

Travail de Bachelor 2021

Augmented Reality for GIS applications



Etudiant-e : Victor Bonny

Professeur : Antoine Widmer

Résumé

Ce travail consiste à développer un prototype d'application de réalité augmentée qui apporte des informations complémentaires au simple paysage que les utilisateurs observent à travers celle-ci. Le but est de proposer une solution utilisable, testable et pouvant servir de base par la suite à un projet plus poussé.

Nous ferons un état de l'art des différentes technologies actuelles qui sont englobées dans la réalité mixte afin de mettre en avant les possibilités d'applications qu'elles proposent. Nous irons plus en détail pour la réalité augmentée qui est utilisée pour notre projet. Nous définirons également les systèmes d'informations géographiques qui sont très utiles lors de projets sur les cartes et leurs données.

Dans ce document, nous définirons le marché actuel de la réalité augmentée et les tendances des applications populaires et plus précisément celles qui traitent des données géographiques.

Ensuite, nous présenterons le prototype ainsi que ses particularités techniques. Dépendamment des fonctionnalités développées, nous détaillerons les aspects théoriques qui leur sont propres. Nous aurons, comme cela, lors de la lecture de ce document, une compréhension globale de l'application.

Nous ferons un compte rendu de tests du prototype et mèneront une enquête auprès d'utilisateurs afin d'avoir des avis externes sur le prototype et les différentes thématiques traitées dans ce document.

Puis, nous détaillerons les futures améliorations qui pourront être apportées ou faites au prototype. Nous mettrons en avant des possibilités trouvées mais pas développées du fait de la durée limitée du travail de Bachelor.

Finalement, nous concluons ce travail avec un bilan divisé en deux parties : une conclusion globale sur le projet ainsi qu'une conclusion plus personnelle.

Mots-clés : Réalité augmentée, GIS, Application, Paysage, Smartphone

Avant-Propos

Ce travail a été effectué dans le cadre du travail de Bachelor en informatique de gestion à la Haute école spécialisée de Suisse occidentale (HES-SO) Valais-Wallis.

Le travail a été proposé par mon professeur responsable, Monsieur Antoine Widmer.

La réalité augmentée est encore très peu utilisée dans des domaines comme le tourisme qui sont des marchés très opportuns pour aider au développement de ces technologies.

Les applications qui permettent d'ajouter des informations virtuelles sur le monde actuel se multiplient petit à petit et c'est pourquoi nous avons décidé de créer un prototype de réalité augmentée dans le cadre de cette thèse.

Le scope du projet a été discuté et défini à la séance de démarrage. Les objectifs ont évolué d'un commun accord durant le développement afin de rajouter de nouvelles fonctionnalités au prototype.

Au travers de ce projet, de nombreux aspects techniques sont abordés en plus de la réalité augmentée. Notre travail propose une base concrète pour de futurs développements sur notre prototype qui emploie de données GIS. Ce document regroupe les explications théoriques et techniques pour réussir à reproduire certaines fonctionnalités de l'application et les problèmes survenus et des possibilités de solutions pour y remédier.

Remerciements

Je tiens à adresser mes remerciements aux personnes qui m'ont, d'une manière ou d'une autre, permis de réaliser ce travail de Bachelor :

Monsieur Antoine Widmer pour ses conseils utiles tout au long du projet.

Madame Nancy Zappellaz qui m'a permis d'utiliser et de relire son code d'un projet antérieur.

Mes proches, mes amis ainsi que mes parents pour leur soutien et leur aide lorsque j'en ai eu besoin.

A mes collègues de l'Institut de recherche en informatique de gestion à Sierre qui m'ont soutenu et conseillé lors de l'élaboration de mon travail.

Toutes les personnes qui ont pris le temps d'échanger au sujet du projet et de son contenu.

Table des matières

1.	Liste des figures	vi
2.	Liste des tableaux.....	viii
3.	Liste des abréviations	ix
4.	Introduction	1
4.1.	Problématique	1
4.2.	Méthodologie de travail	1
4.3.	Marché actuel	2
5.	Technologies en bref.....	3
5.1.	Réalité virtuelle	4
5.2.	Réalité mixte	7
5.3.	Réalité augmentée.....	9
6.	AR et GIS.....	12
6.1.	Système d'information géographique.....	12
6.2.	Possibilités et limites.....	13
6.3.	Applications actuelles	14
7.	Outils de la réalité augmentée.....	17
7.1.	Environnements de développement	17
7.1.1.	Unreal Engine.....	17
7.1.2.	Unity	19
7.1.3.	Récapitulatif et choix pour le prototype	21
7.2.	SDK.....	23
7.2.4.	AR Foundation.....	23
7.2.5.	ARCore	24
7.2.6.	ARKit	25
7.2.7.	Vuforia	26
7.2.8.	Mapbox	27
7.2.9.	Wikitude	27
7.2.10.	Récapitulatif et choix pour le prototype	28
8.	Prototype.....	28
8.1.	Use case	28
8.2.	Mise en place de l'environnement de travail.....	29
8.3.	Rendu en direct de la caméra	30
8.4.	Superposer le virtuel sur le monde réel	30
8.5.	Créer du contenu dans le Canvas	31
8.5.1.	Ajouter un élément virtuel dans le Canvas.....	31
8.5.2.	Dessiner dynamiquement sur le Canvas	32
8.6.	Implémentation de la partie GIS	32
8.6.1.	Outils de géolocalisation	32
8.6.2.	Données géographiques.....	33
8.7.	Texture représentant le paysage.....	34
8.7.1.	Paramètres de base	35
8.7.2.	Accès aux fichiers du dossier StreamingAssets	36
8.7.3.	Créer la texture représentant le paysage	36
8.7.4.	HSV to RGB	37
8.7.5.	Traitement des couleurs.....	40
8.7.6.	Création de l'overlay.....	41
8.7.7.	Transparence globale.....	41

8.8.	Distance de l'utilisateur par rapport à un point	42
8.8.1.	Tentative 1	42
8.8.2.	Tentative 2	43
8.8.3.	Solution trouvée	44
8.9.	Limitations du prototype	45
9.	Tests du prototype.....	48
10.	Enquête	49
11.	Améliorations du prototype	51
12.	Conclusions	52
12.1.	Conclusion projet	52
12.2.	Conclusion personnelle	53
13.	Références	54
14.	Annexe I : Product Backlog	61
	Annexe II : Calendrier	62
	Annexe III : Questionnaire de l'enquête (menée par l'auteur en 2021)	64
	Déclaration de l'auteur	65

1. Liste des figures

Figure 1 : Prédications de l'évolution du marché AR/VR de 2016 à 2025	2
Figure 2 : Résumé des différentes virtualités.....	3
Figure 3 : Illustration du fonctionnement des 3 degrés de liberté	5
Figure 4 : Illustration du fonctionnement des 6 degrés de liberté	5
Figure 5 : Illustration d'un Oculus Quest 2.....	6
Figure 6 : Illustration d'un HP Reverb G2.....	6
Figure 7 : Illustration d'un Microsoft HoloLens 2.....	8
Figure 8 : Illustration d'un pack Zappar Zapbox.....	9
Figure 9 : Illustration d'une paire de Glass Enterprise Edition 2.....	11
Figure 10 : Illustration d'une paire de lunettes AR de Facebook.....	12
Figure 11 : Capture d'écran d'une application fonctionnant avec Argis Lens	14
Figure 12 : Capture d'écran de l'application Hier et Aujourd'hui	15
Figure 13 : Illustration de l'application PeakFinder avec un filtre	16
Figure 14 : Capture d'écran de l'application PeakFinder sans filtre	17
Figure 15 : Illustration d'un objet 3D apparu grâce à un Clip code	26
Figure 16 : Illustration d'un Clip Code	26
Figure 17 : Packages installés dans le projet.....	29
Figure 18 : Maquette du rendu en direct de la caméra	30
Figure 19 : Architecture de base de la scène	30
Figure 20 : Architecture de la scène après ajout de l'overlay.....	31
Figure 21 : Maquette de l'utilisation d'un filtre sur le paysage	31
Figure 22 : Illustration de l'angle appelé Bearing.....	33
Figure 23 : Représentation visuelle des données de nos fichiers .hgt.....	34
Figure 24 : Maquette représentant le paysage découpé par strie	35
Figure 25 : Photo d'une Sandbox avec une Heat Map.....	37
Figure 26 : Représentation conique du modèle HSV	38
Figure 27 : Rendu après création de la texture pour l'overlay	39
Figure 28 : Disposition des pixels sur un écran de smartphone Android	40
Figure 29 : Texture mise à l'endroit	40
Figure 30 : Overlay avec rendu de la caméra en direct	41
Figure 31 : Overlay avec transparence complète	42

Figure 32 : Illustration graphique de l'agorithme de tracé de segment de Bresenham.....	43
Figure 33 : Capture d'écran du prototype avec distance affichée et transparence intégrée..	45
Figure 34 : Overlay si la position de l'utilisateur est hors des fichiers.....	46
Figure 35 : Illustration de différentes surfaces possibles	47
Figure 36 : Erreur de recherche dans le tableau de couleurs	48
Figure 37 : Couleur du point choisi par l'utilisateur introuvable	48

2. Liste des tableaux

Tableau 1 : Comparatif entre Unreal Engine 4 et Unity 2021.1.....	22
Tableau 2 : Moyennes des temps d'exécution en secondes du prototype	49
Tableau 3 : Journal de bord.....	62

3. Liste des abréviations

AR	Augmented Reality
DMS	Degrés (°), Minutes ('), Secondes (")
DOF	Degrees of Freedom
GIS	Geographic Information System
GPS	Global Positioning System
HGT	Height
HSV	Hue – Saturation - Value
HUD	Head Up Display
PNJ	Personnage non-joueur
SDK	Software Development Kit
SIG	Système d'information géographique
VR	Virtual Reality
WWDC	Apple Worldwide Developers Conference
XR	Extended Reality

4. Introduction

4.1. Problématique

L'utilisation de smartphone est en constante augmentation. Nous nous retrouvons de plus en plus connectés mais surtout en possession d'outils de plus en plus puissants. Ces avancées au niveau du matériel permettent aux développeurs de proposer des applications de plus en plus rapides et complètes également. La réalité augmentée, virtuelle et mixte profitent de ces améliorations.

Nous pouvons nous apercevoir que beaucoup de jeux, vidéos ou même des objets 3D peuvent s'utiliser à l'aide de casque de réalité augmentée ou d'autres équipements.

En parallèle de ces avancées technologiques, les acteurs professionnels qui pourraient en bénéficier stagnent. Nous ne pouvons pas encore observer énormément de campagnes marketing orientées sur des applications ou outils de réalité augmentée.

Le domaine du tourisme propice à l'utilisation de ce genre de technologie est un terrain de jeu vaste pour les développeurs. C'est ce dernier qui nous concerne le plus avec notre projet d'application de réalité augmentée. Nous aimerions savoir s'il est possible de développer un prototype d'application de réalité augmentée qui intègre les outils de géolocalisation ainsi que des fichiers de modèles de terrain en 3D ou autres données géographiques.

Nous proposons dans cette thèse les informations utiles qui détaillent les technologies utilisées pour développer les fonctionnalités principales d'une application AR. Elle s'utilise sur smartphone Android et permet d'ajouter des éléments virtuels sur un paysage observé à l'aide d'un smartphone.

4.2. Méthodologie de travail

Aucune méthodologie ne nous a été imposée. Nous avons établi des rendez-vous presque hebdomadaires avec Monsieur Antoine Widmer afin d'avoir un suivi régulier du développement du prototype et de l'écriture de la thèse. La méthodologie Agile a été adaptée aux besoins du projet et en fonction de l'avancée des sprints.

Dans un premier temps, un cahier des charges a été écrit et révisé d'un commun accord avec mon professeur responsable.

Ensuite, le projet a débuté par une étude et des tests d'autres applications proposées sur le marché ainsi que de l'état de l'art de la réalité augmentée afin d'établir les possibles développements pour notre prototype. Des recherches sur les GIS ont également été effectuées. Seulement après cela, nous avons commencé à travailler sur le prototype et ajusté le «Product Backlog» au fur et à mesure du projet pour rajouter ou modifier des fonctionnalités.

En parallèle de la thèse, un journal de bord avec les objectifs à atteindre pour chaque sprint avec les explications théoriques ou techniques correspondantes a été écrit afin de garder un suivi et une trace du travail effectué.

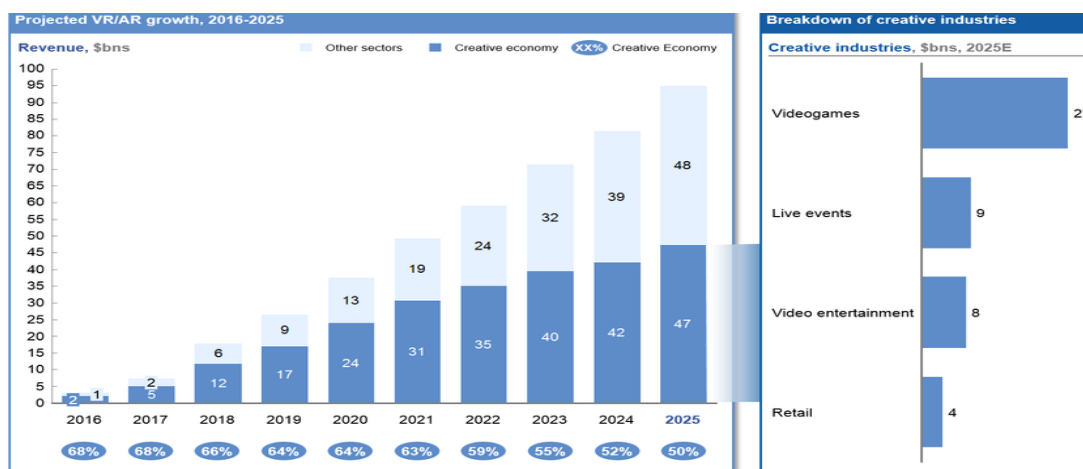
4.3. Marché actuel

La réalité augmentée, virtuelle ou mixte n'est plus un simple gadget. Les dernières années ont prouvé que cela pouvait être plus qu'une mode mais bien un marché prometteur dans son ensemble.

Selon Mirabaud Securities, la taille du marché des technologies immersives ou en anglais « Extended Reality » pourrait atteindre les 209 milliards de dollars en 2025. Dès 2018, des projections faites par Goldman Sachs estiment également que les revenus générés jusqu'en 2025 ne vont cesser d'augmenter avec une croissance régulière. (Mendiratta, 2018)

Que ce soit les ventes de programmes, de jeux vidéos et d'accessoires tels que les casques de réalité virtuelle, tout indique une constante augmentation des chiffres d'affaires.

Figure 1 : Prédications de l'évolution du marché AR/VR de 2016 à 2025



Source : Récupéré sur <https://www.entrepreneur.com/article/307025>

Des secteurs tels que le commerce de détail, les médias, la télécommunication, la santé et l'éducation sont propices à accentuer ce phénomène.

En 2021, si on compare les prédictions et les chiffres actualisés, on peut observer que tout est toujours très optimiste pour les prochaines années. Les sites spécialisés annoncent une possible augmentation du marché de 160 milliards de dollars. (Technavio, 2021)

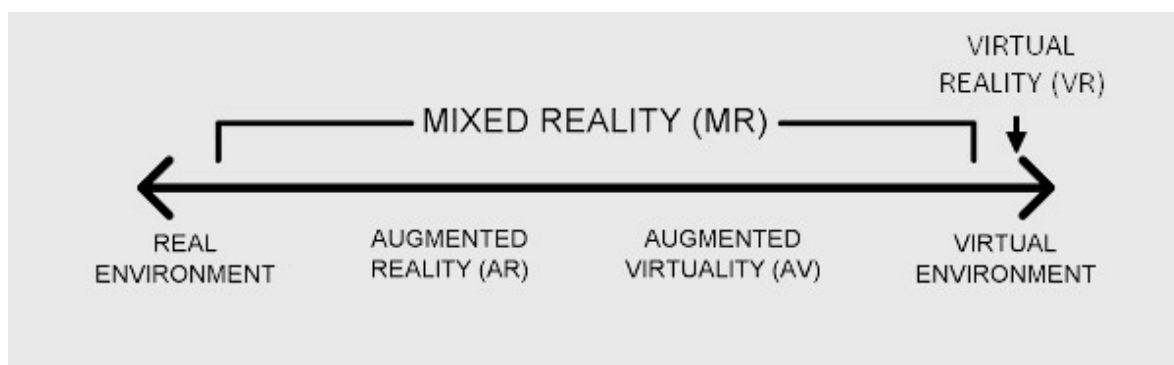
Les effets du Covid-19 peuvent peut-être jouer un rôle de catalyseur pour aider à remplacer certaines technologies ou interactions actuelles par de la réalité virtuelle, augmentée ou mixte. Nous pourrions, peut-être, dans quelques années, assister à des réunions en réalité virtuelle afin de combler cette distanciation amenée par le télétravail. C'est en tout cas ce que sous-entend le président du World XR Forum M. Salar Shahna. (Zaki, 2020)

Pour l'instant au niveau matériel, ce sont des produits coûteux qui peuvent atteindre plusieurs milliers de francs qui sont mis en avant sur le marché pour des clients et domaines très précis. Beaucoup d'entreprises spécialisées comme HTC ou Varjo préfèrent se diriger vers les professionnels et laisser le segment client à des compagnies qui sont déjà très bien placées avec leurs produits à prix inférieurs et pour des utilisations plus courantes. (Zaki, 2020)

5. Technologies en bref

Les différentes « réalités » sont définies par la manière avec laquelle l'utilisateur est immergé ou non dans un monde virtuel et comment il peut interagir avec celui-ci. Plus nous allons vers le monde virtuel, plus nous nous éloignons de l'environnement réel et vice-versa.

Figure 2 : Résumé des différentes virtualités



Source : Récupéré sur <http://www.immersion.fr/realite-virtuelle-realite-mixte-realite-augmentee/>

Nous présenterons ci-dessous la réalité virtuelle, mixte et augmentée. C'est intéressant d'observer les différences entre celles-ci. Cela permet d'avoir les champs des possibles pour de futurs projets en fonction des libertés et contraintes de l'utilisateur avec chaque technologie.

5.1. Réalité virtuelle

La réalité virtuelle englobe les technologies et outils permettant de plonger l'utilisateur dans un monde entièrement généré par un ordinateur. Nous quittons l'environnement réel pour être plongé dans un environnement de synthèse qui peut être créé de plusieurs façons. Nous pouvons, par exemple, permettre à une personne de voyager dans des endroits distants sans pour autant qu'elle ne quitte sa chaise à partir d'images 360° comme nous pouvons permettre à cette personne d'être un personnage de fiction dans un jeu à la 1^{ère} personne dans un monde fictif de A à Z.

Cette technologie est devenue accessible au grand public grâce à des casques de réalité virtuelle commercialisés en 2016. Des entreprises comme Oculus VR rachetée par Facebook en 2014 ont permis de démocratiser des outils qui permettent à des professionnels comme à des particuliers de profiter de cette nouvelle réalité en 2021.

La technologie en tant que telle existe depuis plusieurs décennies. Il faut remonter dans les années 50 pour trouver la première machine permettant à son utilisateur d'être plongé dans un monde imaginaire en ayant ses sens trompés par des odeurs, sons ou autres sensations. C'est un cinéaste nommé Morton Heilig qui a fabriqué cette première machine.

