



Rapport de stage Master 2

Auteur : Nicolas LEOTIER

Parcours : FSIL spécialité ISL

Enseignant : Jean-Luc MASSAT

Entreprise : Nablaware

Tuteur : Xavier TALANDIER

Année : 2016-2017



Table des matières

Rapport de synthèse.....	3
Présentation de l'entreprise.....	3
Organigramme.....	3
Sujet du stage.....	4
Outils de développement.....	5
Unity.....	5
Vuforia.....	7
Méthodologie.....	8
Travail effectué.....	9
Application Haagen Dazs.....	9
Retouche d'une vidéo 360 à l'aide d'Unity.....	10
Prise de vue et montage vidéo 360.....	11
Application de localisation à base de Beacon.....	14
Live streaming 360 Ambisonic.....	17
Application LesJounéesMonSabert2017.....	19
Fauteuil de simulation pour la réalité virtuelle.....	21
Conclusion.....	23
Sources.....	24
Rapport Technique.....	25
Vidéo panoramique 2D et 3D dans Unity.....	25
Développement d'applications Vuforia.....	27
Complément sur la prise de son Ambisonic.....	28
Configuration du serveur NGINX.....	29
Fichier bash permettant le transcodage d'une vidéo.....	30
Complément sur l'algorithme de localisation à partir de signal RSSI.....	31

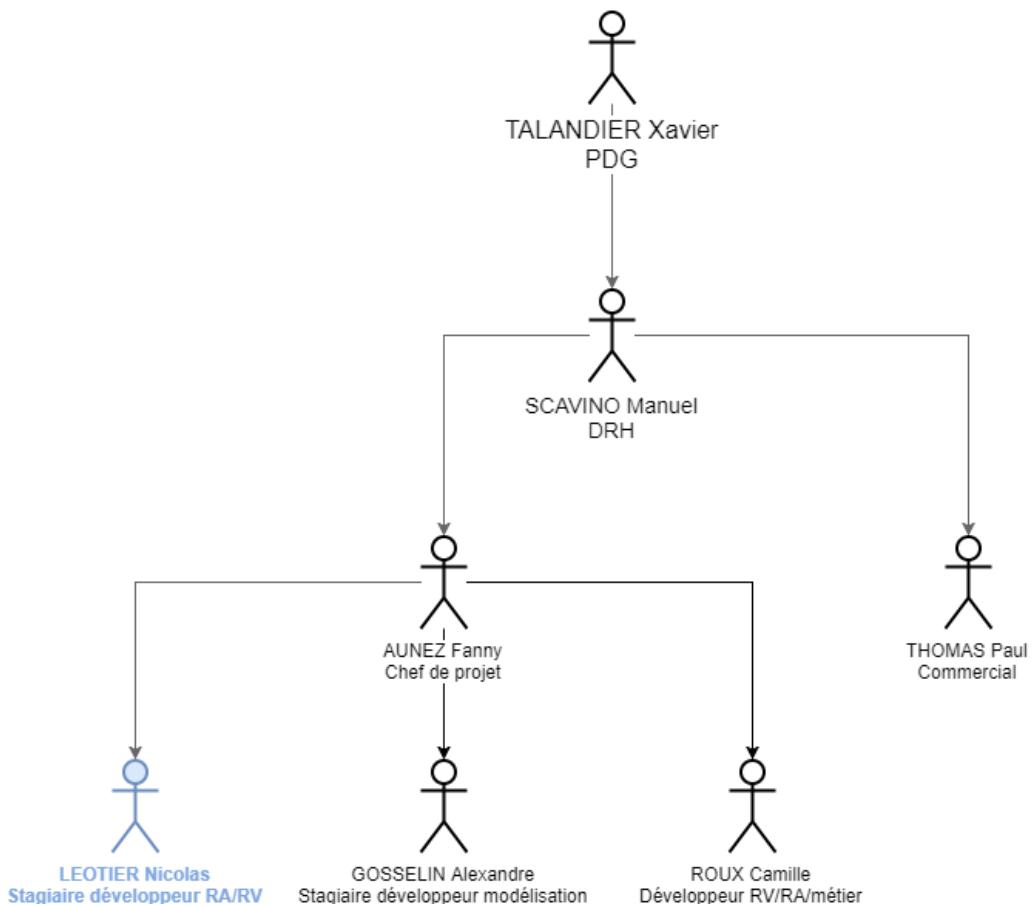
Rapport de synthèse

Présentation de l'entreprise

Nablaware est une entreprise créée en 2015 comportant 7 employés. Son activité principale est la création d'application mobile à destination des entreprises.

Mon recrutement s'est effectué suite à une décision de la part de l'entreprise de s'étendre dans le milieu de la réalité virtuelle et de la réalité augmentée. Cette nouvelle activité partage le cœur de métier de Nablaware, c'est à dire les applications mobiles. Elles offrent cependant des services novateurs grâce à des technologies innovantes. Il a donc fallu faire une séparation au sein de Nablaware afin de ne pas confondre cette nouvelle branche avec l'activité principale, car les outils de développement et la méthodologie diffère des applications métiers classiques. Ainsi NablawareXP est née.

Organigramme



Sujet du stage

Le sujet de mon stage est : développement d'application mobile dans le domaine de la réalité virtuelle. Nous allons voir que le travail réalisé est plus diversifié que prévu et a demandé des connaissances dans plusieurs domaines de l'informatique.

Le fil rouge de mon stage est un évènement organisé par NablawareXP, prévu pour le week-end du 16 et 17 septembre : les journées Monsabert. Au travers d'une exposition de l'armée de terre, le fort Ganteaume ouvre ses portes et invite le public pour des visites libres ou guidées, pendant les journées européennes du patrimoine de la Ville de Marseille. L'exposition retrace 75 ans de l'histoire de la 3e division, l'unité qui a libéré Marseille en 1944 aux côtés des forces françaises de l'intérieur (FFI), et dont l'état-major est situé aujourd'hui dans la cité phocéenne, au quartier Rendu (13009).

NablawareXP possédera son stand dédié à la découverte des technologies mise en place lors de mon stage afin de proposer un contenu ludique et divertissant.



Au travers de mon stage j'ai particulièrement été en contact avec des technologies comme la réalité virtuelle et la réalité augmentée qui sont les domaines phares de l'entreprise. Mais lorsqu'une entreprise emploie 7 salariés, dont 3 développeurs, il faut savoir faire preuve de polymorphisme et procéder méthodiquement afin d'utiliser au mieux les ressources à disposition. On peut donc facilement se retrouver à faire du réseau, des montages vidéos, ou encore être disponible pour des déplacements éventuels.

Outils de développement

Unity

Unity est un moteur de jeu multi-plateforme (smartphone, Mac, PC, consoles de jeux vidéo et web) développé par Unity Technologies. Il est l'un des plus répandus dans l'industrie du jeu vidéo, du fait de sa rapidité aux prototypages pour les très gros studios, aussi pour la sphère du jeu indépendant qui l'utilise pour sortir leurs applications sur tout supports.

Le logiciel a la particularité d'utiliser un éditeur de script compatible mono (C#), UnityScript (un langage proche du JavaScript et inspiré d'ECMAScript) et Boo au lieu de Lua très utilisé dans les jeux vidéo.

Lors de mon arrivé à Nablaware, le premier outil qui m'a été présenté est Unity, expliqué par le biais d'un collègue de travail. L'entreprise n'avait que peu d'expertise dans le domaine. Il m'a fallu donc entreprendre l'apprentissage du logiciel, ainsi que du C#, le langage de programmation qu'il utilise.

La prise en main d'Unity n'a pas été un problème puisque j'avais une idée de ce qu'est un moteur graphique. De part mes recherches personnelles ou part le biais de l'université, comme par exemple un exposé en groupe d'anglais qui était consacré à ce sujet.

Le C# est très proche du Java que l'on a étudié ces 3 dernières années, son apprentissage a donc été relativement facile. Mais j'ai pu constater que la connaissance d'un autre langage orienté objet, couplé avec mes leçon de C de premières années de licence, m'ont permis de comprendre la logique d'un nouveau langage que je n'avais jamais étudié auparavant.

Le duo Unity-C# est une équipe gagnante pour moi, et mon utilisation pendant les 6 derniers mois confirme cette opinion. Unity est déjà un outil très puissant, avec une communauté grandissante, son interface graphique intuitive permet de poser aisément les bases d'une application, comme l'UI et les changements de scènes. Le C# quant à lui permet d'implémenter les règles plus complexe, comme la gestion à l'image près d'une vidéo, la rotation d'un objet, ou encore l'implémentation des règles métiers.

Les objets créés dans l'interface Unity peuvent donc être manipulé en C# pour disposer de toute les ressources d'un langage de programmation.

