accessoires, notamment ceux qui avaient été ajoutés et supprimés.

Nous avons donc, pour corriger ce problème, réinstancié l'ARSessionOrigin à chaque ajout ou suppression d'accessoire.

Les accessoires scannés qui restent sur l'écran même après le passage à la caméra frontale :

Nous avons rencontré quelques problèmes avec le scan des accessoires puisqu'au départ, les objets restaient sur l'écran (en plein milieu et pas du tout sur le visage) au moment du retour à la caméra frontale.

Nous avons donc décidé de désactiver tous les objets instanciés pour l'image tracking dès que la caméra passe de arrière à frontale.

Les évolutions envisagées

1. Pouvoir positionner un accessoires comme on le souhaite en le déplaçant avec les doigts sur l'écran tactile.
2. Faire des catégories d'accessoires : plutôt que de faire défiler tous les accessoires mélangés, nous voudrions faire des catégories que l'utilisateur sélectionne avec de choisir l'accessoire qu'il veut essayer (par exemple : la catégories des chapeaux ou des boucles d'oreilles).
3. Améliorer l'occlusion des accessoires.

Les sources du projets

- <https://free3d.com/fr/3d-models/>
- <https://learn.unity.com/tutorial/tracking-faces-in-ar?uv=2020.2>
- <https://www.youtube.com/watch?v=klcvAi60qII&t=1768s>
- <https://www.youtube.com/watch?v=RgomLumqwCk>
- <https://youtu.be/JQEovMKq2U0>
- <https://medium.com/@harivigneshjayapalan/arcore-cupcakes-5-augmented-faces-api-2ca78681228e>
- <https://youtu.be/6seuSOUf3OU>
- <https://support.wikitudo.com/support/discussions/topics/5000082243>
- <https://www.youtube.com/watch?v=IXvt66A0i3Q>
- <https://youtu.be/I9j3MD7gS5Y>
- <https://www.turbosquid.com/3d-models/>
- <https://developers.google.com/>