

ECOLE POLYTECHNIQUE DE L'UNIVERSITÉ FRANÇOIS RABELAIS DE TOURS

Département Informatique

64 avenue Jean Portalis

37200 Tours, France

Tél. +33 (0)2 47 36 14 14

polytech.univ-tours.fr

Projet Recherche & Développement

2018-2019

Mobile App en Réalité Augmentée polytech



POLYTECH[®]
TOURS

Entreprise

Flagtown



Tuteurs entreprise

Stéphane GRANDJEAN (CEO Flagtown)

Xavier GOUPIL

Étudiant

Wenli YAN (DI5)

Tuteurs académiques

Gilles VENTURINI

Barthelemy SERRES

Liste des intervenants

Entreprise

Flagtown
85 rue Fleurie
37540 Saint Cyr sur Loire
www.flagtown.fr/



Nom	Email	Qualité
Wenli YAN	wenli.yan@etu.univ-tours.fr	Étudiant DI5
Gilles VENTURINI	gilles.venturini@univ-tours.fr	Tuteur académique, Département Informatique
Barthelemy SERRES	barthelemy.serres@univ-tours.fr	Tuteur académique, Département Informatique
Stéphane GRANDJEAN	stephane@flagtown.io	Tuteur entreprise, CEO Flagtown
Xavier GOUPIL	xavier.goupil@gmx.com	Tuteur entreprise



Avertissement

Ce document a été rédigé par Wenli Yan susnommé l'auteur.

L'entreprise Flagtown est représentée par Stéphane Grandjean et Xavier Goupil susnommés les tuteurs entreprise.

L'Ecole Polytechnique de l'Université François Rabelais de Tours est représentée par Gilles Venturini et Barthélemy Serres susnommés les tuteurs académiques.

Par l'utilisation de ce modèle de document, l'ensemble des intervenants du projet acceptent les conditions définies ci-après.

L'auteur reconnaît assumer l'entière responsabilité du contenu du document ainsi que toutes suites judiciaires qui pourraient en découler du fait du non respect des lois ou des droits d'auteur.

L'auteur atteste que les propos du document sont sincères et assument l'entière responsabilité de la véracité des propos.

L'auteur atteste ne pas s'approprier le travail d'autrui et que le document ne contient aucun plagiat.

L'auteur atteste que le document ne contient aucun propos diffamatoire ou condamnable devant la loi.

L'auteur reconnaît qu'il ne peut diffuser ce document en partie ou en intégralité sous quelque forme que ce soit sans l'accord préalable des tuteurs académiques et de l'entreprise.

L'auteur autorise l'école polytechnique de l'université François Rabelais de Tours à diffuser tout ou partie de ce document, sous quelque forme que ce soit, y compris après transformation en citant la source. Cette diffusion devra se faire gracieusement et être accompagnée du présent avertissement.



Pour citer ce document

Wenli Yan, *Mobile App en Réalité Augmentée polytech*, Projet Recherche & Développement, Ecole Polytechnique de l'Université François Rabelais de Tours, Tours, France, 2018-2019.

```
@mastersthesis{
  author={Yan, Wenli},
  title={Mobile App en Réalité Augmentée polytech},
  type={Projet Recherche \& Développement},
  school={Ecole Polytechnique de l'Université François Rabelais de Tours},
  address={Tours, France},
  year={2018-2019}
}
```


Table des matières

Liste des intervenants	a
Avertissement	b
Pour citer ce document	c
Table des matières	i
Table des figures	vi
I Introduction	1
II L'état de l'art	2
1 Réalité augmentée	3
1 Concept	3
2 Caractéristiques	4
3 Historique	4
4 Principe	4
5 Technologies clés	5
5.1 Technologie d'enregistrement de suivi.....	6
5.2 Technologie d'affichage	6
5.2.1 Dispositif d'affichage monté sur la tête	6
5.2.2 Dispositif d'affichage à l'écran	7
5.2.3 Dispositif d'affichage par projection.....	8
5.3 Technologie d'interaction homme-machine.....	8

5.3.1	Technologie interactive basée sur des périphériques matériels traditionnels.....	8
5.3.2	Technologie interactive basée sur la reconnaissance vocale.....	9
5.3.3	Technologie interactive tactile.....	9
5.3.4	Technologie interactive basée sur la reconnaissance de mouvement	9
5.3.5	Technologie interactive basée sur le suivi oculaire.....	9
6	Applications d'AR.....	10
6.1	Éducation	10
6.2	Soins médicaux	10
6.3	Formation en entreprise	11
6.4	Shopping	12
6.5	Marketing publicitaire basé sur la localisation.....	13
6.6	Design d'intérieur.....	13
6.7	Industrie.....	14
6.8	Armée.....	15
6.9	Jeux vidéo.....	16
6.10	Cinéma et la télévision	16
6.11	Culture et Art.....	17
7	Situation actuelle de développement.....	17
2	ARCore	19
1	Introduction d'ARCore	19
2	Appareils supportés	20
3	Environnement de développement	21
4	Principe de fonctionnement.....	21
5	Caractéristiques de base - ARCore 1.0.....	21
5.1	Suivi de mouvement	22
5.2	Compréhension de l'environnement	22
5.3	Estimation de la lumière.....	23
6	Caractéristiques et supériorités puissantes de la version ARCore 1.2	23
6.1	Ancre nuage qui permet à plusieurs personnes partageant la RA	23
6.2	Ajout d'images augmentées pour la reconnaissance de plan vertical	24
6.3	Un nouveau Sceneform SDK basé sur Java	25
7	Conditions et performance.....	25
7.1	Capacité	25
7.2	Qualité des images.....	25
3	AR aujourd'hui : smartphone vs autonome	28
4	Introduction de la société	30

III	Cahier de spécification Système	31
5	Contexte de la réalisation	32
1	Contexte.....	32
2	Objectifs.....	32
2.1	Objectif Général.....	32
2.2	Objectif PRD.....	32
3	Hypothèses	33
4	Bases méthodologiques	33
6	Description Générale	34
1	Environnement du projet	34
2	Caractéristiques des utilisateurs	35
3	Fonctionnalités et structure générale du système	35
4	Architecture générale du système	36
IV	Analyse et conception	38
7	Description des interfaces externes du logiciel	39
1	Interfaces matériel/logiciel	39
2	Interfaces homme/machine	39
2.1	Ergonomie du système :.....	40
2.2	Niveau de interface :.....	41
3	Interfaces logiciel/logiciel	41
4	Activité d'utilisateur	41
8	Diagramme de classes	43
9	Conception de base de donnée	44
V	Mise en oeuvre	46
10	Outils et librairies utilisées	47
11	Déploiement et implémentation	48
1	Fonctionnalités et déviation	48
1.1	Préparation du projet	48
1.1.1	Choix de images.....	48
1.1.2	Configuration de projet	49
1.1.3	Configuration de portable.....	50
1.1.4	Création de base de donnée.....	50

1.2	Version 1	51
1.3	Version 2	56
1.4	Version 3	57
1.5	Version 4	60
2	Limites	61
3	Choix techniques	61
12	Performance	62
1	Stratégie	62
1.1	Temps	62
1.2	Taux de reconnaissances.....	63
1.3	Précision d'interaction.....	63
VI	Plan de développement	64
13	Gestion de projet	65
1	Analyse de projet	65
2	Découpage du projets en tâches.....	65
3	Planning.....	66
3.1	Liste de tâches.....	66
3.2	Gantt	67
VII	Bilan et conclusion	68
VIII	Annexes	69
14	Spécifications fonctionnelles	70
1	Géolocalisation de l'utilisateur	70
2	Connexion et communication avec serveur	70
3	Reconnaissance et amélioration d'images 2D.....	71
4	Reconnaissance et amélioration partielle de l'image.....	72
5	Reconnaissance et amélioration d'objets 3D (optionnel).....	73
6	Ajouts des actions interactives	73
7	Gestion de coupons	74
15	Spécifications non fonctionnelles	76
1	Contraintes de développement, de conception et de maintenance.....	76
1.1	Matériels	76
1.2	Langages de programmation imposés ou adoptés	76

1.3	Logiciels et bibliothèques à utiliser pour le développement.....	76
1.4	Environnements nécessaires : simulateurs, outils logiciels ;.....	76
2	Contraintes de fonctionnement et d'exploitation	77
2.1	Performances.....	77
2.1.1	Du point de vue de l'utilisateur	77
2.1.2	Du point de vue de l'environnement	78
2.2	Capacités.....	78
2.3	Sécurité	78
2.4	Intégrité.....	78
2.5	Conformité aux standards	78
	Bibliographie	79
	GLOSSAIRE	80

Table des figures

1 Réalité augmentée

1	Réalité augmentée	3
2	Structure typique du système AR	5
3	Hololens (a) et Meta2 (b)	7
4	Google Glass(a) et "Magic Leap One"(b)	7
5	JigSpace Ar Education	10
6	Google Glass Ar en soins médicaux	11
7	Ar formation	11
8	Ar shopping - vente au détail.....	12
9	Ar essayage virtuelle.....	12
10	Facebook Ar	13
11	Ikea Ar APP - IKEA Place	14
12	Ar en l'industrie	14
13	Ar en Armée.....	15
14	Ar dans les jeux	16
15	Ar dans le film	16

2 ARCore

1	ARCore.....	19
2	Principe de fonctionnement Tango	20
3	Supported Models	20
4	Environnements de développement	21
5	MotionTracking	22
6	Environmental Understanding	22

7	Estimation de la lumière	23
8	Ancre nuage	24
9	Images augmentée	24
10	Qualité d'image test outil - arcoring	26
11	Oeuvre - art12 (global)	26
12	art01,art02,art03.....	27
13	Qualité des images d'une oeuvre	27
3	AR aujourd'hui : smartphone vs autonome	
1	Support Smartphont.....	28
2	Angle et Direction	29
4	Introduction de la société	
1	Flagtown Logo	30
6	Description Générale	
1	Environnement du projet	34
2	Cas d'utilisation application mobile	36
3	Architecture	36
7	Description des interfaces externes du logiciel	
1	Cartographie Humaine Machine.....	39
2	Warning	40
3	Type de navigation : Drawer	40
4	Activité d'utilisateur	41
8	Diagramme de classes	
1	Diagramme de classe	43
9	Conception de base de donnée	
1	Conception de BD.....	44
11	Déploiement et implémentation	
1	Article : Collection Loto-Québec.....	49
2	Échantillonnage d'œuvres inuits	49
3	Évaluation de qualité des images inuits.....	49
4	ARCore librairies	50
5	Création de base de donnée de multi-images	50