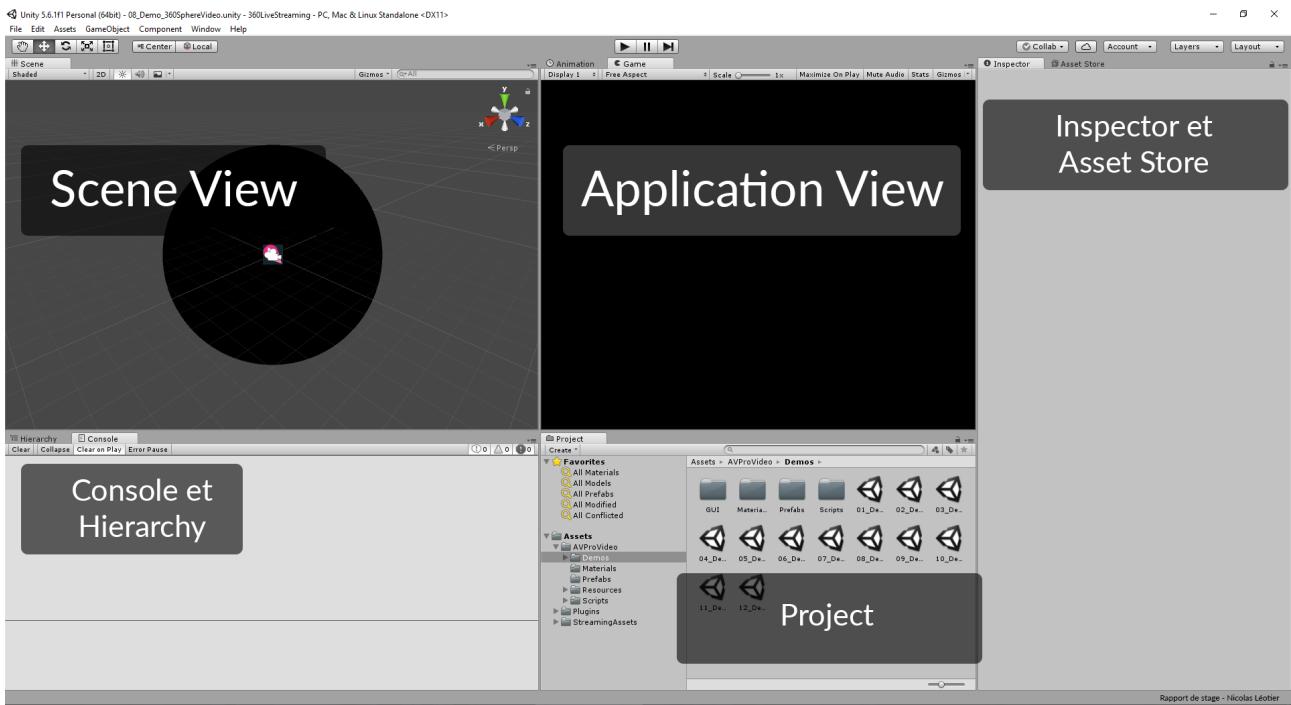


Illustration 1: Capture d'écran du moteur graphique Unity

Scene View : L'endroit où l'on peut déplacer les objets à l'intérieur d'Unity, les 5 boutons en haut à gauche permettent de transformer l'objet : le déplacer, l'agrandir ou le rétrécir, le redimensionner et le tourner. En haut à droite de la vue nous pouvons nous orienter dans le plan 3D, car il est parfois difficile de repérer le haut du bas dans un repère quadrillé.

Application View : Les boutons en haut au centre permettent de lancer l'application, l'application view se met donc à bouger en fonction du projet ouvert. Ici nous avons une sphère dans la scene view qui contient une texture vidéo, si nous appuyons sur le bouton play, la texture vidéo va bouger dans l'application view. Pour avoir un affichage dans l'application view, il est nécessaire de placer une caméra dans la scene view. Ici nous avons placé la caméra au centre de la sphère, et nous avons attaché à la caméra un script permettant de récupérer les mouvements de la souris, puis de faire bouger la boule en fonction de ces mouvements, ce qui nous permet de regarder une vidéo projetée à l'intérieur de la sphère.

Console et Hierarchy : C'est ici que nous retrouvons les fonctionnalités d'un IDE plus classique comme Eclipse. La console affiche les erreurs de script et les debug Unity. La hierarchy est composée des éléments présents dans notre scène.

Project : Tout simplement l'affichage du répertoire du projet. Il faut faire attention à respecter certaines normes fixés par Unity et à ne pas se perdre en manipulant les nombreux plugins mis à disposition.

Inspector et Asset Store : L'inspector permet d'analyser le comportement d'un objet. Des informations s'affichent dans cette fenêtre lorsque l'on clique sur un objet dans la scene view ou dans la hierarchy. L'inspector nous montre tout les scripts rattachés à un objet, ainsi que ses propriétés fondamentale comme sa position, sa rotation, ou encore sa couleur.

L'Asset Store est un « magasin » qui permet d'acquérir de nouveau plugin, à la manière des librairies, les plugins permettent d'ajouter des fonctionnalités à Unity, comme la lecture avancée de vidéo, la création rapide de prototype 3D ou le templating d'application.

Vuforia

Vuforia est un kit de développement logiciel (SDK) pour la réalité augmentée utilisé sur les appareils mobiles qui permet la création d'applications. Il utilise la technologie Computer Vision pour reconnaître et suivre les images planes et les objets 3D simples, comme les boîtes, en temps réel. Cette capacité d'enregistrement d'image permet de positionner et d'orienter des objets virtuels, tels que des modèles 3D et d'autres supports, en relation avec des images du monde réel lorsqu'ils sont visualisés via la caméra d'un appareil mobile. L'objet virtuel suit alors la position et l'orientation de l'image en temps réel afin que la perspective de l'observateur sur l'objet corresponde à leur point de vue sur l'image cible, de sorte qu'il apparaît que l'objet virtuel fait partie de la scène du monde réel.

Vuforia fournit des interfaces de programmation d'applications en C++, Java, et les langages .Net via une extension du moteur de jeu Unity. De cette façon, le SDK prend en charge à la fois le développement natif pour iOS et Android, tout en permettant également le développement d'applications dans Unity facilement portables sur les deux plates-formes. Les applications développées à l'aide de Vuforia sont donc compatibles avec une large gamme d'appareils mobiles, y compris les téléphones et tablettes iPhone, iPad et Android avec Android OS version 2.2 ou supérieure et un processeur ARMv6 ou 7 avec des capacités de traitement en Floating Point Unit.

Vuforia a été le premier outil que j'ai pris en main à l'aide d'Unity. Il a donc fallu comprendre et analyser le monde complexe de la réalité augmentée grâce aux informations fournies de la part d'Unity et de Vuforia. Pour cela j'ai été accompagné par Nablaware tout au long de ma progression, non seulement sur ma compréhension globale de la réalité augmentée, mais aussi concernant les outils qui ont été mis à ma disposition.

Méthodologie

Nablaware s'appuie sur 6 points clefs, nous verrons que la réalité est légèrement différente car surmonter les contraintes passe souvent par l'adaptation à la situation.

1. Analyse & Conseil

- Comprendre, anticiper et définir les besoins.
- Proposer les solutions les plus pertinentes.

2. Définition du projet

- Réaliser un cahier des charges complet
- Définir les ressources nécessaires

3. Planning & Gestion

- Établir le calendrier complet du projet
- Définir les étapes clés de validation

4. Conception

- Valider avec le client l'ensemble de la conception
- Préparer la phase développement

5. Développement

- Engager la phase de développement
- Proposer plusieurs plateformes cibles

6. Livraison

- Mettre en place des nouvelles versions
- Prévoir une évolution fonctionnelle

Travail effectué

Application Haagen Dazs

Pour mes premières semaines de stage, ma mission a été de me familiariser avec mon environnement, que ce soit socialement ou en terme de développement. La première mission qui m'a été confiée fut la reprogrammation d'un ancien jeu codé avec une version d'Unity dépassée, qui ne pouvait pas être mise à niveau automatiquement.

L'application est un jeu mobile qui a pour but de divertir et d'amuser, elle s'adresse à un public large. Basée sur la réalité augmentée, elle se déroule donc dans le monde réel à travers la caméra d'un périphérique.

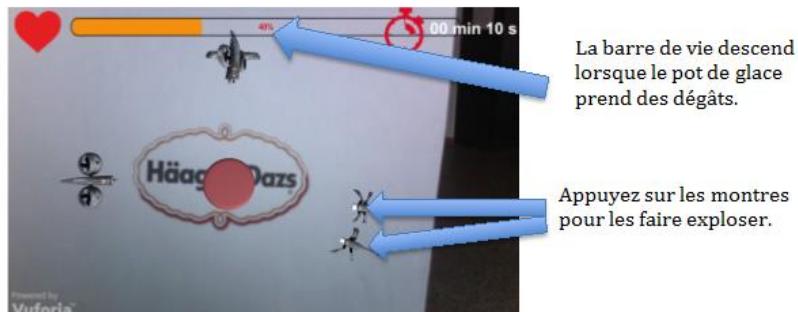


Illustration 2: Capture d'écran de l'application Haagen-Dazs

Le but de ce jeu est de défendre le plus longtemps possible un pot de glace contre des petites créatures. Une barre de vie et un chronomètre sont intégrés afin d'améliorer l'expérience de jeu. L'application fonctionne à l'aide d'une caméra et d'un marqueur, actuellement le logo d'Häagen-Dazs, mais le design peut s'adapter facilement à une variété de format. Il suffit de lancer l'application et d'appuyer sur le bouton « Jouer », la caméra du périphérique s'allume, il est alors possible de cibler le logo d'Häagen-Dazs pour que la partie puisse démarrer.

Un pot de glace en 3D apparaît alors dans le monde réel accompagné de créatures qui viendront attaquer le pot. Appuyez simplement dessus pour les faire exploser avant qu'elles ne parviennent à détruire le pot !

Chaque fois que le pot est touché, il perd de la vie. Une jauge vous indique le pourcentage de vie qu'il lui reste. Un chronomètre est lancé à chaque début de partie et prendra fin lorsque la vie du pot tombe à 0 %. La partie est alors terminée en vous indiquant le temps que vous avez mis avant de céder aux tirs ennemis. Une fois la partie terminée, vous avez la possibilité de retenter votre chance.

Retouche d'une vidéo 360 à l'aide d'Unity

Nablaware est une société de service et peut donc être amenée à aider ses clients à réaliser des tâches particulières afin de monter en compétence et de développer des outils en interne, tout en remplissant un contrat.

Phonomatik Lab, qui est une société de production et d'ingénierie sonore spécialisés dans l'immersion, a réalisé un clip dans un format qui prend en charge 360 degré de lecture vidéo et un son ambisonic qui permet de spatialiser le contenu sonore d'un film. Phonomatik Lab a donc fait appel à Nablaware afin d'incruster des éléments 3D à l'intérieur de la vidéo.

Après que l'on m'ait attribué la mission, le but était simple : rajouter un fantôme sortant d'un placard puis modéliser des danseuses exotiques sortant d'un globe.

Unity ne s'est pas avéré l'outil le plus efficace pour cette tâche, mais l'expérience en reste quand même enrichissante, j'ai appris à diffuser une vidéo en 360 à l'aide d'Unity, puis à rajouter des éléments 3D à l'intérieur d'une sphère possédant une texture vidéo. Le tout ajusté à l'image près afin que le rendu soit le plus réaliste possible. Des logiciels comme Adobe Première ou Adobe After Effect sont d'habitude recommandés pour ce genre de prestations, le choix d'Unity était poussé par le goût de la découverte et de l'exploration, afin de trouver une forme de contenu novateur.

Le clip est disponible sur Youtube à l'adresse suivante :
<https://www.youtube.com/watch?v=iyZoOy1P9EE>

Cette première mission m'a tout de suite fait comprendre le monde dans lequel je venais d'entrer : des deadlines à respecter, des responsabilités quant au travail effectué, une autonomie sur la gestion de mon emploi du temps.

