

LEVILAIN Rémi
Cycle Ingénieur, 2ème année
Année universitaire 2013 – 2014

Enseignant encadrant : M. Paul RICHARD
Maître de stage : M. Fabrice ROBERT

Innov'3D
71 Avenue de Rennes
49100 Angers
06 59 55 99 03
www.innov-3d.com

Développement d'applications 3D et de Réalité augmentée

Soutenu par LEVILAIN Rémi
4^{ème} année ISTIA
Spécialité Automatique et Génie Informatique



ISTIA
Ecole d'Ingénieur de l'Université d'Angers





Mai - Juillet 2014



Développement d'applications 3D
et de Réalité augmentée

Maître de stage :
M. Fabrice ROBERT

LEVILAIN Rémi

Remerciements

Je tiens tout d'abord à remercier Innov'3D de m'avoir accueilli pour mon stage de 4^{ème} année d'école d'ingénieur à l'ISTIA.

Je remercie également l'intégralité du personnel présent au sein de l'entreprise qui m'a permis de vivre une expérience professionnelle et humaine aussi passionnante qu'enrichissante et plus particulièrement mon maître de stage : M. Fabrice ROBERT et mon chef de projet : M. Amaury DEPOORTER.

Pour finir, je remercie mon enseignant tuteur M. Paul RICHARD.

Rémi LEVILAIN



Mai - Juillet 2014



Développement d'applications 3D
et de Réalité augmentée

Maître de stage :
M. Fabrice ROBERT

LEVILAIN Rémi

Sommaire

RAPPORT DE STAGE :

Développement d'applications 3D et de Réalité augmentée

REMERCIEMENTS

I.	INTRODUCTION	
1.	Présentation d'Innov'3D	1
II.	APPLICATIONS 3D ET REALITE AUGMENTEE (RA)	
1.	Présentation	2
2.	Moyens utilisés	3
III.	REALISATION D'UNE APPLICATION SUR FACEBOOK	
1.	Objectifs du projet	4
2.	Choix et installation d'un serveur privé virtuel (VPS)	4
3.	Configuration du VPS	5
4.	Les fonctionnalités de base et la programmation associées à Facebook	6
5.	Ajout d'un mini-jeu et d'un système de scores	9
6.	Difficultés rencontrées et solutions apportées	10
IV.	REALISATION D'APPLICATIONS DE REALITE AUGMENTEE	
1.	Objectifs des projets	11
2.	Les marqueurs en RA	11
3.	Premières réalisations	12
4.	Le projet BEJO	13
5.	L'export sur Android	14
6.	Difficultés rencontrées et solutions apportées	15
V.	ACTIVITES ANNEXES	
1.	Outils d'aide à la formation pour l'IFM3R	16
2.	Réalisation d'un diagramme dynamique pour le Serious Game de Chrysippe	17
VI.	CONCLUSION	19
	TABLE DES FIGURES	20
	WEBOGRAPHIE	21
	TABLE DES ANNEXES	22
	ANNEXES	



Mai - Juillet 2014



Développement d'applications 3D
et de Réalité augmentée

Maître de stage :
M. Fabrice ROBERT

LEVILAIN Rémi

I. INTRODUCTION

Dans le cadre de ma 4^{ème} année d'école d'ingénieur à l'ISTIA, j'ai effectué un stage d'une durée de 3 mois au sein d'Innov'3D. Ce rapport vous présentera tout d'abord les domaines d'applications sur lesquels j'ai travaillé. Ensuite, j'aborderai les deux principaux projets effectués, cela comprend les tâches qui m'ont été confiées, les choix que j'ai faits et les solutions techniques retenues.

1. Présentation d'Innov'3D

Innov'3D, spécialiste des solutions marketing 3D, est porté par une jeune équipe, à échelle humaine, une caractéristique lui assurant une organisation souple et dynamique ainsi qu'une ambiance de travail conviviale. Son offre : des solutions d'aide à la vente 3D et des applications en réalité virtuelle pour accompagner la force de vente en mobilité et accélérer la décision des clients.



Figure 1 : Logo d'Innov'3D

Le pôle de développement que j'ai intégré est composé de trois personnes : deux développeurs et un graphiste. Leur objectif est de répondre à la demande en termes de développement des applications clients ainsi qu'évaluer les possibilités techniques de l'entreprise.

II. APPLICATIONS 3D ET REALITE AUGMENTEE (RA)

Afin d'aborder la suite de ce rapport plus sereinement voici une brève explication à propos des applications 3D et de la réalité augmentée.

1. Présentation

- Les applications 3D en temps réel

La "3D temps réel" est une méthode de représentation de données en 3 dimensions pour laquelle chaque image composant l'animation est rendue dans l'instant qui précède son affichage. Ce procédé ne doit pas être confondu avec un système temps réel pour lequel le respect des contraintes temporelles et au moins aussi important que le résultat.

La vitesse de rendu doit permettre une animation fluide. La "3D temps réel" s'oppose à la 3D pré-calculée dans la mesure où le rendu des données 3D est immédiat. Le temps de rendu en "3D temps réel" doit être inférieur à la persistance rétinienne pour rendre l'affichage fluide, ce qui explique le besoin d'accélération matérielle à l'aide par exemple de moteurs graphiques puissants, mais également de nombreux prétraitements sur les scènes 3D.

Les formats de la "3D temps réel" comportent des données tridimensionnelles (géométrie, texture, animation) et un scénario prédéfini ou interactif (scripts).

La 3D temps réel permet ainsi un grand degré d'interactivité : possibilités d'immersion, de contrôle d'animation, d'interactivité avec la scène 3D, ce qui la met au cœur d'applications telles que la visite virtuelle, la simulation, le jeu vidéo, la visualisation et Conception Assistée par Ordinateur, le Web3D, etc...

- La réalité augmentée ou "RA"

Elle désigne les systèmes informatiques qui rendent possible la superposition d'un modèle virtuel 3D ou 2D à la perception que nous avons naturellement de la réalité, et ceci en temps réel. La réalité augmentée est donc un aspect particulier des applications 3D temps réel. Elle comprend des méthodes qui permettent d'incruster de façon réaliste des objets virtuels dans une séquence d'images le plus souvent tirées du monde réel.

Ces applications sont multiples et touchent de plus en plus de domaines, tels que les jeux vidéo, l'éducation par le jeu, les chasses au trésor virtuelles, les industries ou le médical.

Dans le cas de mon stage les applications sont conçues pour utiliser la caméra intégrée au Smartphone des utilisateurs pour scanner les éléments dits "augmentés".

2. Moyens utilisés

Durant mon stage, j'ai eu accès à différents moyens pour parvenir à réaliser l'ensemble des tâches qui m'ont été confiées. Premièrement, pour créer et mettre en place les applications j'ai utilisé le logiciel Unity.



Figure 2 : Logo d'Unity

Unity est un logiciel de création d'environnements virtuels présentant plusieurs avantages. Tout d'abord sa librairie est assez fournie et complète à la fois pour les mathématiques, la physique ainsi que pour les effets de caméra. Son système d'interface en glisser-déposer est également très intuitif. De plus, la version professionnelle dispose également d'un bon rendu pour les ombres et l'eau.

Pour créer une application, il suffit d'assembler ces différentes fonctionnalités dans l'éditeur grâce à des dépendances et des interactions. Certaines d'entre-elles sont déjà incluses dans Unity3D notamment les interactions physiques.



Figure 3 : Logo de Vuforia

Un des gros avantages d'Unity est sa communauté et ses réalisations. En effet, le module de RA "Vuforia" pour Unity est proposé gratuitement par Qualcomm. Les développeurs peuvent ainsi s'essayer à la réalité augmentée à moindre coût. Beaucoup d'autres modules dans de très nombreux domaines sont ainsi proposés par la communauté. Certains sont payants alors que d'autres sont libres. C'est grâce à ces "add-ons" d'Unity que j'ai pu réaliser différentes fonctionnalités au sein de mes applications.

La grande majorité des applications contient des éléments visuels : modèles 3D, textures, éclairages, etc... L'ensemble de ces objets a été réalisé par M. DEPOORTER, le graphiste de l'entreprise, afin que je puisse par la suite les intégrer dans les applications.

II. REALISATION D'UNE APPLICATION SUR FACEBOOK



Figure 4 : Logo
Facebook

Le principal projet en terme de temps et d'investissement sur lequel j'ai travaillé au sein d'Innov'3D est le passage d'une application réalisée sur Unity sur le réseau social Facebook. Je présenterai d'abord les objectifs initiaux de ce projet pour ensuite décrire les différentes étapes techniques nécessaires à l'aboutissement du projet. Je finirai en abordant les difficultés rencontrées les plus importantes et les solutions retenues pour y pallier.