6	Implémentation dans projet.....	50
7	Charger la base de donnée préexistante	51
8	Créer la base de donnée pendant l'exécution	51
9	Classes importantes.....	51
10	Définition des noeuds d'intérêts.....	51
11	Lier les layouts rendrables pour les trois points d'intérêt.....	52
12	Chargement et configuration des points.....	52
13	Variables nécessaires	53
14	Version check.....	53
15	Créer AR scène	53
16	Charger la base de donnée	54
17	Initialisation de l'interface	54
18	le processus image tracking	55
19	Résultat version 1 : single-image augmentée.....	56
20	Version 1 avec une forme personnage.....	56
21	Mode pour base de donnée.....	57
22	Plusieurs images sont reconnues en même temps.....	57
23	Activité Bienvenue et Menu drawer.....	58
24	Superposition d'informations des points d'intérêt et Interaction : introduction animation video	59
25	Resultat final avec quelques amélioration	60
26	Changement de système de coordonnée.....	60
27	Variable pour contrôler l'état de vidéo	60
28	CompletionListener.....	60
29	Jouer qu'une vidéo.....	61
12	Performance	
1	Évaluation de la performance des images de différentes qualités	62
13	Gestion de projet	
1	Période du projet-Gantt0.....	67
2	Période du projet-Gantt1	67
14	Spécifications fonctionnelles	
1	Reconnaissance d'images 2D.....	71
2	Reconnaissance des points d'intérêt	72
3	Reconnaissance d'oeuvre 3D	73
4	Informations supplémentaire sur l'oeuvre	74

15 Spécifications non fonctionnelles

1	Android Stuido	77
---	----------------------	----

Première partie

Introduction

Ce document est pour but d'aborder l'ensemble de l'état d'art et des spécifications ainsi que la planification concernant le Projet Recherche et Développement (PRD) : Mobile App en Réalité Augmentée. Ce sujet est proposé par la société Flagtown qui est une Start-up mais qui se développe rapidement. Il nous oblige de le faire à partir de zero (from scratch). L'expression du besoin en forme Cahier de Charges est donc réalisée par l'encadrant du projet M. Gilles VENTURINI et l'étudiante Wenli YAN qui est l'auteur de ce document. La supervissions de la rédaction de ce cahier de spécification est réalisée par M. Jean Yves RAMEL. Ce projet se concentre sur la réalisation innovante, mais dans le processus de mise en œuvre, de nombreuses recherches dans ce domaine sont impliquées.

Ce projet vise à mettre en œuvre une application basée sur Android utilisant la dernière technologie AR, qui s'appelle ARCore, pour permettre aux utilisateurs d'en apprendre davantage sur les œuvres artistiques et culturelles d'une ville grâce à une expérience plus réaliste grâce à des interactions réelles.

Dans la première partie de cet article, je vous présenterai d'abord l'état de l'art et dans la deuxième partie je vous présenterai des détails des spécifications, et en plus la troisième partie sur le planning de ce projet.

Deuxième partie

L'état de l'art

1

Réalité augmentée

Ce sujet est basé sur le thème de la technologie de la réalité augmentée et nous allons d'abord présenter le concept, le principe, la méthode de mise en oeuvre de la réalité augmentée et le développement courant du monde dans ce domaine.

1 Concept



Figure 1 – Réalité augmentée

La technologie de la réalité augmentée (AR) consiste à incarner des informations physiques difficiles à expérimenter (tels que des informations visuelles, son, goût, toucher, etc.) dans un espace du monde réel. L'information est appliquée au monde réel et est perçue par les sens

humains pour réaliser une expérience sensorielle qui transcende la réalité. L'environnement réel et les objets virtuels se superposent en temps réel sur la même image ou le même espace. C'est une nouvelle technologie qui intègre de manière transparente des informations du "monde réel" avec des informations du monde virtuel. Il améliore la perception du monde réel par l'utilisateur et offre aux humains un nouveau moyen de communiquer avec le monde.

2 Caractéristiques

La technologie AR présente trois caractéristiques :

- La fusion de réalité virtuelle : Augmentez les scènes réalistes avec la superposition d'objets virtuels et d'environnements réels.
- L'interaction en temps réel : Les personnes interagissent en temps réel à travers l'appareil et la réalité augmentée.
- L'enregistrement en 3D : Il a mis l'accent sur la correspondance entre les objets générés par ordinateur et l'environnement réel. La position et la taille en trois dimensions de l'objet virtuel et de l'environnement réel doivent être parfaitement intégrés.

3 Historique

En fait, AR n'est pas une nouvelle idée. Dans le tableau ci-dessous, nous pouvons comprendre un peu.

- 1968 : Ivan Sutherland crée le premier système de Réalité Augmentée, qui est aussi le premier système de réalité virtuelle (le même Ivan Sutherland qui a créé le premier logiciel graphique le Sketchpad en 1963)
- 1992 : Tom Caudell et David Mizell définissent le terme "réalité augmentée" comme la superposition de matériel informatisé sur le monde réel
- 1994 : Steve Mann commença à porter une webcam pendant presque 2 ans
- 1996 : Jun Rekimoto présente les marqueurs 2D
- 1997 : Steve Feiner et AI présentent la touring machine, qui sera le premier système de Réalité Augmentée portable
- 1999 : Hirokazu Kato et Mark Billinghurst présentent ARToolkit
- 2006 : Reitmayr et AI présentent un modèle de tracking hybride pour de la Réalité Augmentée extérieure
- 2009 : Imberly Spreen et AI développent ARhrrrr(Un jeu de tir à réalité augmentée pour téléphones-appareils photo mobiles).

4 Principe

Architecture

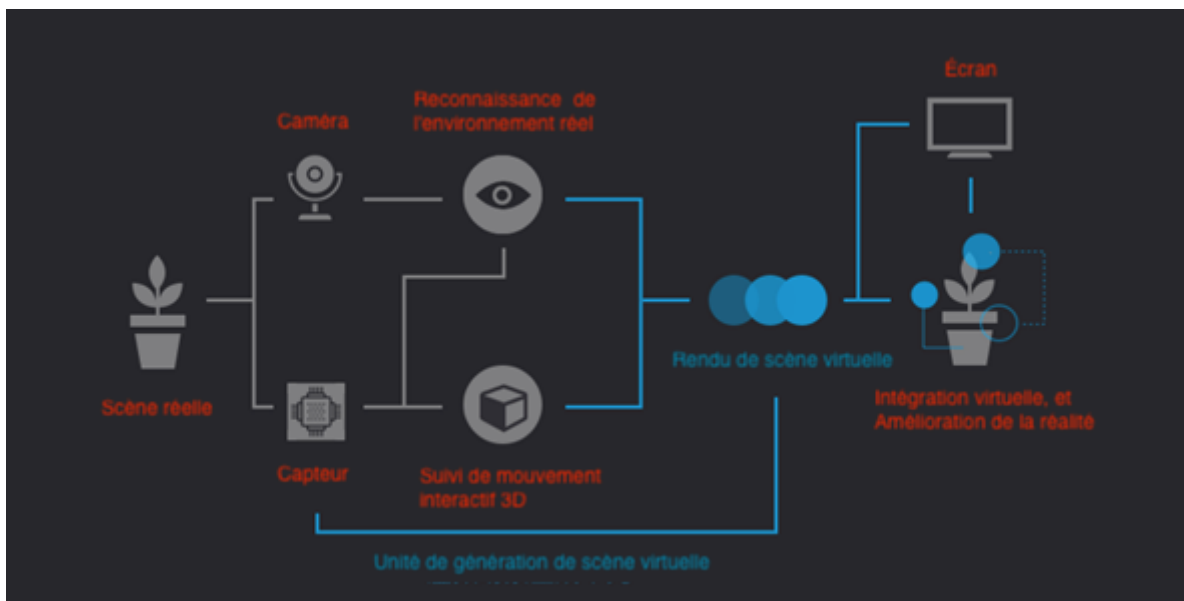


Figure 2 – Structure typique du système AR

(cité de : ARChina)

La figure ci-dessus est une structure de système AR typique composée d'une unité de génération de scène virtuelle et de dispositifs interactifs tels qu'un écran et un casque. L'unité de génération de scènes virtuelles est chargée de la modélisation, de la gestion, du dessin et de la gestion d'autres périphériques de la scène virtuelle, l'écran d'affichage est chargé d'afficher le signal combiné du virtuel et de la réalité, le dispositif de suivi de la tête suit le changement de la ligne de mire de l'utilisateur, le dispositif interactif est utilisé pour mettre en œuvre le signal sensoriel et l'entrée et la sortie du signal d'opération de contrôle d'environnement.

Tout d'abord, la caméra et le capteur collectent la vidéo ou l'image de la scène réelle, et l'unité de traitement en arrière-plan analyse et reconstruit celle-ci. Elle combine les données du dispositif de suivi de la tête pour analyser la position relative de la scène virtuelle et de la scène réelle afin d'aligner le système de coordonnées. Le calcul de fusion de la scène virtuelle est effectué, le dispositif d'interaction collecte un signal de commande externe afin de mettre en œuvre une opération d'interaction entre la scène virtuelle et réelle. Les informations après la fusion du système seront affichées à l'écran en temps réel et affichées dans le champ de vision humain.

5 Technologies clés

En gros, il y a deux concepts de base.

- MarkerDetection (détection de marqueur). C'est-à-dire que l'image (marqueur) est reconnue par l'objectif de la caméra ; et établir une association avec une partie correspondante du serveur pour déclencher l'expérience.
- MarkerTracking (Suivi des marqueurs), c'est la capacité de maintenir la direction en temps réel d'un objet physique ou d'une balise et de mettre à jour en permanence le contenu numérique pour le simuler.

5.1 Technologie d'enregistrement de suivi

Pour le système de réalité augmentée, une tâche importante consiste à obtenir la position et la posture actuelles de la caméra en temps réel et avec précision, et à déterminer la position de l'objet virtuel dans le monde réel, réalisant ainsi la fusion de l'objet virtuel et du monde réel. La méthode d'acquisition de la pose de la caméra est la technologie d'enregistrement de suivi. En termes de mise en œuvre, la technologie d'enregistrement de suivi peut être divisée en trois catégories : technologie d'enregistrement et de suivi par capteur, technologie d'enregistrement et de suivi sur la vision par ordinateur et technologie intégrée de suivi et d'enregistrement de vision et capteur.

Technologie d'enregistrement de suivi par capteur

La technologie d'enregistrement de suivi basée sur des capteurs suit et localise principalement la caméra par le biais de capteurs matériels tels que des capteurs de champ magnétique, des capteurs inertiels, des capteurs à ultrasons, des capteurs optiques et des capteurs mécaniques.

Technologie d'enregistrement de suivi basée sur la vision par ordinateur

La technologie d'enregistrement de suivi identifie et localise l'environnement de scène réel en analysant et en traitant les informations de données d'image capturées, déterminant ainsi la correspondance entre la scène réelle et les informations virtuelles. Cette méthode ne nécessite généralement que des informations d'image capturées par la caméra et requiert peu de matériel.

Technologie intégrée de suivi et d'enregistrement de la vision et des capteurs

Dans certaines applications de réalité augmentée, la technologie d'enregistrement de la Technologie d'enregistrement de suivi basée sur la vision par ordinateur ne permettent pas d'obtenir l'effet de suivi idéal, donc des chercheurs étudient en détail les avantages et les inconvénients des deux méthodes et les combiner pour obtenir un meilleur effet d'enregistrement de suivi.

Les plates-formes connus comme ARKit(Apple) et ARCore(Google) prennent respectivement en charge les systèmes d'exploitation iOS et Android, offrant ainsi des possibilités illimitées d'applications de réalité augmentée sur des périphériques intelligents mobiles.

5.2 Technologie d'affichage

Le but ultime de la technologie de réalité augmentée est de présenter un monde de réalité virtuelle. Par conséquent, la technologie d'affichage est une partie importante des systèmes de réalité augmentée.