Ces aspects m'ont donné une énergie et une confiance que je ne connaissais pas, j'ai donc mené à bien ma mission malgré les obstacles qui se sont dressé sur ma route. Mes collègues ont encore été là pour me soutenir et me donner toutes les ressources dont j'avais besoin afin de finir le développement dans le temps imparti.

Prise de vue et montage vidéo 360

Un secteur principal dans lequel Nablaware veut se développer est la prise de vue et le montage de vidéo 360. Ce format permettant un contenu immersif, il est possible que l'on assiste à une montée de cette technologie dans les prochaines années. Durant mon stage, j'ai effectué des déplacements afin de réaliser des prises de vue en 360.

Dans les premières semaines de mon stage, une seule caméra 360 était disponible, c'est un appareil créé par GoPro spécialement pour la prise de vue en 360. Un cube métallique permet de garder les 6 GoPro HERO4 ensembles. De plus, les 6 caméras sont reliés à une unité centrale qui permet de synchroniser l'enregistrement de photo et de vidéo. La qualité maximale est 4k pour les films et 8k pour les photos.

Une télécommande incluse permet de déclencher tout les caméras via Wi-Fi. Dans des conditions optimales, il est possible de contrôler le dispositif à une distance de 180 m. Ainsi, il est possible de se placer en dehors de la scène pour enregistrer la totalité de son environnement.



Illustration 3: GoPro Omni



Illustration 4: Insta360 Pro

La deuxième caméra qui est arrivé au milieu de mon stage est l'Insta360 Pro, elle dispose de 6 objectifs indépendants de type « Fish-eyes ». Elle peut être utilisée aussi bien pour réaliser des clichés de très haute qualité (60 Mégapixels) ou des vidéos à 360 degrés et/ou en 3D. C'est surtout la définition qui différencie la caméra VR des autres produits sur le marché. La caméra Insta360 Pro propose effectivement une définition de 8K pour la vidéo. Il est aussi possible de filmer en 4K et d'adopter le mode 100 fps. Cette vitesse permet notamment de s'affranchir du phénomène de nausées bien connu des possesseurs de casques de réalité virtuelle.

Il est aussi possible d'utiliser la caméra en mode Live-Stream pour partager un évènement sur Internet via le protocole RTMP .

Les premières prises de vue que j'ai faites sur le terrain ont été réalisées à l'aide de la GoPro Omni, elles m'ont permis de voir comment se comporte le matériel sur le terrain, connaître les limites, les contraintes et les cotés pratiques du film en 360.

Contrairement au film conventionnel, une vidéo prise en 360 n'est pas exploitable immédiatement. Il faut en effet passer par une étape appelée le « stitching ». GoPro propose une suite logiciel appelé Kolor permettant de « stitcher » ou « assembler » les différentes photos/vidéos en une seule photo/vidéo au format 360.



Illustration 5: Capture d'écran du logiciel Autopano créé par GoPro

Après avoir transféré les vidéos de la caméra vers l'ordinateur, il faut utiliser Autopano qui est un logiciel de la suite Kolor proposée par GoPro. Ici nous avons, les « stitching information » qui nous donnent le RMS. La valeur efficace, dite aussi valeur RMS (de l'anglais root mean square, moyenne quadratique) d'un signal est la racine carrée de la moyenne de cette grandeur au carré, sur un intervalle de temps donné ou la racine carrée du moment d'ordre deux (ou variance) du signal :

Le RMS correspond donc à la cohérence d'une vidéo 360.

Le problème résolu par l'optimiseur peut être considéré comme un problème d'ajustement de la courbe : compte tenu d'un modèle de courbe (p. Ex. $Y = a * X + b$), trouver les paramètres (a et b) qui font que la courbe s'adapte au mieux à une série de points de données. Dans le contexte de la couture panoramique, le modèle est l'équation de projection d'un point de scène 3D à un pixel d'image 2D et les paramètres sont les inconnues d'étalonnage et l'orientation de chaque image.

L'entrée de l'optimiseur est en deux parties:

- Le modèle du problème. Dans le cas de l'assemblage des images, il s'agit essentiellement du type d'objectif utilisé (standard ou fisheye).
- Les données. Ils sont un ensemble de points qui sont censés être générés par le modèle. Dans notre cas, ce sont les points de contrôle appariés provenant du détecteur.

Le but de l'optimiseur est d'estimer les paramètres du modèle (a et b) à partir des points de données. Donc plus le RMS est bas, meilleure et la cohérence de la vidéo.

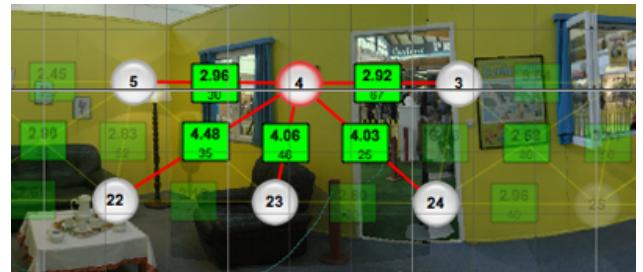


Illustration 6: Zoom sur l'interface du logiciel Autopano

3.2 Optimisation

Dans la « panorama editing area » les cercles représentent les différentes prises de vue capturé par les caméras. Les carrés verts sont les liens que le logiciel a trouvé entre les prises.

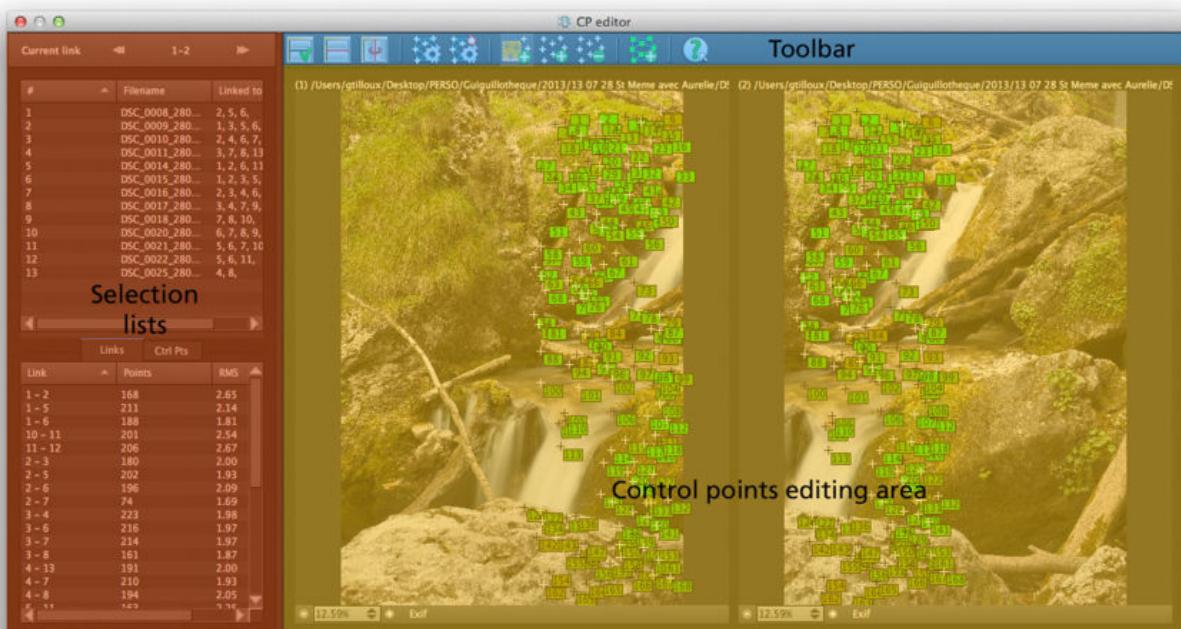


Illustration 7: Capture d'écran du logiciel Autopano

La zone d'édition des points de contrôle est divisée en 2 fenêtres où 2 images sélectionnées dans la liste peuvent être visualisées et les différents outils peuvent être utilisés pour modifier les points de contrôle.

Application de localisation à base de Beacon

Après 2 mois passés au sein de Nablaware, une nouvelle mission m'a été confié : permettre une localisation en intérieur à l'aide d'objets bluetooth connectés appelés beacons. Une application mobile permettant de récupérer les informations émises par les beacons et calculer la position de l'appareil par rapport aux signaux reçus a donc été envisagée .

Les balises Bluetooth sont des émetteurs – ils font partie des périphériques Bluetooth Low Energy (BLE), ils envoient leur identifiant sur les appareils dotés d'une puce bluetooth à proximité. La technologie a pour vocation de permettre aux smartphones, tablettes et autres appareils d'effectuer des actions à proximité d'un beacon.

Une application basique des beacons consiste à distribuer des messages à un point d'intérêt spécifique, par exemple un magasin, un arrêt de bus, une pièce ou un emplacement plus précis, comme un meuble ou un distributeur automatique. Cette méthode est similaire à la technologie Geopush précédemment utilisée basée sur le GPS, mais avec un impact réduit sur la durée de vie de la batterie et une précision accrue considérablement.

Avec plusieurs balises, la trilateration peut être utilisée pour estimer la position d'un utilisateur à environ 2 mètres près. Les balises Bluetooth sont capables de transmettre une Indication de la Force du Signal Reçu (RSSI) en plus d'autres données. Cette valeur appelé RSSI est calibrée par le fabricant de la balise pour être la puissance du signal à une distance connue, typiquement un mètre. En utilisant la puissance du signal de la balise et la puissance du signal observée par le dispositif de réception, une approximation peut être faite sur distance entre la balise et le dispositif. Cependant, cette approximation n'est pas très fiable, donc pour un suivi de position plus précis, d'autres méthodes sont préférées.

Depuis sa sortie en 2010, de nombreuses études ont été menées à l'aide de balises Bluetooth. Quelques méthodes ont été testées pour trouver la meilleure façon de combiner les valeurs RSSI afin d'avoir la position la plus précise possible. Les réseaux de neurones ont été proposés comme un bon moyen de réduire l'erreur dans l'estimation. Une approche dites Fingerprint (empreinte digitale) a également été testée, cette méthode utilise une carte d'intensité (radiomap) pour estimer l'emplacement d'un utilisateur.

