

RAPPORT TECHNIQUE DE PROJET DE FIN D'ÉTUDES
PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
DANS LE CADRE DU COURS GTI795 PROJET DE FIN D'ÉTUDES EN TI

**DÉTECTION DE MODÈLE HUMAIN
POUR APPLICATION DE RÉALITÉ AUGMENTÉE**

Vincent BOITEAU-ROBERT
BOIV14029409
Marc-Antoine HÉBERT
HEBM14019401
Julien LEMONDE
LEMJ20059208
Alexandre MALO
MALA0702940

DÉPARTEMENT DE GÉNIE LOGICIEL ET TI

**Professeur-superviseur
Carlos Vazquez**

MONTREAL, 25 AVRIL 2018
HIVER 2018
UNIVERSITÉ DU QUÉBEC

Résumé

RÉSUMÉ

DÉTECTION DE MODÈLE HUMAIN POUR APPLICATION DE RÉALITÉ AUGMENTÉE

Vincent BOITEAU-ROBERT

BOIV14029409

Marc-Antoine HÉBERT

HEBM14019401

Julien LEMONDE

LEMJ20059208

Alexandre MALO

MALA0702940

Le projet consiste à détecter un être humain à l'aide d'une caméra 3D afin d'être en mesure d'aller récupérer la texture et la profondeur. Pour ensuite convertir l'humain détecté en modèle 3D. Une fois le modèle bâti, le projet consiste aussi à intégrer le modèle humain dans une application de réalité augmentée. Celle-ci repose sur framework ARKit de Apple. Le tout, encapsuler dans une seule et même application unique.

Ce document contient une introduction sur le sujet ainsi qu'une revue de la documentation utilisé pour la réalisation de ce projet. De plus, la méthodologie de travail utilisé et le processus de conception y sont documentés. Le tout se termine avec une discussion et une conclusion.

Table des matières

INTRODUCTION	1
REVUE DE LA DOCUMENTATION	4
Structure Sensor	4
MÉTHODOLOGIE DE TRAVAIL	7
Planification	7
Jalon	8
Clôture	9
PROCESSUS DE CONCEPTION	10
DSCUSSION	14
CONCLUSION	15
BIBLIOGRAPHIE	16

Liste des tableaux

Table des figures

1 Diagramme de classes du Structure Sensor SDK 5

2 Diagramme des composantes de l’application 11

LISTE DES ABBRÉVIATIONS

ÉTS - École de Technologie Supérieure

AR - Augmented Reality ou en français Réalité Augmentée

SDK - Software Development Kit ou en français Trousse de développement logiciel

API - Application Programming Interface ou en français Interface de programmation

AV - audio-vidéo

INTRODUCTION

La performance des caméras des appareils mobiles est en constance croissance depuis quelques années. Ce qui permet d'envisager l'usage de ces appareils dans des nouveaux domaines d'activités. Un de ces domaines est la réalité augmentée, où la superposition d'éléments virtuels sur une capture de l'environnement réel en temps réel. Une démocratisation du développement d'application utilisant cette technologie s'est faite au cours de l'année précédente, 2017. En effet, la nouvelle interface de programmation applicative ARKit conçue par Apple offre un environnement de développement simplifié au développeur d'applications ([Statt, N., 2017](#)). Toutefois, ces technologies permettent l'interaction avec des modèles générés de façon synthétique. Un défi qui n'a pas encore de solution reconnue est l'usage de modèles provenant de l'environnement de l'utilisateur dans la réalité augmentée. Certaines technologies offrent des pistes de solution, mais leur taux d'adoption et le niveau de compatibilité avec les autres outils de l'environnement de développement Apple est limité.

Une des technologies les plus intéressantes est le Structure Sensor([Molitch-Hou, M., 2016](#)). Celle-ci consiste en un périphérique qui doit être attaché à l'appareil. Le périphérique ajoute une caméra et un capteur 3D. Avec l'usage de la caméra de l'appareil, celui de l'appareil peut avoir une meilleure compréhension de l'environnement 3D. L'outil vient avec une trousse de développement logiciel permettant de prendre les coordonnées d'un objet sous quatre dimensions, x, y, z et la couleur. Toutefois cette solution n'est pas très connue et a un nombre de ressources limitées.