Puis, vers 1960, la machine s'est transformée en casque afin que cela devienne plus facile à transporter et à utiliser. Des entreprises telles que Tecmo ont créé des cabines permettant des immersions sensorielles complètes comme imaginé par Heilig pour faire profiter à des joueurs de salles d'arcades japonaises de jeux inimitables ailleurs.

Dès les années 70, au fur et à mesure que les technologies se sont développées, nous avons pu voir l'arrivée de nouvelles machines sur le marché proposant de la réalité virtuelle jusqu'aux casques que l'on connaît en 2021.

Ce sont des domaines techniques tels que l'aviation qui ont bénéficié de ces percées scientifiques avec des cockpits de simulations de vol en VR. Ensuite, le MIT a présenté le Aspen Movie Map (l'ancêtre de Google Map) qui permet de faire voyager un utilisateur de

ville en ville à la 1^{ère} personne. Finalement, en 1991, le Sega VR a lancé la course des casques modernes de VR. (Barnard, History of VR - Timeline of Events and Tech Development, 2019)

Les applications de réalité virtuelle sont présentes dans de nombreux domaines en plus des jeux vidéos. Des simulations permettent à des personnes de soigner des traumatismes mais également à préparer des personnes à réagir dans des situations qu'elles ne peuvent pas vivre tous les jours dans leur métier mais pour lesquelles elles doivent être prêtes. Les visites immobilières peuvent se faire à l'aide de réalité virtuelle également. Il y a même la possibilité d'agrandir ces visites à l'échelle d'une ville ou d'un pays.

Afin de pouvoir utiliser la réalité virtuelle, il faut être en possession d'accessoires comme ceux présentés à la fin de ce chapitre. En fonction du casque et des commandes que l'utilisateur possède, il est limité dans ses actions dans le monde virtuel. Nous parlons de degrés de liberté. Un casque permet par exemple de faire les actions indiquées sur l'image.

Figure 3 : Illustration du fonctionnement des 3 degrés de liberté



Pencher la tête à droite et à gauche

Tourner la tête à gauche et à droite

Pencher la tête en avant ou arrière

Source : Récupéré sur <https://virtual-guru.com/astuces/vr-quelle-est-la-difference-entre-3-dof-et-6-dof/>
Il y a 3 degrés de liberté qui sont définis par les 3 mouvements cités ci-dessus. Il est possible d'en avoir plus car certains casques et commandes permettent de rajouter d'autres interactions.

Figure 4 : Illustration du fonctionnement des 6 degrés de liberté



Pencher la tête à droite et à gauche

Tourner la tête à gauche et à droite

Pencher la tête en avant ou arrière

Avancer vers l'avant, l'arrière, à droite, à gauche, vers le haut et vers le bas

Source : Récupéré sur <https://virtual-guru.com/astuces/vr-quelle-est-la-difference-entre-3-dof-et-6-dof/>

Nous arrivons à 6 mouvements de libertés. Ces critères sont les principales limites aux jeux ou applications créés avec la réalité virtuelle en écartant les limites technologiques liés au développement. Ces degrés de liberté sont également utiles dans les autres réalités que ce soit augmentées ou mixtes. (Barnard, Degrees of Freedom (DoF): 3-DoF vs 6-DoF for VR Headset Selection, 2019)

Il faut bien réfléchir en amont aux problèmes que certaines actions à faire dans le monde virtuel peuvent amener dans le monde réel.

On peut imaginer de devoir se déplacer avec un joystick qui pourrait rendre malade le joueur ou encore faire une action rapide de la tête qui serait handicapante avec un gros casque posé sur le joueur.

Voici, en exemple, une sélection de casques et de manettes disponibles en 2021 pour utiliser des applications de réalité virtuelle :

Figure 5 : Illustration d'un Oculus Quest 2



Oculus Quest 2

Prix : environ 500 francs

Software / Hardware : Plusieurs configurations disponibles

Source : Récupéré sur <https://www.esquire.com/lifestyle/a35339497/oculus-quest-2-vr-headset-review-endorsement/>

Figure 6 : Illustration d'un HP Reverb G2



HP Reverb G2

Prix : environ 600 francs

Software / Hardware : Plusieurs configurations disponibles

Source : Récupéré sur <https://support.hp.com/id-en/document/c06938191>

5.2. Réalité mixte

La réalité mixte est un terme englobant les différentes manières de visualiser des contenus en 3D superposés au monde réel grâce à des outils ou appareils qui y sont dédiés. Le terme de réalité mixte est apparu en 1994 lors de travaux sur la virtualité. Ils déterminaient les technologies qui étaient semblables à la réalité augmentée et virtuelle sans pour autant rentrer dans ces deux catégories. En d'autres termes, ce mot regroupe la réalité virtuelle et augmentée. La différence avec la réalité augmentée est que l'utilisateur peut grâce à différents mouvements et gestes, interagir très facilement avec les éléments en 3D qui sont affichés comme un calque sur les images réelles. Et à la différence de la réalité virtuelle, l'utilisateur n'est pas entièrement immergé dans un monde créé par ordinateur.

En 2015, le terme de réalité mixte a évolué avec la sortie du casque Microsoft HoloLens qui n'est pas un casque de réalité virtuelle mais un casque de réalité augmentée accompagné de fonctionnalités supplémentaires que la réalité augmentée ne propose pas.

Intégrant entièrement la catégorie de réalité mixte, on utilise le casque de Microsoft comme point de comparaison si l'on veut classer et définir un casque ou autre accessoire dans le domaine du virtuel.

En 2021, de nombreux industriels utilisent la réalité mixte dans leurs exploitations ou usines.

Dans le domaine de l'automobile, Ford a remplacé des maquettes en argile par des maquettes 3D qui deviennent modulables et plus détaillées que les modèles qu'elles ont remplacés grâce à la réalité mixte.

Microsoft développe avec Volvo un projet de vente de voitures d'un nouveau genre. Grâce à un casque de réalité mixte, les clients pourront personnaliser leur voiture d'un simple geste afin de la voir en temps réelle devant eux sans pour autant que le concessionnaire ne la propose physiquement dans son garage.

En plus du domaine de l'automobile, on peut observer l'apparition de la réalité mixte dans celui de la construction, de l'immobilier et même de la chirurgie.

Lors d'une opération pour la pose d'une prothèse d'épaule, un médecin de l'hôpital Avicenne s'est aidé d'un casque HoloLens pour obtenir en temps réel des informations nécessaires au bon déroulement de cette opération. C'est bien entendu plus précis et moins déroutant que devoir retrouver ces informations sur des machines ou autres rapports.

L'apport d'images 3D ou d'informations complémentaires sur le monde réel permettent de gagner en temps et en précision. Les entreprises qui créent les casques essaient d'avoir le plus grand Field Of View couvert (FOV). C'est-à-dire le champ du monde réel sur lequel la réalité mixte peut s'appliquer grâce au casque. . (Artefacto, n.d)

La limite principale de cette technologie est le prix car pour l'instant les casques type HoloLens sont très onéreux. L'apparition d'un casque plus accessible permettrait la démocratisation de cette technologie et une utilisation plus grand public.

Un des derniers gros évènements était la mise sur le marché du casque Magic Leap One qui aurait permis d'avoir de très bonnes performances à moindres coûts. Cette sortie s'est transformée en échec et l'entreprise s'est retirée du segment des consommateurs. Le produit annoncé à bas coût n'a pas convaincu les clients.

Voici, en exemple, une sélection de casques de réalités mixtes disponibles en 2021 dont un qui pourrait être intéressant dans les années à venir avec son prix très accessible :

Figure 7 : Illustration d'un Microsoft HoloLens 2



Microsoft HoloLens 2

Prix : environ 4'000 francs

FOV : 52°

Utilisation : Casque autonome

Software / Hardware : Plusieurs configurations disponibles

Source : Récupéré sur <https://www.pinterest.com/pin/514043744971406484/?d=t&mt=login>

Figure 8 : Illustration d'un pack Zappar Zapbox



Zappar Zapbox

Prix : environ 40 francs

FOV : Adaptif

Utilisation : Casque connecté à un smartphone

Source : Récupéré sur <https://androidnewspad.com/tag/zapbox/>

5.3. Réalité augmentée

La réalité augmentée est également une technologie visant à enrichir les connaissances de l'utilisateur à l'aide d'une application ou d'un programme en intégrant des éléments virtuels sur un environnement réel.

Au contraire des deux autres technologies présentées dans ce rapport, elle se place très proche du monde réel. En effet, les applications permettent d'enrichir la vision, des images ou autres éléments captés dans le monde réel.

L'AR, à l'instar des autres technologies, se basent sur les recherches de Morton Heilig et a ensuite évolué sur sa propre branche grâce à des inventions de différents chercheurs tel que Steve Mann.

Il a présenté dans les années 1980 le premier dispositif opérationnel de réalité augmentée appelé EyeTap.

En parallèle de la présentation de cet appareil, on peut également voir l'arrivée des premiers systèmes HUD au sein de différentes armées. Ces systèmes permettent d'afficher des informations sur un écran transparent. Les avancées militaires sont complétées par celles de la NASA qui sortent un casque de réalité augmentée qui est surtout utilisé comme modèle pour le casque de Microsoft présenté dans le paragraphe sur la réalité mixte.

Le terme officiel de la réalité augmentée, tel qu'on le comprend aujourd'hui, prend vraiment sens dans les années 90 lorsque des salariés de Boeing développent un outil destinés à des ouvriers des chaînes de fabrications qui leur permet d'avoir un calque d'informations sur les

éléments de la chaîne en direct. C'est un affichage de plan technique qui se fixe sur un élément physique.

Dans la lancée de cette invention, en 1994, deux scientifiques de Sony Computer Lab. développent NaviCam. C'est la première fois qu'un système de réalité augmentée est capable de lire des marqueurs. Les utilisateurs doivent toujours utiliser un casque et le programme basique ne sert qu'à afficher quelques informations textuelles en fonction des marqueurs que l'utilisateur capte.

Cette étape a été très importante. Elle sert de base pour la suite de l'évolution de cette technologie. Nous présenterons, plus tard dans ce document, les entreprises qui ont suivi cette tendance pour développer et proposer des SDK qui sont spécialement adaptés à la reconnaissance de marqueurs, vidéos, images ou autres objets du monde réels.

Une autre étape qui a permis à la réalité augmentée de se démocratiser est l'évolution technologique des téléphones mobiles. (Artefacto, n.d)

L'amélioration des processeurs, appareils photos et de la mémoire a permis aux développeurs de proposer au grand public des applications de réalité augmentée très complète et détaillée. On ne se contente plus d'afficher seulement des informations textuelles mais on peut afficher des meubles dans notre salon et les placer comme bon nous semble avant de les acheter. Cette application est proposée par IKEA.

Dans de nombreux domaines la réalité est déjà présente. Nous avons des applications pour l'Histoire avec des visites de musées aidées par l'AR, des jeux pour enfants comme Quiver et des applications dans la mode avec la possibilité d'essayer virtuellement des chaussures ou des habits. On trouve également l'utilisation de filtres virtuels dans toutes les applications de réseaux sociaux. Le marché de l'AR comme les autres technologies présentées plus tôt est encore très ouvert et propice aux nouveautés.

En 2020, une estimation a démontré qu'il y avait presque 1 milliard de smartphones sur terre avec une proportion d'au moins 27% d'appareils qui étaient assez avancés pour utiliser l'AR. On remarque que les potentiels utilisateurs sont nombreux surtout que les outils de développement permettent d'alléger les applications afin d'en augmenter le nombre. (Révész, 2020)

En 2021, les outils d'affichages d'information comme la vitesse dans les voitures sur le pare-brise sont intégrés aux machines mais les principales applications s'utilisent avec des tablettes ou des téléphones mobiles.

Les différentes grosses entreprises technologiques telles que Facebook essaient de sortir leur épingle du jeu en proposant des alternatives aux smartphones.

Google a présenté des lunettes de réalité augmentée, en 2013, étant spécialement conçues pour les professionnels afin que des développeurs y proposent des applications ainsi que des améliorations. Le projet a été mis en stand-by quelques années et une nouvelle édition des lunettes appelées Glass Enterprise Edition 2 a été créée. Il est possible de devenir partenaire dans le développement de lunettes ou d'applications leur étant destinées sur le site internet dédié à ce projet. (Google, n.d)

L'alternative de Facebook présente des lunettes de réalité augmentée dévoilées en 2020 au sein du projet Aria. (Announcing Project Aria: A Research Project on the Future of Wearable AR, 2020)

L'objectif du Facebook Reality Lab. est de proposer des outils très poussés allant jusqu'à un assistant virtuel qui nous accompagne au quotidien. Des présentations sur les possibilités des lunettes ont été faites mais très peu de données sur les composants ont été divulguées. Des informations de sites spécialisés sous-entendent que les lunettes pourraient être en vente durant l'année 2021. De plus, en juillet 2021, Mark Zuckerberg a expliqué lors d'un interview que son équipe était en charge d'explorer les possibilités de créer un « metaverse ». Il s'agit d'un réseau social où les personnes se retrouvent sous forme d'avatar dans un univers virtuel. (Newton, 2021)

Voici une petite fiche des lunettes de réalité augmentée des projets cités précédemment :

Figure 9 : Illustration d'une paire de Glass Enterprise Edition 2



Glass Enterprise Edition 2

Prix : non disponible

Source : Récupéré sur <https://www.computerworld.com/article/3396661/revamped-google-glass-2-aims-more-powerful-ar-at-the-enterprise.html>

Figure 10 : Illustration d'une paire de lunettes AR de Facebook



Projet Aria – Lunettes AR de Facebook

Prix : non disponible

Source : Récupéré sur <https://www.hebergementwebs.com/nouvelles/facebook-travaille-sur-les-lunettes-de-realite-augmentee-protect-aria-avec-le-lancement-des-lunettes-intelligentes-en-2021>

6. AR et GIS

6.1. Système d'information géographique

Bien qu'apparus dans les années 60 au Canada et aux Etats-Unis, les GIS ont plusieurs définitions à travers le monde. Les caractéristiques communes sont qu'ils ont pour objectif de rassembler, d'organiser, de gérer, d'analyser, de combiner, d'élaborer et de présenter des informations localisées géographiquement à partir de diverses sources de données. Tout cela en vue d'une utilisation plus développée de ces informations comme pour un aménagement d'un territoire, du contrôle de l'environnement, des fins économiques ou autres applications.

Les informations de base viennent principalement de coordonnées géographiques (adresses postales, latitude, longitude,...), de cartes topographiques, de photos aériennes et d'images satellites.

Beaucoup de ces données comme notre position est enregistrée en temps réel par nos appareils technologiques comme un ordinateur, GPS ou un smartphone. (A.Balasubramanian, 2017)

De plus, il y a également beaucoup de bases de données disponibles sur internet et en libre utilisation. Nous avons, par exemple, OpenStreetMap, Corine Land Cover ou encore Geosignal. Ces référentiels nous permettent d'avoir des données plus ou moins précises et plus ou moins intéressantes selon ce que l'on veut mettre en avant dans nos projets.

Ensuite, ces données sont traitées et utilisées par des programmes comme PostGIS qui permet de faire des requêtes spatiales ou Arc-Map qui permet de faire des cartes-multiplateformes. Il y a d'autres programmes gratuits et open source disponibles sur internet.

L'utilisation de l'AR dans le système d'informations géographiques se fait principalement grâce à des applications disponibles sur smartphones. Nous présenterons dans cette thèse différentes applications.

Ces applications GIS combinent, en général, l'utilisation de technologies comme les senseurs optiques, les accéléromètres et le GPS. Ces informations sont quasiment toutes disponibles lors de l'utilisation d'un smartphone d'où leur popularité dans ce domaine.

En résumé, les systèmes d'informations géographiques amènent des aides à la décision et de la plus-value sur des données brutes disponibles en tout temps par nos appareils. (Kouba, 2019)

6.2. Possibilités et limites

Les applications combinant AR et l'utilisation de données géographiques se multiplient et offrent une infinité de possibilités.

Les informations du monde réel sont captées au travers de la caméra du smartphone qui peuvent être ensuite utilisées par l'application. Les applications peuvent afficher des objets virtuels sur le terrain réel ou effectuer des mesures par exemple. Il y a également la possibilité de rajouter certains filtres visuels sur l'écran entier.

Le fait que le smartphone sait où nous regardons grâce à son inclinaison, son positionnement et la direction dans laquelle il est tourné rend les informations géographiques intéressantes.

D'autres utilisations de GIS peuvent se faire avec l'apport de données en direct avec des relevés de drones ou autres outils technologiques plus récentes. La société Arvizio a annoncé , en 2021, avoir réussi à combiner AR/VR avec des données et modèles 3D fournis par les drones. (Arvizio, 2021)

Les améliorations apportées au réseau internet et à sa couverture permettent également de créer des rendus plus rapidement et à de nouveaux endroits. Certains lieux sont encore trop isolés et l'utilisation d'applications connectées reste impossible. Afin de garantir une

application précise qui respecte les points ci-dessus, nous devons avoir la position exacte de l'utilisateur et en temps réel. Ce qui est impossible en mode avion par exemple. La démocratisation de la 5G pourrait également permettre l'apparition de nouvelles applications utilisant des données GIS de meilleures qualités. En effet, la proximité des antennes 5G entre elles réduisent les différentes interférences qui peuvent apparaître. De plus, la vitesse des calculs et du réseau permettront de rendre les mises à jour plus rapides et précises. Les positions des utilisateurs seront plus précises également. (Bergum, 2020)

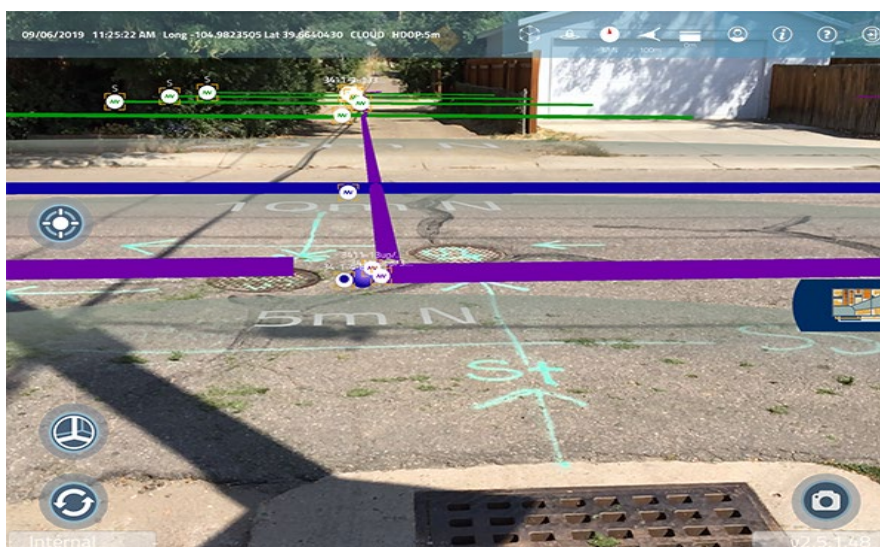
Le traitement d'images ou de données demandent également beaucoup de ressources matérielles. Cela limite le développement d'applications ou d'outils de ce genre car nous ne pouvons demander à un smartphone d'effectuer des calculs complexes qui devraient être dédiés à un serveur. Il faut suivre si dans les années futures, les fabricants considéreront pleinement la réalité augmentée afin qu'ils développent des smartphones dédiés à cet usage.