Dans notre cas, nous avons 6 beacons en notre possession, il a fallu les placer judicieusement dans les 110m² des bureaux de Nablaware. Pour donner un prototype fonctionnel il m'a été donné 3 semaines, autant dire que les délais était très courts. De plus, un nouveau stagiaire en première année de formation dans à l'école Epitech nous a rejoins après la première semaine. J'ai donc eu une semaine pour concevoir un plan de travail s'étalant sur deux semaines, avec la mise à niveau de ce nouveau stagiaire avec qui j'ai collaboré. Il a été jugé qu'en dessous d'une précision à 5m près, le prototype restera sans suite car les champs d'utilisation en serait trop réduit.

Après une semaine de recherche, de sérieux problèmes concernant les beacons sont apparus :

- Une fluctuation du RSSI se retrouve dans tous les appareils bluetooth, le défaut semble inévitable et il se retrouve à plus ou moins grande intensité chez les différents constructeurs.
- Les objets métalliques affectent grandement le signal, causant une distorsion lors de la perception par le récepteur.

Les applications de localisation implémentant des méthodes d'empreinte numérique peuvent être utilisées dans les centres commerciaux, pour connaître le chemin des consommateurs et fournir des alertes marketing; dans les hôpitaux, pour surveiller les patients, les médecins et les équipements critiques; en logistique, pour le suivi des actifs et l'optimisation des espaces vides dans les ports ou les entrepôts intérieurs; et dans les maisons, pour les services Ambient Assisted Living (AAL), autrement dit les maisons connectées. Le principal avantage des techniques d'empreinte numérique est qu'elles peuvent utiliser les infrastructures de communication sans fil existantes, sans avoir à déployer d'équipements spécialisés supplémentaires. L'ensemble de données des empreintes numériques, appelé "radiomap", est à la base des algorithmes de positionnement. Les radiomaps peuvent être générés rapidement et à un coût relativement faible. L'empreinte digitale nécessite une phase hors ligne et une phase en ligne. Pendant la phase hors ligne, la radiomap est générée en enregistrant les valeurs RSS des points d'accès (AP) pour chaque emplacement dans la zone d'intérêt. Des techniques de calibrations à base d'apprentissage automatique sont généralement utilisées pendant cette phase, afin d'améliorer la qualité de la radiomap. Au cours de la phase en ligne, la station mobile (par exemple un smartphone) effectue la découverte du réseau ainsi que des mesures RSS en temps réel. Des algorithmes de positionnement différents sont ensuite appliqués afin d'identifier la meilleure correspondance entre l'empreinte digitale observée et la valeur moyenne respective des empreintes enregistrées pendant la phase hors ligne.

De nombreuses possibilités pour permettre la localisation en intérieur ont été étudiées en utilisant des algorithmes tels que K-Nearest Neighbor (KNN), Maximum A Posteriori (MAP) et Minimum Erreur carrée moyenne (MMSE). Pour notre application, en nous appuyant sur KNN, nous avons implémenté un nouvel algorithme de positionnement (baptisé i-KNN) capable de filtrer l'ensemble de données d'empreinte digitale initiale (c'est-à-dire la radiomap), après avoir considéré la

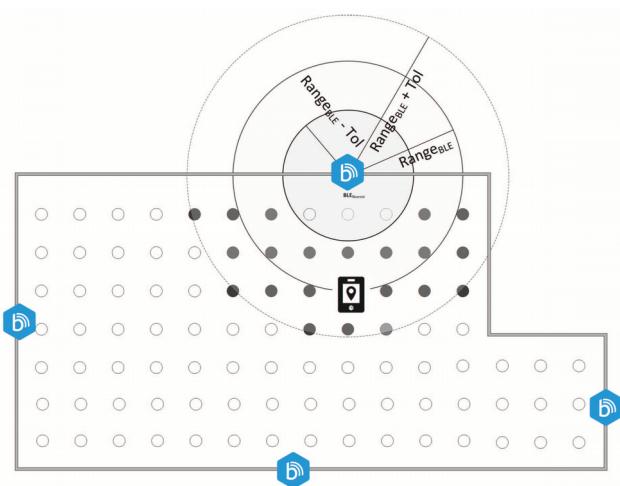


Illustration 8: Implémentation de i-Knn

proximité des empreintes numériques par rapport aux périphériques BLE. De cette façon, i-KNN fournit un petit sous-ensemble optimisé d'emplacements d'utilisateurs possibles, selon lequel il estime enfin la position de l'utilisateur. La méthodologie proposée atteint une estimation de positionnement rapide en raison de l'utilisation d'un fragment de l'ensemble de données d'empreinte digitale initiale, tout en améliorant la précision de positionnement en minimisant les erreurs de calcul.

Grâce à la puissance d'Unity, il a été relativement facile de récupérer les informations fournies par les beacons, un plugin a en effet été construit afin de gérer les protocoles beacons les plus connus (iBeacon pour Apple et Eddystone pour Google). Puis l'implémentation de l'algorithme d'apprentissage automatique k-NN amélioré a été réalisée en C#.

L'algorithme i-KNN a été testé en combinant des mesures et des simulations réelles. Pour accomplir cette tâche, 6 beacons ont été déployés dans un environnement intérieur d'environ 110 m². Chaque appareil était situé dans une pièce différente de l'environnement de test. Au cours des mesures, les empreintes numériques ont été collectées à 110 emplacements espacés (1 m d'espacement), à une hauteur constante de 90 cm. À chaque point de mesure, 30 échantillons RSS (1 échantillon / seconde) ont été enregistrés à l'aide d'une application développée pour les smartphones Android.

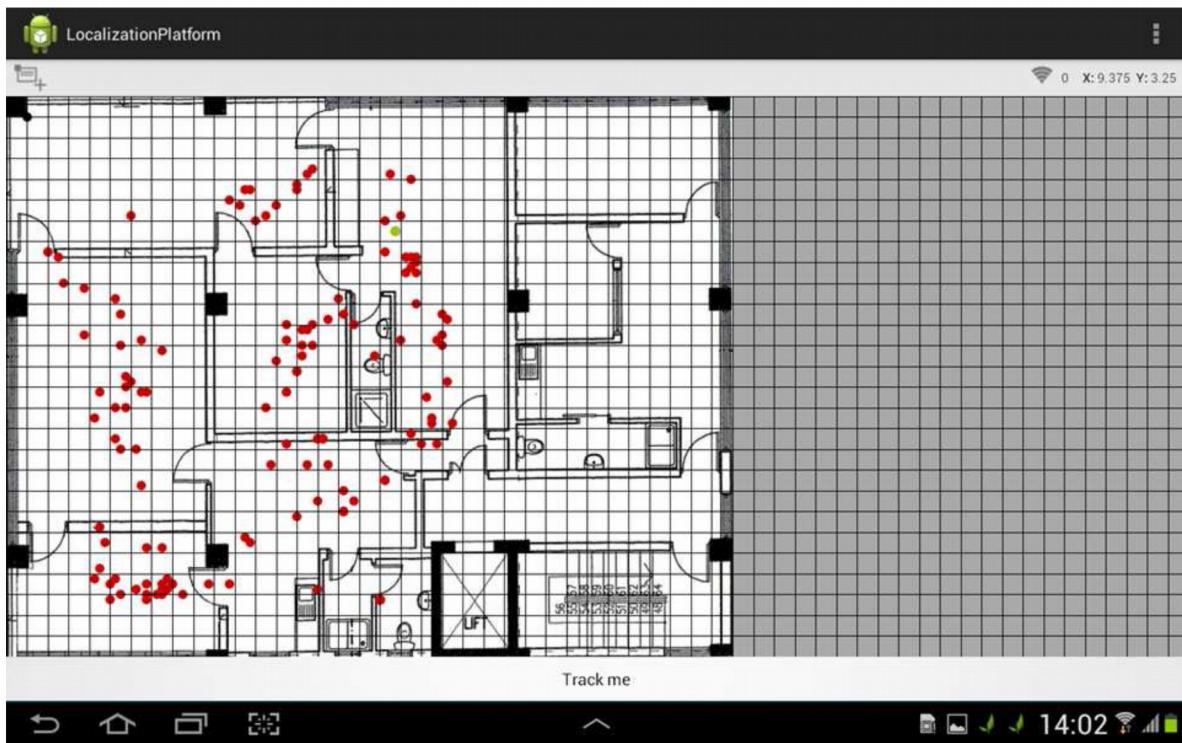


Illustration 9: Capture d'écran d'une application Android pour localisation intérieur à base de Beacon

Le prototype est un succès et une nouvelle implémentation à grande échelle est prévue.

Live streaming 360 Ambisonic

La société Phonomatik Lab avait maintenant besoin d'une nouvelle application. Le prototype devait pouvoir récupérer un flux ambisonic ainsi qu'une vidéo 360 et d'en synchroniser la lecture en direct. L'application pour Android devra aussi être compatible avec les lunettes Gear VR créé par Samsung et Oculus afin d'améliorer le confort de lecture.

Un plan d'action a été mis en place, avec les composants principaux suivants :

Matériels

- Core Sound TetraMic, un micro permettant d'enregistrer du son ambisonic
- Insta360 Pro, une caméra capable de streaming live d'un flux vidéo 360 via le protocole RTMP

Logiciels

- Une Station AudioNumérique (DAW : Digital Audio Workstation).
- Soundflower pour passer le flux audio entre les applications.
- NGINX avec le module RTMP pour servir de proxy RTMP.
- FFmpeg pour transcoder l'audio et la vidéo.

NGINX agit comme un relais RTMP et appelle FFmpeg qui mélange l'audio ambisonic avec la vidéo, puis renvoie le flux RTMP à notre application.