Les deux technologies décrites plus haut apportent chacune des possibilités très intéressantes pour le développement d'application de réalité augmentée. L'usage des fonctionnalités d'une dans l'autre serait très intéressant. Toutefois au moment où l'équipe a débuté le projet, il n'y a pas d'interface une telle combinaison. Les deux interfaces utilisent la caméra, par contre chacun utilise un utilitaire différent pour le contrôle de celle-ci. Les deux utilisent des modèles numériques 3D, mais chacun a choisi une norme différente. En effet, le Structure Sensor permet l'extraction en Modèle I/O. Alors que le ARKit utilise les modèles de type scène qui est un format propriétaire à Apple. Le principal défi

du projet consistera à la combinaison de plusieurs interfaces de programmation pour créer un outil fonctionnel.

Comme décrit dans la section précédente, le principal objectif du projet est la création d'une interface entre deux technologies mobiles, soit un périphérique d'acquisition de modèle 3D, le Structure Sensor et la trousse de développement de réalité augmentée d'Apple. Autrement dit, l'équipe s'attend en fin de session d'avoir produit une application mobile permettant la prise de modèle à l'aide du capteur. L'application permettra d'ajouter un modèle extrait sur une surface plane. L'application permet cette chaîne d'action avec une expérience utilisateur agréable et naturelle à l'utilisateur.

L'application décrite ci-haut représente les attentes pessimistes par rapport aux résultats finaux. Toutefois, une telle application apporte un éventail de possibilités sur la manipulation de l'environnement augmentée. Par exemple, une des retombées envisageables et commercialisables est la numérisation de catalogue de magasin comme Ikea. L'utilisateur peut donc ajouter un item dans une pièce de sa maison. Si plusieurs couleurs sont disponibles pour un modèle, chacune des variantes sera disponible à la modélisation.

Une autre ouverture technologique, qu'une telle application apporterait, est dans un contexte d'animation 3D. En effet, la création d'un modèle permettrait l'ajout d'un squelette dans celui-ci. Une fois le squelette créé l'animation du modèle pourrait être faite. Cette possibilité est particulièrement intéressante pour l'équipe responsable du projet. Dans l'éventualité où la vision de base du projet sera complétée avant la fin de la session d'Hiver 2018, l'équipe compte se concentrer sur la création de ce squelette et possiblement la manipulation de celui-ci. La création de positionnement clé des noeuds du squelette pourra être enregistrée afin de permettre l'animation. L'animation libre est un trop grand défi étant donné les contraintes de temps.

L'objectif principal semble réaliste pour l'équipe de développement étant donné les interfaces déjà présentes pour chacune des technologies. Toutefois, le défi reste important étant donné le manque de cohérence entre les différentes technologies. Le manque de documentation et de support pour le Structure Sensor risque d'amener un ensemble de défi supplémentaire.

Par contre, l'équipe de développement considère le temps disponible à la résolution d'un tel enjeu adéquat. L'expérience en recherche d'information et apprentissage autonome acquise au cours des formations et stages des membres de l'équipe laisse envisager un succès.

Pour ce qui est en la complétion des objectifs accessoire, les limitations de temps rendent l'estimation de succès plus difficile. Par contre, il est clair que la réalisation de ceux-ci apporterait un plus considérable à l'expérience de chacun des membres. La motivation de l'équipe face à l'enjeu améliore les

probabilités de terminer ces objectifs.

Ce rapport consiste donc en un suivi des étapes qui permettront de valider ou invalider les hypothèses décrites si haut. Afin de résoudre un tel défi, l'équipe a dû se fier sur les documentations disponibles en lien avec les technologies ciblées. Un résumé de celles qui ont été utiles à la résolution du problème sera présenté.

Par la suite, une présentation de la méthodologie utilisée par l'équipe sera décrite. Les étapes ayant permis la conception de la solution seront par la suite présentées. Une analyse sur le produit obtenue sera explorée. Afin de clore le projet, les possibilités que la solution apporte à un groupe voulant poursuivre ou implémenter un projet semblable seront expliquées. L'équipe fera s'est recommandation en fonction de l'expérience acquise lors du développement.