6.3. Applications actuelles

Voici quelques applications qui utilisent des données géographiques et le rendu en direct d'un appareil afin d'y apporter des détails et informations supplémentaires.

Dans le domaine de la construction ou de l'urbanisme, il est possible de mettre en évidence des canalisations cachées sous le sol grâce au positionnement de l'utilisateur et de la direction dans laquelle il regarde.

Figure 11 : Capture d'écran d'une application fonctionnant avec Argis Lens



Source : Récupéré sur https://www.youtube.com/watch?v=JKEQUXax8o4&feature=emb_title

La capture ci-dessus est issue d'une application qui est utilisée par la ville d'Englewood dans le Colorado. Elle utilise la technologie Argis Lens qui permet d'amener des visualisations 3D sur le rendu vidéo en direct grâce aux données géographiques. Leurs équipes peuvent bénéficier d'informations utiles au niveau des routes et structures cachées par le bitume. En cas d'erreurs de précision, les données sont mises à jour au fur et à mesure par les employés sur le terrain. Cette répétition de traitement permet d'affiner la précision des positions des objets 3D et les différents marqueurs dans l'environnement réel. (ARGIS, 2019)

Pour son marché touristique, la ville de Paris a créé une application en partenariat avec Geneanet permettant de comparer des images des derniers siècles à des monuments actuels.

Figure 12 : Capture d'écran de l'application Hier et Aujourd'hui



Source : Récupéré sur <https://play.google.com/store/apps/details?id=org.geneanet.vuesHier&hl=fr&gl=US>

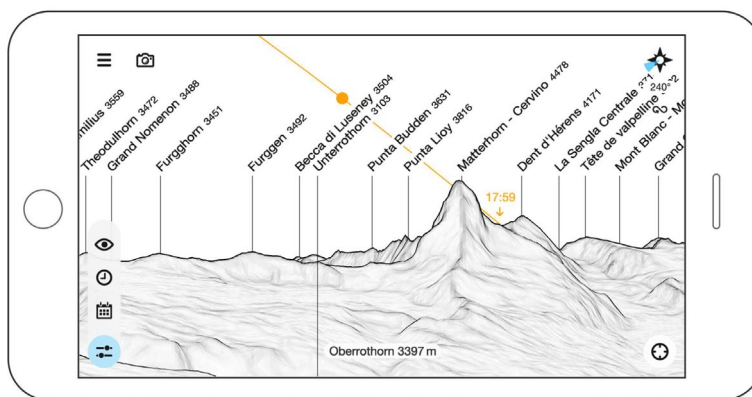
Comme illustré ci-dessus, l'utilisateur prend une photo d'un endroit dans Paris qui lui plaît et, ensuite en fonction de sa position, une image correspondante au lieu où il est mais datée d'il y a quelques années est affichée.

L'utilisateur peut comparer et observer les changements qui sont apparus avec le temps. C'est une manière originale de découvrir un lieu. Les anciennes images sont des cartes postales de l'époque numérisées. Elles sont stockées dans une base de données et les images sont sélectionnées puis affichées sur l'écran grâce aux données de localisation de l'appareil de l'utilisateur. La base de données est actualisée et mise à jour avec les nouvelles

photos et cartes postales rajoutées à chaque fois. Les comparaisons de l'application appelée Hier et Aujourd'hui sont également disponibles sur leur site internet pour les personnes désirant simplement observer les changements de certains lieux sans téléphone mobile. (Geaneat, 2020)

Dans un domaine plus géographique et proche de notre prototype, une application très connue est PeakFinder. Elle permet aux personnes qui l'utilisent d'observer l'horizon au travers de leur téléphone dans le monde entier. Ils ont également des informations supplémentaires grâce à la réalité augmentée. Ils peuvent avec ou sans réseau, afficher les sommets, zoomer sur des endroits moins proéminents, afficher des informations sur le soleil et la lune, utiliser une boussole et bénéficier de pleins d'autres fonctionnalités disponibles. L'application n'est pas gratuite mais est très accessible avec un prix de cinq francs environ. Elle a gagné plusieurs prix et est recommandée par de nombreux médias ou organisations comme National Geographic. C'est un modèle excellent pour un prototype comme le nôtre car ils ont implémenté beaucoup des fonctionnalités intéressantes. Ci-dessous, nous pouvons trouver des exemples de fonctionnalités proposées par PeakFinder. (PeakFinder, 2021)

Figure 13 : Illustration de l'application PeakFinder avec un filtre



Source : Récupéré sur <https://play.google.com/store/apps/details?id=org.peakfinder.area.alps&hl=fr&gl=US>

Figure 14 : Capture d'écran de l'application PeakFinder sans filtre



Source : Récupéré sur <https://play.google.com/store/apps/details?id=org.peakfinder.area.alps&hl=fr&gl=US>

De nombreuses applications à but similaire sont également disponibles sur les différents stores (Peaklens, Peakview, AR Map World Peaks) mais elles sont souvent moins bien cotées. Les remarques récurrentes sont que les données de positions sont trop éloignées de la position réelle et souvent ces applications ne fonctionnent pas sans réseau. Le manque de connexion est quelque chose qui arrive souvent en montagne qui est le terrain adapté à l'utilisation de ces produits.

7. Outils de la réalité augmentée

7.1. Environnements de développement

De nombreux logiciels de développements pour la VR ou AR existent. Nous pouvons citer Amazon Sumerian, Google VR for everyone ou Blender. Cependant, pour nos besoins d'applications nous avons fait une sélection de deux moteurs qui sont présentés ci-dessous afin de définir lequel des deux nous est le plus utile.

7.1.1. Unreal Engine

C'est un moteur de jeu développé par Epic Games. Cette compagnie est très connue pour un de ses jeux nommé Fortnite.

Cette plateforme de développement est également utilisée pour développer des applications multiplateformes en réalité virtuelle et augmentée.

N'ayant pas d'outils adéquats pour créer leurs jeux, les ingénieurs d'Epic Games ont créé Unreal et l'ont amélioré en fonction de leurs besoins lors de leurs développements. En 1998,

une version basique est apparue. La version très light permettait juste la modification de quelques informations et d'attribuer des mouvements à des PNJ.

Au fil des modifications de Unreal et de ses améliorations, la compagnie Epic à mis en place un système de scripts visuels Kismet pour travailler. Cette apparition s'est faite en 2006 avec la version 3 d'Unreal Engine.

Ensuite en 2014, les scripts visuels Kismet ont laissé place aux Blueprints que l'on connaît depuis Unreal Engine 4. Le code source de Unreal est disponible sur Github pour plus de détails.

Le Blueprint permet de travailler différemment que du code pur et dur. Un système de glisser-lâcher (drag and drop dans le milieu informatique ou en anglais) permet à des novices en programmation d'apporter des modifications à tous les étages du jeu. On peut comme cela modifier le son à la suite d'une action comme le mouvement d'un PNJ après une mission très simplement. Pour les plus aguerris, ils peuvent sans autres coder en C++ afin d'apporter des modifications plus précises aux scripts proposés dans les Blueprints. Il faut bien mettre en avant que Unreal Engine 4 est le premier moteur à utiliser cette manière de travailler. Cet aspect de no-code est révolutionnaire dans ce domaine.

La version 5 étant qu'en accès anticipé au moment où nous rédigeons cette thèse, nous détaillons la version 4 de Unreal Engine au niveau du moteur physique. Pour faire interagir convenablement les éléments entre eux, Unreal Engine utilise le logiciel PhysX de Nvidia. Le logiciel permet d'éviter aux développeurs de jeux vidéos de créer leur propre code au niveau de la physique dans les jeux. Un gain de temps et d'argent car les kits de développement sont gratuits pour les usages commerciaux et non commerciaux pour des jeux sur les plateformes comme Microsoft Windows, OS X, Linux, Android et Apple iOS. De nombreux jeux comme la série Tom Clancy's ou différents Batmans utilisent ce moteur physique.

Unreal permet d'optimiser les jeux afin de bénéficier de graphismes réels et de haute qualité sur toutes les plateformes grâce à son moteur graphique. La version 4 permet également de prendre en charge DirectX12, Javascript/WebGL et OpenGL.

UE propose dans ses outils beaucoup d'effets post-traitement pour jouer sur les lumières, les ombres, les sons et autres aspects des jeux pour rendre l'expérience utilisateur meilleure.

Les PNJ et autres éléments qui doivent bouger ou interagir avec le personnage joueur peuvent bénéficier d'une intelligence qui imite les interactions et actions réelles. Cette

intelligence artificielle proposée dans les fonctionnalités de Unreal Engine donne vie aux éléments du jeu et tout peut être ajusté par les développeurs en fonction de leurs besoins.

Pour travailler avec Unreal Engine, les développeurs peuvent avoir plusieurs éditeurs qui sont séparés en fonction des modifications qu'ils apportent. Unreal Editor, par exemple, est l'éditeur principal et permet, sans modification de code, d'ajuster, d'organiser ou modifier des composants de leurs jeux. Depuis cet éditeur, on peut également naviguer vers l'éditeur de matériels ou celui des Blueprints. Pour travailler avec le code C++, nous pouvons utiliser Visual Studio qui est compatible avec Unreal Engine.

Un Marketplace est également disponible afin de télécharger ou acheter des compléments comme des objets, plugins, ambiances sonores et encore bien d'autres choses.

L'utilisation de Unreal Engine est gratuite. Sur leur site internet, nous pouvons trouver deux licences en fonction des besoins. La première, permet juste de créer des projets personnels et surtout d'apprendre. C'est idéal pour les étudiants ou des personnes désirant seulement s'entraîner. (R., Unreal Engine : Tout savoir sur le moteur de jeu d'Epic Games, 2021)

Ensuite, si nous voulons publier notre jeu ou application, nous devons passer à une deuxième licence permettant cela. Elle est toujours gratuite. Par contre, dès que le revenu généré par la vente de l'application dépasse un certain montant, Unreal impose des royalties qui varient en fonction des sommes atteintes.

Ensuite, pour les plus gros développeurs ou entreprises, il y a une première licence payante qui est mise en place sur demande. Il faut contacter Unreal Engine qui proposera une solution sur-mesure pour chaque client.

Il y a également, pour 1'500 dollars par année, par utilisateur, un programme pour les entreprises qui désirent avoir un suivi de projet plus avancé avec des outils proposés en plus de ceux de base. (Unreal Engine, n.d)

7.1.2. Unity

C'est un logiciel qui permet de créer des applications ou jeux vidéo en 2D ou 3D. Il ressemble à d'autres logiciels de modélisation 3D comme Blender. Le programme est composé d'une

partie logicielle et d'une partie programmation. Ce moteur de jeux permet également de créer des applications en VR et AR. C'est un outil multiplateforme qui a été commercialisé en 2005 avec sa première version. Nous pouvons exporter les projets sur PC, Mac, Linux, PS Vita, PlayStation 4, Xbox One, iPhone, iPad, Android, etc... .Il est développé par Unity Technologies et n'a pas cessé d'être mis à jour en fonction des nouveautés et de la demande des jeux vidéos ou applications. En 2021, c'est la version 2021.1.0 qui est la plus stable et la plus avancée. Il y a également des versions en accès anticipé. Unity ne possède pas les Blueprints comme Unreal Engine mais propose de travailler avec du code C#, Javascript ou Boo jusqu'en 2017. Pour développer un jeu avec les scripts, nous pouvons associer Unity avec Visual Studio ou utiliser leur éditeur de développement monoDevelop. Il est très répandu chez les créateurs de jeux par sa facilité d'utilisation.

Dans les jeux connus développés sur Unity, nous pouvons citer Beat Saber et Pokemon Go par exemple.

Unity propose les fonctionnalités adéquates au bon développement de jeux vidéos. Tout ce qui concerne le rendu 3D, la physique de jeu et la détection de collisions.

Un environnement de développement intégré dans Unity permet d'avoir dans une seule GUI les outils de développement nécessaires pour créer un jeu ou une application.

Il y a également un système de drag and drop pour permettre aux développeurs de gagner en temps et en facilité lorsque nous voulons ajuster des propriétés depuis l'éditeur principal sans aller dans le code. C'est surtout lorsque nous manipulons des objets ou scripts pour les associer avec d'autres objets.

Pour les lumières, les sons et autres composants, comme pour Unreal Engine, Unity possède des objets préfabriqués avec des fonctionnalités de base qui peuvent être ajustés au moyen de scripts ou de réglages. Les fonctionnalités sont assez similaires à celles de Unreal Engine.

Il est plus compliqué dans ce moteur de jeux de travailler sans savoir coder. En effet, les principales actions, ajustements ou autres effets sont créés à partir de scripts en langage C#.

On peut également avoir accès à l'Asset Store qui est un shop en ligne d'objets, plugins ou autres packages qui peuvent être achetés ou obtenus gratuitement afin de gagner du temps dans la création d'applications ou de jeux.

Unity peut être obtenu gratuitement mais est limité au niveau de certaines fonctionnalités. Si nous travaillons avec la version gratuite, nous ne pouvons pas bénéficier d'outils

développés par Unity comme des analyses ou diagnostics qui peuvent aider les développeurs. Nous pouvons quand même proposer les applications développées au grand public et les vendre. Pour être éligible à cette licence, il ne faut pas que le revenu du développeur ou de l'entreprise dépasse les 100'000 \$ lors des 12 derniers mois.

Plusieurs autres versions payantes avec des abonnements annuels sont proposées. Dans l'ordre de la moins chère à la plus chère, nous avons la licence Plus, Pro et Enterprise. Il y a aussi des conditions pour bénéficier de chaque licence en fonction du revenu et du nombre de personnes sur le projet. Les licences évoluent surtout avec des outils clouds et collaboratifs supplémentaires à chaque fois. (R., Unity : Tout ce qu'il faut savoir sur le puissant moteur de jeu, 2021)

En résumé, pour un étudiant ou une personne qui débute, la licence gratuite est idéale. Ensuite, pour bénéficier de plus de ressources et ajuster certaines fonctionnalités bloquées, la version Plus à 399\$ est disponible. Pour avoir une meilleure gestion de projet et un suivi plus poussé, il y a une licence Pro à 1800\$. C'est la dernière version avant l'abonnement Enterprise qui est proposé pour les entreprises qui veulent une solution sur-mesure qui répond à tous leurs besoins. Les prix indiqués sont pour une seule licence.

A côté des licences, il y a également la possibilité de s'abonner aux outils collaboratifs Unity Teams qui sont proposés dans certains packs d'abonnement mais également aux personnes qui ont la version gratuite. Lors de l'utilisation de Unity, on possède les outils collaboratifs basiques mais pour 9\$ dollars par mois, on peut débloquer certaines fonctionnalités qui ne sont pas disponibles gratuitement. On peut avoir une plus grande zone de stockage cloud et agrandir l'historique de travail pour un meilleur suivi par exemple. (Unity, 2021)

7.1.3. Récapitulatif et choix pour le prototype

Bien que les deux environnements de travail proposent des outils performants pour faire des projets de bonnes qualités, nous devons choisir l'un ou l'autre environnement.

Dans l'ensemble, il y a beaucoup de similarités entre les deux solutions. Le choix de Unity s'est fait surtout en regardant les préférences personnelles dans le mode de développement par script avec le langage C# et aussi du fait de l'expérience déjà acquise sur cet environnement.

Unity est également plus complet au niveau des fonctionnalités disponibles avec les applications de réalité augmentée. De plus, Unity a une documentation très complète et

utile au niveau de l'AR avec une série de tutoriaux ou d'explications pour se former et apprendre rapidement.

Nous avons, ci-dessous, un résumé des caractéristiques décisives dans la sélection pour chaque environnement.

Tableau 1 : Comparatif entre Unreal Engine 4 et Unity 2021.1

Nom	Unreal Engine 4	Unity 2021.1
Développeur	Epic Games	Unity Technologies
Prix	Gratuit et des licences payantes sont disponibles	Gratuit et des licences payantes sont disponibles
OS	Windows, MacOS, Linux	Windows, MacOS, Linux
Exemple de matériels Windows conseillés pour l'utilisation de la solution	Windows 10 64-bit 32 GB RAM SSD + HDD NVIDIA GeForce RTX 3070 AMD Ryzen 9 5900X	Windows 10, 64-bit 32 GB RAM SSD + HDD NVIDIA GeForce RTX 3070 AMD Ryzen 9 5900X
Langage de programmation favori	C++	C#
IDE disponibles	Visual Studio, Visual Studio Code, JetBrains Rider	Visual Studio, Visual Studio Code, JetBrains Rider
Support	Grande Communauté, documentation officielle sur leur site internet, forums et wikis disponibles, chaîne YouTube	Grande Communauté, documentation officielle sur leur site internet, forums et wikis disponibles, chaîne YouTube
Fonctionnalités supportées (SDK : ARCore)	Réalité augmentée sur les visages et images, ancrage dans le cloud, détection des plans verticaux	Suivi de l'appareil, détection des plans, détection avec nuage de points, détection des visages, détection des images 2D, rayon

		directionnel, flux vidéo
Fonctionnalités supportées (SDK : ARKit)	Détection des images 2D, Détection des objets 3D, détection des visages, expériences persistantes, expériences partagées	Suivi de l'appareil, détection des plans, détection avec nuage de points, détection des visages, détection des images 2D, rayon directionnel, flux vidéo, reconnaissance de l'environnement, détection des images 3D, détection des corps 2D & 3D, segmentation humaine et occlusion, travail collaboratif

Source : Données de l'auteur et ressources internet (Puget Systems, 2021)

7.2. SDK

Le projet est une application mobile Android. Pour nous aider, nous devons choisir un Software Development Kit qui nous permet de bénéficier d'un ensemble d'outils d'aide à la programmation.

Ces logiciels peuvent être importés ou installés sans trop de difficultés depuis la liste de plugins disponibles sur Unity.

Pour notre application, il y a de nombreux SDK disponibles avec leurs caractéristiques propres à chacun. Nous les présenterons ci-dessous et choisirons le kit adéquat pour notre prototype.

7.2.4. AR Foundation

AR Foundation nous permet de travailler avec la réalité virtuelle dans un format multi-plateforme (Android et iOS). Ce SDK regroupe les fonctionnalités d'ARCore et d'ARKit.

L'environnement d'AR Foundation fournit des APIs et des scripts pour des projets qui utilisent les concepts suivants :

- Le suivi du monde

- Permet de suivre la position et l'orientation de l'appareil dans l'espace physique
- La détection de plans
 - Permet la détection des surfaces horizontales et verticales
- Les nuages de points
 - Permet une meilleure détection de l'environnement physique
- Les points de référence
 - Permet de suivre une position et une orientation arbitraires
- L'estimation de la lumière
 - Permet d'estimer les couleurs et la luminosité moyenne dans l'espace physique pour de meilleurs ajustements visuels
- Les sondes d'environnement
 - Permet de générer une carte cubique pour représenter une zone particulière de l'environnement physique
- Le Suivi des visages
 - Permet de détecter et suivre des visages humains
- Le suivi d'images
 - Permet de détecter et suivre des images 2D
- Le suivi d'objets
 - Permet de détecter et suivre des objets 3D

AR Foundation est gratuit et dispose d'une documentation très complète. (Unity, n.d)

7.2.5. ARCore

C'est le SDK gratuit de Google. La plate-forme permet de créer des applications de réalité augmentée fonctionnant sur Android bien que certaines API sont également disponibles sur iOS pour permettre des expériences d'AR partagées. En utilisant ce SDK, on a la possibilité de détecter son environnement, de comprendre le monde et d'interagir avec les informations. Il est assez similaire à ARKit.