À l'heure actuelle, les dispositifs d'affichage couramment utilisés comprennent les dispositifs d'affichage montés sur la tête, les dispositifs d'affichage d'écran d'ordinateur, les dispositifs d'affichage mobiles de poche et les dispositifs d'affichage de projection.

5.2.1 Dispositif d'affichage monté sur la tête

Étant donné que le système de réalité augmentée nécessite que l'utilisateur observe des images en temps réel dans le monde réel, le dispositif d'affichage monté sur la tête est principalement un affichage de casque transparent. La fonction principale de ces dispositifs est de fusionner des informations réelles dans l'environnement de l'utilisateur avec des informations virtuelles générées par ordinateur.

Hololens

Les lunettes de réalité augmentée Hololens de Microsoft utilisent la technologie holographique, associée à de multiples capteurs, pour projeter du contenu virtuel dans des images holographiques pour une fusion virtuelle et réelle. L'objectif intègre une unité de traitement centrale (CPU), une unité de traitement graphique (GPU) et un processeur holographique (HPU) pour permettre une interaction dans le monde réel sans qu'il soit nécessaire de connecter un autre périphérique.

Meta2

Meta2 est également des lunettes de réalité augmentée extrêmement immersif avec un champ de vision plus large que Hololens, mais la fidélité de suivi doit encore être optimisée et connectée à un ordinateur pour les calculs en cours d'utilisation. La figure ci-dessous montre des lunettes Hololens et Meta2.



Figure 3 – Hololens (a) et Meta2 (b)

Google Glass

Google Glass, introduit par Google, est un affichage optique monté sur la tête, transparent, qui est grossi de manière optique pour afficher les données à l'utilisateur via un prisme (Fig. (a)). Google Glass permet de contrôler les photos, les appels vidéo, le positionnement GPS, le traitement de texte, les e-mails et autres grâce au contrôle vocal.

Magic Leap

Magic Leap a publié un dispositif de réalité augmentée monté sur la tête et basé sur un champ lumineux, Magic Leap One (Fig. (b)). Le dispositif utilise une caméra externe et un processeur de vision informatique pour suivre la position de l'utilisateur en temps réel, tout en ajustant en permanence la distance focale des deux yeux pendant le processus de suivi et en affichant l'image contenant les informations de profondeur via l'affichage du champ lumineux.



Figure 4 – Google Glass(a) et "Magic Leap One"(b)

5.2.2 Dispositif d'affichage à l'écran

Un dispositif d'affichage à l'écran d'ordinateur a généralement une résolution plus élevée et un volume plus important qu'un dispositif de sortie conventionnel. Dans les applications de

réalité augmentée, de tels dispositifs conviennent mieux au rendu et à la superposition d'objets graphiques virtuels fins à l'intérieur ou dans un large éventail de scènes. Parce que ces dispositifs sont moins immersifs, mais à un prix inférieur, ils conviennent généralement aux systèmes de réalité augmentée bas de gamme ou multi-utilisateurs.

Dispositif d'affichage mobile de poche

Les terminaux mobiles intelligents se sont développés rapidement ces dernières années : la plupart des appareils portables intelligents existants sont équipés de caméras, de systèmes de positionnement global (GPS) et de gyroscopes, d'accéléromètres et d'autres capteurs, ainsi que de grands écrans haute résolution. La réalité augmentée fournit une bonne plateforme de développement. Comparé au dispositif d'affichage de type casque, le dispositif d'affichage portable tenu dans la main est généralement de petite taille, léger et facile à transporter, mais l'immersion est faible et les performances informatiques de différents dispositifs sont inégales en raison de limitations matérielles.

5.2.3 Dispositif d'affichage par projection

Le dispositif d'affichage par projection peut projeter l'image de réalité augmentée dans un large éventail d'environnements afin de répondre à la demande de l'utilisateur en matière d'affichage sur grand écran.

Étant donné que la focalisation de l'image générée par le dispositif d'affichage par projection ne change pas avec la perspective de l'utilisateur, elle convient mieux aux environnements de réalité augmentée en intérieur.

5.3 Technologie d'interaction homme-machine

L'objectif du système de réalité augmentée est de créer un monde amélioré de réalité virtuelle, permettant aux utilisateurs de se sentir d'objets virtuels quasi réels dans le monde réel et d'offrir aux gens l'interaction avec ce monde amélioré.

Dans ce processus, la qualité de l'interaction homme-machine affecte grandement l'expérience utilisateur. De manière générale, les méthodes d'interaction traditionnelles comprennent principalement un clavier, une souris, un appareil tactile, un microphone, etc. Ces dernières années, certains modes d'interaction plus naturels basés sur la voix, le toucher, les mouvements des yeux, les gestes et le sens du corps sont apparus.

5.3.1 Technologie interactive basée sur des périphériques matériels traditionnels

La souris, le clavier et la poignée sont des outils interactifs courants du système de réalité augmentée : l'utilisateur peut sélectionner un point ou une région de l'image à l'aide de la souris ou du clavier pour effectuer le zoom, le glissement et le déplacement de l'objet virtuel au point ou à la région. Ce type de méthode est simple et facile à utiliser, mais requiert la prise en charge d'un périphérique d'entrée externe, ne fournit pas une expérience interactive naturelle à l'utilisateur et réduit le sentiment de naufrage du système de réalité augmentée.

5.3.2 Technologie interactive basée sur la reconnaissance vocale

La langue est le moyen de communication le plus direct entre les êtres humains. L'interaction linguistique contient une grande quantité d'informations et une grande efficacité. Par conséquent, la reconnaissance vocale est devenue l'une des méthodes d'interaction homme-machine importantes dans les systèmes de réalité augmentée. Ces dernières années, le développement de l'intelligence artificielle et l'amélioration des capacités de traitement informatique ont permis à la technologie de reconnaissance vocale de mûrir et d'être largement utilisée dans les terminaux intelligents, les plus représentatifs étant Siri d'Apple et Cortana de Microsoft. Prend en charge la saisie en langage naturel, obtient des instructions par reconnaissance vocale, renvoie les résultats les plus pertinents en fonction des besoins de l'utilisateur, réalise une interaction homme-machine naturelle et améliore considérablement l'efficacité du travail de l'utilisateur.

5.3.3 Technologie interactive tactile

La technologie interactive tactile est une méthode de saisie humaine plus conviviale que la saisie traditionnelle au clavier et à la souris. La popularité des appareils mobiles intelligents a permis aux technologies interactives tactiles d'évoluer rapidement et d'être plus facilement reconnues par les utilisateurs. Au cours des dernières années, la technologie interactive tactile a évolué pour passer d'une touche unique à une touche tactile, permettant ainsi une interaction doigt unique-multipoint ou multi-utilisateurs, qui peut être utilisé à deux mains. Cliquez, double-cliquez, etc. en reconnaissant différents gestes.

5.3.4 Technologie interactive basée sur la reconnaissance de mouvement

La technologie d'interaction basée sur la reconnaissance de mouvement calcule et traite la position des pièces clés obtenues par le système de capture de mouvement, analyse le comportement des actions de l'utilisateur, le convertit en commandes de saisie et réalise l'interaction entre l'utilisateur et l'ordinateur. Hololens de Microsoft utilise une caméra profonde pour capturer les gestes des utilisateurs et manipule des objets virtuels sur l'interface via un suivi manuel. Meta's Meta2 et Magic Leap's Magic Leap One permettent également aux utilisateurs d'interagir avec des gestes. Ce type d'interaction réduit non seulement le coût des interactions homme-machine, mais se conforme également aux habitudes naturelles de l'être humain. Il est plus naturel et plus intuitif que la méthode traditionnelle d'interaction. Il est actuellement au centre de l'interaction homme-machine.

5.3.5 Technologie interactive basée sur le suivi oculaire

La technologie d'interaction basée sur le suivi oculaire analyse et détermine les points de regard de l'œil humain en capturant les changements subtils autour de l'œil lorsque l'œil humain regarde dans différentes directions. Il les convertit en signaux électriques et les envoie à l'ordinateur pour réaliser l'interaction entre l'homme et l'ordinateur. Il n'est pas nécessaire d'entrer manuellement ce processus. Magic Leap Magic Leap One est équipé d'un capteur permettant de suivre les mouvements des yeux à l'intérieur des lunettes pour contrôler l'ordinateur en suivant les yeux.

6 Applications d'AR

Ces dernières années, avec le développement de la science et de la technologie, la technologie de réalité augmentée et ses applications sont multiples et ont été largement utilisée dans de nombreux domaines tels que :

6.1 Éducation

Les applications AR changent la façon dont elles enseignent de manière plus interactive.

Exemple

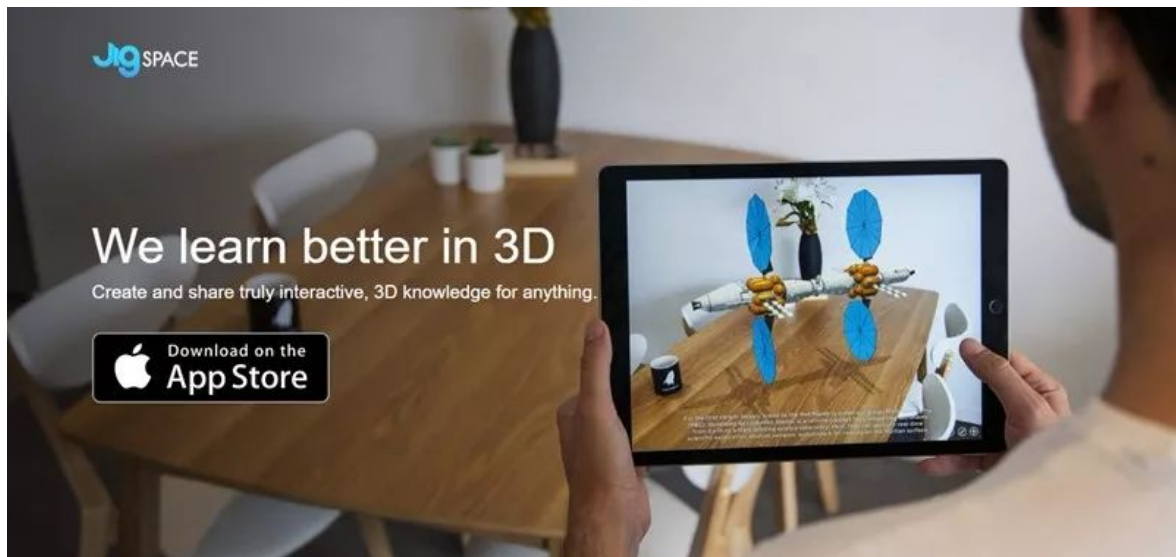


Figure 5 – JigSpace Ar Education

JigSpace est un outil d'apprentissage développé à partir de l'outil Apple ARKit. Les étudiants peuvent utiliser des appareils mobiles pour explorer des modèles virtuels 3D d'objets sur la table afin de comprendre la structure interne de divers objets.

En projetant un modèle 3D interactif dans le RA, les concepts et objets abstraits peuvent être scindés étape par étape, offrant ainsi aux étudiants l'expérience la plus intuitive possible.