1. Objectifs du projet

Ce projet m'a été proposé afin de répondre à différents besoins d'Innov'3D. L'utilité d'avoir une application présente sur Facebook est de toucher un public très large et de connaître différentes informations relatives à leurs habitudes. Cela permet à l'entreprise de se donner une plus grande visibilité et de montrer à ses clients sa capacité de communication au sein des réseaux sociaux.

Mes objectifs personnels sont liés à la découverte de la programmation Facebook et de la nécessité d'héberger l'application de manière sécurisée pour qu'elle puisse être validée.

2. Choix et installation d'un serveur privé virtuel (VPS)

Facebook n'autorise que les applications ayant un protocole de communication HTTPS qui correspond à du HTTP sécurisé avec un cryptage SSL, je reviendrai dessus lorsque j'aborderai la configuration du VPS.

Il faut donc héberger notre application sur un serveur bénéficiant des protocoles de communication et de sécurité adéquats. J'ai donc recherché une solution abordable, durable et modulable en fonction des besoins d'Innov'3D.



Figure 6 : Logo
d'OVH

Mon choix s'est alors orienté vers un modèle de serveur virtuel privé (VPS) relativement léger et abordable sur le site d'OVH. Il possède 10Go de stockage avec 1Go de RAM, cela permet de stocker de nombreuses applications et la RAM n'est quasiment pas exploitée pour de l'hébergement. Son coût est de 2€/mois, ce qui est insignifiant pour une entreprise. L'autre solution aurait été d'utiliser un serveur dédié pour ne pas être gêné par la mutualisation d'un

VPS mais ce choix est plus onéreux pour des avantages non requis afin de réaliser les tâches demandées.

Il faut aussi choisir le système d'exploitation fourni avec le serveur. Pour des raisons d'usage, j'ai pris une licence Linux: Debian 7 (Wheezy) qui est assez répandue et modulable en fonction des besoins que l'on peut avoir.



Figure 5 : Logo Debian

3. Configuration du VPS

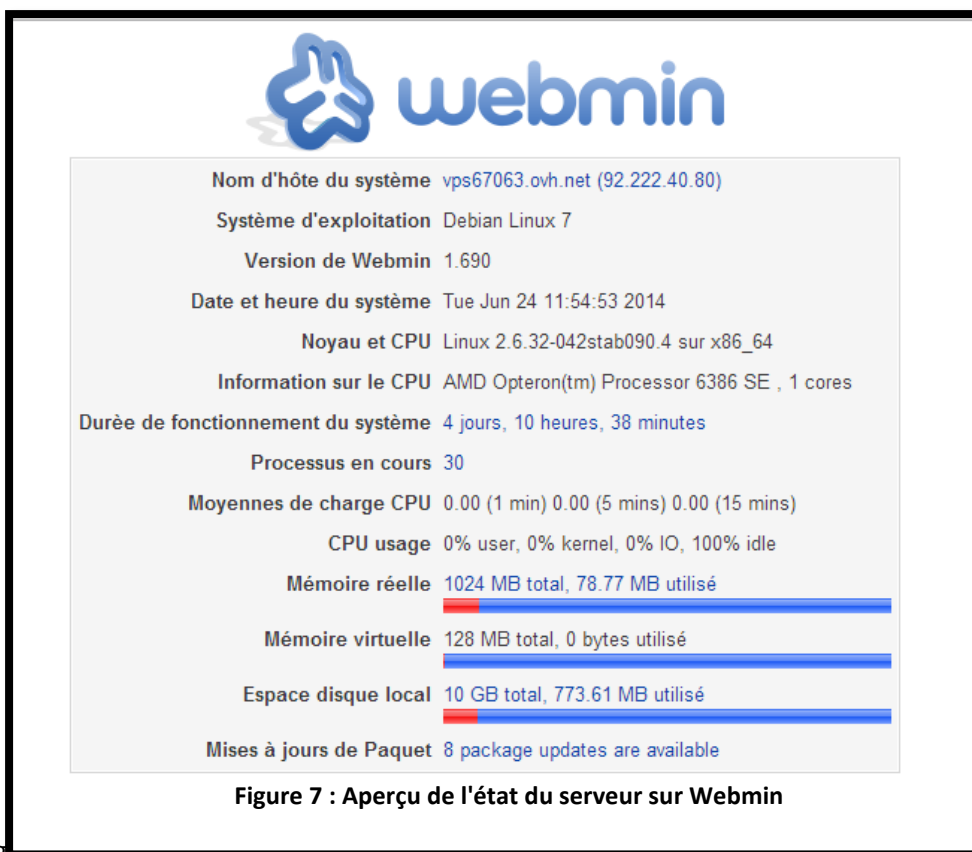
Ayant peu de notions en commandes Linux et ne souhaitant pas perdre de temps sur cet aspect technique de mon stage car il ne concerne pas la programmation d'applications. J'ai demandé à une connaissance de m'expliquer et de me montrer comment configurer le VPS afin qu'il soit sécurisé, configurable par navigateur et doté du protocole HTTPS.

J'ai tout d'abord téléchargé le logiciel Kitty permettant de se connecter à distance au serveur sur le port SSH qui correspond à un protocole de communication particulier. J'ai ensuite utilisé les commandes Linux pour installer Apache2 qui permet d'obtenir le protocole HTTP ainsi que HTTPS en activant module de cryptage SSL. Pour fonctionner correctement il faut attribuer un certificat SSL au serveur. Il a été choisi d'en prendre un auto-signé avec Ubuntu. Pour voir les différentes commandes utilisées, voir l'**Annexe n°1**.

Une fois ces protocoles mis en place, je suis passé à la configuration du pare-feu en consultant "l'iptables" sous Linux. J'ai alors récupéré un fichier contenant les règles du pare-feu que j'ai placé dans le dossier de "iptables" du serveur. Pour plus de détails à propos de ces règles, voir l'**Annexe n°2**.

Afin de pouvoir administrer le serveur via navigateur, j'ai installé Webmin. Cela permet d'avoir un aperçu de l'état du serveur et d'administrer les différents modules installés. Par exemple, pour envoyer et recevoir des mails, j'ai mis en place Postfix. Je peux ainsi via Webmin configurer les noms de domaines, les mails, l'état du serveur, etc...

Le serveur est ainsi configuré, protégé et opérationnel mais un problème d'authentification persistait : si l'utilisateur n'acceptait pas le certificat du serveur manuellement, Facebook ne chargeait pas l'application. Il a donc été nécessaire de rechercher un nouveau certificat SSL.



Le site internet StartSSL propose des certificats SSL vérifiés et gratuits. J'ai donc créé un compte et obtenu la clé SHA sur la boîte du serveur. Cette clé doit être écrite dans le fichier ssl.key et validée par le mot de passe rentré sur StartSSL. Une fois ces informations rentrées, on peut paramétrer le cryptage SSL via Webmin en indiquant la localisation des différents fichiers (ssl.crt, ssl.key).

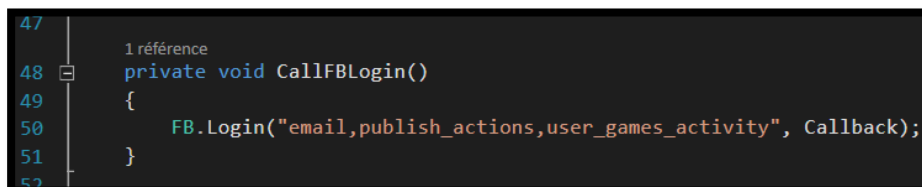
L'installation de ce nouveau certificat a permis une validation automatique lorsqu'on se connecte au serveur via son navigateur internet. Les applications installées sur le serveur sont alors lancées correctement par Facebook. L'erreur que l'on rencontrait au chargement a ainsi été corrigée.

4. Les fonctionnalités de base et la programmation associées à Facebook

Facebook propose un SDK (kit de développement logiciel) pour Unity avec une scène de démonstration afin de pouvoir se familiariser avec les différentes fonctions importantes utilisées dans les applications. Voici ces fonctions :

- **FB.Init()** : Initialise le SDK de Facebook, doit être lancé au démarrage. La fonction CallFBInit() est appelée dans la fonction Awake() d'Unity afin d'être exécutée en premier. FB.Init() contient 2 fonctions : SetInit() qui lance l'application Unity3D et qui vérifie si l'application est autorisée par l'utilisateur et OnHideUnity() qui permet juste de désactiver votre application lorsqu'elle n'est plus au premier plan de votre ordinateur, par exemple si vous changez de fenêtre ou activez le chat FB. Un exemple est donné à l'**Annexe n°3**.

