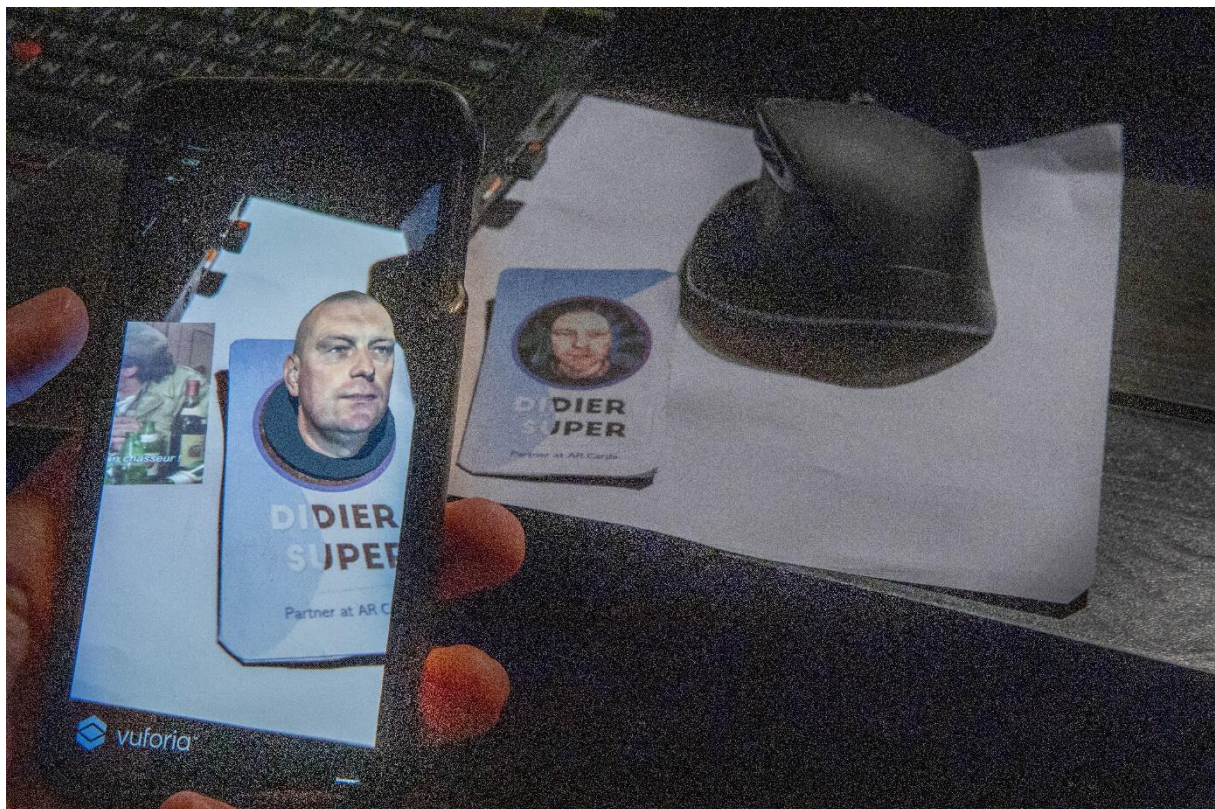


# RAPPORT DE PROJET : LES CARTE DE VISITE AUGMENTÉES

Par Daniel Ikka, Dan Symchowicz, Victor Aymard



## Table des matières

Introduction .....	2
Phase 1 : A la recherche d'un sujet.....	2
Phase 2 : L'idéation .....	2
Phase 3 : la réalisation .....	2
L'incrustation vidéo.....	3
Modélisation 3D.....	4
La gestion des interactions .....	5
Les virtuels buttons .....	5
La gestion pause des vidéos .....	5
Les boutons « physiques » .....	5
Rendu final :.....	6
Axes d'approfondissement :.....	7
Bilan .....	7
Pour aller plus loin.....	7