REVUE DE LA DOCUMENTATION

Afin de réaliser le projet, l'équipe de développement a dû faire des recherches préliminaires. Les technologies utilisées n'étaient pas maîtrisées par l'équipe au démarrage du projet. Donc, les recueils de documentation et les projets exemples présents sur internet ont été une bonne source d'information.

Cette section présentera donc ces ressources pour les grandes parties du projet, soit le périphérique Structure Sensor, la trousse de développement logiciel de réalité augmentée de Apple et la gestion de projet.

Structure Sensor

Occipital, la compagnie propriétaire du périphérique, publie régulièrement une version de leur trousse de développement logiciel exposant des interfaces de programmation. La dernière version publiée par la compagnie est la version 0.7.1 en février 2018. ([Occipital, 2018](#)) La version du SDK étant inférieure à 1.0.0, on comprend que la première version complète est encore en développement.

Dans l'état actuel, le SDK fournit quand même une liste d'interface disponible dans deux fichiers, Structure et StructureSLAM. Le premier est composé des différentes interfaces de contrôle du périphérique. Alors que le deuxième est un regroupement des classes permettant le contrôle des mailles en position tridimensionnelles et les textures de ceux-ci.

Ces interfaces seront la base de la conception du module de prise de modèle grâce au périphérique. Les tâches de chacune des interfaces ont été identifiées lors de la lecture du code des entêtes des interfaces en Objective-C. L'équipe a produit un diagramme de classe afin d'améliorer la compréhension des responsabilités de chacune des interfaces et les relations entre celles-ci. Celui-ci est présenté à la page suivante.

Dans le diagramme, on constate que plusieurs patrons de conception ont été utilisés et exposés dans l'interface. La classe STSensorControllerDelegate servira de point d'ancrage entre notre application et le Structure Sensor. Cette interface permet donc de réduire les communications ou les recherches d'informations dans la logique d'affaires. La plupart de l'usage du SDK dans l'application devra être fait par cette classe. ([Best-Practice-Software-Engineering, 2013](#))

Le Structure Sensor utilise les DepthFrame et ColorFrame pour comprendre l'environnement dans lequel il est utilisé. Ces concepts ont été introduits par Kinect un des premiers Senseurs 3D ayant connu un succès commercial étant donné son intégration dans l'environnement Xbox 360 de Microsoft. (Microsoft, 2014)

Le Structure SDK contient aussi plusieurs exemples d'application afin de guider les développeurs dans le développement de leurs applications. Le plus intéressant pour l'objectif du projet est le Scanneur. Celui-ci permet de convertir un objet dans l'espace de test en un modèle 3D et l'envoyer par courriel. D'autres

fonctions incluses dans cet exemple sont la colorisation du modèle et les différentes options de capture. L'espace de capture est aussi indiqué par un cube et un plan utilisant les couleurs pour indiquer les objets qui peuvent être capturés.

Les principales sections du SDK utilisé dans cet exemple sont le STStructure-SensorDelegate pour le contrôle du périphérique. Les STDepthFrames et les STColorFrames ou en français les grilles de profondeurs et les grilles de couleurs permettant de créer le modèle STMesh. Une fois le modèle scanner en maille celui-ci est présenté dans le MeshViewController grâce au rendu de maille, ST-MeshRenderer. Le langage de programmation utilisé dans la globalité du SDK est le Objective-C.

Dans le but d'utiliser des technologies qui sont à jour avec les pratiques de l'industrie, l'équipe de développement souhaite utiliser le nouveau langage de programmation Swift. Celui-ci est le nouveau langage proposé par Apple pour le développement mobile. Le langage est déjà à sa quatrième version majeure et a pour but de simplifier la syntaxe qui était auparavant utilisée soit le Objective-C. ([Kremenek, T., 2017](#)) Un membre de la communauté de développeur Structure a produit un port de l'exemple de Scanneur en Swift 2. L'équipe devra donc faire les changements nécessaires à cette version pour la compiler avec la version 4. ([Worley, C., 2016](#))

L'exemple de scanneur qui sera utilisé comme base du module de scan utilise plusieurs autres APIs par Apple. Ces APIs sont documentées sur le site de Apple dans la section pour développeurs. ([Apple, 2018c](#)) L'équipe utilisera donc cette ressource en ligne pour accomplir la tâche.