- **FB.Login()** : A ne pas confondre avec une authentification, FB.Login() demande à l'utilisateur d'autoriser votre application ainsi que tous les droits dont elle a besoin pour fonctionner (liste d'amis, droits de publier, etc...). Cette fonction est indispensable.



```

47 |
48 | 1 référence
49 | private void CallFBLogin()
50 | {
51 |     FB.Login("email,publish_actions,user_games_activity", Callback);
52 | }

```

Figure 8 : Exemple de FB.Login()

- **FB.API()** : Cette fonction permet de communiquer avec les serveurs de Facebook. Elle permet de récupérer les informations relatives à l'utilisateur si vous lui avez demandé l'autorisation (cf : Login). Cette fonction est basée sur des requêtes HTML : GET, POST et DELETE. Chacune permet de récupérer, d'éditer ou de supprimer des informations.

FB.API("/me/scores", Facebook.HttpMethod.POST, null, ScoreData);

L'exemple ci-dessus permet tout simplement de poster le score qu'a obtenu l'utilisateur à votre jeu. Les paramètres sont :

- Le chemin de l'API que vous souhaitez atteindre
- La méthode http si vous souhaitez éditer, lire ou effacer
- Le callback, pas forcément utile en POST mais plus en GET si vous souhaitez vérifier le résultat
- La variable que vous enverrez

- **FB.Feed()** : Permet à l'utilisateur de poster sur son mur ou à des personnes un flux contenant ou non des médias. Nécessite l'autorisation de publication (cf : Login). Pour plus de détails sur cette fonction, consultez l'**Annexe n°3**.

Nous avons parlé précédemment de FB.API() permettant de récupérer ou d'éditer des informations. Lorsque l'on code sur Unity3D, il est difficilement possible de déboguer cette fonction car aucun lien avec Facebook n'est fait. Afin de palier à ce problème, Facebook a développé une console pour tester vos requêtes : le Graph Explorer. Dans le but d'utiliser correctement le Graph Explorer il faut d'abord comprendre ce que sont les Access Token.

- **Les Access Token** : C'est une chaîne de caractères qui correspond à la façon qu'a une application d'identifier un utilisateur. Quand celui-ci appelle la fonction FB.Login, un Access Token lui est donné en accord avec les autorisations qu'il a associées à l'application. Il faut donc toujours penser à vérifier le niveau d'autorisation d'un Access Token avant de faire des requêtes API.

- **Le Graph Explorer** : Il permet de simuler les requêtes API que l'application enverra. Tout d'abord, il faut choisir dans quelle application les futures commandes seront exécutées. En effet, si l'on demande le score d'un joueur dans une application n'ayant aucun résultat la réponse sera alors erronée. Ensuite, il est nécessaire de générer un Access Token adapté à nos besoins. Par exemple, si nous avons besoin des scores de l'utilisateur nous devons demander l'accès à "user_games_activities". Le plus fréquemment utilisé est l'autorisation de publication : "publish_actions".

Une fois ces deux paramètres rentrés, on peut commencer à rentrer nos premières requêtes GET. La plus classique étant "/me" qui affiche les champs contenant les informations relatives à l'utilisateur en cours. D'autres telles que "/me/score" pour les jeux ou "/app" pour les informations de l'application qui pourront être utiles. Voici un exemple présentant comment récupérer le score du joueur actuel :

```
FB.API("app/scores?fields=score,user", Facebook.HttpMethod.GET, CallbackMyScore);
```

Si on rentre "`app/scores?fields=score,user`" dans le Graph Explorer, on récupère (si l'application est un jeu) le score ainsi que le node "user" (qui contient un champ "name" et "id") de l'utilisateur.

Pour finir, il y a les requêtes POST où il faut remplir des champs particuliers. Par exemple, le champ "score" de "/me/scores" correspond au score obtenu dans l'application sélectionnée. Bien entendu dans le Graph Explorer la syntaxe est simplifiée, ainsi dans Unity3D vous devrez écrire la ligne suivante pour poster un score :

```
FB.API("/me/scores", Facebook.HttpMethod.POST, null, ScoreData);
```

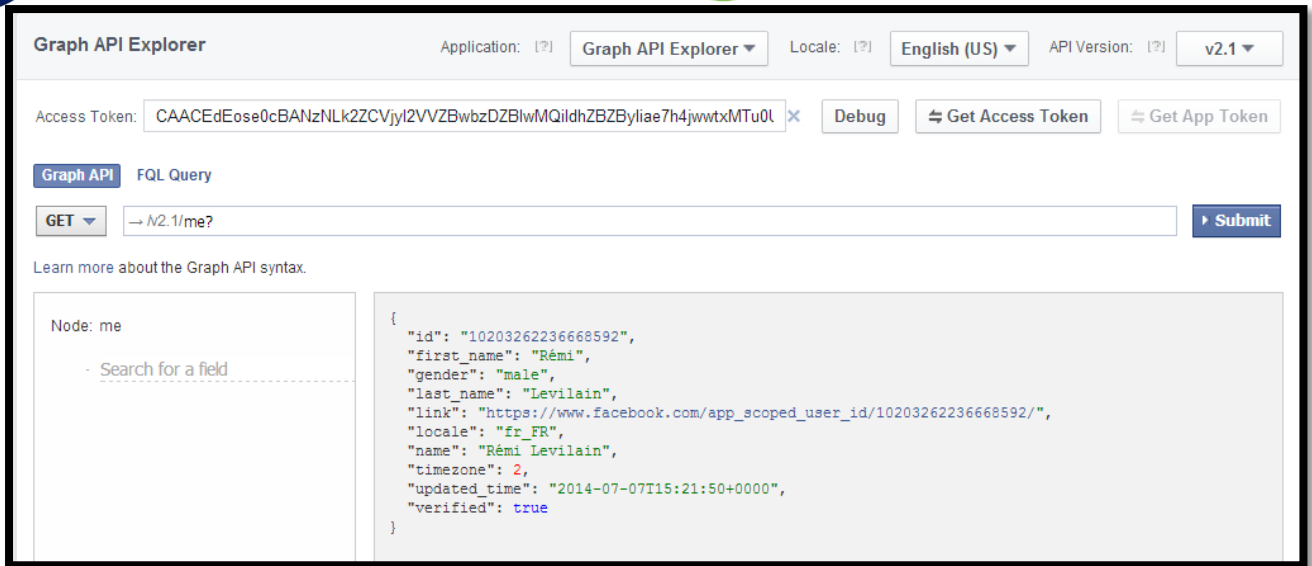


Figure 9 : Requête GET : /me dans le Graph API Explorer

- **Les JSON (JavaScript Object Notation)**

Le résultat des requêtes GET que vous enverrez via le Graph Explorer seront au format FB.Result qui correspond soit à un JSON, soit à une image, soit une erreur. Le JSON est un format d'écriture allégé pour le transfert de données. Pour illustrer, si vous faites un GET "/me" avec les champs nom, id et amis, vous obtiendrez un JSON similaire à :

```
{ "name": "John Doe", "user_id": "75782347",
  "friends":
    { "data": [ { "first_name": "Sally", "user_id": "98198298" } ] }
}
```

Ainsi pour récupérer les différentes valeurs des champs il faut "déserialiser" le JSON. Il sera alors au format d'un dictionnaire. Dans l'exemple actuel le dictionnaire contiendra tout d'abord les cases name, user_id et friends. Il faudra donc prendre la valeur associée à la clé "friends" pour la remettre dans un dictionnaire et ainsi obtenir la liste des amis comme dans l'exemple ci-dessous :

```
var dict = Json.Deserialize(jsonString) as Dictionary<string,object>;
    string userName = dict["name"];

    object friendsH;
    var friends = new List<object>();
    string friendName;
    if(dict.TryGetValue ("friends", out friendsH)) {
        friends = (List<object>)((Dictionary<string, object>)friendsH) ["data"];
        if(friends.Count > 0) {
            var friendDict = ((Dictionary<string,object>)(friends[0]));
            var friend = new Dictionary<string, string>();
            friend["id"] = (string)friendDict["id"];
            friend["first_name"] = (string)friendDict["first_name"];
        }
    }
```

5. Ajout d'un mini-jeu et d'un système de scores

A ce stade du projet nous avons une application Facebook fonctionnelle avec une scène de plage et quelques fonctionnalités disponibles comme le partage de l'application sur le mur d'amis.

Afin de proposer à l'utilisateur un minimum d'interactions dans cette application il a été décidé de rajouter un mini-jeu de memory sur les cabines de plage. Un système de score et de partage sera ajouté afin qu'il soit possible de concourir contre ses amis incitant ainsi des personnes supplémentaires à rejoindre l'application.