## Introduction

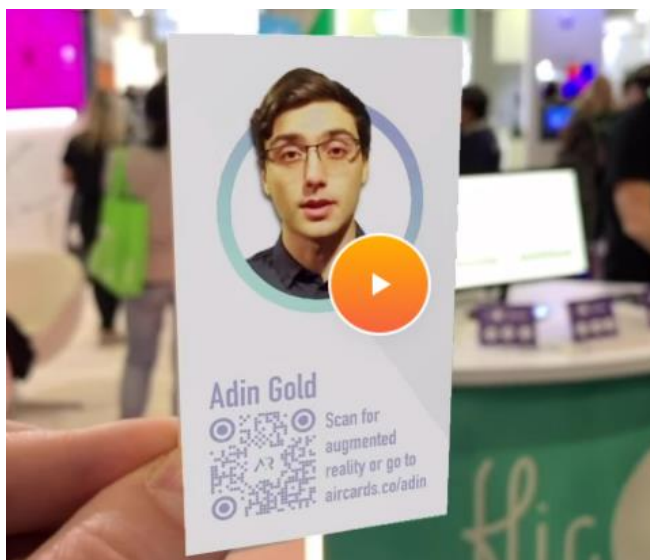
Ce projet a eu pour but de nous faire prendre en main l'application Unity avec le plugin Vuforia, afin de créer une application de réalité augmentée (RA), mêlant réel et virtuel, sur une application mobile.

## Phase 1 : A la recherche d'un sujet

Il y a peut-être deux ans, Daniel avait vu passer sur linkedIn, une courte vidéo présentant des cartes de visite en réalité augmentée. Celle-ci l'avait vraiment marqué, à tel point que lorsque nous avons dû trouver un sujet, cette idée m'est immédiatement venue en tête. Nous avons donc commencé à nous documenter sur les fonctionnalités que nous souhaiterions avoir et durant nos recherches, nous avons trouvé le site d'Air Cards. Une entreprise proposant des cartes de visite en réalité augmentée sans plateforme. <https://www.aircards.co/augmented-reality-business-cards>

## Phase 2 : L'idéation

Nous nous sommes grandement inspirés de leurs idées, car le but de ce projet était avant tout de nous familiariser avec le logiciel Unity et son plugin Vuforia. Ainsi, voici nos deux premiers modèles : Ar living card à gauche, et l'holographic card à droite.



Cependant, comme nous étions 3, il nous fallait une troisième idée. Les deux précédentes comportant principalement de la 2D, nous avons décidé d'intégrer de la 3D dans cette solution. L'idée de modéliser notre visage en 3D pour ensuite l'intégrer nous a paru judicieuse et pertinente.

## Phase 3 : la réalisation

Passé les tp 1 et 2, la gestion de l'image target et de l'exportation sur smartphone nous était devenue élémentaire. Mais deux défis s'offraient à nous :

- Comment réaliser et intégrer une vidéo transparente
- Comment réaliser et intégrer un scan de visage 3D



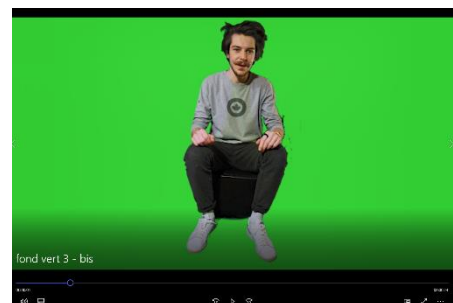
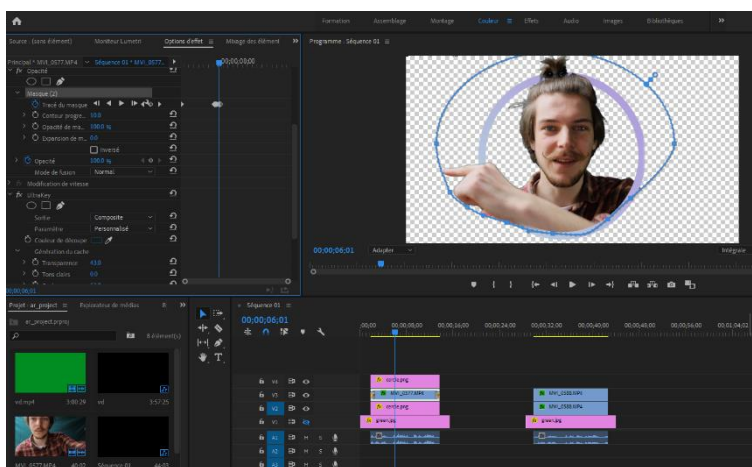
## L'incrustation vidéo

Pour la réalisation de l'AR living card, il nous fallait donc une photo, ainsi qu'une vidéo toutes deux détourées. Heureusement pour nous, Victor a quelques notions en photo et vidéo. Il s'est donc débrouillé comme il a pu pour faire un fond vert (en l'occurrence, avec un drap). Il a ensuite fallu éclairer la scène de tel sorte que le moins d'ombres possible apparaissent sur le drap (pour que le détourage en post production puisse être bien réussi).



Une fois la photo prise et la vidéo tournée, il a fallu les intégrer sur une carte de visite. Nous avons repris certaines idées du design de l'AR living card. Quelques dizaines de minutes plus tard, nous avons notre première carte.