MÉTHODOLOGIE DE TRAVAIL

La démarche entreprise par l'équipe de développement pour compléter le projet a été basée sur la combinaison des acquis obtenus dans les projets résolus en stage et au cours du processus universitaire. Toutefois la nature du projet de fin d'études comprend des contraintes et conditions qui ont dû être prises en compte.

Planification

Tout d'abord, aucun des membres de l'équipe n'avait des connaissances dans les technologies qui devaient être traitées dans les projets. Il a donc été nécessaire de faire une recherche préliminaire afin de gagner une compréhension nécessaire à la planification du projet. La recherche a permis de répondre à des enjeux sur lesquels étaient les principales parties du projet ? Quelles technologies étaient incluses implicitement au ARKit et Structure Sensor ? Quel environnement de développement devait être utilisé ? Est-ce que les objectifs étaient réalistes dans les contraintes de temps et de ressources ?

Suite à cette recherche, le projet ayant été jugé comme réalisable a été séparé en plusieurs modules. C'est à dire des parties du projet qui peuvent être développées indépendamment et être assignées à différents étudiants. Ayant conscience de la possibilité de blocages au cours du développement cette technique permet de ne pas limiter l'avancement du projet complet, dès qu'il y a un problème. Les différents modules identifiés sont : le développement du scanneur avec le Structure Sensor, le développement de l'environnement AR et le développement du squelette dans le modèle humain. Le dernier module a été identifié comme accessoire étant donné l'importance de l'enjeu. De plus, le développement des deux premiers modules et leur combinaison est déjà importante en taille et complexité.

Une fois les deux modules principaux identifiés, une séparation de l'équipe de développement a été faite. La séparation visait à permettre à chacun de se concentrer sur un type d'enjeu et de diminuer la grosseur des défis. De plus, cette méthode a permis de répondre à des contraintes qui entourait le

projet et les technologies disponibles. En effet, un seul périphérique Structure Sensor était disponible. De plus, trois développeurs sur quatre avaient accès à un appareil Apple mobile.

Une fois les sous-équipes identifiées, celles-ci ont dû séparer et lister les tâches à accomplir, les ressources documentaires disponibles autour de leur technologie et lister les risques que leurs technologies impliquaient. Ces informations ont été ajoutées au plan de projet avec la description du projet et la mise en situation. Les tâches ont été par la suite séparées en trois jalons de développement.

Pour le module de Structure Sensor, les trois jalons consiste en le transfert du projet Scanneur de Swift 2 à Swift 4 pour le premier. Pour le second, la cible était l'intégration dans la structure de l'application AR et le transfert du modèle dans l'application. Le troisième consistait en la conception d'une interface visuelle adéquate et une possibilité de faire un début de squelette sur le modèle humain.

Pour le module de réalité augmentée, les trois jalons sont la sélection d'un modèle dans une liste, l'ajout dans la scène et la suppression de celui-ci pour le premier. Pour le second, la modification à l'aide des gestes classiques d'éditions d'objets sur un appareil mobiles. Pour le troisième, la conception de l'interface visuelle et une possibilité de contrôle des poses du squelette humain.

Jalon

Chaque jalon représentant une partie du travail à faire contiendra toujours un certain nombre commun de regroupements de tâches. Pour commencer, avant d'entreprendre toute forme de travail, des recherches doivent être faites. Les éléments concernant les différentes méthodes de réalisation d'une tâche et les avantages et désavantages de chacune de celles-ci doivent être identifiés.

Par la suite, une solution doit être sélectionnée en comparant les avantages et désavantages. Le contexte de développement de l'application doit être aussi pris en cause. Une fois la solution choisie, l'adaptation de la conception de la solution doit être faite pour le cadre du projet. Si la solution s'intègre au sein d'une autre solution, les documents de conceptions de la solution parentes sont adaptés. La conception est donc souvent itérative à travers les jalons et les tâches.

Par la suite, la solution est intégrée dans l'application à l'aide de l'éditeur spécialisé XCode. La compilation et la mise en marche est testé sur un appareil mobile Apple au fur à mesure du développement afin d'éviter les mauvaises surprises en fin de jalon. Une fois une tâche terminée, le cas d'utilisations prin-

cipal de l'application est testé pour s'assurer qu'aucune autre fonctionnalité n'a été brisée.