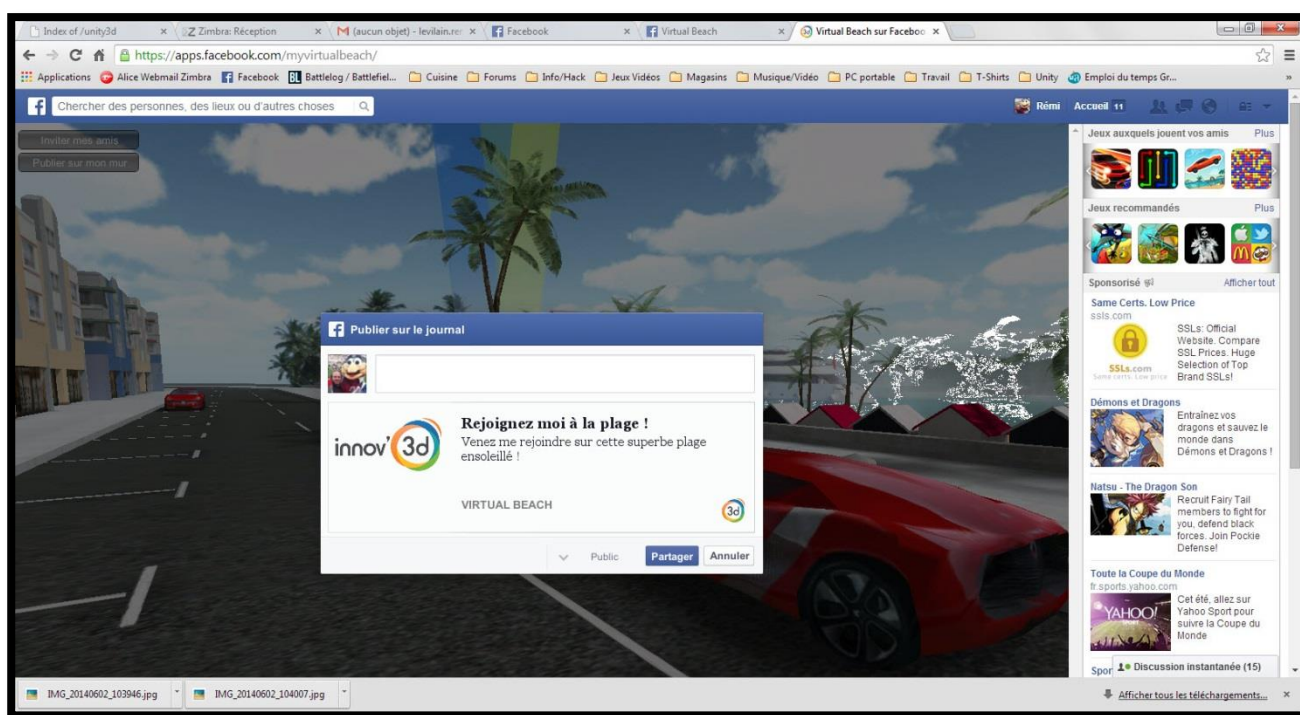


Figure 10 : Aperçu du partage de l'application Facebook

Le mini-jeu a entièrement été programmé en C# comme l'ensemble des applications et des scripts réalisés. Ce langage est le plus utilisé et le plus accessible pour la programmation sur Unity. Lors de la conception d'un jeu, il y a un script "Manager" qui veille au bon déroulement du jeu. Dans le cas du Memory, il gère le score et les actions à effectuer en fonction des combinaisons de portes ouvertes. D'autres scripts sont présents pour gérer l'identification et l'ouverture des portes, la position des avatars, etc...

Le système de score est basé sur le temps mis par le joueur pour retrouver toutes les paires. Plus il sera rapide, plus son score sera élevé. Le score est ensuite transmis à Facebook par le biais de la fonction vue précédemment :

```
FB.API("/me/scores", Facebook.HttpMethod.POST, null, ScoreData);
```

L'information est ainsi stockée en ligne par Facebook et peut être utilisée pour faire un tableau des scores de ses différents amis.

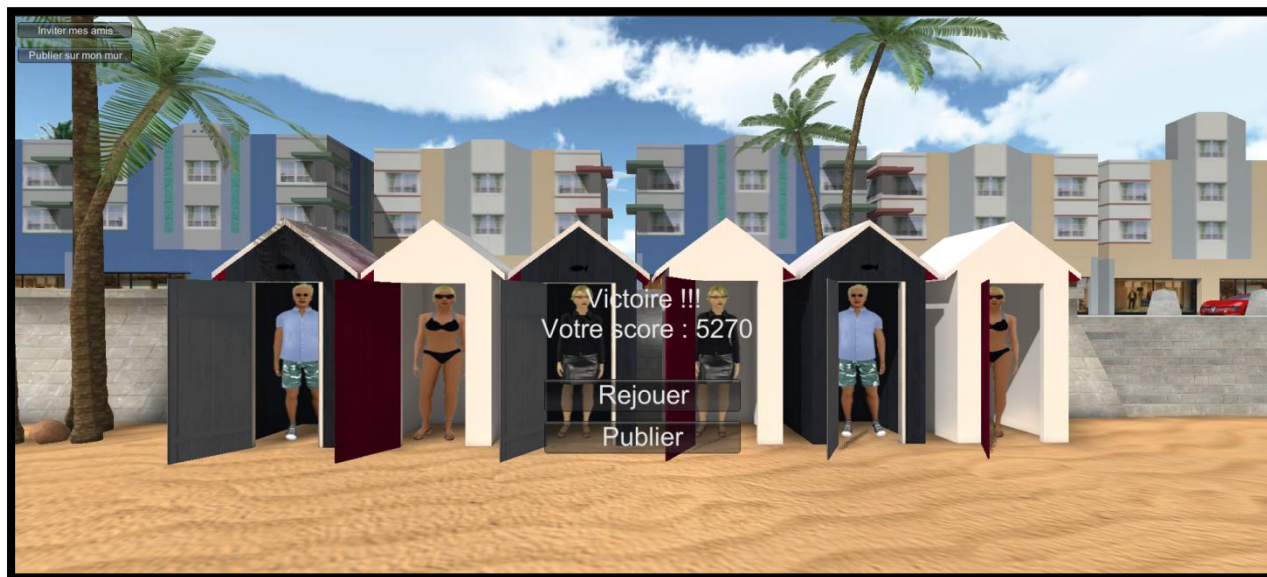


Figure 11 : Aperçu du score de l'application Facebook

6. Difficultés rencontrées et solutions apportées

La difficulté la plus importante dans ce projet aura été l'absence de connaissances initiales. En effet, l'ensemble des éléments liés à Facebook appartenait au domaine de la R&D. Afin de palier à cette difficulté, je me suis longuement renseigné sur les sites de documentation de Facebook à propos des fonctionnalités que je devais utiliser au sein de l'application. Cela comprend le login avec le système d'Access Token, le Graph Explorer qui permet d'essayer les différentes requêtes et aussi comment poster et partager du contenu à un autre utilisateur. La documentation de Facebook est claire mais pas encore totalement complète à propos du développement sur Unity.

La seconde difficulté aura été l'hébergement de l'application sur un serveur HTTPS. L'objectif principal était d'avoir rapidement une application fonctionnelle, or Facebook n'accepte que les applications hébergées en HTTPS. Il a donc été nécessaire de trouver rapidement comment obtenir ce type d'hébergement. J'ai eu la chance d'avoir l'aide d'une connaissance dans ce domaine, car dans le cas contraire j'aurais perdu énormément de temps à installer et configurer le VPS.

La dernière difficulté est liée à la gestion de projet. J'ai remarqué que ne connaissant pas les différents éléments à mettre en place pour un projet, je suis incapable de me prononcer sur le temps nécessaire à sa réalisation, ce qui peut poser problème dans une petite entreprise où les plannings sont bien remplis.

III. REALISATION D'APPLICATIONS DE REALITE AUGMENTEE

La seconde partie la plus importante de mon stage a été la réalisation d'applications en Réalité Augmentée (RA). Je me suis tout d'abord formé sur le module "Vuforia" pour Unity en créant une application contenant quelques objets augmentés. Ensuite on m'a confié un projet client avec des exigences bien précises.

1. Objectifs des projets

Le premier projet de RA consistait à comprendre l'utilisation du module Vuforia. Une fois ces connaissances acquises, il m'a été demandé d'augmenter quelques objets publicitaires afin de les montrer au public de l'entreprise. Cela permet de présenter rapidement une application RA et ainsi susciter l'intérêt d'un futur client potentiel.

Le second projet est commandé par le groupe Bejo qui possède une activité de vente de légumes et propose différents catalogues à ses clients. L'application RA doit être un support supplémentaire à utiliser sur les articles / images de leur catalogues. Cela donne alors accès à une fiche descriptive du produit, une vidéo de présentation et des liens vers le support client ainsi que le commercial associé au produit.