Pour ce qui est du détourage vidéo, il a été fait sous Adobe Premiere Pro (le logiciel de vidéo de la suite Adobe) avec l'effet ultrakey ainsi qu'un masque de détourage (pour permettre une bonne intégration avec le cercle). Cependant, le passage sur Unity fut plus compliqué que prévu. En effet, après quelques recherches, nous nous sommes rendu compte que Unity n'acceptait que les vidéos transparentes qu'au format .mov. Ce format étant propriétaire d'Apple, il a été impossible de l'intégrer sous Windows. Cependant Unity permet également de gérer la clé de chrominance. De retour sous Premiere, il a fallu rajouter une image verte avant de ré-exporter la vidéo vers Unity. Cette fois-ci, il a été possible d'obtenir un résultat relativement satisfaisant.

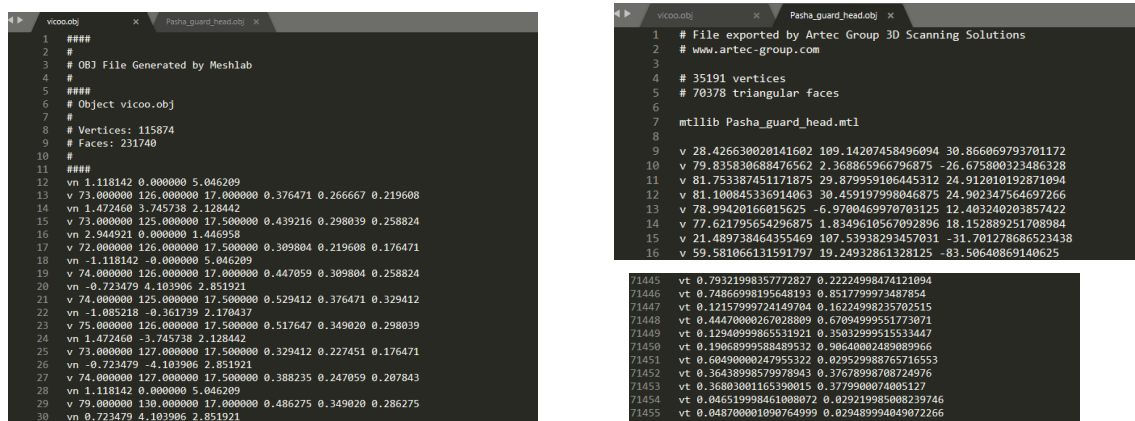


En ce qui concerne la vidéo holographique, le même principe a été utilisé. Il a seulement fallu rajouter des éléments 3D, comme la plateforme créée sur Unity, ou le tabouret importé.

## Modélisation 3D

Pour la dernière carte, nous avons donc essayé d'intégrer un buste, ou une tête de l'un de nous en 3D. Pour ce faire, nous avons d'abord eu recours au site web suivant, qui permet d'obtenir des .obj avec un rendu des couleurs, à partir d'une image 2D <https://cvl-demos.cs.nott.ac.uk/>

Cependant, lorsque nous avons essayé de l'importer sur Unity, il était impossible d'accéder au mesh. Après de nombreuses heures de recherches sur les forums, nous avons réalisé avec regret que notre fichier .obj ne comportait effectivement pas les données de texture. Le nôtre, celui de gauche, comporte des sommets (v) et de normales (vn) mais pas de texture (vt), alors que l'on retrouve bien les couleurs sur MeshLab. Nous n'avons malheureusement pas réussi à résoudre ce problème. Sur le projet ouvert à droite (trouvé sur le net pour comparer), on retrouve bien des informations de textures.



Nous avons donc essayé d'autres solutions, comme des applications de scan 3D par la caméra frontale d'un Iphone X – celui-ci possédant un capteur infrarouge pouvant scanner en 3D pour l'utilisation de Faceld. Après quelques essais sur différentes applications, nous avons été confronté à un problème similaire : Il nous était impossible d'insérer nos fichiers sur Unity avec de la texture, même avec l'utilisation de .mlt couplée à celle d'un .obj. Nous nous sommes résolus à prendre un modèle de tête en 3D disponible sur internet. La déception était grande, mais nous avons une solution.



FIGURE 3 RENDU VIA IMAGE 2D



FIGURE 2 RENDU VIA SCANDY PRO



FIGURE 1 RENDU VIA CAPTURE

## La gestion des interactions

Pour rendre nos cartes de visite en réalité augmentée plus intéressantes, nous avons pensé à rendre les interactions possibles en utilisant différents procédés.