Les 3 fonctionnalités mises en avant pour ajouter du contenu virtuel sur le retour de la caméra sont :

- Le suivi de mouvement
 - Permet de comprendre et suivre sa position par rapport au monde
- La compréhension de l'environnement

- Permet au téléphone de détecter la taille et l'emplacement de tous les types de surfaces : surfaces horizontales, verticales et inclinées comme le sol, une table basse ou des murs
- L'estimation de la lumière
 - Permet au téléphone d'estimer les conditions d'éclairage actuelles de l'environnement.

Google résume lui-même ARCore en quatre mots : « see » « move » « light » « play ».
(Google, n.d)

7.2.6. ARKit

C'est "l'équivalent" iOS de ARCore proposé par Apple gratuitement.

AR Kit fournit une boîte à outils aux développeurs iOS pour programmer des applications en réalité augmentée.

Ce SDK propose également des outils pour travailler avec la profondeur de l'image, la position géographique, le Face tracking, l'occlusion et beaucoup d'autres possibilités.

Lors du dernier WWDC, en juin 2021, Apple a présenté sa nouvelle version ARKit 5. De nouvelles fonctionnalités et améliorations ont été apportées depuis la version 4. Nous pouvons citer, par exemple, une amélioration aux positionnements d'informations virtuelles sur des positions réelles et des améliorations lors du suivi du visage et des mouvements. Une nouveauté est l'apparition de Clip Codes au sein des applications qui permettent de déclencher des fonctionnalités comme l'apparition d'un objet virtuel. Cela permet d'avoir une application plus légère et plus rapide. Comme le montre l'exemple ci-dessous, en scannant le code qui fonctionne un peu comme un QRCode, l'utilisateur peut observer d'une façon plus immersive le produit avec sa représentation en 3 dimensions.

Figure 16 : Illustration d'un Clip Code

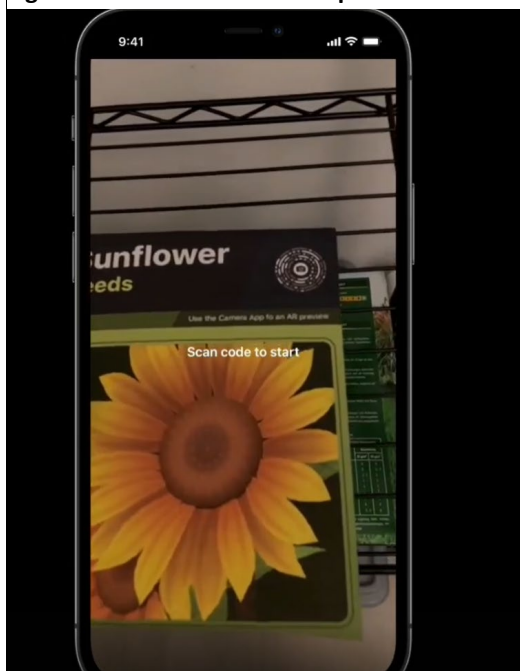
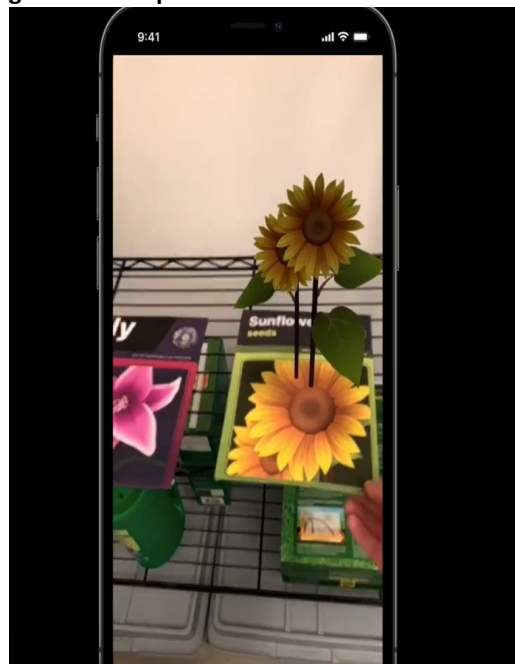


Figure 15 : Illustration d'un objet 3D apparu grâce à un Clip code



Sources : Récupérés sur

https://developer.apple.com/documentation/app_clips/interacting_with_app_clip_codes_in_ar

Les appareils supportés sont ceux qui possèdent les versions iOS 11 ou plus récentes ainsi qu'un processeur A9 ou plus récent également. Sans ces configurations, certaines fonctionnalités ne peuvent fonctionner et limitent l'expérience de réalité augmentée.

Ce kit est un choix adapté pour les créateurs d'applications sur iOS. (Apple, n.d)

7.2.7. Vuforia

Vuforia est le SDK proposé par PTC qui est un fournisseur mondial de technologies dédiées à l'internet des objets.

L'utilisation de Vuforia combinée à Unity permet de créer des applications et jeux pour Android et iOS.

Vuforia est très compétent dans le suivi des objets ou formes qui peut être combiné à d'autres technologies comme le GPS, l'accéléromètre et d'autres encore.

Très utile pour les développeurs qui souhaitent créer des applications de réalité augmentée combinant des objets réelles et virtuelles.

Ce SDK est disponible gratuitement, mais il propose également des versions payantes qui permettent d'accéder à des fonctionnalités avancées. (Vuforia, n.d)

7.2.8. Mapbox

Mapbox propose un SDK pour Unity afin de travailler avec les cartes et données géographiques.

Nous pouvons grâce à lui créer différentes applications mêlant données de positions et utilisation des composants standards de Unity. Mapbox propose d'autres outils sur différentes plateformes qui permettent d'interagir avec leurs APIs pour les services de cartes, de localisations, de navigations ou autres.

Lors de l'installation du SDK, il y a la possibilité d'essayer les fonctionnalités basiques dans des jeux exemples qui peuvent être utilisés ensuite comme projet de base. Nous en avons fait le test avec leur scène appelée AstronautGame. Nous pouvons facilement situé un astronaute virtuel à une position réelle et le faire apparaître sur notre écran lorsque nous scannons cette position avec le smartphone et l'application ouverte.

C'est très facile à faire et à comprendre le fonctionnement des scènes proposées pour bien démarrer.

En fonction des technologies et environnement de travail utilisé sur leur site, Mapbox propose différents SDK afin de répondre aux besoins de tous.

Le SDK est gratuit mais son prix change en fonction du nombre d'utilisateurs mensuels actifs sur l'application. Par exemple, le Maps SDK pour Unity est gratuit jusqu'à 25'000 utilisateurs et ensuite coûte environ quatre francs pour 1'000 utilisateurs de 25'001 à 125'000. Ensuite, les montants varient avec d'autres fourchettes de prix.

Nous pouvons également profiter des différentes APIs qu'ils proposent sans travailler avec leurs SDK. Ils ont également un système de prix évolutif en fonction du nombre de requêtes envoyées à leurs services. (Mapbox, n.d)

7.2.9. Wikitude

Le SDK de Wikitude est également très complet et dispose de nombreuses fonctionnalités déjà présentes dans d'autres kits cités précédemment.

Le SDK permet d'étendre les fonctionnalités des autres kits comme ARKit et ARCore.

En 2020, ils ont annoncé de nouveaux outils pour la branche de la construction comme l'utilisation de modèles 3D et des données du logiciel CAD. Cela offre des possibilités de réalité augmentée à 360° autour d'objets physiques.

Le SDK n'est pas gratuit. Nous pouvons bénéficier d'une version très limitée ou d'un essai gratuit mais ensuite il faut acheter une licence. Différents prix sont proposés en fonction des besoins car ils ont une version Pro 3D, une version Cloud et une version sur demande pour les entreprises. Il faut déboursier aux alentours de 2'700 francs pour bénéficier de la licence à vie sans les mises à jour ou s'inscrire à l'abonnement annuel coûtant environ 3'300 francs qui permet d'avoir un kit régulièrement amélioré. (Wikitude, n.d)

7.2.10. Récapitulatif et choix pour le prototype

Nous avons effectué plusieurs tests et nous sommes aperçus qu'après nos expériences qu'AR Foundation convenait parfaitement.

Fonctionnant sur Android, il est totalement adapté pour le développement de prototype sur cet OS. Ensuite, nous avons apprécié le fait qu'il est gratuit et bénéficie d'un support très complet sur internet aux travers de vidéos, blogs ou encore forums. La documentation est également bien fournie sur le site officiel. Grâce à ses APIs et outils nous étendons les fonctionnalités disponibles d'ARCore qui est aussi utilisé dans ce projet.

Le SDK offre la possibilité de produire une application multiplateforme Android – iOS si le prototype évolue dans ce sens.

Avec une installation très facile, nous pouvons assez vite prendre en main les différents outils proposés.

8. Prototype

8.1. Use case

Comme expliqué lors de la présentation des différentes applications dans le domaine géographique avec de la réalité augmentée, nous proposons un prototype de type PeakFinder qui permet à un utilisateur d'observer le paysage au travers de son smartphone à l'aide d'une application. Notre contrainte principale est l'utilisation de fichiers .hgt en tant que données du projet. Différentes fonctionnalités ont été développées à la suite du « Product Backlog » établi en amont du projet.

Notre application de base nous permet d'avoir le paysage observé avec un filtre de couleur qui démontre l'éloignement des élévations avec l'utilisateur. Ensuite, nous avons transformé

le filtre de couleur en filtre transparent qui représente plus que les bordures des montagnes ou collines. Finalement, une méthode d'affichage de distance a été ajoutée.

Nous présentons ci-dessous chaque fonctionnalité au niveau technique et théorique dans l'ordre chronologique de développement.

8.2. Mise en place de l'environnement de travail

Afin de débiter le développement du prototype, nous avons installé Unity et démarré un projet avec la version 2020.1.16f1. Après plusieurs essais, c'est la version qui s'est montrée la plus adaptée et la plus stable pour notre prototype.

Avec cela, nous avons pu installer les différents packages nécessaires depuis le Package Manager.

Figure 17 : Packages installés dans le projet

▶ AR Foundation	3.1.10	✓
▶ ARCore XR Plugin	3.1.10	✓
▶ ARKit XR Plugin	3.1.10	✓
▶ Test Framework	1.1.19	ⓘ
▶ TextMeshPro	3.0.1	ⓘ
▶ Timeline	1.3.6	ⓘ
▶ Unity Collaborate	1.3.9	✓
▶ Unity UI	1.0.0	✓
▶ Visual Studio Code Editor	1.2.3	✓
▶ Visual Studio Editor	2.0.9	ⓘ

Source : Données de l'auteur

Nous utilisons AR Foundation avec le plugin permettant d'accéder aux fonctionnalités du SDK ARCore. Nous avons également testé quelques fonctionnalités pour iOS donc le plugin ARKit est également installé. Ensuite, comme interface de développement pour nos scripts, nous utilisons Visual Studio tout au long de ce projet. Le débogage s'effectue toujours depuis le fichier de log de notre smartphone Android physique utilisé pour tester l'application.

8.3. Rendu en direct de la caméra

L'objectif de ce point est de permettre à l'utilisateur de l'application d'accéder au rendu en direct de sa caméra lorsqu'il utilise l'application comme le montre la maquette.

Figure 18 : Maquette du rendu en direct de la caméra

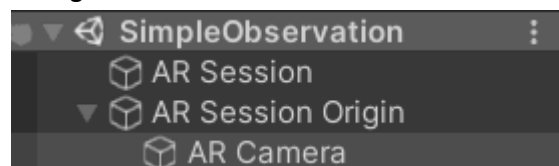


Source : Données de l'auteur

Pour y arriver, nous avons dû créer une scène et y ajouter deux éléments qui sont l'AR Session et l'AR Session origin. Le premier cité permet de contrôler le cycle de vie de notre application en activant ou désactivant la réalité augmentée. Le deuxième objet permet de transposer les différents éléments perçus dans la réalité au sein de notre scène Unity. Comme il y a des échelles différentes entre le monde réel et virtuel, par exemple, AR Session Origin permet de gérer et à adapter cela automatiquement. Nous lui avons attribué l'objet AR Camera qui renvoie directement le retour de la caméra de l'appareil Android utilisant notre application. Elle est la caméra principale de notre projet.

Il n'y a pas de réglages particuliers sur notre objet AR Camera à part l'ajout d'un composant ARCameraBackground qui permet de s'abonner aux événements de la caméra AR et d'afficher différents objets virtuels qui apparaissent plus tard dans le développement sur le rendu en direct de la caméra.

Figure 19 : Architecture de base de la scène



Source : Données de l'auteur

8.4. Superposer le virtuel sur le monde réel

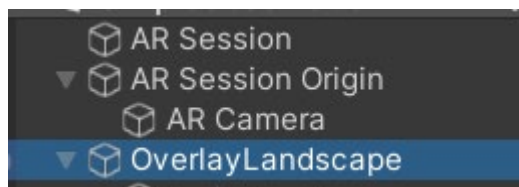
Nous devons facilement pouvoir insérer notre contenu virtuel sur le retour de la caméra. Pour cela, nous avons essayé tout d'abord de le faire avec des images et textes indépendants. Nous avons créé à chaque fois un objet et l'avons rajouté à la scène. Ce

n'était pas optimal et compliqué de gérer toutes les dépendances. Nous avons opté pour la création d'un Canvas qui est l'écran virtuel dans lequel sont insérés les éléments affichés et qui est ensuite appliqué devant le monde réel sur l'écran.

Le terme, en anglais, lorsque l'on parle d'éléments virtuels affichés sur un retour du monde réel est un overlay. Nous avons repris ce terme pour le nom en précisant que c'est un calque qui concerne la partie du paysage.

Dans le réglages, nous avons dû préciser que le Canvas sert d'overlay, lui définir une échelle avec une taille et qu'il serait utilisé avec l'élément décrit précédemment appelé AR Camera.

Figure 20 : Architecture de la scène après ajout de l'overlay

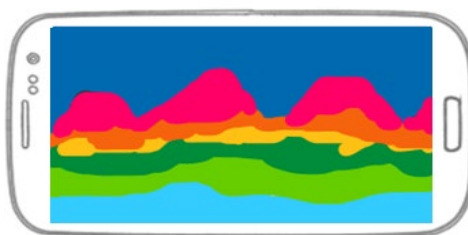


Source : Données de l'auteur

8.5. Créer du contenu dans le Canvas

Nous sommes prêts au niveau des réglages pour ajouter des éléments sur notre rendu en direct. Comme expliqué, nous voulons afficher des éléments virtuels sur le paysage observé par l'utilisateur. La maquette suivante illustre très bien le filtre désiré sur notre rendu en direct.

Figure 21 : Maquette de l'utilisation d'un filtre sur le paysage



Source : Données de l'auteur

8.5.1. Ajouter un élément virtuel dans le Canvas

Premièrement, nous avons pensé que l'élément à afficher serait une image, nous essayons d'intégrer au Canvas une image 2D qui serait ensuite redessinée en fonction de la position de l'utilisateur et de la direction qu'il regarde.

Nous avons commencé par récupérer une image avec un arrière-plan transparent pour nos tests et l'avons rajoutée dans la scène comme objet enfant du Canvas créé précédemment.

Cela n'a pas fonctionné. Nous avons dû transformer notre image en Sprite 2D. C'est un objet qui permet d'afficher des éléments 2D à partir d'images ou de textures.

8.5.2. Dessiner dynamiquement sur le Canvas

Comme expliqué, nous voulons modifier l'image virtuelle en fonction des données géographiques. Nous devons être capables de la redessiner dynamiquement afin de mettre à jour l'overlay avec les mouvements de l'utilisateur.

Pour cela, nous avons dû corriger l'étape précédente. Nous ne devons pas utiliser un Sprite à partir d'une image mais d'une texture qui serait redessinée à chaque mise à jour.

Une texture est utilisée pour définir l'aspect graphique d'élément 3D ou 2D.

Comme premier essai, nous avons créé un script qui permet de créer une texture aléatoire et ensuite de l'appliquer comme overlay sur le rendu en direct de la caméra. Pour cela, nous avons dû créer un nouveau Sprite qui utilise la texture comme représentation graphique et qui est ensuite appliqué sur le Canvas.

L'évolution dynamique fonctionne et l'overlay évolue à chaque mise à jour vraiment très fluidement.

8.6. Implémentation de la partie GIS

Nous avons une base pour créer un overlay dynamique. Nous pouvons commencer l'intégration des éléments concernant la géolocalisation et les données géographiques.

8.6.1. Outils de géolocalisation

Grâce à un script, nous pouvons utiliser les fonctionnalités GPS des appareils utilisant notre application.

Nous avons tout centralisé dans un script appelé GPS qui permet d'avoir une instance utilisable dans tout notre projet. Nous avons accès aux informations de positionnement et d'orientation depuis d'autres scripts en cas de nécessité.

Le script GPS est attaché à la caméra principale pour qu'il fonctionne dès l'initialisation du projet et ensuite se mette à jour tout au long de l'utilisation de l'application.

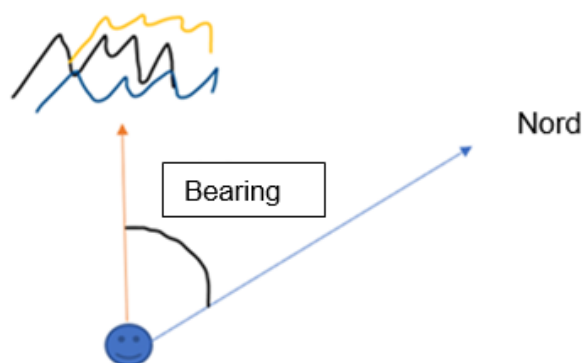
Ces services ne se lancent pas forcément directement. Il se peut qu'il y ait un petit délai avant que les données soient disponibles en fonction des smartphones des utilisateurs. Nous avons laissé une marge de 10 secondes avant l'utilisation d'une de ces données GPS.

Les données se récupèrent facilement car ces services sont disponibles grâce à la classe « LocationService.cs » de Unity et de notre appareil Android. Il suffit de les lancer en activant les outils nécessaires au prototype donc le compas et la location pour notre cas.

la location reprend la position de l'appareil et fourni la latitude, la longitude et l'altitude de l'appareil.

Le compas permet de savoir dans quelle direction que nous regardons. Nous récupérons grâce à lui la direction relative par rapport au vrai nord ou au nord géographique. Cela se sait à partir de l'angle appelé Bearing ou Relèvement en français.

Figure 22 : Illustration de l'angle appelé Bearing



Source : Données de l'auteur

Cet angle se calcule depuis le vrai nord dans notre projet et se calcule dans le sens horaire. Il est compris entre 0 et 359.999 degrés. Par exemple, 90 degrés est l'est, 135 degrés le sud-est et 180 degrés le sud.