2. Les marqueurs en RA

En réalité augmentée, l'interactivité se fait par l'intermédiaire de marqueurs permettant, via une caméra, de repositionner un objet virtuel dans un flux vidéo en temps réel. Plus concrètement, lorsque la caméra détecte l'image du marqueur, elle fait apparaître l'objet virtuel associé.

Il y a des règles à respecter pour réaliser de bons marqueurs et pour les détecter. Ne pas prendre d'images à motifs répétitifs ou unis, il n'y aurait pas assez d'éléments distinctifs pour les reconnaître. Les surfaces réfléchissantes ou plastifiées sont à éviter car l'éclairage fait varier leur apparence. Il faut donc avoir une image assez variée et mate.

Durant mon stage, les marqueurs étaient réalisés via le site internet de Vuforia sur lequel il est nécessaire de s'inscrire. Il convertit les images envoyées en une base de données de marqueurs que l'on peut importer sur Unity. Ce système permet ainsi d'avoir plusieurs marqueurs pour une seule application.

Sur l'image ci-contre, le marqueur est la carte de visite d'Innov3D et l'augmentation est un modèle 3D visible sur le téléphone.

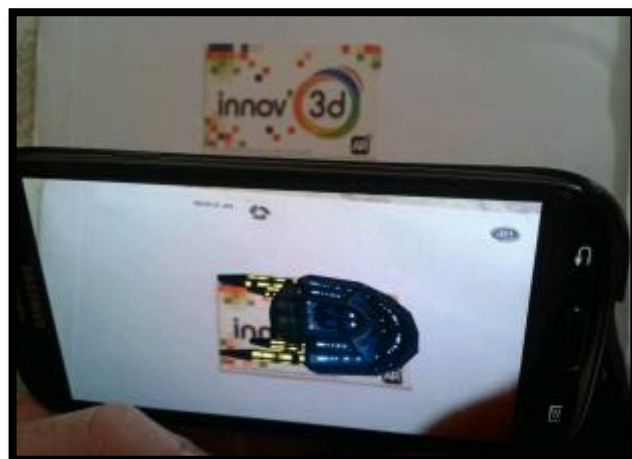


Figure 12 : Mon 1er test de RA

3. Premières réalisations

Le premier élément de démonstration que j'ai réalisé est une "ToDoList" qui s'affiche lorsque l'utilisateur cible l'image associée. L'objet virtuel qui est affiché est alors un post-it sur lequel sont affichées les différentes tâches de l'utilisateur. Dans cette démonstration, l'édition des tâches n'était pas disponible mais il était tout de même possible de cocher / décocher les "checkbox".

D'un point de vue de la réalisation sur Unity, il faut tout d'abord associer les éléments virtuels (le post-it) au marqueur de la base de données de Vuforia. Ensuite, il s'agit de rajouter des scripts contenant le code pour cocher les cases. La programmation associée à Vuforia pour la RA est déjà installée, il est tout de même nécessaire de la paramétrer.

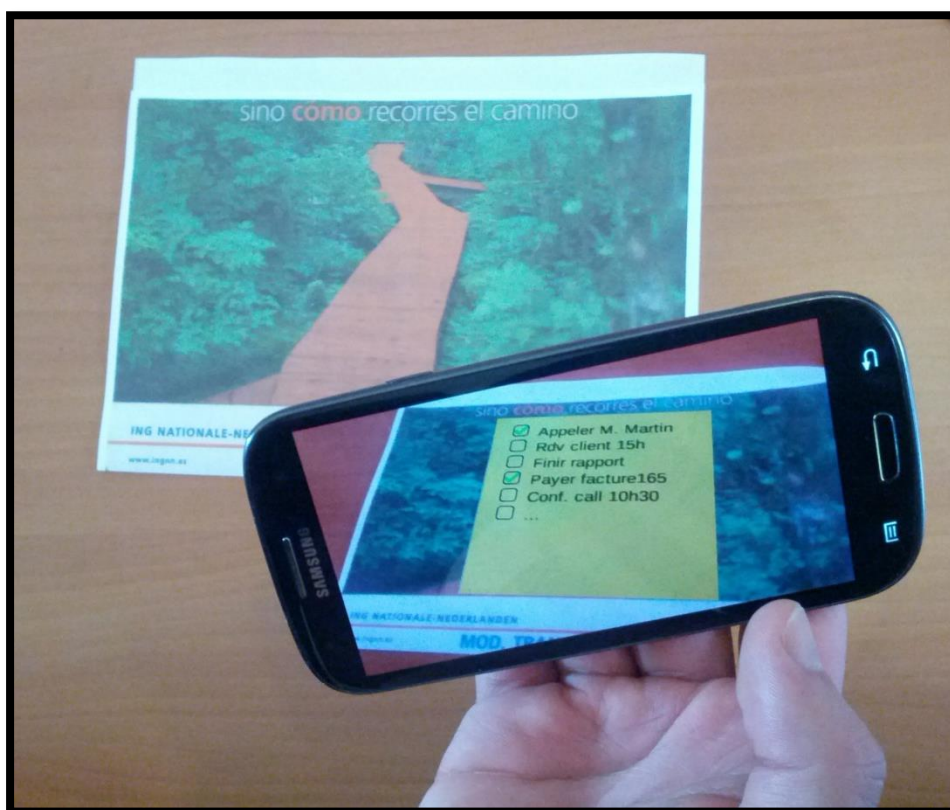


Figure 13 : Aperçu de l'application "ToDoList"

La seconde application de démonstration utilise un autre module gratuit pour Unity3D : CoherentUI. Il permet entre autres d'afficher une page internet sur des objets 3D au sein d'un environnement. La combinaison de cette technologie avec les marqueurs RA m'a permis de faire apparaître une page internet sur un tapis de souris lorsqu'il est ciblé avec une caméra. De plus, il est possible de naviguer et d'interagir avec la page comme lorsqu'elle est ouverte sur un navigateur internet. Malheureusement, après une mise à jour importante d'Unity ce module ne fonctionnait plus correctement et les pages web ne s'affichaient plus en combinaison avec une application en réalité augmentée.

4. Le projet BEJO

Dans le cadre de cette application le marqueur n'a pas encore été défini précisément par le client. Ce sera très certainement des images issues du catalogue Bejo. Trois produits nous ont été fournis pour travailler dessus : carottes, oignons et poireaux. J'ai choisi de me concentrer sur un seul. Les opérations étant identiques, le résultat sera dupliqué à la fin du projet.

Le marqueur de ce produit fera apparaître 2 objets : une représentation 3D animée en rotation du produit ainsi qu'un lien vers une vidéo YouTube qui s'ouvrira dans l'application associée. La finalité serait que la vidéo soit lue In-App (dans l'application), chose qui était possible jusqu'au problème de compatibilité entre CoherentUI et Unity.

Le modèle 3D du produit sera un lien cliquable vers une fiche de description du produit qui s'ouvre dans l'application. Deux liens associés au produit sont disponibles dans cette description : la page web du support et la fiche descriptive du commercial.

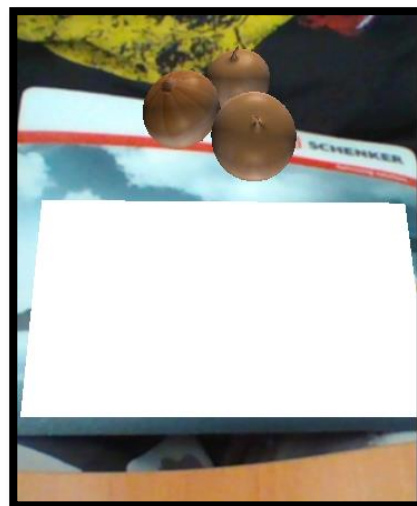


Figure 14 : Aperçu de l'objet 3D et
du cadre pour YouTube

- **Les liens :**

Pour atteindre une page internet dans une application, il existe une fonction nommée "Application.OpenURL(votreURL.fr)" qui est déjà présente initialement dans Unity. Cette fonction ouvrira alors la page web dans votre navigateur internet lorsque vous êtes sur ordinateur. Sur téléphone le lien est ouvert par l'application concernée, cela peut être votre navigateur ou bien l'application YouTube si le lien mène vers une vidéo de ce site.

- **Les fiches descriptives :**

L'affichage des fiches se fait différemment. Contrairement aux objets 3D qui s'affichent au-dessus du marqueur, les fiches s'affichent directement face à l'écran de manière à le remplir. Pour cela on utilise le système d'interface utilisateur (UI) d'Unity : le GUI. Dans notre cas il s'agit d'y afficher une image, on s'oriente alors vers une "GUI Texture". Ainsi lorsque l'utilisateur clique sur le modèle de l'oignon, le script détecte ce clic et instancie la "GUI Texture" à l'écran masquant ainsi l'affichage de la caméra.