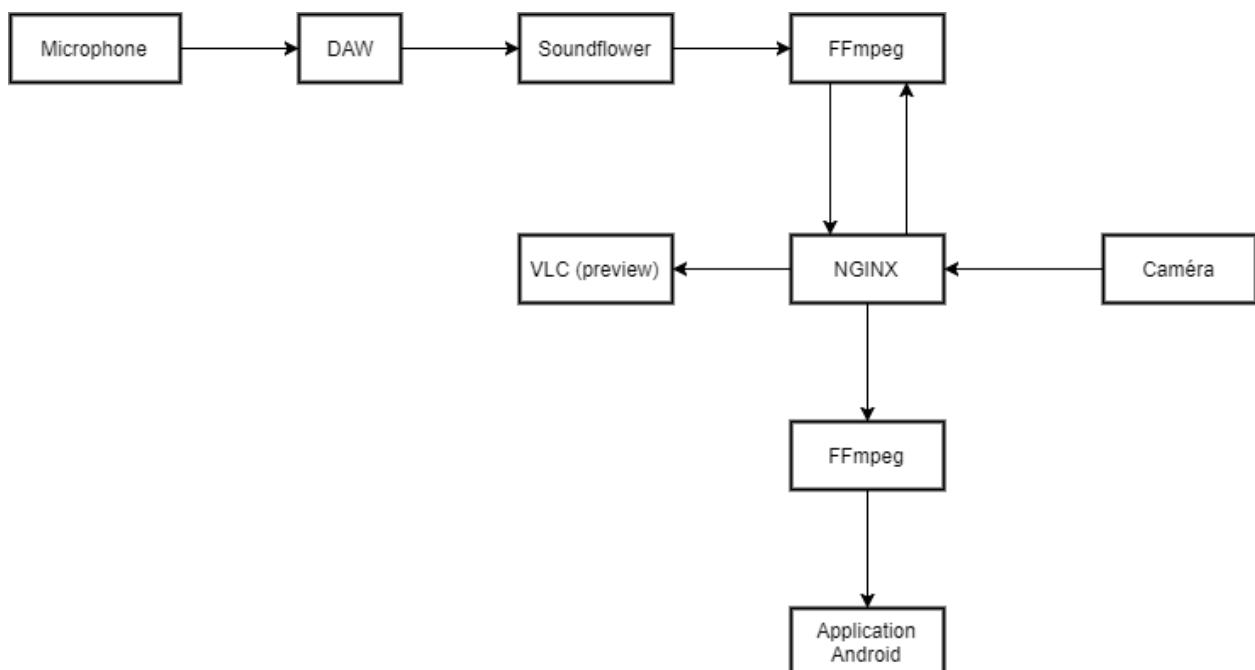


Illustration 10: Workflow du projet streaming live 360 ambisonic

Microphone → DAW

Récupérer le flux audio du microphone dans la station audionumérique a été effectué par PhonomatikLab. L'application prend en charge le format ambiX de premier ordre (4 canaux en ordre ACN avec la normalisation SN3D).

DAW → Soundflower

Le signal ambisonic doit être acheminé vers Soundflower où FFmpeg peut le lire. La gestion de Soundflower a aussi été livré par PhonomatikLab.

La vidéo de l'appareil photo et l'audio du microphone ne sont pas synchronisés; cela doit se faire manuellement. Si les tampons sont suffisamment grands, la synchronisation est stable une fois établie, mais elle doit être rétablie pour chaque flux en direct. Pour paramétrier les délais de façon précise, nous avons utilisé Reaper JS.

Camera → NGINX

NGINX est le serveur RTMP installé sur un ordinateur distance que j'ai configuré. Il appelle FFmpeg pour le transcodage, il est aussi extrait par VLC pour l'aperçu et FFmpeg pour le streaming sur Facebook.

Une fois que NGINX est en cours d'exécution, il faut lancer la diffusion RTMP sur l'insta360 Pro vers l'adresse IP du serveur prévu à cet effet, accompagné d'un nom de session afin de pouvoir lancer plusieurs flux en même temps. Le serveur a aussi la possibilité d'enregistrer les sessions grâce à un script bash.

NGINX → VLC

Il est maintenant possible d'ouvrir le flux RTMP provenant du serveur sur VLC ou tout autre logiciel capable d'accueillir le protocole.

NGINX → Application Android

L'application Android se charge de récupérer le flux RTMP en direct, puis le joue sur Android. Pour le moment l'ambisonic n'est supporté que sur l'application pour bureau à l'aide d'un HTC Vive, un autre casque pour la réalité virtuelle qui fonctionne avec Windows.

Les premiers tests effectués lors d'un concert à Marseille ont été fructueux sur un navigateur internet fonctionnant sous Windows. Il est prévu de revenir vers ce prototype pour intégrer la plateforme Android.

Application LesJounéesMonSabert2017

L'application LJM2017 est le seul projet tout au long de mon stage où toute l'équipe a été réunis afin de travailler ensemble. L'équipe a donc unis les technologies maîtrisés par les différents membre pour aboutir à un résultat dans un délais d'un mois.

Le schéma suivant représente les différentes parties de l'application.

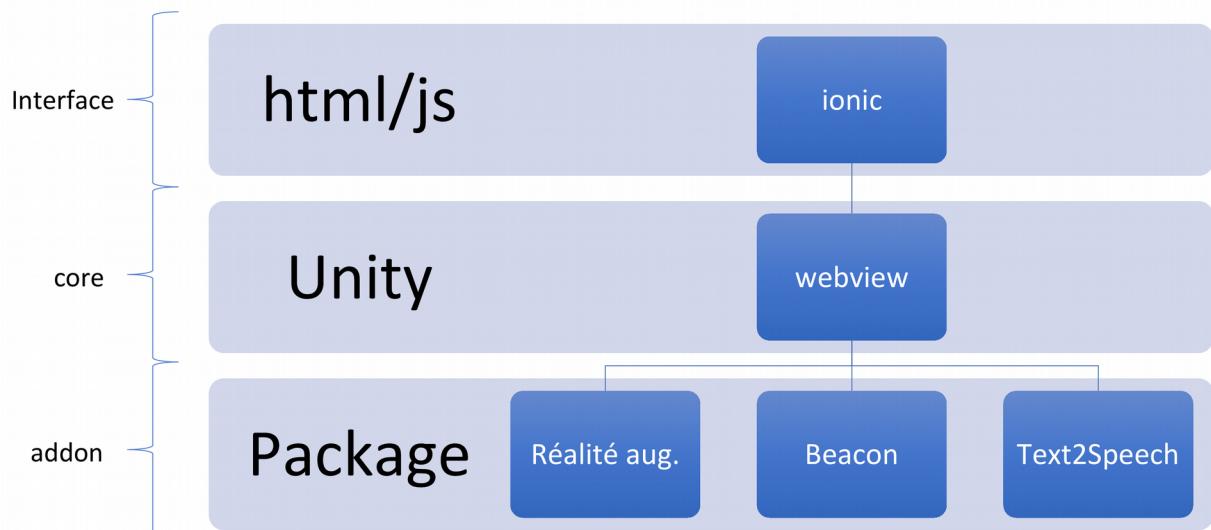
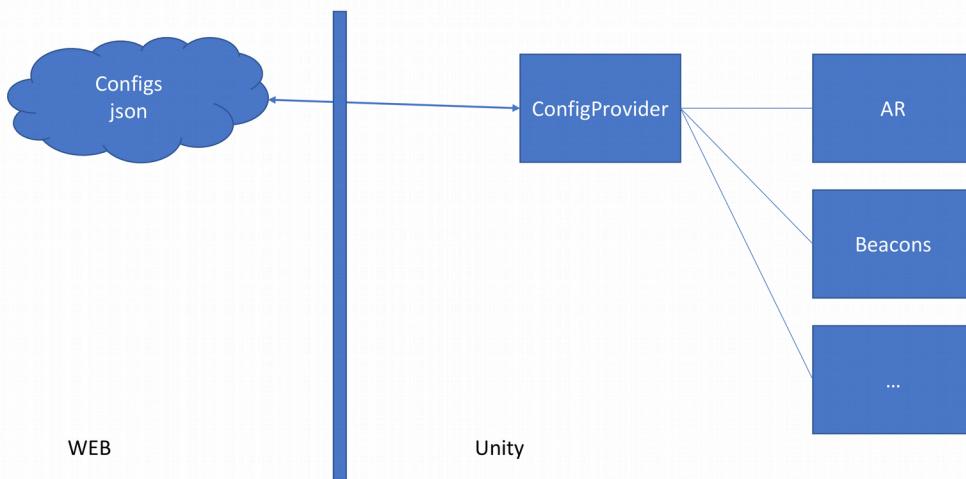


Illustration 11: Architecture de l'application Monsabert

Après ma monté en connaissance avec Unity et la technologie beacon sur mes précédents projets, il était naturel que je m'occupe de ces parties. Il a donc fallu concevoir un gestionnaire de scène permettant de passer de la partie de mon collègue en HTML/JS/CSS à mon application développé avec Unity. Une intégration grâce à Ionic et la webview disponible dans Unity permettant d'afficher une page web a été réalisé en premier lieu, pour attester du bon fonctionnement du squelette.

Puis les parties secondaires ont été développé, la technologie beacon qui n'avait été porté que sur Android jusque là nous a causé des difficultés sur le système d'exploitation iOS. La détection des beacon était aléatoire, beaucoup trop pour espérer porter notre prototype de localisation dans l'application. Il a donc été décidé d'abandonner cette partie.

Une autre contrainte présente lors de l'élaboration de l'application était le manque d'information du client. En effet l'application est sensé présenter une carte des différents stands installés dans le fort Ganteaume pour l'évènement. Une carte 3D a donc été réalisé par un camarade avec des marqueurs permettant de retrouver les différents points d'intérêts.



Pour pouvoir placer correctement les marqueurs avec des informations qui peuvent changer à tout moment, il a fallu prévoir une configuration sur un serveur, qui actualise le contenu de l'application.

Pour cela un serveur HTTP simple a été mis en place, sur lequel on pouvait récupérer une configuration JSON. Il a donc fallu récupérer le JSON grâce au C#, puis donner les bonnes informations de configuration aux différents éléments de l'application.

Git a été utilisé lors de ce projet, le répertoire git c'est vu découpé en plusieurs partie : le projet master, qui englobe

la totalité de l'application, et contient les gestionnaires de configuration et de scène. Les sous-répertoires représentent les parties annexes qui sont raccordé à l'application.

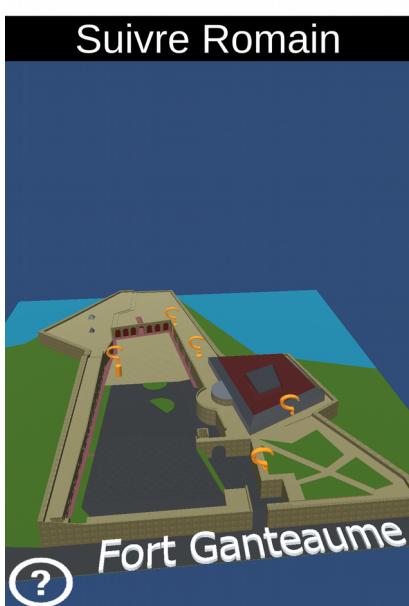


Illustration 13: Capture d'écran de l'application LJM2017



Illustration 12: Capture d'écran de l'application LJM2017

L'utilisation de git avec Unity c'est assez mal passé puisque aucun membre de l'équipe n'avait d'expérience avec son utilisation. Malgré les connaissances de chacun envers git, il a fallu une grande précision dans les méthodes de déploiement pour arriver à un workflow stable, qui peut être établie en temps raisonnable.