6.2 Soins médicaux

Les soins de santé sont également l'un des principaux domaines d'application des AR, et les applications des AR en médecine sont de plus en plus nombreuses. Elles ont été appliquées avec succès dans les domaines de l'éducation, de la formation, de l'analyse des patients et de la chirurgie.

Exemple



Figure 6 – Google Glass Ar en soins médicaux

Dès 2015, les chirurgiens de l'Institut de cardiologie de Varsovie ont utilisé Google Glass pour faciliter le traitement chirurgical afin de comprendre l'occlusion coronaire du patient. Des étudiants de l'école de médecine Case Western Reserve University utilisent HoloLens pour disséquer des tissus virtuels sur des cadavres numérisés.

6.3 Formation en entreprise

L'application de la réalité augmentée dans le domaine de la formation technique a attiré l'attention de nombreuses entreprises. Honeywell International a par exemple introduit des outils de réalité mixte pour la formation des employés.



Figure 7 – Ar formation

«De nombreuses études sur la génération millennials montrent qu'elles recherchent un apprentissage plus expérientiel.» La société guide les employés à travers les lunettes HoloLens afin de former et de tester les capacités opérationnelles d'un véritable environnement d'usine.

6.4 Shopping

Avec le développement rapide des smartphones, la réalité augmentée offre davantage de possibilités. Presque tout le monde peut utiliser un téléphone portable pour faire l'expérience du monde de la réalité augmentée.

Pour plusieurs fabricants de produits de base, les étagères des détaillants physiques vont devenir un nouveau champ de bataille pour les promotions AR.

Exemple - Application de vente au détail

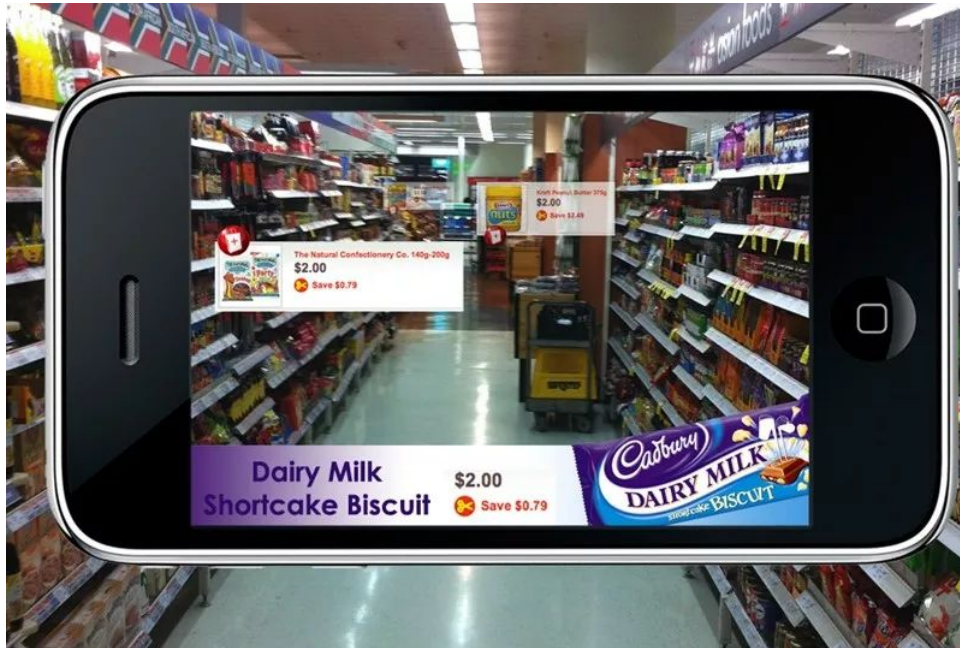


Figure 8 – Ar shopping - vente au détail

Une application développée par Aurora Reality, une société de développement de technologies de réalité augmentée, permet aux consommateurs de visualiser en temps réel des informations sur les produits en magasin. Elle utilise la technologie de vision par ordinateur et le suivi en magasin pour aider les clients à trouver les produits dont ils ont besoin.

Exemple - Essayage Virtuelle



Figure 9 – Ar essayage virtuelle

La pose est un processus fastidieux lors de l'achat de vêtements dans un centre commercial. De plus en plus de magasins ont adopté la technologie AR pour l'expérience de l'ajustement.

La cabine d'essayage virtuelle développée par Bevond, un développeur de technologies de réalité augmentée aux États-Unis, permet aux clients de faire l'expérience du port d'un vêtement sans avoir à le changer. La technologie peut également résoudre un problème pour les achats en ligne : l'écart entre les photos des modèles de site Web et l'effet de pansement.

6.5 Marketing publicitaire basé sur la localisation

Combinée au support du GPS mobile et d'autres systèmes de positionnement, la publicité géolocalisée n'est pas nouvelle, mais la publicité associée à la RA est encore nouvelle.

Exemple



Figure 10 – Facebook Ar

Facebook a récemment lancé une application AR qui permet aux développeurs de créer des éléments déclenchés par des déclenchements géographiques.

6.6 Design d'intérieur

La décoration intérieure est une tâche très complexe, qui doit prendre en compte l'espace et la taille, le matériau, la couleur et la correspondance des différents meubles. Et AR permet aux gens ordinaires de concevoir facilement des intérieurs et des meubles de maison.

Exemple



Figure 11 – Ikea Ar APP - IKEA Place

IKEA a développé l'application AR IKEA Place basée sur Apple ARKit, qui permet aux consommateurs de visualiser les effets des produits IKEA placés à la maison via des appareils mobiles afin de faire des choix plus satisfaisants et de les enregistrer ou de les ajouter au panier d'achat.

Similaire à IKEA, Sotheby's International Real Estate a également lancé l'application Curate, qui permet aux acheteurs potentiels de placer virtuellement leur choix de mobilier dans un espace commercial ou chez eux qu'ils souhaitent acheter ou louer pour obtenir les meilleurs résultats.

6.7 Industrie

(conception, design, maintenance, assemblage, pilotage, robotique et télérobotique, implantation, étude d'impact, etc.)



Figure 12 – Ar en l'industrie

6.8 Armée

La réalité augmentée a également été largement utilisée dans l'armée.

Transmission de renseignements secrets

Les agents peuvent utiliser AR pour indiquer des cibles spécifiques, visibles uniquement par des personnes spécifiques, et améliorer la sécurité.

Dépistage de l'amitié

Une large gamme de couvertures AR distingue les forces amies et les civils, marquant les forces amies et les civils, les véhicules ennemis et ennemis de couleurs ou de symboles différents pour minimiser les pertes.

Améliorer la vision du pilote

Identifier les cibles, éviter les obstacles, naviguer et améliorer la précision des cibles.

Détection de bombe

Utilisez AR pour détecter les bombes, recevoir des informations en retour, marquer l'emplacement de la bombe et éviter tout danger.

Opérations sur les armes

Fournit des instructions pour l'utilisation d'armes, de véhicules, de drones et d'autres ressources.

Affichage de l'état

Affiche les munitions restantes, de la nourriture et de l'eau, la santé de l'individu et de toute l'équipe, les détails de la mission et d'autres informations clés.

Exemple



Figure 13 – Ar en Armée

L'armée aérienne australienne teste l'utilisation de Microsoft HoloLens. Aider au développement de stratégies, à la gestion des menaces et à la formation en fournissant une vision en réalité augmentée (ARV). La technologie de RA aura un «impact significatif» sur les communications, la visualisation et l'entraînement militaires. Dans l'environnement de formation, les formateurs et les stagiaires peuvent partager une expérience interactive riche en images et le monde réel peut superposer des images et des informations numériques sous forme d'hologrammes.

6.9 Jeux vidéo

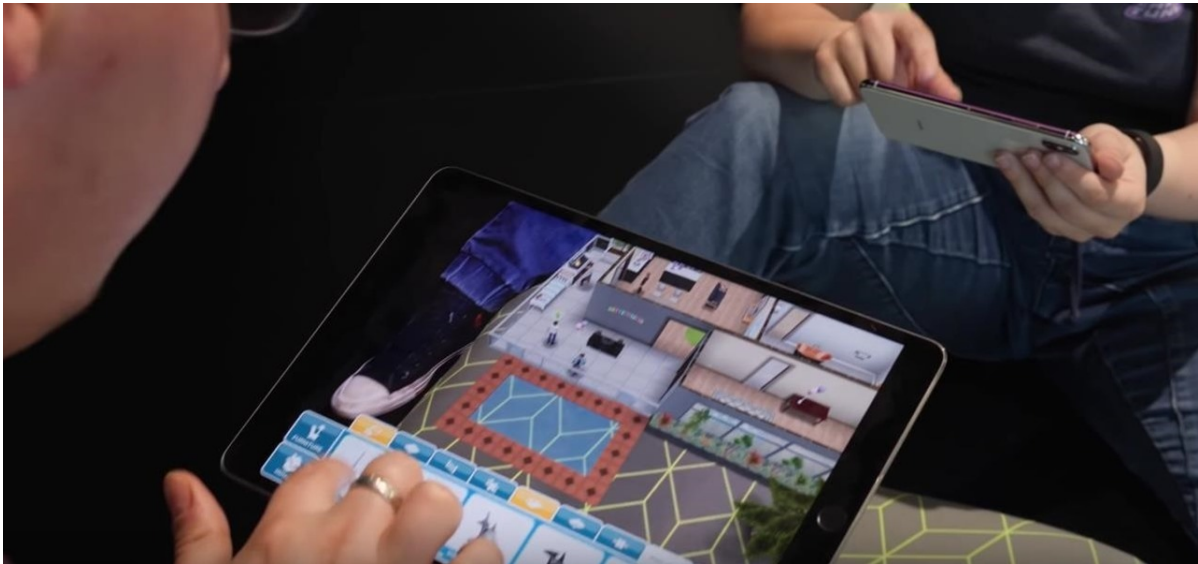


Figure 14 – Ar dans les jeux

La réalité augmentée joue un rôle important dans le jeu et se développe depuis longtemps. Il apporte une toute nouvelle expérience au jeu. Bien qu'il soit moins immersif que les jeux de réalité virtuelle, il possède les atouts de la combinaison du monde réel. Il existe également de nombreuses applications de ce type. Un grand nombre de jeux AR sont apparus dans le récent app store.

6.10 Cinéma et la télévision



Figure 15 – Ar dans le film

AR est appliquée au post-production, studios virtuels, retransmissions sportives etc dans le film ou la télévision.

Dans de nombreux films, nous pouvons également voir les applications de la réalité augmentée des gens dans le futur, telles que le contrôle de l'écran dans les airs et l'interaction les uns avec les autres.

6.11 Culture et Art

Visite guidée du musée et de la ville

Lorsque on visite un musée ou passez devant les monuments de la ville, vous pourrez instantanément en savoir plus sur les expositions du musée et sur le contexte culturel des monuments.

Art Analyse

Comprenez les développements antérieurs de l'œuvre et les œuvres finales créées sur ces fondations, ainsi que les techniques et techniques utilisées par les artistes.

Enseignement de l'art

Selon les étapes et les points indiqués par AR, apprenez à peindre et à graver pas à pas.

La restauration de l'histoire

La restauration des sites historiques grâce à la technologie AR, a permis de voir l'aspect original des monuments.