De plus, les fiches descriptives comportent deux autres éléments invisibles qui sont les "GUI Textures" transparentes positionnées au-dessus des boutons "Support" et "Contact" servant à détecter lorsque l'utilisateur les sélectionne.



Figure 15 : Aperçu d'une fiche descriptive Bejo

- **L'auto-orientation :**

C'est un des éléments importants pour réaliser une bonne application sur Smartphone. Dans ce projet l'auto-orientation concernait les fiches descriptives, il n'y a donc pas besoin d'orienter la RA étant donné qu'elle est positionnée en fonction du marqueur.

Premièrement il a été nécessaire de créer un double de chaque fiche descriptive pour la seconde orientation de l'écran. On obtient ainsi deux "GUI Textures" : portrait et paysage. Pour les "boutons invisibles" la même solution a été retenue car leur position n'est pas la même en fonction de l'orientation du téléphone. Il y a donc deux paires de boutons, une lorsque le téléphone est à la verticale et l'autre quand il est à l'horizontale.

Ensuite la détection du changement d'orientation du téléphone s'effectue à l'aide d'un script en C# où un événement est programmé. Plus simplement lorsque l'orientation à l'instant T ne correspond pas avec celle de l'instant T-1 l'événement est validé. J'ai alors lié une action à cette validation qui consiste à afficher la "GUI Texture" et les boutons associés à l'orientation du téléphone. Bien entendu il faut aussi effacer les éléments correspondants à l'ancienne orientation. Pour plus de détails à propos de la programmation de cet événement, consultez **l'Annexe n°4**.

5. L'export sur Android

L'ensemble de ces applications étant prévu pour être utilisé sur Smartphone, il a été nécessaire de les exporter au format adapté. Possédant un téléphone Android et la procédure pour iPhone étant relativement complexe, je me suis directement axé sur la méthode d'export Android.

Unity permet d'exporter sous de nombreuses plateformes : PC, Mac, Linux, iOS, Android, Xbox, Playstation, etc... Il faut donc sélectionner la bonne plateforme au début du projet Unity afin que les conditions de travail soient émulées correctement. Une fois l'environnement adapté, différents paramètres vont devoir être remplis pour effectuer correctement l'export, principalement des données d'identification de l'application (version, entreprise, etc...). Il ne reste plus qu'à "builder" (construire) l'application pour obtenir un .apk qui s'installe sur un téléphone Android.



Figure 16 : Différents paramètres d'export Android

6. Difficultés rencontrées et solutions apportées

La principale difficulté pour les projets de RA aura été la configuration de mon ordinateur pour réussir à exporter les applications en .apk pour Android. Pour cela, il faut télécharger le kit de développement (SDK) Java pour Android et récupérer les différents niveaux API (interface de programmation). Cela donne un volume conséquent de fichiers mais ce n'est pas tout, il faut aussi respecter certaines règles pour l'installation. Il ne faut pas mettre d'espace dans le nom de dossier du SDK et choisir la version 32 bits car la version d'Unity fonctionne en 32 bits et non en 64 comme l'ordinateur. Si ces conditions ne sont pas respectées, l'export ne fonctionne pas et aucune indication n'est donnée, il faut se renseigner sur internet.

La seconde difficulté que j'ai observée concerne les problèmes de compatibilité avec les modules lorsqu'une nouvelle version d'Unity est déployée. C'est récurrent et les équipes de développement des modules importants sont réactives à chaque mise à jour. Parfois le problème peut être lié à un conflit entre deux modules qui ne peuvent fonctionner ensemble. Durant mon stage, j'ai changé de version d'Unity, malheureusement le module d'interface CoherentUI qui permettait d'afficher des pages internet dans une application ne fonctionnait plus après ce changement. La solution est de trouver un autre module ou d'attendre une version corrective.

L'application du groupe Bejo a soulevé quelques difficultés vis-à-vis de l'orientation des fiches et la détection du clic sur les boutons. Au début je n'avais aucune idée de la méthode pour faire apparaître correctement l'image à l'écran et gérer sa rotation tout en ayant les boutons qui s'adaptent à leur position. Au final, comme expliqué précédemment, j'ai utilisé les "GUI Texture" et un évènement pour la détection de l'orientation.

IV. ACTIVITES ANNEXES

1. Outils d'aide à la formation pour l'IFM3R

L'IFM3R (Institut Régional de Formation aux Métier de Rééducation et de Réadaptation des Pays de la Loire) souhaitait une démonstration d'application 3D pour l'enseignement de l'anatomie à leurs étudiants. Nous avons décidé de présenter un corps humain sur lequel on sélectionne la partie du corps que l'on souhaite étudier. Un zoom est fait et les muscles et/ou les os apparaissent. L'étudiant doit alors associer un nom aux éléments présentés à l'écran.

Cette application a été réalisée uniquement à l'aide d'Unity et de scripts associés aux interactions de l'utilisateur. Tout d'abord nous avons récupéré un modèle 3D conséquent représentant un corps humain complet. Il contenait le squelette, le système nerveux, lymphatique, les muscles, etc... Le but étant de faire une présentation à un client, nous nous sommes concentrés sur la zone du coude uniquement.

Le 1^{er} script réalisé est un script d'avancement permettant de connaître la progression de l'utilisateur. Ce script contient une variable publique qui s'incrémente au fur et à mesure de l'avancement. En fonction de cette variable, ce script affiche l'interface utilisateur adapté à la situation, c'est-à-dire les noms des différents éléments du coude (muscles, os...) que l'utilisateur doit replacer au bon endroit.

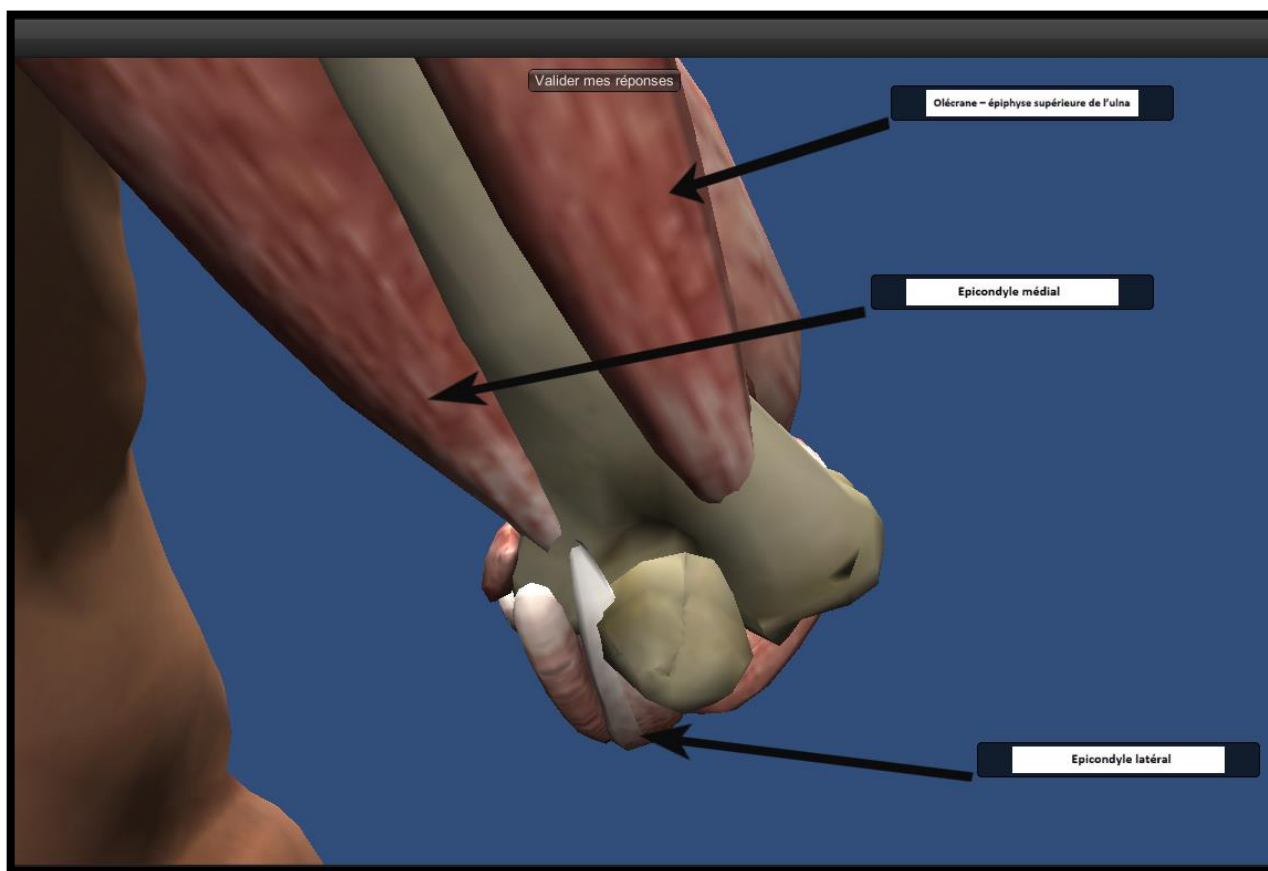


Figure 17 : Aperçu de l'application pour l'IFM3R

Le 2nd est destiné à la caméra. En effet, la zone du coude est cadrée sous différents angles afin de pouvoir observer l'intégralité de la zone étudiée. Ainsi, toujours en fonction de la variable d'étape, la caméra va être amenée à se déplacer. Le travelling de caméra a été réalisé à l'aide du module d'animation d'Unity.