L'application est disponible sur le Google Play et Apple Store.

Fauteuil de simulation pour la réalité virtuelle

Le dernier projet de mon stage chez Nablaware a été la création d'un fauteuil de simulation pour la réalité virtuelle.

Un simulateur de mouvement est un mécanisme qui encapsule les occupants et a pour vocation de simuler l'effet / les sensations d'être dans un véhicule en mouvement. Le mouvement est synchrone avec l'affichage visuel et il est conçu pour ajouter un élément à la réalité virtuelle. Lorsque le mouvement est appliqué et synchronisé aux signaux audio et vidéo, le résultat est une simulation complète de la vue, du son et du toucher. Le simulateur de mouvement déplacent l'ensemble du compartiment occupant et peut transmettre des changements d'orientation et l'effet de fausses forces gravitationnelles.

Ces indices de mouvement trompe l'esprit en pensant qu'il est immergé dans l'environnement simulé et subit des changements dans la position, la vitesse et l'accélération. L'incapacité de l'esprit à accepter l'expérience peut entraîner un mal de mouvement lorsque la calibration est mauvaise. Les plates-formes de mouvement peuvent fournir des mouvements jusqu'à six degrés de liberté (DOF) trois degrés de liberté de rotation (roulis, tangage, lacet) et trois degrés de liberté translationnels ou linéaires (cavalement, l'embardée, le pilonnement).

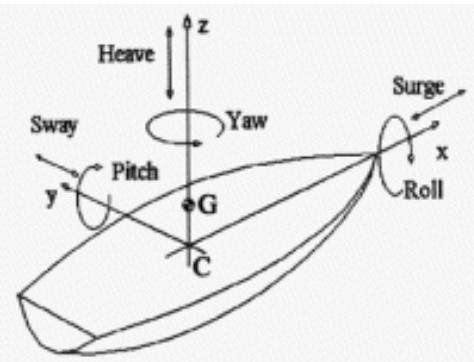


Illustration 14: Les 6 degrés de liberté

Le siège devait respecter des règles simples : tout d'abord respecter un certain budget, ce qui a réduit le champ des recherches. Puis le fauteuil devait s'intégrer parfaitement avec la VR, et proposer un contenu pour tout type de public.

Le fauteuil étant une technologie ludique et divertissante, il est l'élément phare de Nablaware pour l'exposition de l'armée de terre. Il a donc été décidé de monter une vidéo à l'aide des films en 360 que j'ai tourné auprès de l'armée, durant plusieurs déplacements. La vidéo ainsi créée doit être incorporé dans le HTC Vive, puis une application doit relier le fauteuil et la vidéo présente dans le casque.

Ayant déjà travaillé sur de la VR à l'aide d'Unity, il était évident d'utiliser une fois de plus cet outils afin de réaliser notre application.

Le fauteuil est contrôlé par un Arduino qui contrôle à son tour des puces électroniques qui font offices de driver pour les moteurs. Nous sommes passé par un logiciel SimTools pour envoyer des commandes à l'Arduino pour avoir un délai de livraison plus court, car le logiciel supporte déjà les 6 degrés de liberté.

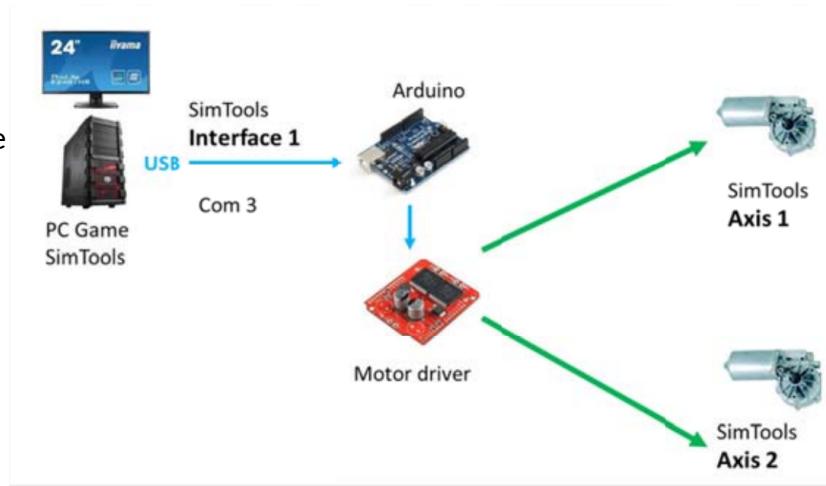


Illustration 15: Fonctionnement de SimTools

Afin de communiquer avec Simtools, il a fallu utiliser le protocole UDP, à l'aide du C# et d'Unity, j'ai créé un programme qui permet de lancer une vidéo sur l'HTC Vive, puis, en fonction de la frame en cours, d'envoyer des paquets UDP à SimTools afin que le fauteuil bouge en fonction de ce qu'il se passe dans la vidéo.

Pour cela j'ai d'abord créé un programme d'enregistrement qui permet, grâce à un joystick ou une souris, de déplacer le siège en même temps que la vidéo s'affiche à l'écran. Puis, ce programme convertit les mouvements de l'objet à l'intérieur d'Unity en degré de liberté. Enfin il permet de lire en synchronisation avec la vidéo les enregistrements précédents.

Lors du montage de la vidéo qui c'est fait en même temps que la création du fauteuil, il a été décidé d'incorporer une scène filmé au dessus d'un tank. Suite à cette décision, nous avons constaté que les 3 moteurs présents dans le fauteuil n'était pas suffisant pour une immersion totale. Puisque un char vibre constamment, on m'a confié la tâche de rajouter à notre installation un système de vibration.

Les délais étant de plus en plus courts, nous avons eu recours à un autre logiciel : SimVibe, qui permet de filtrer les basses fréquences d'une bande son pour les envoyer dans des « bass shaker ». Grâce à Unity j'ai réussi à créer une sinusoïde à basse fréquence redirigé vers une deuxième carte son (la première étant relié à l'HTC Vive qui passe la musique du clip). Ainsi le fauteuil peut maintenant vibrer et l'immersion est totale.

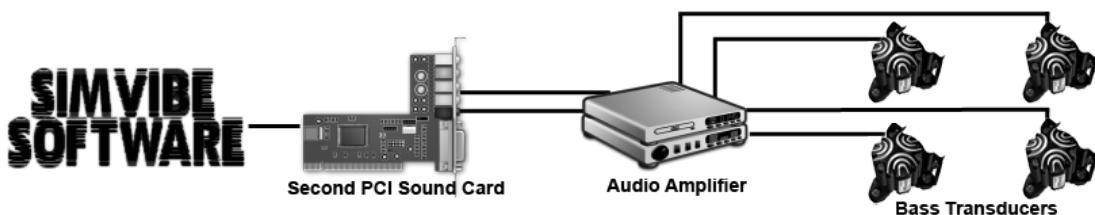


Illustration 16: Fonctionnement de SimVibe

Conclusion

Travaillé chez Nablaware a été pour moi un grand pas vers le monde professionnel, non seulement par l'acquisition de connaissance, avec des technologies comme la réalité virtuelle et la réalité augmenté, mais aussi un mode de travail et une ambiance qui sont radicalement différentes de ce que l'on peut retrouver dans une faculté.

L'équipe a su m'intégrer au sein de son organisation à travers les différents projets que nous avons réalisé.

La diversité de l'équipe de travail m'a permis de constater la palette de connaissance à avoir pour mener un projet à bien, et malgré l'apport de la faculté dans cette vision, j'avais du mal à comprendre le fonctionnement interne d'une entreprise.

Les connaissances acquises tout au long de mon master prennent sens maintenant, comme les pièces d'un puzzle que l'on assemble, au fil des années une compréhension du monde de l'informatique est née.

Au cours de mon stage j'ai su mettre en place les enseignements proposés durant mes cinq années de fac, et il ne s'agit pas de mettre en relation directement les connaissances apprises en classe et le travail demandé, il s'agit plutôt de gagner une capacité à analyser une situation puis agir en conséquence, tout en respectant des délais. Il faut donc agir avec précautions mais aussi avec méthode, pour ne pas se retrouver bloquer dans une voie qui ne fonctionne pas.

Une bonne cohésion au sein de l'équipe a été la clef déterminante quant à la réussite de mon stage, un dialogue simple mais efficace permet de réaliser un projet en minimisant les risques et en maximisant la qualité.

Je tiens à remercier tout mes professeurs durant ces 5 années de facultés qui ont été très enrichissante, d'un point de vue relationnel et professionnel.

Je remercie aussi l'intégralité de l'équipe Nablaware qui a su tirer parti de mes précédents enseignements tout en me donnant envie d'avancer dans ma voie.