Le jalon est par la suite terminé avec une revue du travail fait et avec une présentation préparée par l'équipe pour le superviseur du projet. Dans ces revues on discute du travail fait, des bloquants et du travail à faire. Les artéfacts de conception et documents de projets doivent aussi être mis à jour.

Clôture

Le développement applicatif terminer, certaines tâches doivent être faites afin de clore le projet. Parmi celles-ci est la revue des résultats obtenue par l'équipe de développement. Cette tâche a pour but d'évaluer l'état de l'application par rapport au résultat qui était attendu en début de projet. Dans le cas de différences entre les attentes et le produit final, un bilan de justification doit être tenu.

Par la suite, l'équipe devra lister ces recommandations qui pourraient servir à une autre équipe voulant poursuivre le projet. Le contexte de projet de fin d'études permet en effet la poursuite d'un tel projet. Alors un transfert de connaissance doit être préparé afin de limiter les frictions des futures parties prenantes au projet. Les deux dernières tâches permettront la préparation de la présentation qui sera faite en fin de projet devant d'autres étudiants finissants.

PROCESSUS DE CONCEPTION

La démarche décrite dans la dernière section permettra de structurer les tâches de conception exécutées par chaque membre de l'équipe. Ainsi qu'avec les informations obtenus dans la revue documentaire, une solution à l'enjeu soulevé en début de projet a été conçu. Les besoins comblés, le fonctionnement de la solution, les différentes solutions explorées et la solution retenue et prototypée seront présentés.

Les responsabilités que la solution développée devra couvrir consiste en : La prise de modèle dans l'environnement réel, la présentation de ces modèles dans l'environnement réel pris par la caméra, la compréhension des coordonnées de l'environnement capturé, la modifications du modèle et la reconnaissance des mouvements de l'appareil. La technologie Structure Sensor peut répondre au responsabilité de prise de modèle dans l'environnement et compréhension des coordonnées de l' environnement réel. Par contre, les autres responsabilités devront être résolues par d'autres moyens.

Deux possibilités se sont révélées comme intéressantes lors des recherches. La première consistant en le développement d'une solution personnalisée permettant la gestion du modèle déjà scanner. Cette gestion devrait permettre l'ajout du modèle dans un environnements contrôlées à l'aide des coordonnées acquise grâce au Structure Sensor. Une fois ajouté dans l'environnement le modèle doit pouvoir être déplacer, mis à l'échelle et être pivoter. Ce modèle devra pouvoir être supprimer de l'environnement. Toutes ces différentes tâches demandent donc un cadre de développement permettant la transformation des données provenant des périphérique de capture en scène tri-dimensionnelle avec lesquels les développeurs interagissent.

Un outil répondant à ce besoin a déjà été publié et il consiste en la deuxième option. Cet outil est la trousse de développement de Apple, le ARKit. Celui-ci fait le pont entre les interface du périphérique de capture et la trousse de gestion de scène SceneKit. Cette solution est très intéressante puisqu'elle diminue grandement le poid du travail à faire. Par contre, en ajoutant une interface avec une nouvelle frontière ont ajoute aussi des contraintes au développement. De plus, le ARKit permettant sont usage avec des appareil n'ayant pas de Structure Sensor représente des possibilités fonctionnelles plus limité. Un exemple

de ces limitations est la détection d'objet en premier plan de la scène. Autrement dit, cacher un modèle numérique par un objet réel en le plaçant dans le monde réel est malheureusement impossible.

Afin de diminuer la grosseur du travail à faire, la solution du ARKit a été sélectionnée. Les limites que celle-ci importe ne seront pas atteintes dans le contexte du projet. Toutefois, une poursuite de projet pourrait les rencontrer.