Pour finir, le dernier script permet de comptabiliser le score obtenu par l'utilisateur. Il s'agit de récupérer un booléen dans le script principal qui renvoie si oui ou non l'utilisateur a correctement répondu. Cette fonctionnalité a été placée dans un script à part afin d'avoir une architecture en couches et ainsi un meilleur agencement et répartition des actions au sein du projet Unity.

2. Réalisation d'un diagramme dynamique pour le Serious Game de Chrysippe

Le cabinet Chrysippe réalise un programme de flexibilisation cognitive par immersion en RV. La situation test est "Faire face à une situation stressante de charge supplémentaire de travail". Nous allons donc réaliser une application dans laquelle notre avatar sera au bureau le vendredi soir et recevra du travail supplémentaire. Différents choix seront proposés à l'utilisateur et le scénario s'adaptera à la décision prise.

Dans cette application, les résultats seront affichés sous la forme d'un diagramme dynamique en colonne. Le diagramme n'étant pas présent durant l'ensemble de l'application, il a été nécessaire de réaliser un objet instanciable à n'importe quel moment où l'on a besoin d'afficher des résultats. De plus le nombre de résultats affichés, le type de colonnes, le titre et les légendes doivent être modifiables par l'utilisateur. Qui plus est, la taille du cadre doit aussi s'adapter à celle du tableau en fonction de son nombre de colonnes.

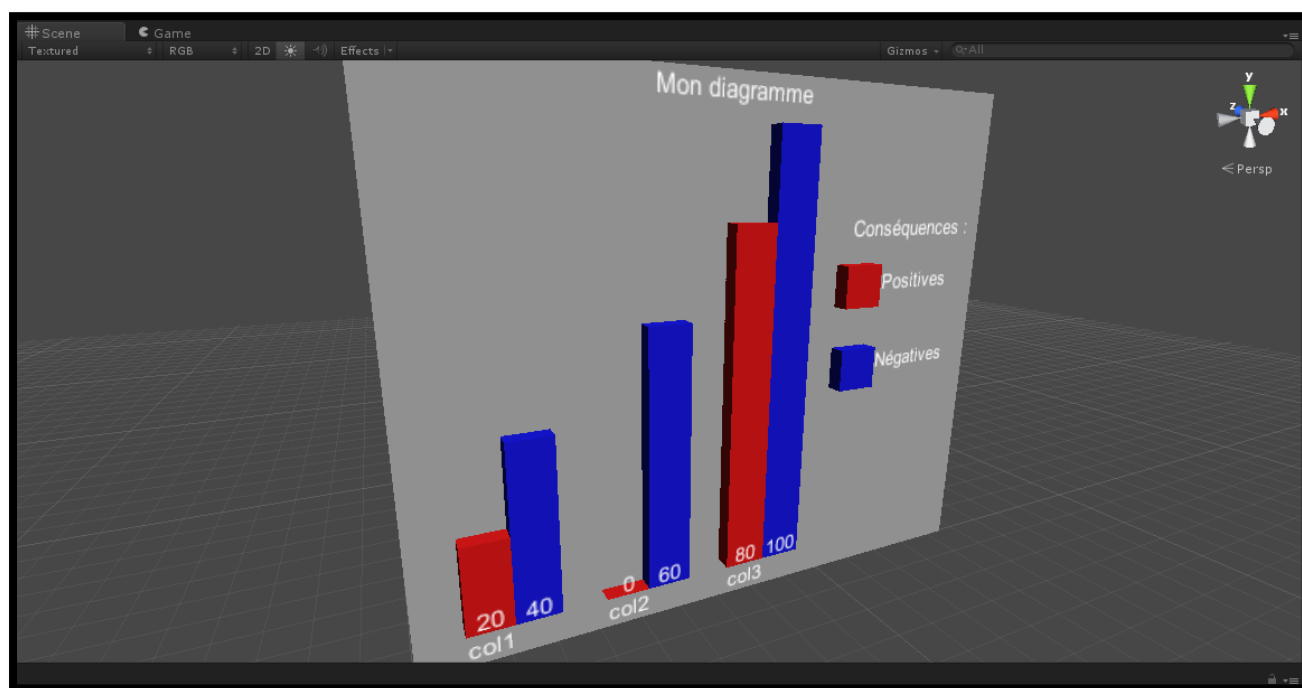


Figure 18 : Aperçu du diagramme en double colonnes

L'ensemble de ces fonctionnalités est regroupé dans le script propre au diagramme. Le fonctionnement est simple, il suffit de donner l'ensemble des paramètres du diagramme et de l'instancier. Les paramètres sont : le nombre de colonnes, le type de colonnes, leur valeur, la légende, le titre du diagramme et la description des couleurs du diagramme.

Le diagramme apparaît ensuite à l'écran et les colonnes s'animent jusqu'à atteindre leur valeurs respectives. Ce diagramme est un prototype, certaines fonctionnalités restent à améliorer telles que le retour à la ligne pour les différents textes, apporter de nouveaux types de diagrammes, etc... Il y a tout de même l'avantage qu'il peut être utilisé dans d'autres applications.

V. CONCLUSION

Durant ce stage de 3 mois j'ai pu réaliser différents projets de réalité augmentée, exporter une application sur Facebook, faire un peu de R&D mais aussi répondre à des demandes clients. J'ai donc eu la chance d'avoir des tâches diverses et enrichissantes. Sur le plan technique je peux conclure que j'ai appris beaucoup de choses à propos d'Unity grâce à la présence des autres développeurs de l'entreprise. A l'aide des recherches effectuées, j'ai acquis les bases de la programmation d'une application Facebook sans compter l'installation et la configuration du serveur. Le plus intéressant selon moi reste la réalité augmentée qui est un domaine en plein essor et dont les applications futures seront vraiment incroyables.

La vie au quotidien dans une petite entreprise m'a aussi montré différents aspects d'une start-up. Le cadre y est très agréable mais il y a des impératifs sur la réalisation des applications pour les clients. On ne compte pas ses heures pour réussir à obtenir un contrat et parfois on passe une soirée à travailler sur un projet qui doit être rendu prochainement. Il n'empêche que terminer un projet et apprendre que le client est satisfait reste toujours quelque chose de gratifiant.

En parlant du cadre, je ne crois pas avoir fait un stage plus agréable sur le plan humain que celui-ci. J'ai très vite été accepté par l'équipe présente et les relations sont rapidement devenues amicales avec le personnel de l'entreprise. Tout le monde se connaît, l'ambiance est bonne, cela permet de travailler et de communiquer plus efficacement.

Je peux ainsi conclure que ce stage a été enrichissant sur tous les plans que ce soit technique et humain. J'espère garder le contact avec les différentes personnes rencontrées dans cette entreprise tant professionnellement que personnellement.