Sources

<http://simxperience.com/Products/SimVibe/SimVibeSoftware.aspx>

<https://www.xsimulator.net/co-ordinate-system/>

<https://www.xsimulator.net/simtools-quickstart/>

<https://www.xsimulator.net/community/faq/which-way-to-set-simtool-axis-movements.230/>

https://fr.wikipedia.org/wiki/Station_audionum%C3%A9rique

<https://facebookincubator.github.io/facebook-360-spatial-workstation/KB.html>

<http://www.mdpi.com/1424-8220/17/4/812/htm>

<https://www.ncbi.nlm.nih.gov/pubmed/28394268>

<https://fr.shop.gopro.com/EMEA/virtualreality/omni---all-inclusive/MHDHX-006-EU.html>

<http://www.kolor.com/360-videos/>

http://www.kolor.com/wiki-en/action/view/Autopano - Panorama_Editor - Control_Points_Editor

https://fr.wikipedia.org/wiki/Valeur_efficace

<http://www.realite-virtuelle.com/insta360-pro-vr-vente-2104>

Rapport Technique

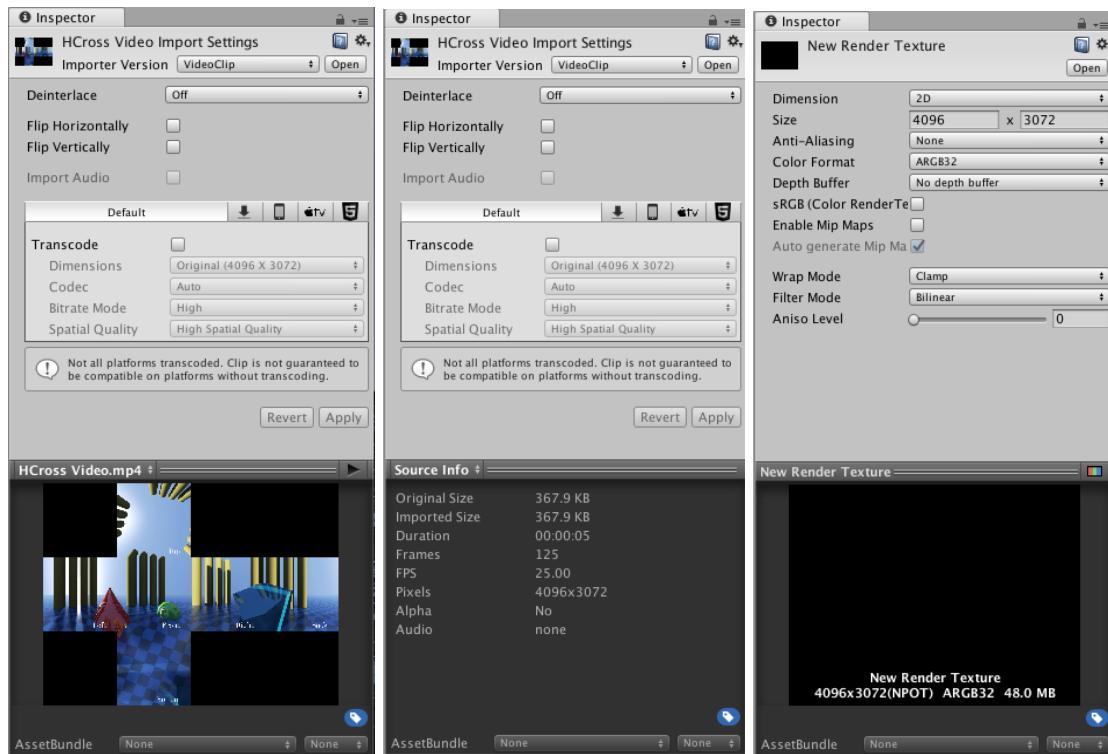
Vidéo panoramique 2D et 3D dans Unity

Grâce à un shader développé, Unity prend en charge des vidéos de 180 et 360 degrés dans des configurations équirectangulaires (longitude latitude) ou Cubemap (6 images).

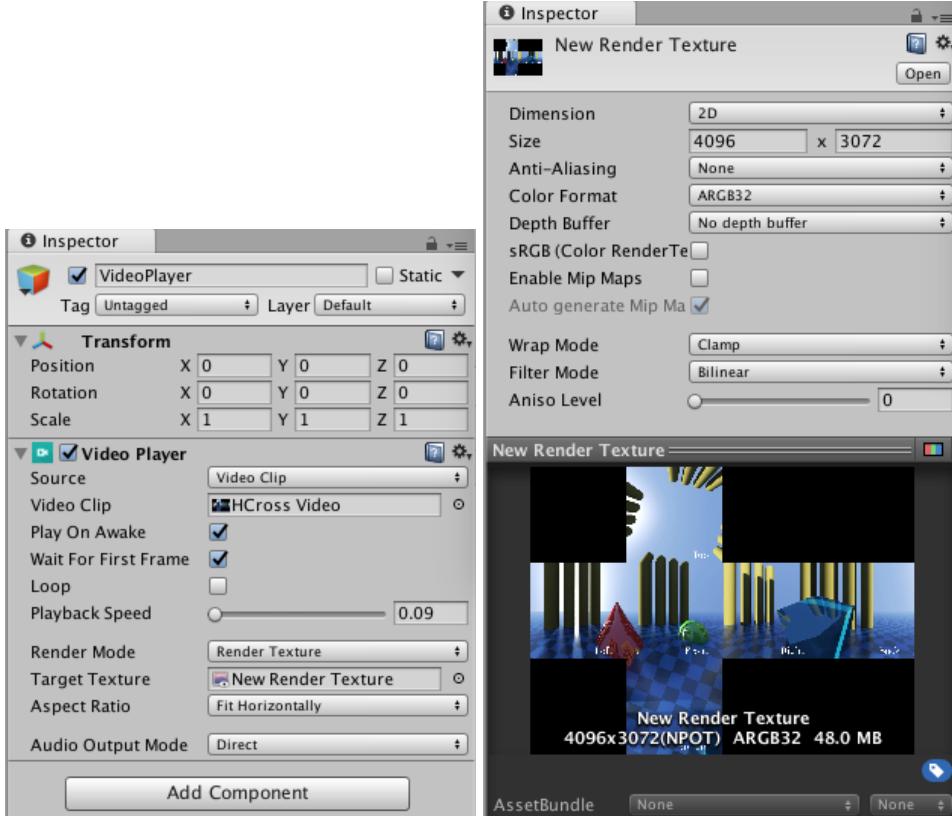
Conceptuellement, l'affichage de la vidéo panoramique dans Unity peut être réalisé en 3 étapes indépendamment du type de vidéo panoramique.

Un lecteur vidéo doit être configuré pour lire la source vidéo sur une texture de rendu. Un matériau Skybox recevra le RenderTexture. La scène doit être réglé pour utiliser le matériel Skybox.

Une fois que la vidéo a été importée dans Unity, un lecteur vidéo peut être créé. Par défaut, cela configurera le composant pour lire la vidéo en plein écran pour la caméra par défaut. Nous modifierons ce comportement afin qu'il soit appliqué à une texture de rendu afin que nous puissions contrôler exactement la façon dont il s'affiche. Pour ce faire, il faut d'abord créer une texture. Il faut ensuite définir la taille de la texture de rendu pour correspondre à celle de la vidéo.



Maintenant, dans l'inspecteur du lecteur vidéo, il faut activer le mode rendu pour rendre la texture. Il est maintenant possible de vérifier que cela fonctionne correctement en entrant en mode Play. Rien ne sera rendu à la vue de la séquence à ce stade, mais en sélectionnant l'élément Render Texture, le contenu est mis à jour avec les images vidéo.



Afin de rendre la vidéo panoramique comme toile de fond de notre scène, nous remplacerons le Skybox par défaut avec le contenu vidéo. Pour ce faire, il faut créer un nouveau matériel. Ensuite il faut régler le Shader du matériel sur Skybox / PanoramicBeta. Pour que la vidéo panoramique soit correctement affichée, il faut identifier correctement le type de contenu dans la vidéo.

Une fois que tout est correctement configuré, un aperçu est disponible en bas de l'inspecteur des matériaux.

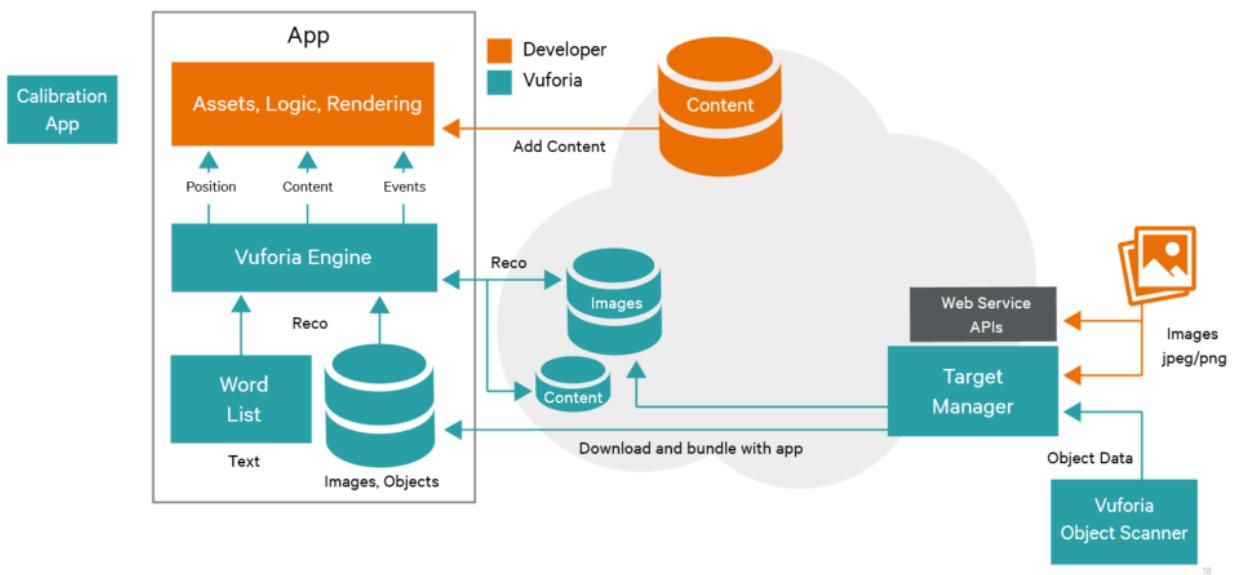
La dernière étape consiste à connecter le matériel Skybox créé à la scène. À ce stade, en appuyant sur Play, la vidéo 360 est joué en tant que scène sur Skybox. La modification de l'orientation de la caméra de la scène montrera une partie différente de la Skybox et donc une partie différente de la vidéo panoramique.

Pour le format Cubemap (6 Frame Layout), les vidéos 2D devraient avoir une échelle 1: 6, 3: 4, 4: 3 ou 6: 1 selon la disposition des faces. Les vidéos 2D de type equirectangulaire (latitude longitude) devraient avoir une échelle d'exactement 2: 1 pour un contenu 360 ou 1: 1 pour un contenu 180. L'inclusion de contenu 3D nécessitera de doubler soit la largeur, soit la hauteur de la vidéo.