Lecture

Vous pouvez verrouiller vos livres dans votre champ de vision, que vous soyez en train de marcher ou de courir, vous pouvez les lire à tout moment, n'importe où, et vous pouvez les lire et les partager avec d'autres à distance.

Nature Tour

Identifiez les types de plantes et d'animaux et découvrez-les.

7 Situation actuelle de développement

Au cours des dernières années, la technologie de la réalité augmentée a retenu l'attention des chercheurs. Poussée par la technologie de la vision par ordinateur et de l'intelligence artificielle, la technologie de réalité augmentée a montré un fort élan de développement, qu'il s'agisse de suivre l'exactitude de l'enregistrement, les performances du dispositif d'affichage ou la nature naturelle de l'interaction homme-machine. En plus, le développement de terminaux intelligents (tels que les téléphones mobiles intelligents) a créé davantage d'opportunités et plus de possibilités pour la réalité augmentée. Plus de gens auront l'occasion d'expérimenter la réalité augmentée.

En plus, bien que la méthode d'enregistrement en réalité augmentée pour les images 2D ait été plus étudiée dans les milieux universitaires, la méthode d'enregistrement en réalité augmentée pour les objets 3D a encore de grandes limites. Les méthodes existantes de reconnaissance et d'enregistrement tridimensionnelles d'objets basées sur la vision par ordinateur posent généralement des problèmes tels qu'un espace de recherche important et un temps de calcul long.

La technologie d'enregistrement visuel traditionnelle convient à un environnement réel relativement simple, par exemple lorsque la caméra est immobile par rapport à l'objet et qu'une fois

que la scène réelle est complexe ou que la caméra se déplace rapidement par rapport à l'objet, la quantité de calcul de l'algorithme d'enregistrement est considérablement augmentée et l'effet d'enregistrement est considérablement modifié. Mauvais, même si certains objets sont occlus, l'enregistrement échoue.

Donc cependant, on peut voir qu'il reste encore beaucoup de problèmes à résoudre dans la technologie de la réalité augmentée. Du point de vue de la technologie d'enregistrement de suivi, la méthode actuelle d'enregistrement de suivi ne peut utiliser qu'une petite quantité d'informations dans la scène, telles que des informations sur les points caractéristiques, ce qui amène le système à comprendre l'environnement de manière incomplète et, grâce à la technologie d'affichage, elle peut donner aux utilisateurs une grande perte. Le sens des lunettes de réalité augmentée ne peut pas répondre aux besoins du public en termes de volume et de prix ; de manière interactive, la technologie interactive plus naturelle de la réalité augmentée prenant en charge plusieurs utilisateurs reste à étudier.

2

ARCore

ARCore est la nouvelle plate-forme de réalité augmentée de Google et il est le cœur de notre projet.

1 Introduction d'ARCore



Figure 1 – ARCore

ARCore est la plateforme de construction d'expérience de réalité augmentée de Google. ARCore utilise différentes API pour informer votre téléphone de son environnement, comprendre le monde réel et interagir avec les informations.

Il support la plateforme Android et iOS.

En fait, avant la naissance d'ARCore, l'investissement de Google dans la réalité augmentée était réservé à Tango, qui est presque le prédécesseur d'ARCore. La fonction de réalité augmentée de Tango est très puissante, mais elle a de grandes exigences pour les appareils des utilisateurs. Avec un équipement de caméra en profondeur, qui limite considérablement les développeurs et son marché, beaucoup moins compétitif que ARKit d'Apple, dès le début de 2018, Google s'est tourné vers l'investissement ARCore, il prend en charge les smartphones ordinaires, sans besoin de la caméra de profondeur. Comme le Tango, ARcore est en mesure d'exécuter parfaitement des fonctions puissantes telles que la détection de plan.



Figure 2 – Principe de fonctionnement Tango

2 Appareils supportés

ARCore peut être utilisé sur une variété de téléphones Android éligibles sous Android 7.0 (Nougat) et plus. Une liste complète de tous les périphériques pris en charge est présentée ci-dessous :

Manufacturer	Model	Notes
Asus	Zenone AR	
	Zenfone ARES	
Google	Nexus 5X	Requires Android 8.0 or later
	Nexus 6P	Requires Android 8.0 or later
	Pixel, Pixel XL	
	Pixel 2, Pixel 2 XL	
HMD Global	Nokia 6 (2018)	Also known as Nokia 6.1
	Nokia 8 Sirocco	
Huawei	P20, P20 Pro	
	Mate RS Porsche Design	
LG	G6	Requires Android 8.0 or later
	G7 ThinQ	
Motorola	Moto G6 Plus	
	Moto Z2 Force	
OnePlus	OnePlus 3T	Requires Android 8.0 or later
	OnePlus 5	
Samsung	Galaxy A5 (2017)	Samsung devices with a model number ending in 0 or 8 are not supported , e.g. SM-G9600.
	Galaxy A7 (2017)	
	Galaxy A8, Galaxy A8+ (2018)	
	Galaxy Note8	
	Galaxy S7, Galaxy S7 edge	
	Galaxy S8, Galaxy S8+	
Sony	Galaxy S9, Galaxy S9+	
	Xperia XZ Premium	Requires Android 8.0 or later
Xiaomi	Xperia XZ1, Xperia XZ1 Compact	Requires Android 8.0 or later
	Mi Mix 2S	

Figure 3 – Supported Models

La liste des périphériques pris en charge est toujours en cours de mise à jour.

3 Environnement de développement

ARCore fournit des SDKs pour la plupart des environnements de développement les plus courants.

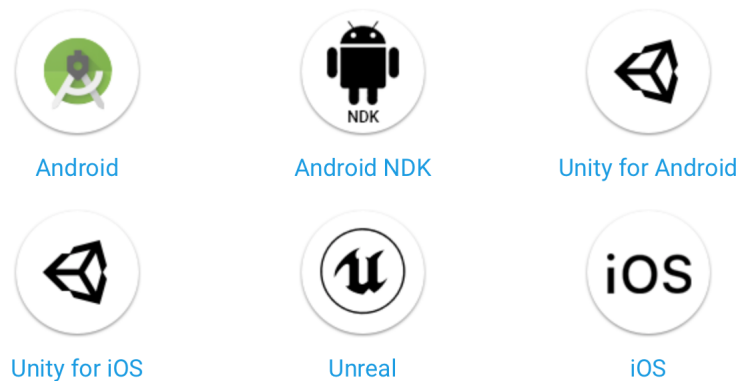


Figure 4 – Environnements de développement

Ces SDKs fournissent des API natives pour toutes les fonctionnalités essentielles de la réalité augmentée, telles que le suivi de mouvement, la compréhension de l'environnement et l'estimation de la lumière. Ces fonctionnalités nous permettent de créer de toutes nouvelles expériences de réalité augmentée ou d'améliorer les applications existantes avec les caractéristiques de la réalité augmentée.

4 Principe de fonctionnement

Pour résumer, ARCore a deux objectifs : suivre sa position à mesure que les appareils mobiles se déplacent et développer leur propre compréhension du monde réel.

La technologie de suivi des mouvements d'ARCore utilise une caméra de téléphone portable pour identifier les points d'intérêt (appelés points caractéristiques) et suivre le mouvement de ces points dans le temps. En combinant le mouvement de ces points avec les lectures du capteur inertiel du téléphone, ARCore peut déterminer sa position et l'orientation de l'écran à mesure que le téléphone se déplace.

En plus d'identifier les points clés, ARCore détecte également les surfaces planes (telles que les tables ou les sols) et estime l'intensité lumineuse moyenne dans les environs. Ensemble, ces fonctionnalités permettent à ARCore de développer sa propre compréhension du monde qui vous entoure.

Grâce à sa compréhension du monde réel, ARCore peut ajouter des objets, des notes ou d'autres informations de manière transparente au monde réel. Vous pouvez placer un chaton ronflant dans le coin de la table basse ou utiliser la biographie de l'artiste pour ajouter une note à un tableau. Le suivi du mouvement signifie que vous pouvez déplacer et afficher ces objets sous n'importe quel angle, même si vous vous détournez de la pièce. À votre retour, le chaton ou les notes se trouveront à la place que vous avez ajoutée.

5 Caractéristiques de base - ARCore 1.0

En février 2018, Google publiait officiellement ARCore 1.0, un kit de développement d'outil logiciel de réalité augmentée pour la plate-forme Android. Il prend en charge 13 téléphones

phares Android et dispose de trois fonctions similaires à celles de Apple ARKit 1.0 : détection de plan horizontal, suivi du mouvement et estimation de l'éclairage.

5.1 Suivi de mouvement

Le suivi des mouvements permet au téléphone de comprendre et de suivre sa position par rapport au monde réel.

Lorsque votre téléphone se déplace dans le monde réel, ARCore comprend sa position par rapport au monde qui l'entoure via un processus appelé télémétrie et mappage parallèles (ou COM). ARCore détecte les caractéristiques visuellement distinctes (appelées points de caractéristique) dans l'image capturée de la caméra et utilise ces points pour calculer leurs changements de position. Ces informations visuelles, combinées aux mesures inertielles de l'IMU du dispositif, permettent d'estimer la pose (position et orientation) de la caméra par rapport au monde environnant au fil du temps.

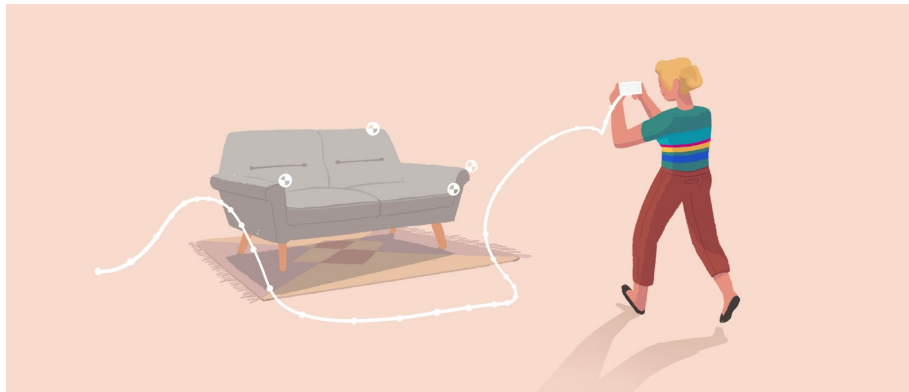


Figure 5 – MotionTracking

En alignant la pose de la caméra virtuelle qui restitue le contenu 3D sur celle de la caméra de périphérique fournie par ARCore, le développeur peut restituer le contenu virtuel dans la perspective correcte. L'image virtuelle rendue peut être superposée à l'image extraite de la caméra du périphérique, ce qui donne au contenu virtuel l'apparence du monde réel.

5.2 Compréhension de l'environnement

La compréhension de l'environnement permet au téléphone de détecter la taille et la position de diverses surfaces telles que des surfaces horizontales, verticales et en pente telles que des sols, des tables basses ou des murs.