8.6.2. Données géographiques

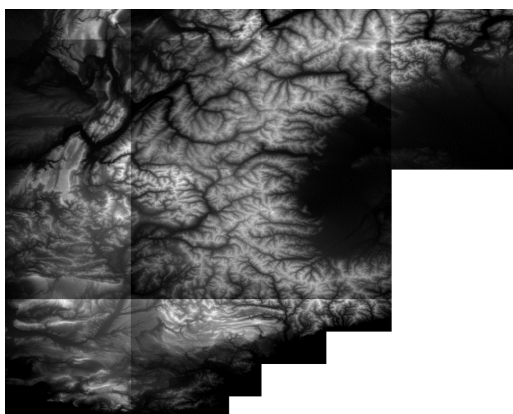
Il n'y a pas de bases de données en ligne dans ce projet. Les informations sur le paysage pour créer l'overlay proviennent de fichiers .hgt.

Ces fichiers sont en libre accès sur internet et représentent l'élévation des terrains dans certaines régions du monde.

La mission Shuttle Radar Topography Mission menée par la NASA et la NGA dans les années 2000, a permis de recueillir, depuis l'espace, les modèles numériques de terrain (MNT) du globe terrestre. Il y a plusieurs types de fichiers disponibles suivant les résolutions voulues. (de Ferranti & Hormann, 2014)

Pour notre projet, nous utilisons les fichiers qui représentent les Alpes suisses.

Figure 23 : Représentation visuelle des données de nos fichiers .hgt



Source : Données de l'auteur

Les fichiers .hgt ont directement été intégrés au projet en les glissant dans le dossier des « StreamingAssets » qui permet d'utiliser des fichiers, données ou autres informations dynamiquement pendant l'utilisation de l'application.

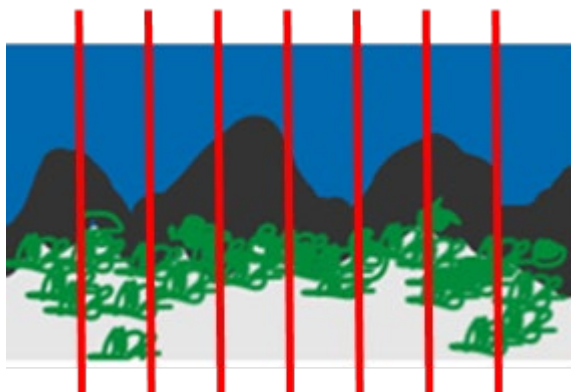
8.7. Texture représentant le paysage

Nous avons les paramètres et données nécessaires pour générer une texture en fonction de la position de l'utilisateur. Pour cette tâche, nous utilisons la classe « Panorama.cs ». Ce script a été fourni par Monsieur Antoine Widmer et réadaptée en fonction des besoins durant le développement du prototype.

La classe « Panorama.cs » permet le traitement des fichiers .hgt en fonction des paramètres de l'appareil de l'utilisateur pour créer un tableau de bytes. Ces bytes sont, en fait, la représentation numérique des couleurs du paysage. Ces couleurs assemblées forment une texture qui est ensuite appliquée sur le rendu du Canvas.

Le principe du traitement est que nous définissons des paramètres représentant l'endroit et sur quelle distance nous devons extraire les couleurs des élévations présentes sur nos fichiers .hgt. Ensuite, notre script échantillonne les dimensions sur toute la largeur en stries verticales tout en récupérant les couleurs de chaque pixel. Puis une méthode les assemble et nous retournons un tableau de bytes contenant les couleurs correspondantes.

Figure 24 : Maquette représentant le paysage découpé par strie



Source : Données de l'auteur

Nous présentons, dans le point suivant, les fonctionnalités clés de la transformation des données en couleurs puis en overlay.

8.7.1. Paramètres de base

Pour que le prototype fonctionne, des paramètres doivent être fixés. Nous n'avons pas pu tout rendre dynamique comme souhaité au début du projet. Certaines limites comme la zone couverte par les fichiers .hgt nous obligent à restreindre l'utilisation de l'application à un endroit. Depuis cet endroit, l'utilisateur peut quand même faire varier la direction vers laquelle il regarde pour mettre à jour le calque du paysage.

Nous situons l'utilisateur à une position proche de Sion avec les coordonnées suivantes :

- **Latitude** : 46.2941311
- **Longitude** : 7.5335362
- **Altitude** : 600
- **Bearing** : Compas du smartphone

Ensuite, des mesures sont établies par défaut afin que nous puissions lire les élévations des fichiers correctement par exemple. Certaines de ces mesures sont exprimées dans le système sexagésimal ou appelé également « DMS ». C'est nécessaire lorsque nous travaillons avec des coordonnées géographiques.

Ensuite, dans la classe « Panorama.cs », en plus des données géographiques statiques comme le rayon de la terre et les paramètres nécessaires aux fonctions utilisant le système DMS, nous devons déterminer la portée de notre observation. C'est-à-dire, jusqu'à combien de mètres nous allons échantillonner nos fichiers .hgt pour créer notre texture. Il est fixé à 25'000 mètres. Puis, le dernier autre paramètre important est le choix de l'angle d'échantillonnage qui permet de délimiter la distance de traitement de nos fichiers en largeur cette fois-ci. Cet angle est fixé à 120 degrés.

De nombreux tests ont été effectués pour trouver le meilleur paramétrage possible. Si un critère est défectueux, l'application peut crasher.

8.7.2. Accès aux fichiers du dossier StreamingAssets

Nous avons eu des problèmes à ce niveau. Une méthode existante dans la classe « Panorama.cs » lisait les fichiers qui étaient stockés en local sur une machine. Pour faire fonctionner l'application sur un appareil externe et utiliser les fichiers du dossier StreamingAssets, nous avons dû installer un module présent sur l'Asset Store de Unity.

Le plugin trouvé est « Better Streaming Assets ». Il permet la lecture de ces fichiers en toute sécurité et est très simple d'utilisation.

Après l'avoir installé, il suffit de l'initialiser avant l'appel d'une des méthodes qu'il fournit. Nous l'avons fait juste avant de débiter la création de notre texture pour être sûr qu'il soit bien démarré.

Ensuite, en fonction de la latitude et longitude d'une position, nous choisissons le fichier adéquat. Les noms des fichiers sont établis de manière à ce que nous puissions les trouver facilement avec les données géographiques. Ils sont composés d'une partie Nord ou Sud puis Ouest et Est variant selon la latitude et longitude.

8.7.3. Créer la texture représentant le paysage

Nous avons toutes les informations nécessaires pour extraire les couleurs de chaque position présente sur les fichiers filtrés. Nous récupérons les reliefs sur une distance grâce à des

méthodes qui étaient déjà codées et qui n'ont pas été changées. Le programme parcourt les fichiers .hgt en effectuant des sauts d'une certaine taille sur les latitudes et les longitudes. Les stries sont comme cela régulières.

Il est également important de préciser que l'on récupère la hauteur des points trouvés grâce aux élévations de ceux-ci sur les fichiers. Grâce à cela, on peut dessiner les reliefs et contours sur notre texture 2D en plus des couleurs représentant la distance des points depuis l'utilisateur.

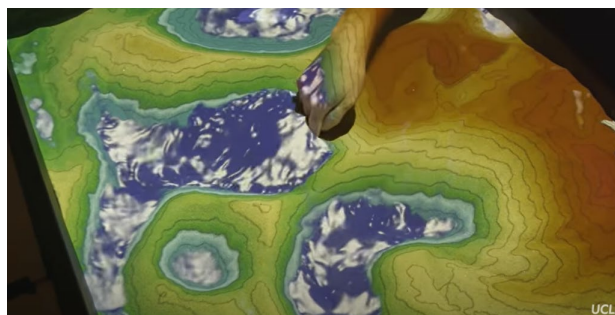
Ensuite, dès que tous les points sont ordrés avec leur hauteur, position et autres informations utiles, nous créons les couleurs. Cette partie se fait dans une fonction qui n'a pas non plus été modifiée dans son fonctionnement. Elle permet de contrôler quel point apparaît après un autre et de vérifier que le point se situe encore sur le paysage que nous voulons afficher ainsi que si c'est une limite de relief ou l'océan.

8.7.4. HSV to RGB

Lorsqu'un point est dans le champs voulu, une couleur doit lui être attribuée. Nous utilisons le principe des « Heat Map » ou cartes de chaleur en français. Cela permet de représenter des données à l'aide de couleurs sans avoir à préciser de hauteur, longueur ou distance. Beaucoup d'exemples sont disponibles sur internet mais le plus concret dans notre thématique est le bac à sable en réalité augmentée créé par l'université de Los Angeles.

Il permet de représenter les reliefs du sable avec des couleurs pour exprimer la différence de hauteur entre chaque point.

Figure 25 : Photo d'une Sandbox avec une Heat Map



Source : Récupéré sur <https://www.youtube.com/watch?v=CE1B7tdGCw0>

De plus, à chaque fois que de nouveaux reliefs sont créés dans le bac, le programme remet à jours les couleurs et les reporte à nouveau sur le sable à l'aide d'un projecteur.

Pour notre projet, nous utilisons le même principe. Les couleurs sont calculées par rapport au point précédent afin de présenter la distance de chacun au fur et à mesure que l'on s'éloigne de l'utilisateur.

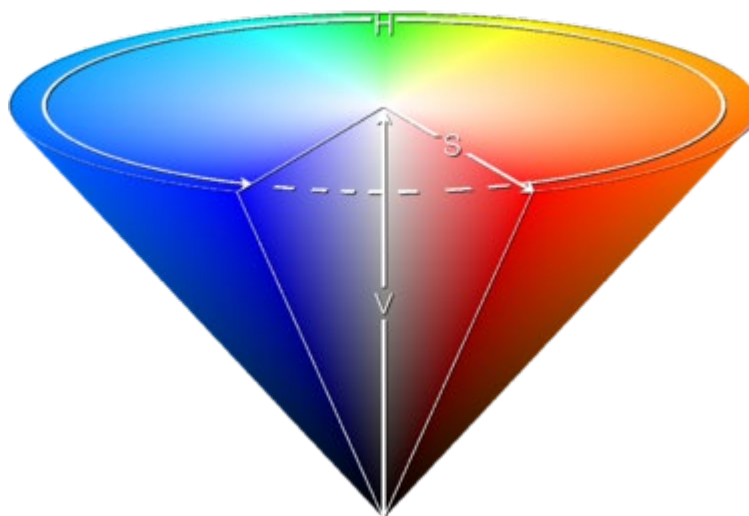
Pour attribuer une couleur aux points géographiques de notre paysage scanné, nous avons trois possibilités. Premièrement c'est une bordure donc nous lui attribuons la couleur noire. Deuxièmement, nous traitons le point à l'aide d'une conversion de HSV (en français, Teinte-Saturation-Valeur) en couleur RGB. Finalement, certains points sont moins hauts que la hauteur totale du cadre, il faut combler ces espaces avec la couleur blanche. Ce sont des zones vides par rapport à nos données géographiques.

Le traitement HSV est un modèle qui permet de gérer des couleurs en informatique. En résumé, grâce à une roue de couleur, on peut exprimer une information. Dans notre cas, l'éloignement des points par rapport à l'utilisateur. Les couleurs sont composées de trois paramètres qui sont la teinte, la saturation et la brillance. (UVED, n.d)

La teinte permet de trouver une couleur sur le cercle des couleurs par rapport à l'angle choisi entre 0 et 360°. La saturation, quant à elle, varie entre 0 et 100% afin de définir l'intensité de la couleur. Elle permet de rendre la couleur plus ou moins fade selon la valeur de saturation. Finalement, la valeur représente la brillance de la couleur. Elle varie entre 0 et 100%. Plus la valeur est haute, plus la couleur est lumineuse.

En résumé, nous pouvons visualiser de manière conique la répartition et possibilité de couleurs pour notre projet.

Figure 26 : Représentation conique du modèle HSV



Source : Récupéré sur https://fr.wikipedia.org/wiki/Teinte_Saturation_Valeur

Pour trouver une couleur, nous avons une méthode qui utilise justement la teinte, la saturation et la valeur de chaque point. Une méthode était présente dans la classe de base.

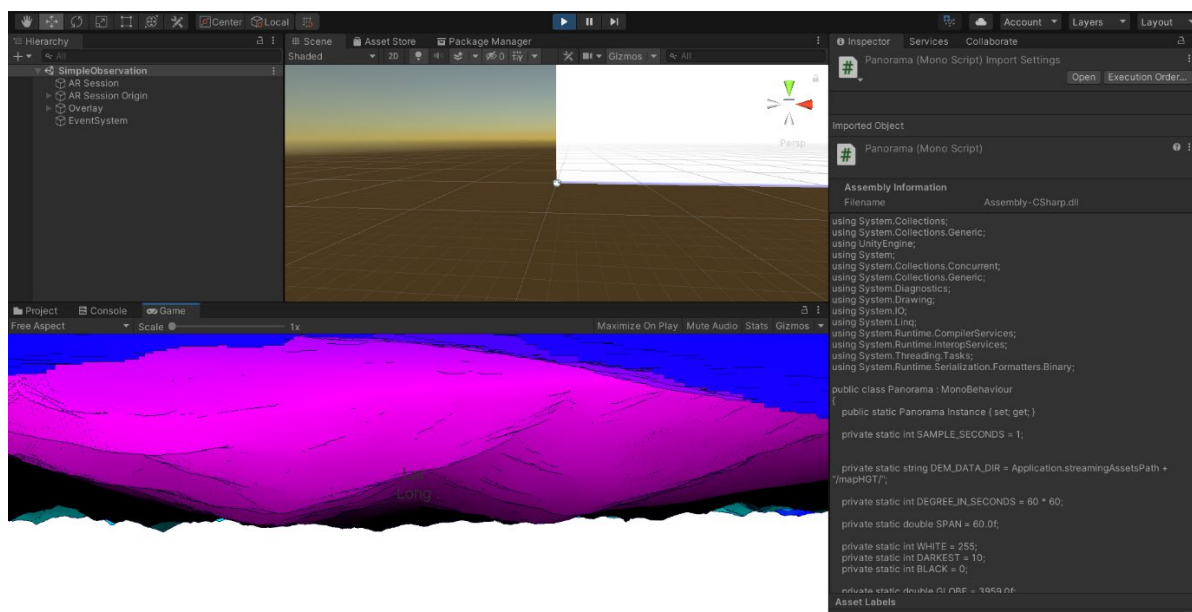
Les paramètres de la méthode proviennent d'informations acquises précédemment comme cela :

- **Teinte** : Valeur définie grâce à l'ordre d'apparition du point par rapport à la taille de la liste totale des points à traiter puis transformée en degrés.
- **Saturation** : Valeur directement définie à 1.0f
- **Valeur** : Valeur directement définie à 1.0f

A la suite du traitement, nous obtenons une couleur RGB qui est organisée par rapport à sa position dans le tableau de bytes que nous obtenons de la classe « Panorama.cs ».

Si nous appliquons directement le tableau de bytes sur la texture pour notre overlay dans notre application, nous obtenons un paysage coloré mais à l'envers.

Figure 27 : Rendu après création de la texture pour l'overlay



Source : Données de l'auteur

Nous définissons un cadre dans lequel les points sont sélectionnés et transformés. Notre classe « Panorama.cs » commence à traiter une image depuis un point avec des coordonnées $x = 0$ et $y = 0$ (en bas à gauche) jusqu'à ce qu'elle ait parcouru tout le cadre défini (tout en haut à droite). Cependant, quand nous appliquons la texture sur notre application, les couleurs sont inversées car les coordonnées ne sont pas dans le même ordre sur un appareil Android.

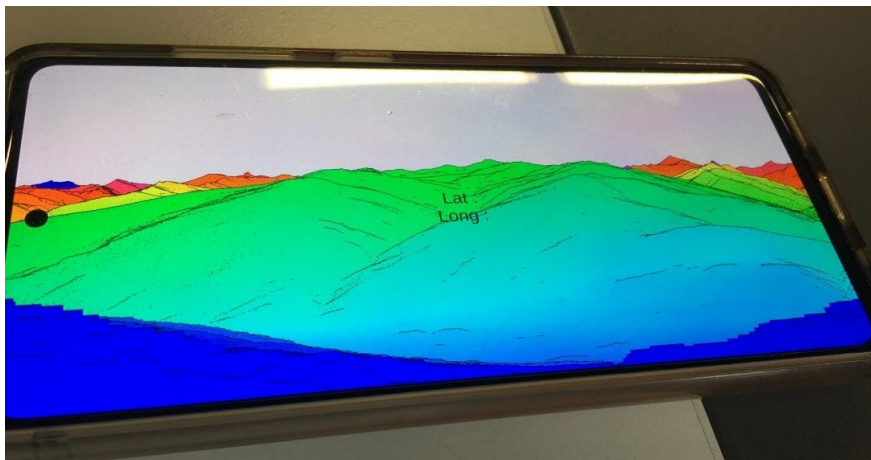
Figure 28 : Disposition des pixels sur un écran de smartphone Android



Source : Récupéré sur <https://stackoverflow.com/questions/11483345/how-do-android-screen-coordinates-work>

La solution trouvée a été de rajouter une ligne de code qui inverse le tableau de bytes avant de le retourner dans la classe qui dessine la texture. Si nous appliquons à nouveau le tableau de bytes à la texture pour contrôler le rendu, nous avons cette fois-ci un paysage bien représenté.

Figure 29 : Texture mise à l'endroit



Source : Données de l'auteur

8.7.5. Traitement des couleurs

Nous recevons un tableau de bytes de la classe « Panorama.cs » dans la classe « DrawTexture.cs » pour construire la texture.

Comme le montre l'image au point précédent, nous avons réussi la représentation du paysage avec les échelles de couleurs. Cependant, nous voulons appliquer le calque sur le rendu de la caméra. Pour cela, il faut éliminer le blanc qui couvre le paysage réel afin que notre overlay n'occupe pas tout l'écran. Pour cela, nous avons découpé le tableau de bytes pour récupérer les couleurs au format ARGB qu'ils représentent. Chaque couleur est ensuite

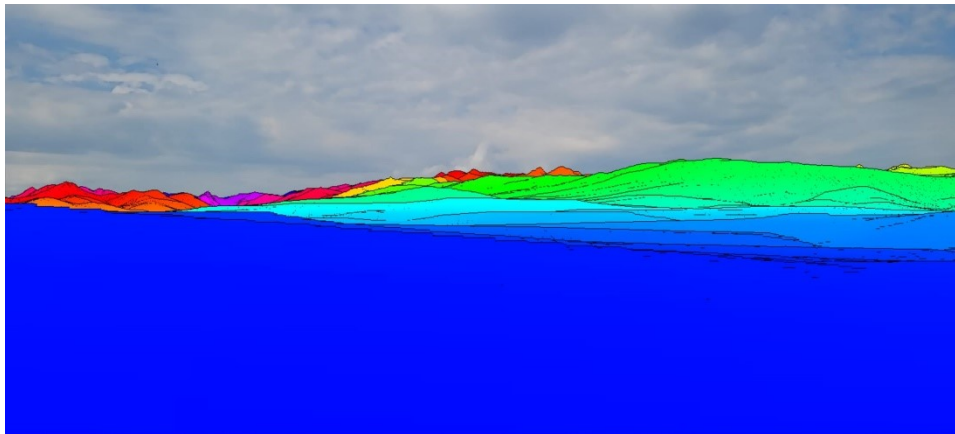
vérifiée afin de voir si c'est du blanc ou non. Si oui, nous utilisons une fonction disponible avec les couleurs de ce format pour la rendre transparente. Si non, nous la laissons telle quelle. Ensuite, la couleur est stockée dans un tableau et nous répétons l'opération jusqu'à ce que tous les bytes aient été traités.