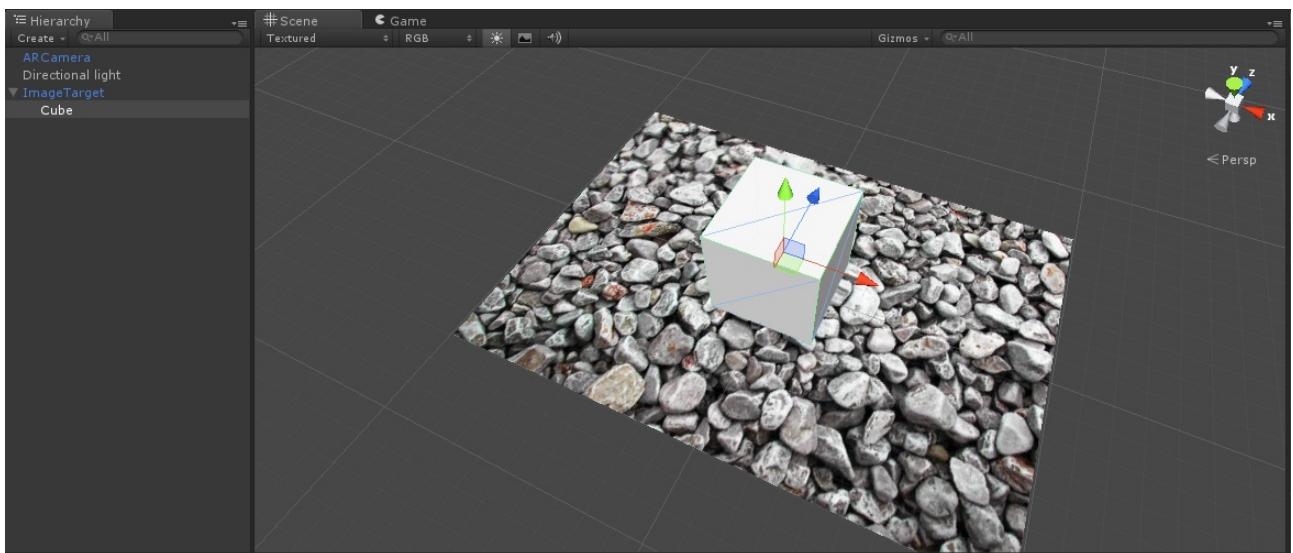
Développement d'applications Vuforia

L'extension de Vuforia pour Unity permet de créer des applications pour Android et iOS en utilisant un simple flux de production de création de glisser-déplacer dans Unity.

Pour utiliser Vuforia il faut créer une base de donnée sur leur site permettant de transformer des images classiques en marqueur pour que le SDK de réalité augmenté détecte l'image.

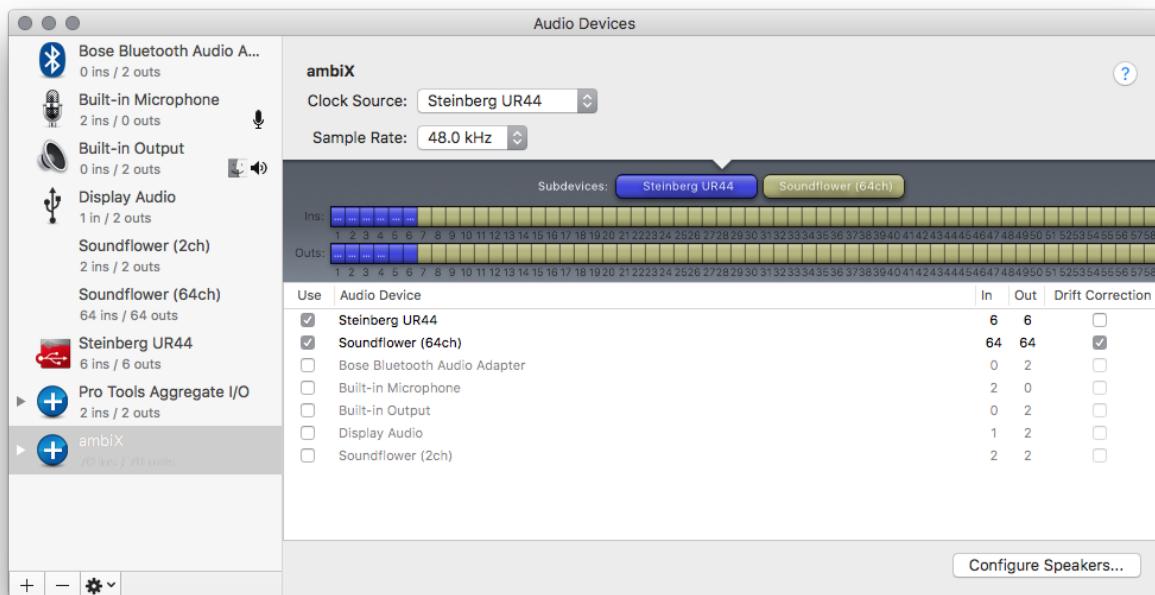


Une fois dans Unity il est possible d'utiliser une forme géométrique comme surface au marqueur, puis d'ajouter n'importe quel type d'objet par dessus afin qu'il apparaisse lorsque le marqueur est détecté par un appareil.



Complément sur la prise de son Ambisonic

Il faut acheminer les 4 canaux de l'audio ambiX en direct vers les quatre premiers canaux de "Soundflower (64ch)". Dans Reaper, cela se fait dans la boîte de dialogue de routage de la piste en ajoutant une sortie de matériel audio pour le premier canal de Soundflower (64ch), avec 4 canaux de sortie.



Nous pouvons surveiller le microphone avec de l'audio spatial en ajoutant un plugin à une autre piste et en utilisant le rendu binaural. Il suffit de régler l'entrée dans le plugin Spatialiser sur ambiX et s'assurer que "decode binaural" est sélectionné dans le plugin Control.

Configuration du serveur NGINX

```
# nginx clears environment, except what we pass through. Pass through PATH  
env PATH;  
  
# nginx insists on this existing  
events {  
}  
  
rtmp {  
    # kill ffmpeg gracefully  
    exec_kill_signal term;  
  
    interleave on;  
    wait_key on;  
  
    # You can play with this. Bigger is safer, smaller is "more live".  
    Besides  
        # the usual risk of dropped packets, if the buffer is too small  
    audio and  
        # video may get out of sync.  
        buflen 2s;  
  
    server {  
        # Default RTMP port  
        listen 1935;  
  
        application live {  
            live on;  
  
            # This script transcodes and mixes in the Soundflower  
            audio.  
            # It should be on your path.  
            exec transcode.sh rtmp://localhost/$app/$name  
            rtmp://localhost/landing/$name;  
        }  
  
        # For preview/synchronization, point VLC at this  
        (rtmp://localhost/landing/$name).  
        # Then run fblive.sh with the destination stream URL and $name  
        application landing {  
            live on;  
            record_path /tmp/rec;  
            record_unique on;  
            #record all;  
        }  
    }  
}
```

Fichier bash permettant le transcodage d'une vidéo

```
#!/bin/bash

# This script is executed by nginx. It must be in your PATH (as seen
by nginx).

SPADLIVE_LOG=${SPADLIVE_LOG:-/usr/local/var/log/spadlive.log}

video_kbps=2000
video_height=1080
audio_kbps=256
buffer_secs=3

# rtmp stream to pull
src=$1
test -n "$src" || exit 1
# rtmp stream to push
dst=$2
test -n "$dst" || exit 2

# mark the time
date >> "$SPADLIVE_LOG"

# audio device names
ur44='Steinberg UR44'
soundflower64='Soundflower (64ch)'

exec ffmpeg -nostats \
-i "$src" \
-f avfoundation -i "none:$soundflower64" \
-map_channel 1.0.0 \
-map_channel 1.0.1 \
-map_channel 1.0.2 \
-map_channel 1.0.3 \
-map 0:v,1:a -map 1:a \
-c:v h264 -preset fast -maxrate ${video_kbps}k -bufsize $[
$buffer_secs * $video_kbps ]k -vf scale=-2:$video_height -r 30 -g
60 \
-threads 4 \
-c:a aac -b:a ${audio_kbps}k \
-copyts -start_at_zero -af aresample=48000:async=1 \
-f flv "$dst" \
>>"$SPADLIVE_LOG" 2>&1
```

Complément sur l'algorithme de localisation à partir de signal RSSI

Les algorithmes de positionnement récupèrent le RSS à l'emplacement de l'utilisateur et mettent en œuvre des méthodes déterministes ou probabilistes pour estimer l'emplacement réel de l'utilisateur. Le concept qui se cache derrière ces deux méthodologies est courant; en recherchant une base de données d'empreintes numériques et en identifiant une ou plusieurs positions dont la signature RSS a la plus grande similitude avec celle observée. Plus précisément, les méthodes de positionnement déterministes estiment la localisation $\hat{\ell}$ comme une combinaison convexe des sites de référence K avec la distance la plus courte entre r_i et s dans l'espace n-dimensionnel. La déclaration susmentionnée est mathématiquement exprimée par l'équation suivante:

$$\hat{\ell} = \sum_{i=1}^K \left(\frac{w_i}{\sum_{j=1}^K w_j} \ell'_i \right).$$

L'ensemble $\{\ell'_1, \dots, \ell'_l\}$ désigne l'ordre des emplacements de référence par rapport à la distance croissante D_i , qui est mesurée entre l'empreinte numérique dans la base de données respective r et la mesure observée lors du positionnement s , c'est-à-dire $D_i = \|r - s\|$. La distance peut être calculée en utilisant des normes standard, telles que Manhattan (1-norme), la norme euclidienne (2-norme) ou Mahalanobis. En se concentrant sur la distance euclidienne, D_i peut s'exprimer par l'équation suivante:

$$D_i = \sqrt{\sum_{j=1}^N (\bar{r}_{ij} - s_j)^2}.$$

Le coefficient de pondération non négatif w_i dans l'équation représente une valeur qui peut être attribuée à chaque emplacement de référence dans la radiomap et différencie son poids, d'où son importance par rapport aux autres empreintes numériques. En d'autres termes, la valeur de ce coefficient peut varier d'une manière telle que chaque empreinte influence différemment l'estimation de positionnement. Dans un cas d'allocation de poids, l'équation exprime l'algorithme pondéré K-voisin le plus proche (WKNN). Une valeur typique pour w_i peut être l'inverse de $\|r-s\|$. En simplifiant l'algorithme précité, on peut supposer que des poids égaux sont attribués à toutes les empreintes digitales utilisées. Une telle supposition entraîne l'élimination de w_i et l'équation est convertie en la méthode K-Nearest Neighbor (KNN). Enfin, en définissant $K = 1$, la formule conduit à la méthode du voisin le plus proche (NN). Selon des études scientifiques, les méthodes KNN et WKNN offrent une précision de positionnement plus élevée par rapport au NN, pour $K = 3$ et $K = 4$. D'autre part, NN semble fonctionner de manière satisfaisante et fournit également de bons résultats dans des scénarios avec des radiodiffusions RSS haute densité.

FIN