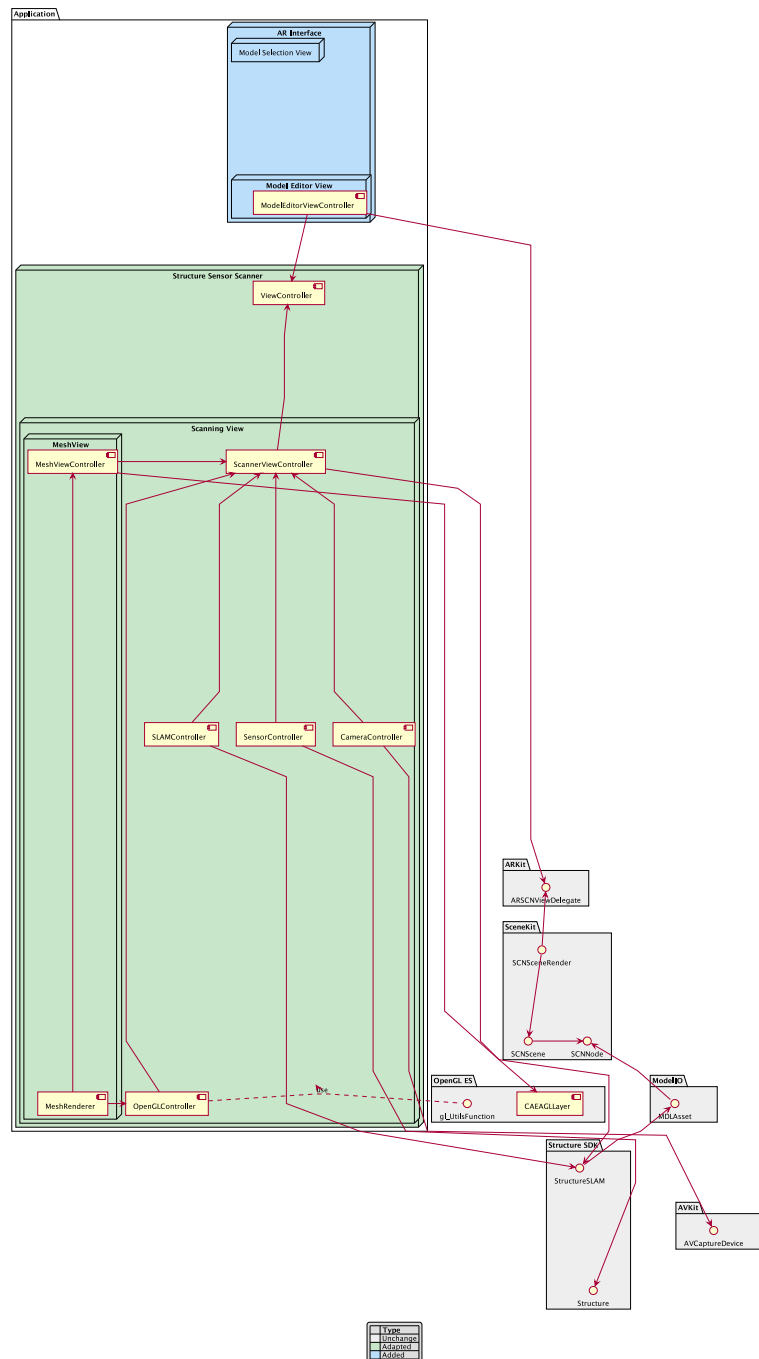


Figure 2 – Diagramme des composants de l'application

Comme on peut le voir le produit lorsqu'implémenter avec le ARKit consiste en une application intégrant plusieurs interfaces. Ce qui représente définitivement un aspect facilitant au développement. Toutefois, un tel choix représente des limites et demande une adaptation de l'application pour réduire la friction au frontière de chacune.

Les différentes interfaces adressent une variété d'enjeu et intègre une variété de technologies produite non seulement par Apple. L'interface de réalité augmentée récemment développée par Apple, ARKit. Celle-ci a pour principale tâche d'intégrer des fonctionnalité de la caméra de la plateforme et les fonctionnalité de détection de mouvement afin de créer une expérience de réalité augmentée.([Apple, 2018a](#))

Cette API est utilisée en coopération avec le SceneKit l'interface de gestion d'environnement 3D dans iOS aussi produit par Apple. Celle-ci simplifie les principaux défis du développement 3D comme les animations, la simulation physique, les effets de particules et le rendu réaliste avec gestion d'effet de lumière. L'usage d'une interface produite par le propriétaire de l'appareil hôte de l'application permet de s'assurer d'une optimisation fournie.([Apple, 2018f](#))

Les modèles utilisés par les Scène proviennent de modèle 3D importé par l'interface de Model I/O, il sont par la suite converti en ScnNode. Le modèle permet la gestion des textures, des assets, des caméras et des matériaux.([Apple, 2018d](#)) Ce genre de modèle n'est pas propriétaire à Apple, il peut donc être utilisé par Unity([Unity, 2016](#)) et Blender([Marklew, R., 2013](#)).