### Les virtuals buttons

La première idée renforce le côté « waouh » de l'application, était l'intégration de virtuals buttons. Ceux-ci ont pu être possibles par un design réfléchi, permettant de créer des zones de points qui ont pu être utilisés comme marqueurs (les trackings points). Il a ensuite été possible de créer des boutons virtuels exécutant une action spécifique définie par un script en C#.



### La gestion pause des vidéos

Il nous a également fallu nous occuper des vidéos qui se lançaient lors du déclenchement du bouton play. Là encore, nous sommes passés par un script en C#.

### Les boutons « physiques »

Dans l'idée de vouloir améliorer l'expérience utilisateur, nous avons également pensé à faire une dernière carte pour pouvoir ajouter des boutons cliquables directement sur le téléphone. Ceci, contrairement aux boutons virtuels, nous a permis de les placer en dehors de la carte. Cette solution nous a également semblée plus intuitive pour l'utilisateur. De plus il n'y a besoin de se soucier de la qualité de l'image et des tracking points.





## Rendu final :

Par ce projet, nous avons essayé d'explorer un grand nombre d'options offert par Unity et Vuforia.

Voici quelques captures d'écran issues de notre vidéo, des différentes cartes réalisées.



Voici le lien vers la vidéo de rendu :

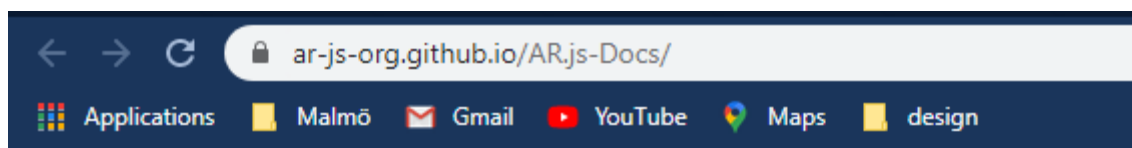
[https://drive.google.com/open?id=1jQAPRHxdxfG0fZheVfe\\_bDMF2k0bQTUJ](https://drive.google.com/open?id=1jQAPRHxdxfG0fZheVfe_bDMF2k0bQTUJ)

## Axes d'approfondissement :

Pour aller plus loin, nous aurions pu travailler les animations, pour rendre nos cartes plus immersives, par des interactions plus fluides. Nous aurions également pu nous concentrer sur une seule carte et intégrer sur celle-ci tous les différents éléments.

Pour pousser notre projet plus loin, nous aurions souhaiter exporter l'application sur un serveur web et intégrer des QR codes sur nos cartes de visite. Il aurait ainsi été possible d'accéder à l'application via un navigateur web, sans avoir besoin de télécharger une quelconque application. Tous les téléphones actuels étant équipés d'un lecteur de qrcode intégré, la fonction serait donc universelle.

Nous voulions donc exporter le projet au format WebGL, mais nous nous sommes encore une fois heurtés à un problème. Après quelques recherches sur des forums, Vuforia ne supporte tout simplement pas le format WebGL. , il n'est donc pas possible d'accéder à l'application via un navigateur simple. Il semblerait qu'une des meilleures alternatives à vuforia pour soit Ar.js, cependant, sa documentation ne nous était pas accessible.



GitHub Pages is temporarily down for maintenance.

Après d'autres recherches, nous en avons conclu que l'objectif n'était pas atteignable avec le temps qui nous était imparti.

Ainsi, le fait que l'intégration dans des navigateurs web soit complexe, on comprend mieux pourquoi la RA est si peu présente autour de nous.

## Bilan

Par ce projet, nous avons réussi à prendre en main Unity et Vuforia, et surtout de comprendre l'intérêt de la réalité augmentée, qui sera au cœur des technologies et de l'économie de demain. On a d'ailleurs pu voir qu'Apple avait mis un capteur LIDAR, dédié à la RA sur son dernier iPad pro, preuve que cette technologie arrive bel et bien et qu'elle est en train de franchir le « Chasm ».

Bien que l'outils Unity soit déjà très puissant, nous avons rencontré de nombreux problèmes avec ce logiciel. Premièrement : son instabilité, faisant crasher le logiciel de temps à autre. Le logiciel est peu intuitif. L'intégration de Github reste à améliorer, ainsi que l'UI.

## Pour aller plus loin

Voici quelques liens si vous souhaitez avoir un aperçu de ce qu'il est possible de faire :

<https://www.youtube.com/watch?v=gYGdCRMhz48>

<https://www.youtube.com/watch?v=4cCEBFDeX8A>

[https://www.youtube.com/watch?v=W48\\_2ebuvv8](https://www.youtube.com/watch?v=W48_2ebuvv8)