Le résultat est que nous avons toutes les couleurs nécessaires pour appliquer un overlay sur le rendu de la caméra.

8.7.6. Création de l'overlay

Tout d'abord, une texture vide est créée au même format que le cadre de traitement des points dans la classe « Panorama.cs ». Le format de la texture choisi est le même que les couleurs obtenues par le tableau de bytes. Ensuite, nous pouvons utiliser les propriétés de l'objet texture pour lui appliquer le tableau de couleurs afin de modifier son aspect graphique. Ensuite, un objet graphique 2D Sprite est créé aux mêmes dimensions que la texture pour que nous puissions lui appliquer l'aspect de la texture. Toute cette partie se déroule en parallèle du traitement de la transparence de la couleur blanche dans la classe « DrawTexture.cs ». Nous pouvons comme cela créer un overlay directement applicable sur le rendu de la caméra.

Figure 30 : Overlay avec rendu de la caméra en direct



Source : Données de l'auteur

8.7.7. Transparence globale

En seconde attention, nous avons rajouté une nouvelle fonctionnalité sur notre prototype afin de proposer aux utilisateurs un filtre plus avancé.

Il s'agit de rendre également transparent le calque lorsqu'il représente une couleur et de ne garder que les bordures.

Pour arriver à faire cela, nous avons juste rajouté une étape après le traitement de la transparence pour le blanc. Nous utilisons la même méthode mais l'appliquons lorsque l'on détecte une couleur autre que le noir qui représente les bordures. Cela permet d'avoir les

élevations mises en évidence. Pour plus de visibilité, nous avons utilisé l'application avec un fond blanc derrière pour les images d'illustrations mais le filtre fonctionne parfaitement avec le rendu de la caméra.

Figure 31 : Overlay avec transparence complète



Source : Données de l'auteur

8.8. Distance de l'utilisateur par rapport à un point

Nous avons remarqué que nous perdons l'utilité principale des couleurs lorsque nous appliquons le filtre transparent. Elles servent à indiquer l'échelle de distance. Nous avons décidé d'ajouter une nouvelle fonctionnalité au prototype. Nous voulons que lorsque l'utilisateur clique sur un point de l'overlay, ça lui indique la distance à laquelle il se situe par rapport à celui-ci.

Plusieurs essais ont été nécessaires avant de trouver la façon la plus précise.

L'idée de base est qu'il faut trouver la couleur du point choisi afin de faire la transformation RGB vers HSV pour remonter jusqu'à la distance totale. Une valeur appelée « Depth » dans la classe « Panorama.cs » est la profondeur qui correspond à chaque point. En trouvant tous les points qui sont sur la chemin vers le point choisi, nous pouvons additionner un nombre défini de « Depth » et ainsi avoir la distance totale par rapport à l'utilisateur.

8.8.1. Tentative 1

Premièrement, nous avons essayé d'utiliser la fonction appelé « GetPixel » fournie par l'objet texture que l'on utilise pour créer l'aspect graphique de l'overlay. La fonction retourne une couleur par rapport aux coordonnées x et y du point choisi. Par exemple, un point se situant aux coordonnées 1'300 x et 120 y sur l'écran aura son équivalence en couleurs sur la texture aux mêmes coordonnées.

Des problèmes de précisions sont apparues. Nous n'arrivons pas à retourner les couleurs à tous les coups.

Pour combler ce manque de précision, nous avons récupéré sur internet une classe « Zoom.cs » pour gagner en précision. Des adaptations ont été faites pour qu'elle fonctionne sur notre projet. Il s'est produit l'effet inverse. Nous avons perdu en précision car le décalage entre les coordonnées de l'écran et leurs équivalences sur la texture étaient encore plus faussées par l'effet de zoom. La classe n'est pas utilisée mais nous l'avons laissée dans le projet car elle pourrait être utile pour de prochaines fonctionnalités. (Weimann, 2017)

Pour utiliser cette fonction de récupération de couleurs par coordonnées, il faudrait que la texture et l'écran soient identiques au niveau du nombre de pixels et de leur positionnement sinon il y aurait encore des décalages et des erreurs.

Dans notre cas ce n'est pas possible. Nous expliquons dans les limites du prototype pourquoi nous n'arrivons pas à mettre l'overlay à la même taille que l'écran.

Il n'est également plus possible d'utiliser la méthode « GetPixel » si nous appliquons un filtre de couleurs à notre texture comme dans notre cas avec la transparence.

8.8.2. Tentative 2

Pour corriger les erreurs de notre premier essai, nous avons créé, lors de la construction de la texture avec le tableau de bytes, un tableau avec la position x et y des couleurs ainsi que leur format. Nous pouvons comme cela trouver la couleur correspondante à un point sans problème d'échelle.

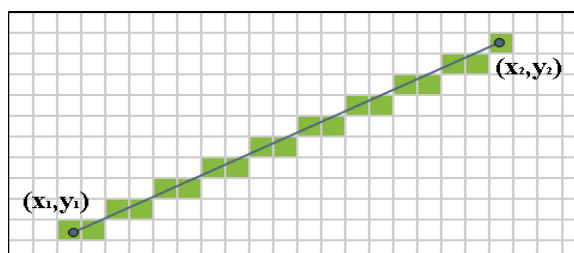
Lorsqu'un utilisateur choisit un point nous avons une coordonnée x et y qui permet de chercher dans le tableau précédemment cité la couleur du pixel.

Ensuite, nous utilisons une méthode pour transformer la couleur RGB en HSV puis inversons les calculs qui sont faits pour transformer les positions sur les fichiers .hgt en couleurs. Nous pouvons trouver la profondeur du point sélectionné par l'utilisateur.

A cette étape il nous manque tous les points qui sont avant celui choisi pour cumuler toutes les profondeurs et obtenir l'éloignement total.

Pour arriver à cela, nous avons eu l'idée d'utiliser l'algorithme de tracé de segment de Bresenham. Après des recherches dans les algorithmes qui permettent de trouver le plus court chemin entre deux points nous avons pensé que c'est le plus adapté à notre problème.

Figure 32 : Illustration graphique de l'algorithme de tracé de segment de Bresenham



Source :

sur https://fr.wikipedia.org/wiki/Algorithme_de_trac%C3%A9_de_segment_de_Bresenham#:~:text=L'algorithme%20de%20trac%C3%A9%20de,attach%C3%A9%20%C3%A0%20une%20console%20texte.

Récupéré

En résumé, nous choisissons un point de départ et d'arrivée puis l'algorithme trace une ligne imaginaire la plus courte possible et retourne les points qu'elle traverse. Des variantes permettent de retourner un nombre de points définis au préalable.

Dans notre cas, nous avons défini le point de départ comme étant le milieu de la texture qui est la position théorique de l'utilisateur sur l'écran et le point d'arrivée comme étant le point choisi par l'utilisateur. Si nous avons quatre points qui sont retournés avec cet algorithme, il suffit de remonter de la couleur jusqu'à la profondeur du point et de répéter cela pour toutes les coordonnées trouvées. Finalement, nous n'avons qu'à cumuler toutes les profondeurs pour trouver la distance totale.

Nous sommes arrivés à des résultats mais beaucoup d'erreurs sont sorties. Nous avons eu un problème au niveau de la redondance des couleurs. Pour cela, nous avons dû trier les couleurs des points retournés afin qu'il n'y ait plus de répétitions car les mêmes couleurs ont la même distance.

Au final, des imprécisions au niveau des résultats retournés sont apparus. Nous n'avons pas réussi à les corriger et avons cherché une autre solution.

8.8.3. Solution trouvée

Pour gagner en précision, nous avons pensé à enregistrer, lors du traitement des élévations dans la classe « Panorama.cs », un tableau des couleurs avec leur profondeur. Cela nous permet d'avoir une référence de l'éloignement pour chaque couleur. Nous ne prenons que les couleurs uniques afin d'éviter les répétitions inutiles.

Nous avons deux tableaux de données comme ceci :

Tableau 1 : coordonnée x – coordonnée y – couleur

Tableau 2 : couleur – profondeur

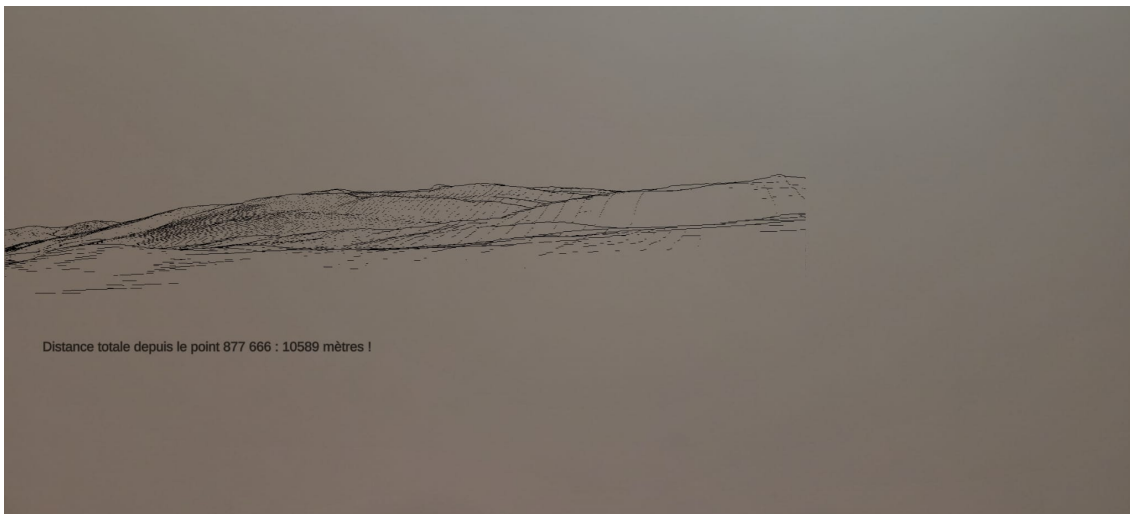
Le principe est simple. Nous récupérons la position sur l'écran du point que l'utilisateur sélectionne. Grâce à ses coordonnées nous interrogeons le Tableau 1 et trouvons la couleur correspondante. Puis, nous interrogeons le Tableau 2 pour en avoir la profondeur.

Il nous suffit de poser une équation pour trouver une estimation de la distance proportionnellement aux autres couleurs. Nous savons que la distance maximum est la portée que nous avons définie comme paramètre dans la classe Panorama. Cette distance est appelée « viewrange » dans notre script. Nous savons également le nombre total de couleurs que nous avons. La dernière est la plus éloignée. Comme nous avons dû inverser le tableau de bytes pour créer la texture, nous inversons également la valeur de la distance du point sélectionné. Pour cela, nous soustrayons cette valeur à la distance maximum.

$$\text{distance du point choisi} = \text{viewrange} - (\text{distance de la couleur trouvée} * \text{distance totale} / \text{nombre totale de couleurs})$$

Avec cela, nous arrivons à des valeurs qui semblent cohérentes. Nous pouvons même, lors de la création de la texture, enregistrer les couleurs dans le tableau Tableau 1 puis gérer la transparence comme expliqué plus haut dans ce rapport afin d'avoir un filtre supplémentaire tout en gardant la fonctionnalité de distance.

Figure 33 : Capture d'écran du prototype avec distance affichée et transparence intégrée



Source : Données de l'auteur

Cette méthode a un défaut majeur qui est que nous ne pouvons pas cliquer sur les bordures noires et calculer leur distance comme elles apparaissent à des profondeurs différentes. De plus, parfois elles sont sélectionnées car elles ne sont pas très visibles sans zoom et empêchent l'utilisateur de sélectionner un point valide.

Pour les éviter, nous pourrions rajouter la classe permettant d'agrandir le Canvas disponible dans le projet et affecter les rapports de zoom à la position des pixels sur la texture pour qu'ils correspondent aux coordonnées du point choisi.

Par manque de temps, cela n'a pas été testé sur le prototype.

8.9. Limitations du prototype

Le prototype, bien que fonctionnel, a beaucoup de limitations qui le ralentissent ou qui déclenchent des erreurs.

Premièrement la classe « Panorama.cs » était une version pour une utilisation sur un ordinateur avec des fichiers en locaux et des bibliothèques fournies par Visual Studio. Le passage sur Unity et l'utilisation de l'application avec un Android ont passablement ralenti tous les calculs et des méthodes très rapides ont dû être modifiées pour fonctionner avec les bibliothèques de Unity. Ces modifications ont rajouté des étapes à de nombreuses reprises ce qui a encore ralenti le traitement des fichiers .hgt. Notre application prend du temps à se lancer car nous devons laisser le temps aux services de localisation du smartphone de s'activer puis nous lançons le traitement de l'overlay. Comme expliqué, nous mettons un délai de 10 secondes avant le premier traitement. Ensuite, la création du calque prend également du

temps suivant les configurations du téléphone de l'utilisateur. Lorsque l'application crée l'overlay, nous ne pouvons pas utiliser l'application. Pour permettre à l'utilisateur d'essayer les fonctionnalités, nous empêchons l'application de relancer un traitement avec de nouveaux paramètres avant 60 secondes. Nous nous retrouvons avec un prototype très lent. Nous avons effectué différents tests et établi un tableau récapitulatif disponible dans le point « Tests » de cette thèse.

Ensuite, le plus gros problème est que les fichiers .hgt testés ont une zone limitée. Il s'agit des Alpes valaisannes. Des essais pour élargir le champ de l'application ont été faits mais l'application devenait trop lourde pour le smartphone et ralentit le tout. Nous devons toujours nous assurer que l'utilisateur observe des endroits qui sont disponibles sur nos fichiers lors de l'utilisation de notre prototype sinon le calque n'est plus uniforme et n'est plus représentatif du paysage. Il se peut, en fonction de la portée totale indiquée dans la classe Panorama combinée à la direction observée par l'utilisateur, que nous sortions des fichiers et obtenions un crash du prototype.

Figure 34 : Overlay si la position de l'utilisateur est hors des fichiers



Source : Données de l'auteur

Nous avons trouvé un lieu proche de Sion qui permet de réduire ces erreurs. De plus, un autre facteur rentre en jeu. Il s'agit de l'angle de vision que nous définissons également dans la classe Panorama.cs. Beaucoup d'erreurs sont apparues lorsque certains paramètres étaient indiqués.

Pour corriger provisoirement cela, nous avons, comme expliqué précédemment, placé l'utilisateur à une position et nous lui permettons juste de changer d'orientation pour observer différentes directions. Nous avons également testé différents paramètres au niveau de l'angle de vision et la distance observée maximum pour choisir la combinaison émettant le moins d'erreurs lors de l'utilisation du prototype. Tous ces paramètres décrits précédemment servent à dénombrer la taille du tableau de couleurs que l'on retourne pour créer la texture. Les erreurs interviennent lorsque nous débordons. Le nombre de couleurs déterminent également la taille de la texture. C'est pourquoi, nous devons avoir assez de couleurs pour remplir la texture sinon une erreur nous indiquera qu'il en manque et l'inverse

est également possible. Nous n'avons pas trouvé l'équation parfaite pour que nous puissions créer une texture qui est à la largeur de l'écran du smartphone de l'utilisateur. De ce fait, notre overlay a une taille définie par avance.

La prochaine limitation combine le problème de taille de l'overlay avec la dernière fonctionnalité ajoutée qui est le calcul de distance. Nous avons trouvé qu'il était possible d'étendre le Canvas pour qu'il puisse être sur tout l'écran de l'utilisateur même si la texture n'a pas les dimensions maximales de l'écran. Cependant, lorsque nous avons voulu récupérer les coordonnées des points choisis pour connaître sa distance, nous n'avions plus de correspondance.

Comme démontré sur la prochaine image, notre texture possède la taille du rectangle bleu et peut être étendue sur toute la surface comme l'illustre le rectangle orange grâce au Canvas qui l'utilise. La recherche de distance totale ne fonctionne que si la sélection se fait dans une surface au mêmes proportions taille Canvas et texture.

Figure 35 : Illustration de différentes surfaces possibles



Source : Données de l'auteur

La dernière limitation concerne entièrement la fonctionnalité de calcul de distance totale. Lorsque nous gardons une référence de profondeur pour chaque couleur, il y a des traitements qui se font en parallèle. De ce fait, il est possible que des fonctions s'exécutent avant une autre dépendamment de la vitesse de chaque processus.

Nous vérifions que chaque couleur ne soit pas déjà dans le tableau pour éviter les redondances car une couleur a toujours la même profondeur. C'est à cette étape que des erreurs peuvent survenir. Plusieurs processus s'exécutent en parallèle. De ce fait, un processus essaie de vérifier si la couleur est déjà dans le tableau mais il n'arrive pas à obtenir la taille de la liste de couleurs pour la parcourir en entier car des nouvelles couleurs sont rajoutées simultanément.

Figure 36 : Erreur de recherche dans le tableau de couleurs



```
[23:49:09] IndexOutOfRangeException: Index was outside the bounds of the array.
(wrapper stelemref) System.Object.virt_stelemref_class_small_iddepth(intptr,object)
```

Source : Données de l'auteur

Elles ne font pas crasher le prototype mais la fonction de distance totale n'est plus disponible. Nous nous retrouvons avec une couleur qui est introuvable car elle n'a pas été rajoutée dans le tableau de référence suite à l'erreur expliquée précédemment.

Figure 37 : Couleur du point choisi par l'utilisateur introuvable

```
07-29 20:02:35.856 27731 30227 E Unity : (Filename: <a6a0db324ac34f8292330f65930dd875> Line: 0)
07-29 20:02:35.856 27731 30227 E Unity :
07-29 20:02:35.880 27731 30227 E Unity : NullReferenceException: Object reference not set to an instance of an object
07-29 20:02:35.880 27731 30227 E Unity : at UpdateDistanceText.Update () [0x0004d] in <a6a0db324ac34f8292330f65930dd875>:0
07-29 20:02:35.880 27731 30227 E Unity :
07-29 20:02:35.880 27731 30227 E Unity : (Filename: <a6a0db324ac34f8292330f65930dd875> Line: 0)
07-29 20:02:35.880 27731 30227 E Unity :
07-29 20:02:35.922 27731 30227 E Unity : NullReferenceException: Object reference not set to an instance of an object
07-29 20:02:35.922 27731 30227 E Unity : at UpdateDistanceText.Update () [0x0004d] in <a6a0db324ac34f8292330f65930dd875>:0
07-29 20:02:35.922 27731 30227 E Unity :
```

Source : Données de l'auteur

Cette erreur intervient aléatoirement en fonction du nombre de couleurs à traiter. Une solution trouvée pour combler ce problème est de relancer l'application en changeant de direction observée. Le paysage traité change et la représentation des couleurs également.

9. Tests du prototype

Afin de connaître les différents temps d'exécutions des différentes fonctionnalités, nous avons effectué des tests de l'application à différents stades de développement. L'appareil de test est celui qui est utilisé lors de l'élaboration du prototype. Il s'agit d'un Samsung Galaxy S20 FE 5G. Nous avons comparé les essais sur le smartphone à ceux obtenus avec le rendu obtenu sur Unity avec un ordinateur DELL modèle XPS 15 9570. Il possède un processeur Intel Core i7 8750H, 16Go de RAM et une carte graphique nVidia GeForce GTX 1050Ti.