L'import des modèles 3D se fait à l'aide du capteur Structure qui est une des principal technologie du projet. Ce périphérique utilise la caméra de la tablette et la caméra sur le périphérique pour avoir une meilleur compréhension de l'environnement 3D. Le périphérique contient aussi un capteur infrarouge pour faciliter l'interprétation de l'environnement. Un kit de développement logiciel est fourni par les développeur du périphérique permettant le contrôle du capteur et l'export en modèle.([Occipital, 2018](#)) Les images capturées par le capteur et la caméra sont par la suite présentées par l'interface de programmation AVKit. La classe AVCaptureDevice est ainsi utilisée pour présentée à l'utilisateur l'environnement pris par l'appareil.([Apple, 2018b](#)) Afin d'offrir un rendu plus réaliste, la librairie OpenGL est aussi utilisée.([Apple, 2018e](#))

Les liens entre la logique de notre application et celle des API se fait par des extensions aux contrôleurs des vues principales de notre application et par l'intégration de certaines interfaces. Les interfaces de Structure, OpenGL, AVCaptureDevice se font par l'extension du contrôleur de vue du scanneur. Pour ce qui est de ARKit et SceneKit, les "delegates" pour chacun ont été implémentés dans le contrôleur de la vue de manipulation de modèle.

Chacun des contrôleurs est lié à une vue dans le storyboard et par les liens

créer par la logique ou par ceux créés dans le storyboard, ce qui permet à l'utilisateur de transférer d'une à l'autre.

DISCUSSION

discussion text

CONCLUSION

conclusion text

Bibliographie

- Apple. (2018a). ARKit. <https://developer.apple.com/documentation/arkit>.
- Apple. (2018b). AVKit. <https://developer.apple.com/documentation/avkit>.
- Apple. (2018c). Apple Developer Documentation. <https://developer.apple.com/documentation>.
- Apple. (2018d). Model I/O. <https://developer.apple.com/documentation/modelio>.
- Apple. (2018e). OpenGL ES. <https://developer.apple.com/documentation/opengles>.
- Apple. (2018f). SceneKit. <https://developer.apple.com/documentation/scenekit>.
- Best-Practice-Software-Engineering. (2013, Octobre, 04). Delegation Pattern [University of San Francisco, Lecture slides]. <http://best-practice-software-engineering.ifs.tuwien.ac.at/patterns/delegation.html>.
- Kremenek, T. (2017, septembre, 19). Swift 4.0 Released! <https://swift.org/blog/swift-4-0-released/>.
- Marklew, R. [Richard Marklew]. (2013, Février, 10). Importing .obj to blender 2.6 [Forum]. <https://blenderartists.org/forum/showthread.php?281322-Importing-obj-files-to-blender-2-6>.
- Microsoft. (2014, Octobre, 21). WindowsPreview.Kinect Namespace [Documentation]. [https://docs.microsoft.com/en-us/previous-versions/windows/kinect/dn758774\(v%3dieb.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/kinect/dn758774(v%3dieb.10)).
- Molitch-Hou, M. (2016, Janvier, 06). Occipital Unveils New 3d Sensor & Mixed Reality Engine at CES [3D Printing Industry, Article]. <https://3dprintingindustry.com/news/64337-64337/>.
- Occipital. (2018, Février, 05). Structure SDK version 0.7.1 [SDK Download Link]. <https://developer.structure.io/sdk>.

Statt, N. (2017, Septembre, 12). Apple shows off breathtaking new augmented reality demos on iPhone 8 [The Verge, Article]. <https://www.theverge.com/2017/9/12/16272904/apple-arkit-demo-iphone-augmented-reality-iphone-8>.

Unity. (2016). How do I import Models from my 3D app? <https://docs.unity3d.com/540/Documentation/Manual/HOWTO-importObject.html>.

Worley, C. [n6xej]. (2016, septembre, 29). StructureScannerSwift. <https://github.com/n6xej/StructureScannerSwift>.