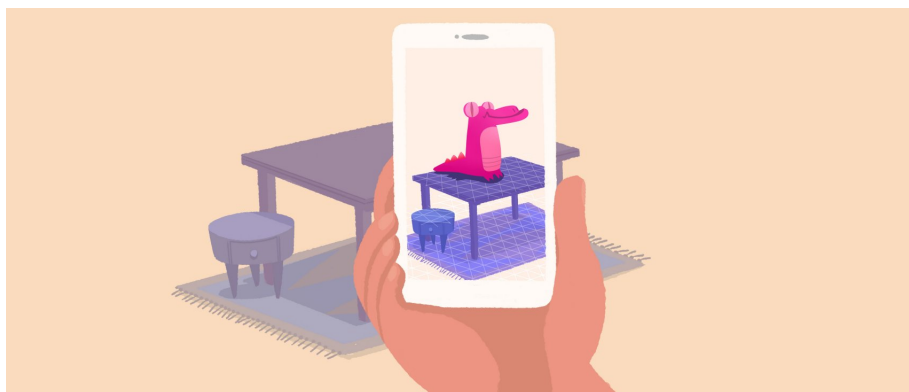


Figure 6 – Environmental Understanding

ARCore améliore continuellement sa compréhension de l'environnement réel en détectant des points et des plans.

ARCore peut rechercher des points de caractéristique en cluster qui semblent se trouver sur des surfaces communes horizontales ou verticales, telles que des tables ou des murs, et permettre à ces surfaces d'être utilisées comme des surfaces plates par votre application. ARCore peut également déterminer les limites de chaque plan et fournir ces informations à votre application. Vous pouvez utiliser ces informations pour placer un objet virtuel sur une surface plane.

Dans la mesure où ARCore utilise des points de repère pour détecter les plans, les surfaces planes dépourvues de texture, comme les murs blancs, risquent de ne pas être détectées correctement.

5.3 Estimation de la lumière

L'estimation de la lumière permet au téléphone d'estimer les conditions d'éclairage actuelles de l'environnement.

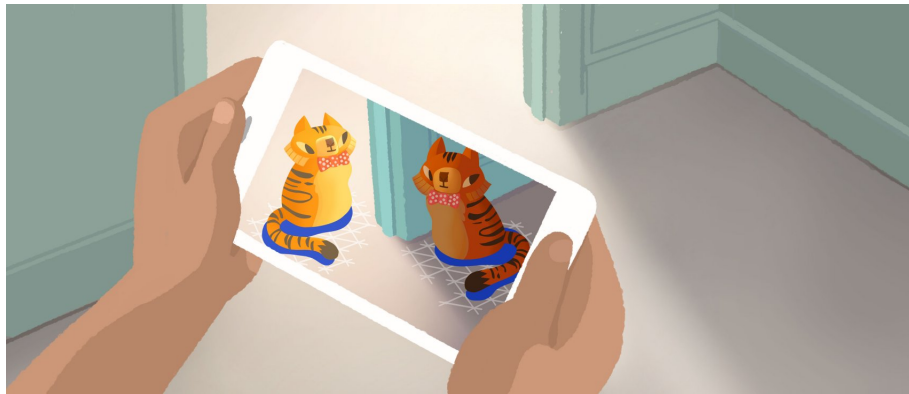


Figure 7 – Estimation de la lumière

6 Caractéristiques et supériorités puissantes de la version ARCore 1.2

Depuis qu'ARCore a été publié par Google, chaque mise à jour a fait une avancée décisive, en particulier ARCore 1.2. Elle offre des fonctionnalités plus puissantes, il y a trois mises à jour majeures : Les ancres de nuage prenant en charge le partage AR pour plusieurs personnes, les images augmentées avec une nouvelle reconnaissance de plan vertical et le nouveau SDK Sceneform basé sur Java.

6.1 Ancre nuage qui permet à plusieurs personnes partageant la RA

ARCore implémente le suivi de mouvement en identifiant les caractéristiques planaires et en créant des nuages de points clairsemés. Les ancres de nuage peuvent stocker ces informations dans le nuage de Google et fournir aux développeurs l'identifiant de ces informations. Cet ID peut être partagé avec d'autres utilisateurs afin que vous puissiez comparer deux nuages de points privés et coller les objets AR au même endroit. À l'aide d'ancres dans le nuage, un appareil peut envoyer des points d'ancrage et des points de fonctionnalité proches au nuage pour l'hébergement. Ces ancres peuvent être partagées avec d'autres utilisateurs sur des appareils dans le même environnement. Cela permet à l'application de restituer le même objet 3D connecté à ces ancres, ce qui permet aux utilisateurs de bénéficier de la même expérience de synchronisation en synchronisation.



Figure 8 – Ancre nuage

Cloud Anchors prend également en charge partiellement ARKit, vous pouvez donc synchroniser les données AR entre Android et iOS pour partager l'expérience, quelle que soit la plate-forme que vous utilisez.

6.2 Ajout d'images augmentées pour la reconnaissance de plan vertical

Cette fonctionnalité ajoute une reconnaissance de plan vertical. Par exemple, dans l'image ci-dessous, vous pouvez présenter le produit contenu dans l'emballage de manière AR en identifiant la surface d'emballage verticale du produit.



Figure 9 – Images augmentée

Les images augmentées sont une nouvelle fonctionnalité similaire au QRCode et de numérisation AR, un système de reconnaissance d'image universel qui utilise l'appareil photo du téléphone pour pointer sur un objet afin de lancer l'expérience AR. Les développeurs peuvent spécifier jusqu'à 1 000 images 2D pour leurs applications, ce qui permet à ARCore de détecter des

éléments tels que des boîtes de produits ou des affiches de films sans le balisage AR carré complexe. Cette fonctionnalité propose un large éventail de scénarios professionnels, tels que la création de posters de films, de dépliants et d'instructions d'assemblage de produits.

6.3 Un nouveau Sceneform SDK basé sur Java

Sceneform est un nouveau SDK conçu pour aider les développeurs Java à créer des scénarios. Comme AR nécessite l'affichage des objets 3D, les développeurs doivent avoir la possibilité d'utiliser des NDK Android, OpenGL et des moteurs de jeu comme Unity, ce qui peut paraître décourageant pour les développeurs Java Android.

Le kit de développement Sceneform est un plug-in et une API Android Studio permettant aux développeurs de créer facilement une expérience AR sans avoir à gérer l'intégralité de la pile de développement 3D. Sceneform contient un certain nombre de widgets d'interface utilisateur 3D courants permettant aux utilisateurs de sélectionner des objets ou de déplacer des éléments sans que le développeur ait à recréer la roue.

7 Conditions et performance

7.1 Capacité

Stockage d'image dans une base de données ARCore

Chaque base de données d'images peut stocker des informations sur les points caractéristiques pour un maximum de 1000 images de référence.

Capacité de suivi

ARCore peut suivre jusqu'à 20 images simultanément dans l'environnement, mais il ne peut pas suivre plusieurs instances de la même image.

Environnement d'exploitation

Tout le suivi se produit sur l'appareil, donc aucune connexion Internet n'est requise. Les images de référence peuvent être mises à jour sur l'appareil ou sur le réseau sans nécessiter de mise à jour d'application.

7.2 Qualité des images

Caractéristique

Des images qui contiennent de nombreux points caractéristiques uniques, moins de répétabilité.

L'image physique dans l'environnement doit mesurer au moins 15 cm sur 15 cm et doit être plate (par exemple, pas froissée ou enroulée autour d'une bouteille).

Taille

Au moins 300*300px

Format

Format : png, jpeg

Tests de la qualité de l'image

On a déjà l'outil pour tester la qualité des images : `arcoring`. On peut le trouver dans le package et (choisir le bon dossier : pour linux, macOS ou windows) :

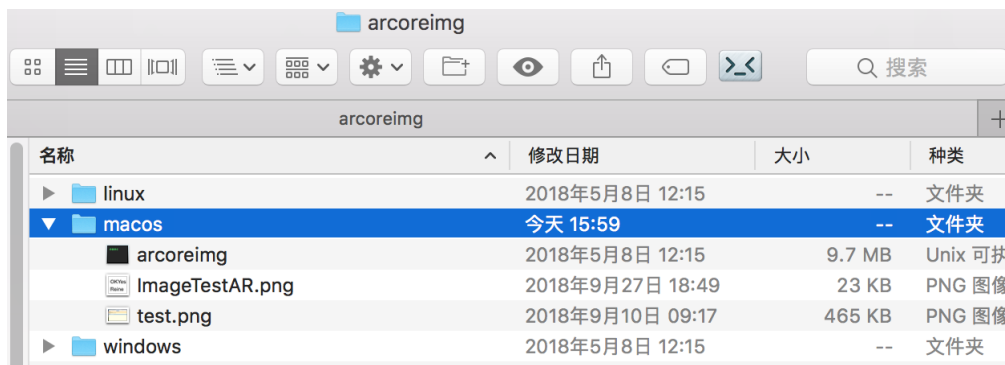


Figure 10 – Qualité d'image test outil - `arcoring`

Avec cet outil, on a testé plusieurs images pour des recherches sur :

- Comment choisir différentes images pour différentes qualités d'image
- L'effet de la qualité de l'image sur la reconnaissance. Y compris : vitesse de reconnaissance, stabilité de rotation, etc.

On a découpé une oeuvre en plusieurs morceaux par des points d'intérêt :



Figure 11 – Oeuvre - art12 (global)

Figure 12 – *art01,art02,art03*

Le résultat de test est comme indiqué ci-dessous :

```
YanwenlideMacBook-Pro:macos yanwenli$ ./arcoreimg eval-img --input_image_path=./images/art01.jpg
35
YanwenlideMacBook-Pro:macos yanwenli$ ./arcoreimg eval-img --input_image_path=./images/art02.jpg
50
YanwenlideMacBook-Pro:macos yanwenli$ ./arcoreimg eval-img --input_image_path=./images/art03.jpg
100
YanwenlideMacBook-Pro:macos yanwenli$ ./arcoreimg eval-img --input_image_path=./images/art12.jpg
20
YanwenlideMacBook-Pro:macos yanwenli$
```

Figure 13 – *Qualité des images d'une oeuvre*

On peut trouver que dans une peinture, la qualité d'image de différentes régions est très différente. L'image 'art03' a obtenu 100 points car elle est plus riche en couleurs (faible répétabilité).

3

AR aujourd'hui : smartphone vs autonome

Côté AR mobile C'est la première fois que la plupart des gens dans le monde font l'expérience de la réalité augmentée. En fait, le développement rapide des smartphones a effectivement favorisé le développement des industries de la RV et de la RA. En effet, certaines parties du smartphone fonctionnent : les gyroscopes, les accéléromètres, les écrans miniaturisés à haute résolution et les casques anti-bruit et réalité virtuelle nécessitent également les mêmes composants.