Nous avons placé l'utilisateur à la même localisation (Latitude de 46.2941311, Longitude de 7.5335362, Altitude de 600 et Bearing de 160) pour tous les tests que ce soit sur l'ordinateur ou le smartphone. Nous avons répété les opérations 15 fois et calculé une moyenne.

Nous avons regroupé la synthèse des tests dans le tableau ci-dessous.

Tableau 2 : Moyennes des temps d'exécution en secondes du prototype

	Unity	Samsung
Rendu en direct	0,500	3,350
Calque couleur	40,045	61,205
Calque transparent	30,450	60,385
Calque couleur + distance	170,700	195,410
Calque transparent + distance	150,315	186,905

Sources : Données de l'auteur

En résumé, nous pouvons facilement observer que l'ajout de fonctionnalité ralentit le processus d'exécution à chaque fois indépendamment du matériel. Nous constatons également que l'affichage du calque transparent prend moins de temps que le calque couleur. L'affichage des couleurs prend du temps car c'est plus coûteux en ressources matérielles. Lors de l'affichage du calque transparent combiné avec la possibilité de calculer la distance totale, nous perdons beaucoup de temps. Nous avons également testé les différentes fonctionnalités avec le GPS activé pour le smartphone. Nous n'avons pas synthétisé les résultats car ils sont très aléatoires à l'inverse de ceux sans GPS. En effet, lorsque l'angle de vue change, beaucoup de paramètres sont modifiés et les calculs deviennent plus ou moins longs à chaque changement sans aucune relation.

10. Enquête

Nous avons mené une enquête divisée en deux parties. Premièrement, nous avons envoyé un questionnaire regroupant des informations sur les différentes thématiques abordées dans cette thèse et ensuite nous avons fait essayer le prototype au panel de personnes sélectionnées. Nous avons trouvé 18 personnes pour répondre au questionnaire et 15 d'entre elles ont pu essayer le prototype par la suite. Les volontaires viennent d'horizons différents et ont des âges différents également afin d'avoir un panel varié.

Le but de l'enquête est de comprendre si ces personnes savent déjà ce qu'est la réalité augmentée et comment l'utiliser, d'observer l'intérêt ou non pour notre prototype et finalement de l'améliorer avec leurs suggestions.

Au niveau de la réalité augmentée, 16 personnes sur 18 ne se rendent pas compte qu'ils utilisent cette technologie tous les jours au travers d'application telles que Snapchat ou

Instagram. Sur les deux individus qui ont affirmé avoir utilisé de la réalité augmentée, un a confondu l'AR avec de la VR. Cela démontre qu'un travail doit être fait pour démocratiser ces technologies et les présenter au grand public.

Les personnes interrogées ne sont pas forcément prêtes à investir de gros montants dans des casques ou commandes. La moyenne se situe à 209,375 francs. Beaucoup étaient également certains que ce matériel était nécessaire pour l'utilisation d'applications de réalité augmentée. Ils étaient étonnés quand nous leurs avons dit que leur smartphone pouvait suffire.

Au niveau de l'application, nous avons eu des remarques identiques aux résultats des tests que nous avons déjà effectués. L'application est trop lente à démarrer et limitée avec la position fixe. En moyenne, ils trouvent acceptable un délai d'attente de 6,59 secondes avant que l'application soit utilisable. Cependant, l'idée de l'affichage avec les couleurs sur le paysage à la place du calque transparent est plus appréciée. Nous avons eu aussi des retours positifs sur la fonctionnalité de distance. Le panel de test serait prêt à payer pour une application telle que PeakFinder qui est présentée dans notre thèse mais la nôtre devrait avoir plus de fonctionnalités pour qu'ils soient prêts à l'utiliser. Nous leurs avons demandé le prix qu'ils seraient prêts à payer pour une version commercialisable de notre prototype. Le montant moyen est à huit francs. Certaines personnes trouvent normal de payer une somme plus élevée pour l'obtenir mais d'autres la voudraient gratuitement.

Nous avons eu des retours négatifs à cause de son développement pour smartphone Android. Cela peut être changé assez rapidement lorsque le prototype sera terminé avec les outils d'AR Foundation.

Finalement, le prototype de base est très bien reçu et l'idée de l'utilisation bien perçue. Il faudra quand même bien enrichir le nombres de fonctionnalités si nous voulons concurrencer d'autres applications déjà développées dans le domaine. En effet, dans les idées reçues, des testeurs ont proposé de rajouter des affichages de lieux, de noms des montagnes, de la signalisation routière et de l'altitude en direct. En cas de développement de fonctionnalités plus urbaines, une personne a demandé si il était possible d'afficher la date de construction des bâtiments, faits et événements historiques liés aux lieux et même pourquoi pas reconstituer virtuellement des ruines observées.

11. Améliorations du prototype

D'après les tests établis et les retours des utilisateurs de notre prototype, nous avons essayé à chaque fois de trouver une solution théorique pour résoudre les différentes limites que nous rencontrons encore.

Nous pensons qu'il serait bien de déléguer le traitement des données et création de la texture à un serveur ou machine dédiée. Cela permettrait d'accélérer tous les calculs et de seulement utiliser les ressources matérielles du smartphone pour récupérer les informations de position et autres paramètres nécessaires. Il faut beaucoup moins de puissance pour la partie affichage de l'overlay et ce point ne devrait plus créer de problèmes.

Ensuite, nous proposons d'augmenter le nombre de surface couverte par les données. Il faudrait ne plus se limiter aux Alpes valaisannes. Les fichiers des autres terrains sont disponibles gratuitement sur internet.

Il faudrait également appliquer l'overlay sur l'écran en entier. Pour l'instant à cause des paramètres fixés par défaut et le manque de couverture, nous limitons la taille de l'overlay pour minimiser les erreurs. Nous pensons qu'avec plus de fichiers d'élévations, nous pourrions augmenter le champs de couverture et augmenter le nombre de bytes pour notre texture. En résolvant ce problème, nous pourrions également rajoutée la fonctionnalité qui permet d'ajuster l'affichage sur l'axe verticale. En fonction de la hauteur de la direction dans laquelle l'utilisateur regarde, nous devons modifier l'affichage du paysage. Nous n'avons pas pu la développer comme il faut à cause des erreurs de bytes. Nous avons joué avec la hauteur de l'horizon pour les différents tests et le prototype crashe avec certaines valeurs. Nous n'avons pas trouvé de redondances entre les tests afin préenregistrer des données valides et nous avons tout simplement supprimé la fonctionnalité testée.

Nous trouvons intéressant également les suggestions faites par les testeurs. De nombreuses informations comme l'altitude et autres données seraient faciles à rajouter sur l'overlay mais nous pensons que l'idée des informations sur les bâtiments et autres lieux historiques mériterait d'être creusée.

Une application à part pourrait être créée avec ces nouvelles fonctionnalités mais sinon nous pourrions ajouter des boutons permettant d'afficher ou non certains calques et de bénéficier de différentes échelles de précisions et d'informations.

Au niveau technique, nous avons utilisé le kit AR Foundation couplé avec les fonctionnalités d'AR Core. Nous avons trouvé durant nos recherches, vers la fin du prototype, un nouvel SDK proposé par l'entreprise ESRI. C'est une entreprise spécialisée en GIS et qui propose des outils pour ses clients dans ce domaine. Nous n'avons pas eu le temps d'essayer les outils qu'ils offrent mais il serait bien de tester leur SDK afin de voir s'il correspondrait mieux à nos besoins pour une version en production et ses futurs développements.

Nous avons essayé de chercher également des thèses ou travaux plus développés sur les sujets qui sont abordés dans ce travail de Bachelor. Nous conseillons de lire la publication appelée *Automated Outdoor Depth-Map Generation and Alignment* de Cadik, Sykora & Lee (Cadik, Sykora, & Lee, 2018). Ils abordent le sujet de l'estimation des profondeurs avec les cartes et les modèles 3D. Une autre publication que nous avons trouvée intéressante est *Interactive Landscape Design and Flood Visualisation in Augmented Reality* écrite par Tomkins et Lange (Tomkins & Lange, 2019). Ils effectuent un travail de représentation de terrain en réalité augmentée. Ces deux thèses peuvent aider les futurs développements sur le prototype avec leurs contenus ou les références citées dans celles-ci. Nous avons également trouvé que l'EPFL possède dans ses archives un cours de programmation orienté-objet (CS-108) avec comme thématique les modèles HGT et les profils altimétriques. Nous n'avons pas pu trouver le cours complet.

12. Conclusions

12.1. Conclusion projet

Nous sommes contents de l'état final du prototype au niveau des fonctionnalités développées. Nous répondons par l'affirmative à la problématique principale de la thèse qui est de savoir s'il est possible d'utiliser des données GIS au sein d'une application de réalité augmentée. Cela n'était pas facile et quelques problèmes à résoudre au niveau du développement ont ralenti la progression de l'application. Le projet final est bon et la portée des aspects théoriques couverts également.

Il y a quand même une petite frustration au niveau de l'utilisation globale du prototype. Nous aurions bien aimé la publier sur un store pour qu'un plus grand nombre d'utilisateurs donnent leur avis. Le fait que le prototype soit long à lancer et lent d'utilisation nous a découragés de mettre l'application en ligne.

12.2. Conclusion personnelle

Personnellement, nous sommes heureux du travail accompli. Cette année n'a pas été de tout repos avec la fin de nos études accompagnée du COVID-19. Cette situation nous a appris d'apprendre à nous adapter aux imprévus. En parallèle, le travail de Bachelor nous a permis un peu de nous évader de ces problèmes en nous plongeant dans la création d'un prototype et d'une thèse dont la thématique nous passionnait dès le début.

Les heures passées à essayer de comprendre, de résoudre et de créer nous ont confirmé nos attentes sur les aspects abordés ainsi que la technologie de la réalité augmentée. Nous étions déjà sensibles à ses possibilités mais lorsque nous avons creusé un peu, nous avons eu notre champs de vision élargi. Ces découvertes nous ont motivés à essayer de continuer de travailler dans ce domaine. Nous avons eu de la chance de pouvoir jongler entre différents sujets tout au long du projet. Cela a empêché trop de redondances au niveau de l'apprentissage et de tomber dans de la monotonie.

Nous sommes également reconnaissants envers Monsieur Antoine Widmer qui a supervisé le projet. Nous avons eu la liberté et l'encadrement nécessaires dans nos choix et avons pu proposer ce qui nous intéressait. Nous avons de ce fait plus appris que si nous avons été contraints à suivre un fil rouge établi du début à la fin en ne laissant peu de place aux envies personnelles.

13. Références

- A.Balasubramanian. (2017, Septembre 03). Data sources and input in GIS. (35). Mysore: Université de Mysore. Récupéré sur <https://fr.slideshare.net/bala1957/data-sources-and-input-in-gis>
- Alsop, T. (2021, Juillet 26). *Virtual reality (VR) - statistics & facts*. Consulté le Juillet 27, 2021, sur Statista: https://www.statista.com/topics/2532/virtual-reality-vr/#dossierSummary__chapter1
- Aniwaa team. (2021, Janvier 12). *Les 9 meilleurs casques de réalité mixte en 2021*. Consulté le Mai 22, 2021, sur Aniwaa: <https://www.aniwaa.fr/guide-achat/vr-ar/meilleurs-casques-realite-mixte/>
- Announcing Project Aria: A Research Project on the Future of Wearable AR*. (2020, Septembre 16). Consulté le Mai 22, 2021, sur About.Facebook: <https://about.fb.com/news/2020/09/announcing-project-aria-a-research-project-on-the-future-of-wearable-ar/>
- Apple. (n.d). *ARKit*. Consulté le Mars 06, 2021, sur Apple Developer: <https://developer.apple.com/documentation/arkit/>
- ARGIS. (2019, Octobre 28). *The City of Englewood, CO Achieves Downtime Reduction with AR*. Consulté le Juillet 19, 2021, sur ARGIS: <https://www.argis.com/success-stories/2019/10/28/reduced-downtime-with-ar>
- Argis Solutions. (2019, Mars 28). *ARGIS Public Works*. Récupéré sur <https://www.youtube.com/watch?v=1jYmv6gRDUo>
- Arora, S. K. (2020, Septembre 15). *Unity vs Unreal Engine: Which Game Engine Should You Choose?* Consulté le Mai 30, 2021, sur Hackr: <https://hackr.io/blog/unity-vs-unreal-engine>
- Artefacto. (n.d). *Qu'est-ce que la réalité mixte ?* Consulté le Juillet 5, 2021, sur Artefacto Augmented Reality: <https://www.artefacto-ar.com/realite-mixte/>
- Artefacto. (n.d). *Qu'est-ce que la réalité augmentée ?* Consulté le Juillet 5, 2021, sur Artefacto Augmented Reality: <https://www.artefacto-ar.com/realite-augmentee/>
- Arvizio. (2021, Mai 4). *Arvizio*. Consulté le Juillet 23, 2021, sur Arvizio Brings Drone Mapped 3D Models & Point Clouds to Life in Augmented Reality: <https://www.arvizio.io/2021/05/03/arvizio-brings-drone-mapped-3d-models-point-clouds-to-life-in-augmented-reality/>

- Association Francophone de Promotion de la Réalité Augmentée. (n.d). *Qu'est-ce que la Réalité Augmentée ?* Consulté le Juillet 5, 2021, sur Augmented Reality: <https://www.augmented-reality.fr/rapro/>
- Barnard, D. (2019, Mai 05). *Degrees of Freedom (DoF): 3-DoF vs 6-DoF for VR Headset Selection*. Consulté le Avril 11, 2021, sur Virtual Speech: <https://virtualspeech.com/blog/degrees-of-freedom-vr>
- Barnard, D. (2019, Juin 06). *History of VR - Timeline of Events and Tech Development*. Consulté le Avril 10, 2021, sur Virtual Speech: <https://virtualspeech.com/blog/history-of-vr>
- Bergum, M. (2020, Août 10). *The Geospatial Convergence: 5G and the Future of Wireless Technology*. Consulté le Juin 28, 2021, sur East View Geospatial: <https://geospatial.com/the-geospatial-convergence-5g-and-the-future-of-wireless-technology/>
- Cadik, M., Sykora, D., & Lee, S. (2018). Automated Outdoor Depth-Map Generation and Alignment. *Elsevier*. Récupéré sur <http://cphoto.fit.vutbr.cz/depth/cadik18automated.pdf>
- Cameron, E., & Foster, J. (2019). ArcGIS Runtime An Introduction. Récupéré sur <https://www.esri.com/content/dam/esrisites/en-us/about/events/media/2019-european-developers-summit/eurodev-30.pdf>
- Coding Rush. (2020, Juin 26). Unity AR Foundation- How to Setup AR Core in Unity For Android. Récupéré sur <https://www.youtube.com/watch?v=NPBnEchvLu0>
- de Ferranti, J., & Hormann, C. (2014, May 26). *Digital Elevation Data*. Consulté le 05 07, 2021, sur View finder panoramas: <http://viewfinderpanoramas.org/dem3.html#alps>
- EPFL. (n.d). *Modèles HGT et profils altimétriques*. Consulté le Juillet 20, 2021, sur cs108: https://cs108.epfl.ch/archive/17/p/04_elevation-profile.html
- FME. (n.d). *SRTM HGT (Shuttle Radar Topography Mission Height) Reader*. Récupéré sur FME Readers and Writers: https://docs.safe.com/fme/2018.1/html/FME_Desktop_Documentation/FME_ReadersWriters/srtmhgt/srtmhgt.htm
- Geaneat. (2020, Octobre 2020). *Now and Then*. Consulté le Juin 18, 2021, sur Google Play: https://play.google.com/store/apps/details?id=org.geneanet.vuesHier&hl=en_US&gl=US

- Geneanet. (2020, Octobre 19). *Google Play*. Consulté le Mars 13, 2021, sur https://play.google.com/store/apps/details?id=org.geneanet.vuesHier&hl=en_US
- GIScience. (2018, Juillet 13). OSM to QGIS to ArcGIS to City Engine to Unity Tutorial. Récupéré sur <https://www.youtube.com/watch?v=FmRU0mowvew>
- Glärbo. (2021, Février 8). *How can I calculate distance from two pixels HSV?* Récupéré sur Mathematics: <https://math.stackexchange.com/questions/4016084/how-can-i-calculate-distance-from-two-pixels-hsv>
- GlobalData Technology. (2020, Juillet 15). *History of virtual reality : Timeline*. Consulté le Avril 16, 2021, sur Verdict: <https://www.verdict.co.uk/history-virtual-reality-timeline/>
- Google. (n.d). *Discover Glass Enterprise Edition*. Consulté le Juin 12, 2021, sur Google: <https://www.google.com/glass/start/>
- Google. (n.d). *Présentation d'ARCore*. Consulté le mars 6, 2021, sur ARCore: <https://developers.google.com/ar/discover>
- Hall, S., & Takahashi, R. (2017, Septembre 08). *Augmented and virtual reality: the promise and peril of immersive technologies*. Consulté le Juillet 11, 2021, sur World Economic Forum: <https://www.weforum.org/agenda/2017/09/augmented-and-virtual-reality-will-change-how-we-create-and-consume-and-bring-new-risks/>
- Hugues, O., Cieutat, J.-M., & Guitton, P. (2011). GIS and Augmented Reality: State of the Art and Issues. Dans B. Furht, *Handbook of Augmented Reality (pp.721-740)* (pp. 721-740). Floride: Springer. doi:10.1007/978-1-4614-0064-6_33
- Jeary Alzahrani, N. (2019, Juin 8). Augmented Reality ARKit-CoreLocation. Récupéré sur <https://www.youtube.com/watch?v=A2mqQdxnlqw>
- Kouba, Y. (2019). *Cours de système d'information géographique*. doi:10.13140/RG.2.2.32533.78564
- Kugler, L. (2021, Février). *The State of Virtual Reality Hardware*. Consulté le Juillet 20, 2021, sur Communications of the ACM: <https://cacm.acm.org/magazines/2021/2/250071-the-state-of-virtual-reality-hardware/fulltext>
- Kuprenko, V. (2021, Janvier 20). *How to develop a location-based Augmented Reality app*. Consulté le Mai 28, 2021, sur GEOSPATIAL WORLD: <https://www.geospatialworld.net/blogs/location-based-augmented-reality-app-development-a-complete-guide/>
- L'équipe Dynamique Entrepreneuriale. (2020, Juillet 01). *La réalité augmentée ou virtuelle, un marché prometteur ?* Consulté le Avril 10, 2021, sur Dynamique Mag:

<https://www.dynamique-mag.com/article/realite-augmentee-virtuelle-marche-prometteur.10615>

Les Numériques. (2020, Septembre 16). *Oculus Quest 2*. Consulté le Mai 22, 2021, sur Les Numériques: <https://www.lesnumeriques.com/casque-realite-virtuelle/oculus-quest-2-p59017.html>

Les Numériques. (2021, Février 17). *HP Reverb G2*. Consulté le Mai 22, 2021, sur Les Numériques: <https://www.lesnumeriques.com/casque-realite-virtuelle/hp-reverb-g2-p58071.html>