Mai – Juillet 2014



Développement d'applications 3D
et de Réalité augmentée

Tables des figures

Figure 1 : Logo d'Innov'3D	1
Figure 2 : Logo d'Unity	3
Figure 3 : Logo de Vuforia	3
Figure 4 : Logo Facebook	4
Figure 5 : Logo Debian	4
Figure 6 : Logo d'OVH	4
Figure 7 : Aperçu de l'état du serveur sur Webmin	5
Figure 8 : Exemple de FB.Login()	6
Figure 9 : Requête GET : /me dans le Graph API Explorer	8
Figure 10 : Aperçu du partage de l'application Facebook	9
Figure 11 : Aperçu du score de l'application Facebook	10
Figure 12 : Mon 1er test de RA	11
Figure 13 : Aperçu de l'application "ToDoList"	12
Figure 14 : Aperçu de l'objet 3D et du cadre pour YouTube	13
Figure 15 : Aperçu d'une fiche descriptive Bejo	13
Figure 16 : Différents paramètres d'export Android	14
Figure 17 : Aperçu de l'application pour l'IFM3R	16
Figure 18 : Aperçu du diagramme en double colonnes	17

Webographie

Site internet d'Innov'3D : <http://www.innov-3d.com/>

Forums officiels d'Unity3D : <http://answers.unity3d.com/questions/index.html>

Site Unity3D Francophone: <http://www.unity3d-france.com/>

Documentation officielle d'Unity3D : <http://unity3d.com/learn/documentation>

Wiki Unity3D : <http://wiki.unity3d.com/>

Documentation officielle relative à Facebook : <https://developers.facebook.com/docs>

Site de Vuforia pour les marqueurs : <https://developer.vuforia.com/target-manager>

Site de Coherent UI : <http://coherent-labs.com/>

Table des annexes

1. Différentes commandes Linux utilisées pour le VPS	A1
2. Règles de pare-feu du VPS	A1
3. Exemples des fonctions Facebook	A2
4. Programmation d'un évènement lié à l'orientation du téléphone	A3

Annexes :

Annexe n° 1 : Différentes commandes utilisées pour le VPS

Mise en place d'Apache2 et du module SSL :

```
Apt-get install apache2
```

```
Sudo a2enmod ssl
```

Obtention du certificat pré-signé d'Ubuntu :

```
Sudo make-ssl-cert/usr/share/ssl-cert/ssleay.cnf/etc/ssl/private/localhost.pem
```

Autorisation des connexions SSL :

```
CP /etc/apache2/sties-availables/default-ssl ../sites-enable
```

Annexe n° 2 : Règles de pare-feu utilisées

```
:INPUT DROP [1:84]
```

```
:FORWARD ACCEPT [0:0]
```

```
:OUTPUT ACCEPT [56:5933]
```

```
-A INPUT -i lo -j ACCEPT
```

```
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
```

```
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
```

```
#-A INPUT -p tcp -m tcp --dport 25 -j ACCEPT
```

```
#-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
```

```
-A INPUT -p tcp -m tcp --dport 443 -j ACCEPT
```

```
#-A INPUT -p tcp -m tcp --dport 110 -j ACCEPT
```

```
#-A INPUT -p tcp -m tcp --dport 143 -j ACCEPT
```

```
#-A INPUT -p tcp -m tcp --dport 1194 -j ACCEPT
```

```
#-A INPUT -p tcp -m tcp --dport 161 -j ACCEPT
```

```
#-A INPUT -p udp -m udp --dport 161 -j ACCEPT
```

```
COMMIT
```

Annexe n° 3 : Exemples des fonctions Facebook

• FB.Init()

```

9  #region FB.Init() example
10
11  private bool isInit = false;
12
13  1 référence
14  private void CallFBInit()
15  {
16      FB.Init(SetInit, OnHideUnity);
17  }
18
19  1 référence
20  private void SetInit()
21  {
22      Debug.Log("SetInit");
23      enabled = true; // "enabled" is a property inherited from MonoBehaviour
24      if (FB.IsLoggedIn)
25      {
26          Debug.Log("Already logged in");
27      }
28      isInit = true;
29  }
30
31  1 référence
32  private void OnHideUnity(bool isGameShown)
33  {
34      Debug.Log("OnHideUnity");
35      if (!isGameShown)
36      {
37          // pause the game - we will need to hide
38          Time.timeScale = 0;
39      }
40      else
41      {
42          // start the game back up - we're getting focus again
43          Time.timeScale = 1;
44      }
45  }

```

• FB.Feed()

Les principaux paramètres sont :

told : A remplir par le FB.Id de la personne à qui appartient le mur sur lequel on souhaite publier.

Link : Si on souhaite mettre un lien internet

Picutre : Pour l'ajout d'une image

```

131 #region FB.Feed() example
132
133 public string FeedToId = "";
134 public string FeedLink = "";
135 public string FeedLinkName = "";
136 public string FeedLinkCaption = "";
137 public string FeedLinkDescription = "";
138 public string FeedPicture = "";
139 public string FeedMediaSource = "";
140 public string FeedActionName = "";
141 public string FeedActionLink = "";
142 public string FeedReference = "";
143 public bool IncludeFeedProperties = false;
144 private Dictionary<string, string[]> FeedProperties = new Dictionary<string, string[]>();
145
146 1 référence
147 private void CallFBFeed()
148 {
149     Dictionary<string, string[]> feedProperties = null;
150     if (IncludeFeedProperties)
151     {
152         feedProperties = FeedProperties;
153     }
154     FB.Feed(
155         toId: FeedToId,
156         link: FeedLink,
157         linkName: FeedLinkName,
158         linkCaption: FeedLinkCaption,
159         linkDescription: FeedLinkDescription,
160         picture: FeedPicture,
161         mediaSource: FeedMediaSource,
162         actionName: FeedActionName,
163         actionLink: FeedActionLink,
164         reference: FeedReference,
165         properties: feedProperties,
166         callback: Callback
167     );
168 }

```

Annexe n° 4 : Programmation d'un évènement lié à l'orientation du téléphone

Script de l'évènement :

```
//Declaring ChangedEventHandler type
public delegate void ChangedEventHandler(object sender, EventArgs e);

1 référence
public class OrientationChangedEventHandlerScript : MonoBehaviour
{
    public event ChangedEventHandler Changed;

    private DeviceOrientation currentDeviceOrientation;

    0 références
    void Start()
    {
        //Setting the initial device orientation
        currentDeviceOrientation = Input.deviceOrientation;
    }

    0 références
    void Update()
    {
        //Condition to trigger the event
        if (currentDeviceOrientation != Input.deviceOrientation)
            currentDeviceOrientation = Input.deviceOrientation;
        OnChanged(EventArgs.Empty);
    }

    1 référence
    void OnChanged(EventArgs e)
    {
        if (Changed != null)
        {
            //Parameters are sent to "DefaultChangedEventHandler.ToDoOnEvent"
            Changed(currentDeviceOrientation, e);
        }
    }
}
```

Script de l'action associée :

```
1 référence
void ToDoOnEvent(object sender, EventArgs e)
{
    //Action to run on event

    DeviceOrientation devOrientation = (DeviceOrientation)sender;

    //Landscape
    if (devOrientation == DeviceOrientation.LandscapeLeft || devOrientation == DeviceOrientation.LandscapeRight)
    {
        this.GetComponent<GUITexture>().texture = ficheHorizontale;
        foreach (Transform tr in children)
        {
            if (tr.tag == "Vertical")
                tr.gameObject.SetActive(false);
            else
                tr.gameObject.SetActive(true);
        }
    }

    //Portrait
    else if (devOrientation == DeviceOrientation.PortraitUpsideDown || devOrientation == DeviceOrientation.Portrait)
    {
        this.GetComponent<GUITexture>().texture = ficheVerticale;
        foreach (Transform tr in children)
        {
            if (tr.tag == "Horizontal")
                tr.gameObject.SetActive(false);
            else
                tr.gameObject.SetActive(true);
        }
    }
}
```



2013 - 2014

Développement d'applications 3D et de Réalité augmentée

Résumé :

Durant ce stage de 3 mois au sein de la start-up angevine Innov'3D j'ai réalisé plusieurs applications sur Unity que ce soit pour de la R&D ou des clients. Les plus importantes comportaient de la réalité augmentée utilisée ensuite sur Smartphone. J'ai aussi effectué l'export d'une application 3D sur Facebook où j'ai utilisé la programmation associée pour profiter des fonctionnalités de ce réseau social. Le travail au sein d'une jeune entreprise reste une très bonne expérience humaine et technique que je recommande à tout étudiant.

Mots-Clés : Unity, Réalité augmentée, Facebook, 3D temps réel

Thème : Applications 3D en temps réel

Summary:

I have spent my 3-months internship in a new company named Innov'3D. My main task was to create several applications on Unity. It was R&D and sometimes for a client. Some of the most important applications contained augmented reality that was used with Smartphones. I have exported a 3D application on Facebook and I have used the associated code in order to have access at several functionalities of this social network. Working in a new company is a pretty good technic and human experience for a student.

Keywords: Unity, Augmented reality, Facebook, real time 3D

Theme: Real time 3D applications

Zusammenfassung :

Ich habe meine 3-monatiges Praktikum in einem neuen Unternehmen mit dem Namen Innov'3D verbracht. Meine Hauptaufgabe war es, mehrere Anwendungen auf Unity erstellen. Es war R & D und manchmal für einen Kunden. Einige der wichtigsten Anwendungen enthielten "Augmented Reality", die mit "Smartphones" verwendet wurde. Ich habe eine 3D-Anwendung auf Facebook exportiert und ich habe den zugehörigen Code verwendet, um Zugriff auf mehrere Funktionalitäten dieser sozialen Netzwerk haben. Arbeiten in einer neuen Gesellschaft ist eine ziemlich gute Technik und der menschlichen Erfahrung für einen Studenten.

Stichworte: Einheit, Augmented Reality, Facebook, Echtzeit-3D

Thema : Echtzeit-3D Anwedungen