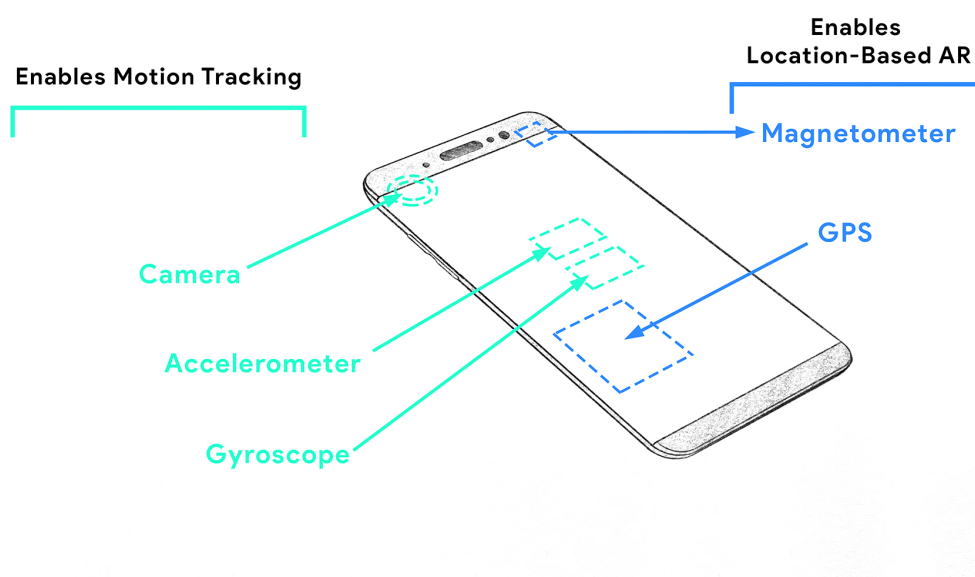


Figure 1 – Support Smartphone

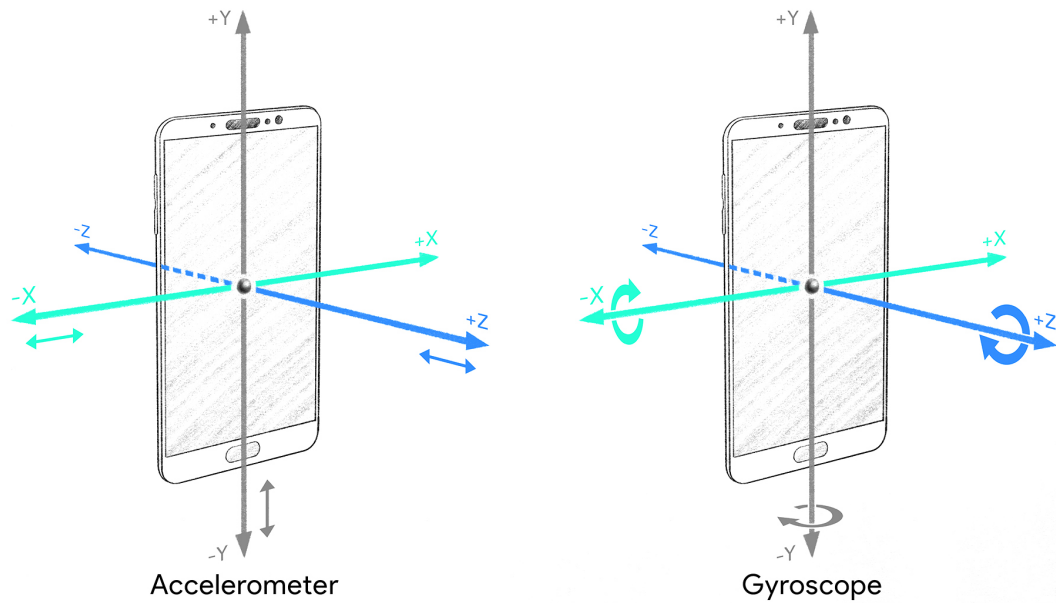


Figure 2 – Angle et Direction

Au cours des 10 dernières années, la forte demande pour les smartphones a entraîné la production en masse de ces composants, ce qui a entraîné une innovation matérielle et une réduction des coûts. À la base, AR crée un appareil photo qui utilise l'avant et l'arrière de notre téléphone. Nous décrochons notre téléphone et l'écran peut afficher des objets numériques et des informations intégrées à notre monde réel. Notre téléphone peut maintenant être la passerelle vers de nouvelles expériences et informations du monde.

4

Introduction de la société



Figure 1 – *Flagtown Logo*

FLAGTOWN, société créée à 2016 en France, installée à SAINT CYR SUR LOIRE (37540), elle est le concepteur et fabricant français d'enseignes professionnelles, spécialisant dans le secteur d'activité de l'édition de logiciels applicatifs. Elle possède nombreux produits et services connectés pour une bonne expérience de visiteur. Le fondateur et le CEO est monsieur Stéphane Grandjean.

Troisième partie

Cahier de spécification Système

5

Contexte de la réalisation

1 Contexte

Chaque ville a ses propres réalisations artistiques, mais malheureusement, les citoyens ordinaires (que ce soit les touristes ou les résidents locaux) ont une compréhension très limitée de ces œuvres d'art d'une ville. En dehors de la visite de musées, il est difficile pour nous d'avoir l'occasion de connaître plus sur les cultures d'art de différentes villes.

Mais la technologie de réalité augmentée (AR) sur mobile en plein essor, combinée à une pensée novatrice, nous a offert davantage de possibilités.

La société Flagtown, jeune start-up créée en 2016 développe une solution mobile grand public basée sur des connecteurs intégrés à l'environnement urbain, afin de dynamiser le tissu culturel et économique local. Pour cela, elle propose du contenu (informations, promotions) à ses utilisateurs en fonction de leurs préférences, sans collecter et stocker de données de ses utilisateurs.

2 Objectifs

2.1 Objectif Général

L'objectif de ce Projet Recherche et Développement est de développer un prototype d'application mobile en Réalité Augmentée visant à mettre en avant des œuvres culturelles.

Cette application permet à l'utilisateur de se localiser via des dispositifs spéciaux, puis d'obtenir une gamme de bases de données de travaux du serveur.

2.2 Objectif PRD

Après une géolocalisation obtenue par un service tiers, le prototype devra obtenir une liste des points d'intérêts avoisinants (POI). Il devra ensuite permettre de visualiser en RA ces POIs. Ces points seront activés lors de la présence proche de l'utilisateur. Le contenu (objet 3D) associé à

ce POI sera affiché intégré dans l'application. Un cas d'étude sera présenté avec une mise en place dans une salle test équipée. Le travail sera mené en étroite collaboration avec l'équipe de développement de Flagtown.

3 Hypothèses

Nous préférons utiliser ARCore, qui associe plus de meilleure technologies de Réalité augmentée. Mais il est limité par le modèle de téléphone qu'il peut utiliser et par la pénurie de matériaux de référence sur ARCore jusqu'à présent, la réalisation sera plus difficile.

Si on n'arrive pas avec ARCore, nous pouvons également passer à une autre plate-forme déjà mature comme Vuforia.

On peut aussi utiliser des outils comme Viro media qui repose sur l'utilisation de ARCore pour android.

4 Bases méthodologiques

Langage de programmation

Dans ce projet on programme en JAVA et C/C++ pour le Développement Native Android.

Développement Android

On développe cette application Android en utilisant AndroidStudio de la version 3.1 ou plus.

Notion de réseaux et services web

Pour la communication avec le serveur, il faut des connaissances de réseaux comme le protocole socket.

Outil de la gestion de projet

La gestion de projet est une partie vitale du projet et joue un rôle essentiel dans la planification du projet, la mise en œuvre du projet, le moment et la qualité de la livraison finale.

- Selon les souhaits de la société, on utilisera le Github pour gérer le projet.
- Pour gérer la découpage de tâches, le Gantt et autres opérations générales, j'ai utilisé un outil "Projet.zoho".
- Pour des diagrammes, j'ai utilisé plusieurs outils comme "Gliffy Diagrams", "LucidChart" et "ProcessOn".
- Pour éditer des cahiers, j'ai utilisé Latex.

Librairie

Ce projet est basé la technologie de la Réalité augmentée, et avec la librairie ARCore.

6

Description Générale

1 Environnement du projet

Lors de la réalisation de ce projet, aucun existant n'est présent.

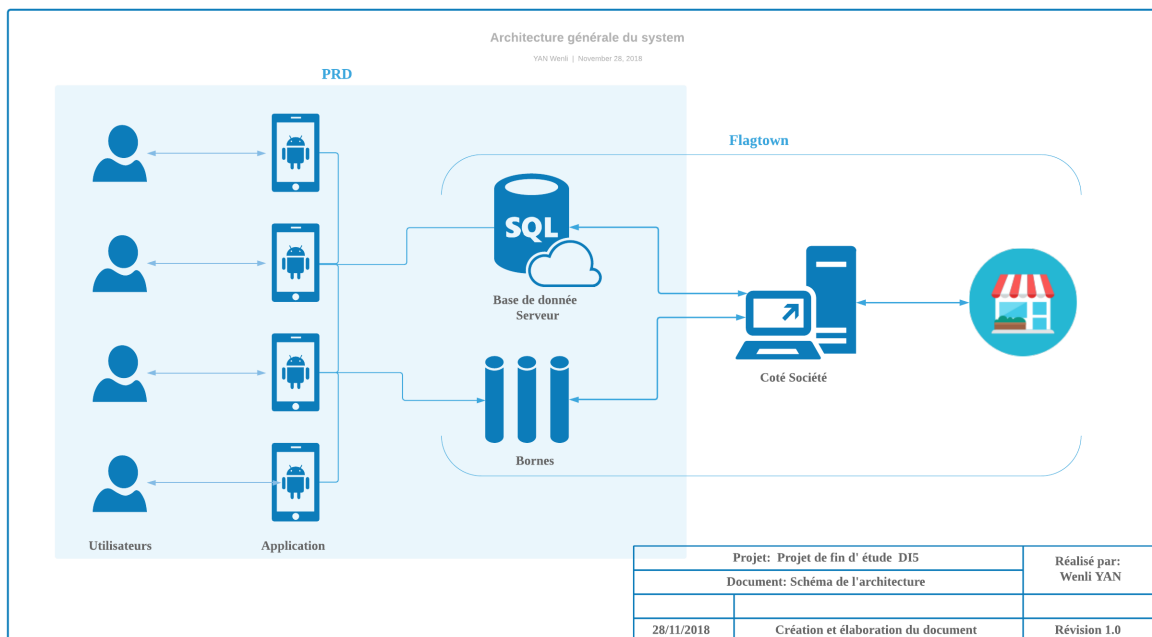


Figure 1 – Environnement du projet

Le projet fonctionne de la façon suivante, un ou plusieurs clients utiliseront leur périphérique mobile (téléphone, tablette sous Android ou en iOS) sur lequel ils peuvent se localiser pour télécharger les données correspondantes à ces œuvres dans l'environnement réel sur lequel ils veulent en savoir plus. Les utilisateurs pourront interagir avec les œuvres via l'application mobile : voir des informations supplémentaires ou jouer des jeux. Dans ce processus, notre sujet implique trois aspects suivants :

- Environnement matériel : téléphones mobiles compatibles avec ARCore.
- Environnement logiciel : utilisateurs ayant installé APP.

- Environnement réseau : fonction Bluetooth et réseau requis pour que les utilisateurs puissent localiser et communiquer avec le serveur.

La base de données de cette application est hébergée sur un serveur distant qui autorise les téléchargements pour tous les utilisateurs.

2 Caractéristiques des utilisateurs

Clients

Ce profil est destiné aux abonnés des structures utilisant notre application. Aucune authentification n'est requise pour ce type d'utilisateur.

Ils peuvent être n'importe qui, soit un touriste dans une nouvelle ville, soit un amateur qui est passionné de culture, soit un aîné ou un jeune. Ils utiliseront l'application mobile qui se doit d'être la plus intuitive et ergonomique possible, qui est assez facile à comprendre et utiliser.