Leswing, K. (2021, Février 20). *The tech industry is looking to replace the smartphone — and everybody is waiting to see what Apple comes up with*. Consulté le Juillet 12, 2021, sur CNBC: <https://www.cnbc.com/2021/02/20/apple-facebook-microsoft-battle-to-replace-smartphone-with-ar.html>

Lynch, G. (2020, Septembre 16). *Facebook working on Project Aria augmented reality glasses — with smart glasses launching in 2021*. Consulté le Juin 12, 2021, sur Techradar: <https://www.techradar.com/news/facebook-working-on-protect-aria-augmented-reality-glasses-with-smart-glasses-launching-in-2021>

Mann, K. (2021). *Making GIS content available in interesting and useful ways*. ESRI. Récupéré sur <https://www.esri.com/news/arcuser/0311/files/arandgis.pdf>

Mapbox. (2018, Juin 14). Build a Unity 3D game with Mapbox (3/29) Setting up the project. Récupéré sur <https://www.youtube.com/watch?v=sK3lr3Yh6XI>

Mapbox. (n.d). *Mapbox pricing*. Consulté le Mars 20, 2021, sur Mapbox: <https://www.mapbox.com/pricing/>

Mapbox. (n.d). *Maps SDK for Unity*. Consulté le Mars 20, 2021, sur Mapbox: <https://docs.mapbox.com/unity/maps/guides/>

Mari, J.-L. (2010, Mai). *Algorithmes de tracé de segment*. Marseille. Récupéré sur https://www.irem.univ-mrs.fr/IMG/pdf/Algos_trace_segment_draft_2_-2.pdf

Mendiratta, G. (2018, Janvier 5). *#10 Digital Marketing Trends To Embrace in 2018*. Consulté le Avril 20, 2021, sur Entrepreneur India: <https://www.entrepreneur.com/article/307025>

Netbrain.es. (2016, Octobre 13). *Senditur*. Consulté le Mars 13, 2021, sur Google Play: https://play.google.com/store/apps/details?id=com.netbrain.senditur&hl=en_US

Newton, C. (2021, Juillet 22). *Mark in the metaverse*. Consulté le Juillet 27, 2021, sur The Verge: <https://www.theverge.com/22588022/mark-zuckerberg-facebook-ceo>

- metaverse-
interview?scrolla=5eb6d68b7fedc32c19ef33b4&fbclid=IwAR2wPW4t4ehyycjcmoKUq
O9ygnP5jo1r77kE5rA8EZbgFutPfc9j9c75Asg
- PeakFinder. (2021). *PeakFinder App*. Consulté le Mars 13, 2021, sur PeakFinder:
<https://www.peakfinder.org/fr/mobile/>
- PeakLens. (n.d). *PeakLens*. Consulté le Mars 14, 2021, sur PeakLens: <https://peaklens.com/>
- Programming Algorithms. (n.d). *HSV To RGB*. Récupéré sur Programming Algorithms:
<https://www.programmingalgorithms.com/algorithm/hsv-to-rgb/>
- Programming Algorithms. (n.d). *RGB To HSV*. Récupéré sur Programming Algorithms:
<https://www.programmingalgorithms.com/algorithm/rgb-to-hsv/>
- Puget Systems. (2021). *Recommended Systems for Unity*. Consulté le mars 20, 2021, sur
Puget Systems: <https://www.pugetsystems.com/recommended/Recommended-Systems-for-Unity-188/Hardware-Recommendations>
- Punni, D. (2019, Août 11). AR Foundation & Unity 01: Setup for Android. Récupéré sur
<https://www.youtube.com/watch?v=0mpsiO2lCx0&t=1s>
- R., S. (2021, mai 25). *Unity : Tout ce qu'il faut savoir sur le puissant moteur de jeu*. Consulté le
Mai 29, 2021, sur Réalité-virtuelle: <https://www.realite-virtuelle.com/unity-tout-savoir/>
- R., S. (2021, Mai 25). *Unreal Engine : Tout savoir sur le moteur de jeu d'Epic Games*. Consulté
le Mai 29, 2021, sur Réalité-virtuelle: <https://www.realite-virtuelle.com/unreal-engine-tout-savoir/>
- Révész, R. (2020, Juillet 11). *How many AR enabled phones are in the world?* Consulté le
Juillet 12, 2021, sur Aron Platform: <https://www.aronplatform.com/mobile-ar-penetration/>
- Rossi, J. (n.d). *What's new in ARKit 5 ?* Consulté le Juillet 05, 2021, sur Orange loops:
<https://orangeloops.com/2021/06/whats-new-in-arkit-5/>
- Schmid, F., & Langerenken, D. (2014, Juin 3-6). Augmented Reality and GIS: On the
Possibilities and Limits of Markless AR. *Connecting a Digital Europe through Location
and Place*. Castellón. Récupéré sur [https://agile-
online.org/conference_paper/cds/agile_2014/agile2014_87.pdf](https://agile-online.org/conference_paper/cds/agile_2014/agile2014_87.pdf)
- Software Testing Help. (2021, Juin 28). *Future Of Virtual Reality – Market Trends And
Challenges*. Consulté le Juillet 11, 2021, sur Software Testing Help:
<https://www.softwaretestinghelp.com/future-of-virtual-reality/>

- Technavio. (2021, Juin 07). *Augmented Reality (AR) and Virtual Reality (VR) Market to grow by USD 162.71 Billion|Key Drivers and Market Forecasts|17000+ Technavio Research Reports*. Consulté le Juillet 20, 2021, sur Cision PR Newswire: <https://www.prnewswire.com/news-releases/augmented-reality-ar-and-virtual-reality-vr-market-to-grow-by-usd-162-71-billionkey-drivers-and-market-forecasts17000-technavio-research-reports-301306368.html>
- Tomkins, A., & Lange, E. (2019). *Interactive Landscape Design and Flood Visualisation in Augmented Reality*. Sheffield. doi:10.3390/mti3020043
- Tutor, T. O. (2018, Avril 8). Bearings vs Direction - Trigonometry Word Problems. Récupéré sur <https://www.youtube.com/watch?v=jyi5zW8mavs>
- Unity. (2014, Octobre 15). *Android returning white in Texture2D.GetPixel(x,y)*. Récupéré sur Answers.Unity: <https://answers.unity.com/questions/810113/android-returning-white-in-texture2dgetpixelxy.html>
- Unity. (2021). *Getting started with AR development in Unity*. Consulté le Mars 13, 2021, sur Unity: <https://docs.unity3d.com/2021.1/Documentation/Manual/AROverview.html>
- Unity. (n.d). *AR Foundation*. Consulté le Mars 6, 2021, sur Unity: <https://unity.com/fr/unity/features/arfoundation>
- Unity. (n.d). *Choose the plan that is right for you*. Consulté le Mai 29, 2021, sur Unity Store: <https://store.unity.com/compare-plans?currency=USD>
- Unity. (n.d). *Offres et tarifs*. Consulté le Mai 29, 2021, sur Unity Store: https://store.unity.com/fr/?_gl=1*1i7pazo*_ga*MTIwNjQ5OTA4My4xNjEzMDUwMzYz*_ga_1S78EFL1W5*MTYyNzYzNDczNi40MC4wLjE2Mjc3NzQ3MzYuNjA.&_ga=2.90682844.795656973.1627374737-1206499083.1613050363
- Unreal Engine. (n.d). *Download Unreal Engine*. Consulté le Mai 29, 2021, sur Unreal Engine: <https://www.unrealengine.com/en-US/download>
- Unreal Engine. (n.d). *Frequently asked questions (FAQ)*. Consulté le Mai 29, 2021, sur Unreal Engine: <https://www.unrealengine.com/en-US/faq?active=release>
- UVED. (n.d). *Le modèle TSL*. Consulté le Juillet 10, 2021, sur ENVCAL - Suivi de l'environnement par télédétection: <https://e-cours.univ-paris1.fr/modules/ued/envcal/html/compositions-colorees/representations-couleur/modeles-perceptuels/tsl.html>
- Vuforia. (n.d). *Getting Started*. Consulté le Mars 10, 2021, sur Vuforia Developer Library: <https://library.vuforia.com/>

- Weimann, J. (2017, Septembre 15). Unity3D - How to Zoom an Image / Create a zoomable pannable sprite. Récupéré sur <https://www.youtube.com/watch?v=BFX3FpUnoio>
- Wen, Q., & olprod. (2021, Avril 10). *Qu'est-ce que la réalité mixte ?* Consulté le Avril 10, 2021, sur Microsoft: <https://docs.microsoft.com/fr-fr/windows/mixed-reality/discover/mixed-reality>
- Wikipédia. (2020, Avril 20). *Shuttle Radar Topography Mission*. Consulté le Avril 25, 2021, sur Wikipédia: Shuttle Radar Topography Mission
- Wikipédia. (2020, Novembre 19). *Teinte Saturation Valeur*. Consulté le Juin 15, 2021, sur Wikipédia: https://fr.wikipedia.org/wiki/Teinte_Saturation_Valeur
- Wikipédia. (2021, Mai 27). *Algorithme de tracé de segment de Bresenham*. Consulté le Juin 25, 2021, sur Wikipédia: https://fr.wikipedia.org/wiki/Algorithme_de_trac%C3%A9_de_segment_de_Bresenham
- Wikipédia. (2021, Juin 02). *Cercle chromatique*. Consulté le Juin 15, 2021, sur Wikipédia: https://fr.wikipedia.org/wiki/Cercle_chromatique
- Wikipédia. (2021, juin 10). *Liste des logiciels SIG*. Consulté le Juillet 14, 2021, sur Wikipédia: https://fr.wikipedia.org/wiki/Liste_des_logiciels_SIG
- Wikipedia. (2021, Juillet 10). *Système d'information géographique*. Consulté le Juillet 14, 2021, sur Wikipédia: https://fr.wikipedia.org/wiki/Syst%C3%A8me_d%27information_g%C3%A9ographique
- Wikipédia. (2021, Mai 05). *Unity (moteur de jeu)*. Consulté le Mai 29, 2021, sur Wikipédia: [https://fr.wikipedia.org/wiki/Unity_\(moteur_de_jeu\)](https://fr.wikipedia.org/wiki/Unity_(moteur_de_jeu))
- Wikitude. (n.d). *Wikitude Documentation*. Consulté le Mars 11, 2021, sur Wikitude Support: <https://www.wikitude.com/documentation/>
- Wikitude. (n.d). *Wikitude Documentation - Expert Edition*. Consulté le Mars 11, 2021, sur Wikitude: <https://www.wikitude.com/external/doc/expertedition/#key-features>
- Zaki, M. (2020, Septembre 07). *Les promesses multiples de la "réalité étendue"*. Consulté le Avril 10, 2021, sur Bilan: <https://www.bilan.ch/finance/les-promesses-multiples-de-la-realite-etendue>
- Zappar. (2021, Juin 21). *All-new ZapBox: 6-DoF Mixed Reality for \$40*. Consulté le Juillet 17, 2021, sur Kickstarter: <https://www.kickstarter.com/projects/all-newzapbox/all-new-zapbox-awesome-mixed-reality-for-40>

14. Annexe I : Product Backlog

US#	Thème	En tant que ...	Je veux ...	Afin de ...	Critères d'acceptance	Priorité	Story Poi	Status
1	Préparation	Développeur	Définir l'objectif du travail de Bachelor	Déterminer le scope du Use Case de base proposé	Définition de l'objectif de la thèse / Définition du Use Case pour le prototype	Must have	3	●
2	Préparation	Développeur	Etudier les applications existantes dans le domaine GIS	Déterminer les fonctionnalités utiles au prototype	Fonctionnalités définies / Validation de M. Antoine Widmer	Must have	13	●
3	Préparation	Développeur	Ecrire état de l'art dans le domaine de la réalité virtuelle, mixte et augmentée	Comprendre les enjeux du domaine et les possibilités	Etat de l'art du domaine général / Comprendre les spécificités techniques	Must have	34	●
4	Préparation	Développeur	Réaliser une étude sur les technologies AR existantes	Définir les outils que l'on va utiliser pour le développement	Tests réussis / Solutions techniques trouvées pour les fonctionnalités à développer	Must have	21	●
5	Préparation	Développeur	Réaliser une étude sur les GIS	Comprendre quels sont leurs utilités	Définition des enjeux et place du GIS dans le projet	Must have	21	●
6	Préparation	Développeur	Mettre en place environnement de développement	Débuter la programmation du prototype	Technologie finale pour le projet définie / Validation de M. Antoine Widmer / Développement de l'application possible	Must have	5	●
7	Préparation	Développeur	Pouvoir réutiliser le projet existant	Récupérer les éléments utiles aux prototypes	Solution Parcours donnée par M. Antoine Widmer	Must have	5	●
8	Technique	Développeur	Partager le projet sur Github	De pouvoir partager le code	Intégrée au projet actuel Création Repository Github / Documentation technique du projet	Must have	2	●
9	Développement	Utilisateur	Avoir le rendu de ma caméra sur l'application	Pouvoir observer en direct le paysage	Implémentation de la fonctionnalité de rendu / Aucune latence	Must have	8	●
10	Développement	Utilisateur	Avoir un calque virtuel sur mon écran	Bénéficier d'informations supplémentaires sur le paysage que l'observe	Overlay créé avec une texture / Rendu en direct de la caméra toujours visible / Distances affichées grâce aux couleurs	Must have	21	●
11	Développement	Développeur	Avoir accès aux données GIS	Créer une texture dynamique pour l'overlay	Utiliser les fichiers HGT comme données / Pouvoir mettre à jour la texture	Must have	21	●
12	Développement	Développeur	Créer le calque virtuel depuis la texture dynamique	Ajouter des informations supplémentaires grâce aux données GIS	Overlay correspond au paysage observé / Rendu en direct de la caméra toujours visible	Must have	34	●
13	Développement	Développeur	Utiliser le positionnement de l'utilisateur	Modifier l'overlay en fonction des données géographiques du smartphone	L'altitude, longitude et altitude du smartphone utilisées / Mise à jour du calque en fonction des données géographiques	Must have	34	●
14	Développement	Développeur	Utiliser l'orientation de l'utilisateur	Modifier l'overlay en fonction de l'orientation vers laquelle est pointée le smartphone	Bearing utilisé / Mise à jour du calque en fonction des données géographiques / Rendu caméra en direct	Must have	21	●
15	Préparation	Développeur	Tenir un journal de bord	Répertorier les tâches effectuées	Dates pour laquelle ça doit être fait / Problèmes	Must have	8	●
16	Préparation	Développeur	Effectuer une enquête sur ma thématique et mon prototype	Connaitre l'avis des utilisateurs et leurs préférences	Réponses enregistrées	Must have	5	●
17	Préparation	Développeur	Rédiger la thèse	Terminer la partie rédactionnelle du travail de Bachelor	Points de la Table des matières validée par M. Antoine Widmer complets et détaillés	Must have	34	●
18	Développement	Développeur	Rendre le filtre de couleur transparent	D'avoir un calque seulement pour représenté les bordures	Bordures et reliefs indiqués en noir / Overlay sans caméra toujours visible	Could have	21	●
19	Développement	Utilisateur	Pouvoir sélectionner un pixel de l'écran	De connaître ma distance par rapport à ce point	Distance affichée avec un vent / Répétition de l'action disponible / Rendu en direct de la caméra et	Could have	89	●
20	Développement	Développeur	Rendre le prototype plus fluide	permettre des tests de l'application par un panel d'utilisateur	Erreurs diminuées / Création de la texture plus rapide	Could have	13	●

User Stories rajoutées en cours de projet

Sources : Données de l'auteur

Annexe II : Calendrier

Tableau 3 : Journal de bord

Date fin sprint	Quoi ?	Statut	Conséquence
25.02.2021	Démarrage TB	Ok	Lecture projet
04.03.2021	Scope du projet réflexion	Ok	Démarrage recherches état de l'art
11.03.2021	Scope et objectifs définis	Ok	Début des tests sur les technologies AR
01.04.2021	Fin des tests des technologies AR	Ok	Recherches sur la thématique des GIS et découverte projet existant
15.04.2021	Recherches bien abouties PanoramaCS testé	Ok	Début rédaction brouillon état de l'art
22.04.2021	Etat de l'art bien avancé	OK	Démarrage développement
29.04.2021	Intégration des tests dans le prototype	OK	Fonctionnalités de rendu caméra et texture du paysage à terminer
06.05.2021	Ajout des fichiers .hgt dans le projet Etat de l'art au propre Corrections orientation overlay	Chemin des fichiers à corriger	Modification de l'accès aux fichiers à faire
13.05.2021	Fichiers .hgt fonctionnels	OK	Ajouter les paramètres de localisation de l'utilisateur
20.05.2021	Latitude, longitude et altitude intégrées dans projet	Ok sauf altitude	Altitude à corriger
27.05.2021	Altitude fonctionnelle	OK	Ajouter le paramètre de direction de l'utilisateur

03.06.2021	Bearing intégré au projet	Pas fonctionnel Corrections d'autres points qui ont été faits	Continuer fonctionnalité de direction
10.06.2021	Bearing du smartphone intégré au projet	OK mais avec des erreurs possibles	Rendre transparent l'overlay
17.06.2021	Overlay transparent	OK	Calculer la distance depuis 1 pixel choisi
08.07.2021	Calcul de la distance fonctionnel	Ok mais avec des erreurs possibles	Fin rédaction de la thèse Corrections des erreurs et optimisation du prototype

Les tâches en cours sont indiquées en jaune

Sources : Données de l'auteur

Annexe III : Questionnaire de l'enquête (menée par l'auteur en 2021)

- 1) Quel âge avez-vous ?
- 2) Connaissez-vous la réalité augmentée ?
- 3) Pouvez-vous citer une application utilisant de la réalité augmentée ? Si oui, laquelle ?
- 4) Avez-vous déjà utilisé une application de réalité augmentée ?
- 5) Quel était son support ?
- 6) Seriez-vous prêts à acheter le matériel nécessaire pour utiliser ce genre d'application ?
- 7) Quel montant maximal seriez-vous prêts à investir ?
- 8) Quel est le temps d'ouverture acceptable d'une application mobile avant l'utilisation (secondes) ?
- 9) Seriez-vous intéressés à utiliser une application qui ajoute des informations virtuelles sur le paysage lorsque vous l'observez avec votre smartphone ?
- 10) Quelles sont les informations que vous aimeriez avoir ?
- 11) Quelles informations virtuelles affichées par le prototype trouvez-vous utiles ?
- 12) Utiliseriez-vous une version commercialisable de notre prototype ?
- 13) Quel montant paieriez-vous pour une version commercialisable de notre prototype ?
- 14) Suggestions pour le prototype ?

Sources : Données de l'auteur

Déclaration de l'auteur

"Je déclare, par ce document, que j'ai effectué le travail de Bachelor ci-annexé seul, sans autre aide que celles dûment signalées dans les références, et que je n'ai utilisé que les sources expressément mentionnées. Je ne donnerai aucune copie de ce rapport à un tiers sans l'autorisation conjointe du RF et du professeur chargé du suivi du travail de Bachelor, y compris au partenaire de recherche appliquée avec lequel j'ai collaboré, à l'exception des personnes qui m'ont fourni les principales informations nécessaires à la rédaction de ce travail et que je cite ci-après :

- Monsieur Antoine Widmer, professeur à la HES-SO Valais "

Fribourg, le 05 juillet 2021, Victor Bonny.