- Ne nécessite pas de connaissance particulière en informatique.
- Ne doit pas posséder d'expérience de l'application.
- Utilisateur occasionnel ou régulier.
- Possède un portable qui supporte ARCore et avec cette application installée.

Informaticiens

Pour le développement et maintenance de ce projet, un informaticien qui possède des connaissances de Développement Android, réseaux, ARCore API et de communications entre services web (API) et qui connaît très bien les projets et la société Flagtown.

3 Fonctionnalités et structure générale du système

Cette application sera une terminale utilisateur et l'utilisateur mobile utilisera des bornes autour pour se localiser et obtenir les données à proximité de l'emplacement correspondant à partir du serveur. À leur tour, les utilisateurs peuvent scanner le travail (y compris les dessins 2D sur plan et les sculptures 3D) et voir les informations de superposition étroitement liées au travail depuis l'écran de leur téléphone mobile, voire même certaines interactions et jeux. Selon les besoins du client, l'application fournira également des coupons de réduction pour le magasin en guise de récompense pour le jeu à l'avenir.

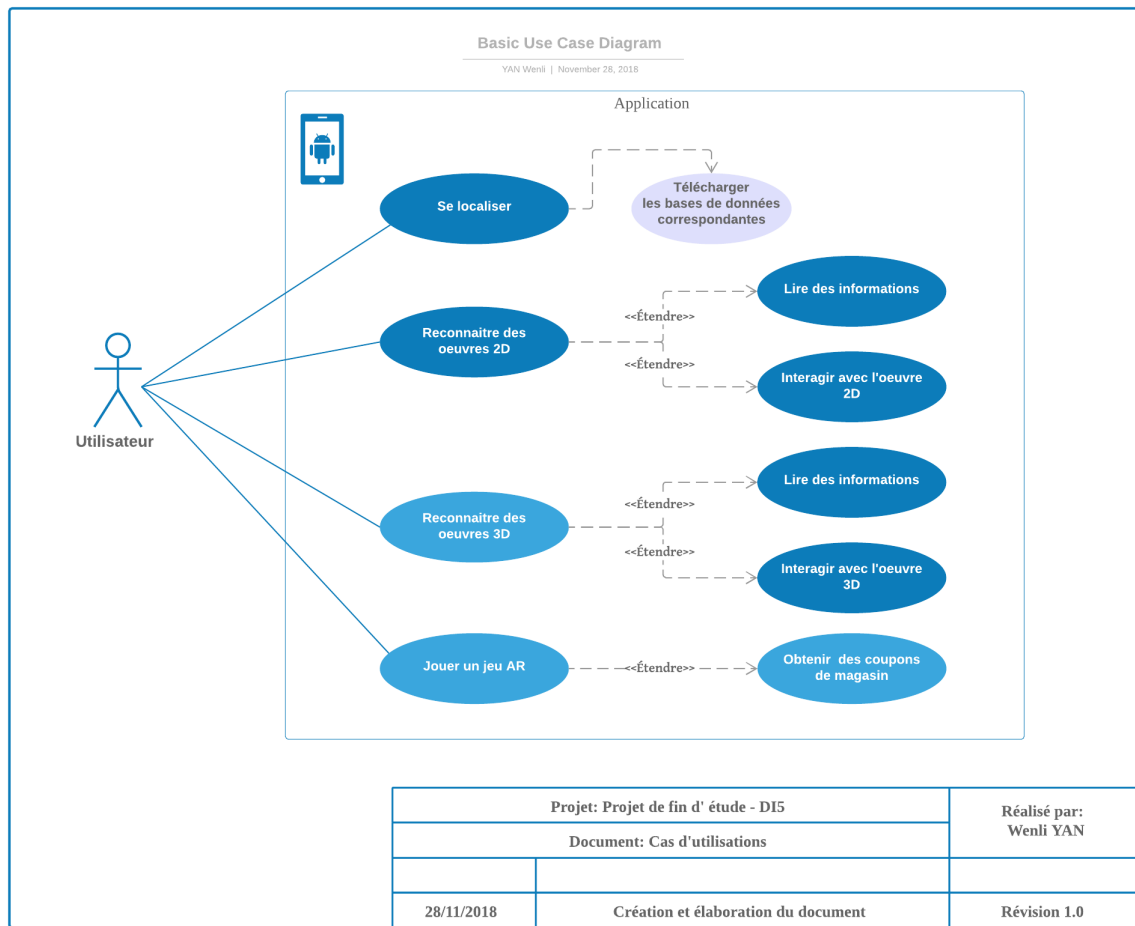


Figure 2 – Cas d'utilisation application mobile

Le cas d'utilisations est comme indiqué dans le figure ci-dessus, les fonctionnalités en bleu profonde sont nos objectives et les fonctionnalités en bleu clair sont dans le futur.

4 Architecture générale du système

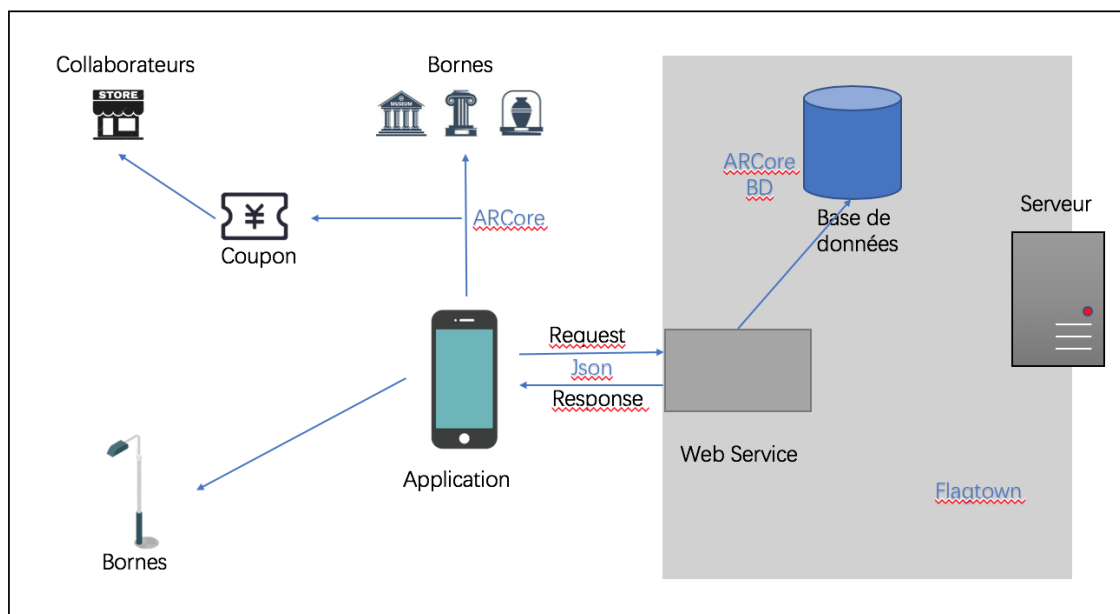


Figure 3 – Architecture

Le système complet est composé par quatre parties principales : le serveur qui gère les bases de données, l'utilisateur qui possède un portable avec cette application installée, les bornes qui aident à localiser des utilisateurs pour obtenir la géolocalisation plus précise et des partenaires d'affaires avec la société Flagtown.

Explication en détail

- Base de donnée : La base de donnée en MariaBD est stockée dans le serveur à distance. Qui contient des emplacements géographiques, des oeuvres, des informations des points d'intérêt sur des oeuvres, des informations de partenaires et leurs coupons (ou promotions). Le seul responsable de la base de données est la société Flagtown.
- Serveur : Le serveur est à Flagtown, qui donne des services à son côté par Web. Dans notre projet, on s'en fiche de la mise en œuvre de cette partie. La communication entre l'application et le serveur sera réalisée par des requêtes "Request" et "Reponse" en Json.
- Partenaires : En général, des partenaires sont des magasins qui veulent faire des promotions par notre service. Donc ils peuvent payer pour être la partenaire.
- Bornes : Les bornes sont des dispositifs conçus par la société Flagtown pour aider les utilisateurs à obtenir un positionnement précis.
- Application : Cette partie est le centre de notre sujet. L'application Android sera réalisée en JAVA avec AndroidStudio. Et il y a un point important c'est il faut lancer sur un portable qui supporte l'ARCore et il faut demander aux utilisateurs de télécharger et d'installer ARCore à partir du Google Store avant de l'utiliser. La communication avec le service web de serveur est par Json, l'interaction avec des oeuvres dans le monde réel est avec ARCore.

Quatrième partie

Analyse et conception

7

Description des interfaces externes du logiciel

1 Interfaces matériel/logiciel

Dans le cadre de ce projet il est nécessaire d'avoir des bornes utilisés pour localiser l'utilisateur, au lieu de GPS, car le GPS est parfois inexact à l'intérieur.

Des bornes sont des périphériques associés à ce projet, qui sont fabriqués par la société Flagtown.

En plus, ce projet nécessite la communication entre l'application et le serveur pour échanger des données, par exemple : transmettre la géolocalisation de l'utilisateur et télécharger les bases de données, etc.

2 Interfaces homme/machine

Lors de la réalisation de ce projet l'application mobile sera multiplateforme (Android, iOS), concernant la charte graphique de l'application mobile elle sera orientée Android (iOS pourra être envisagé dans une version future). Les interfaces humaine-machine sont comme des étapes ci-dessous :

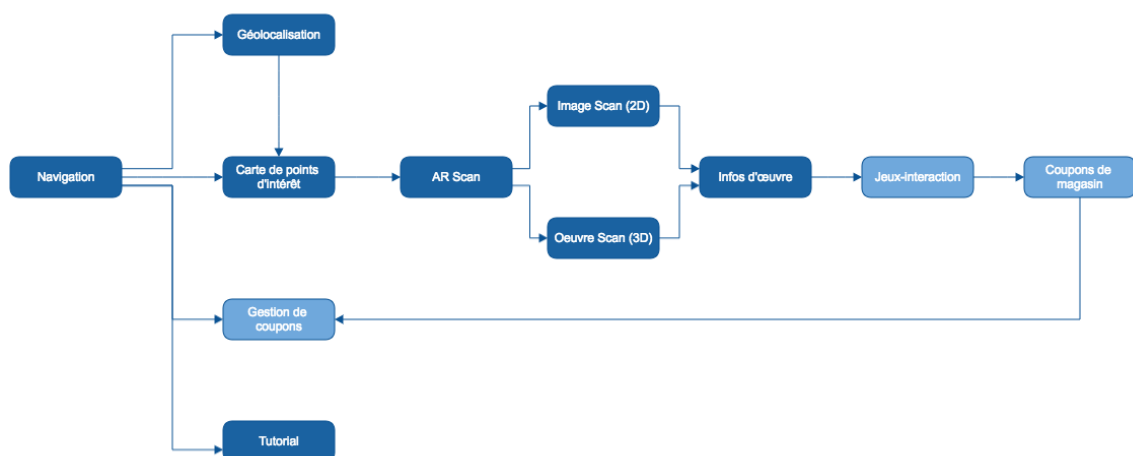


Figure 1 – Cartographie Humaine Machine

Des conceptions à préciser

2.1 Ergonomie du système :

Avertissement :

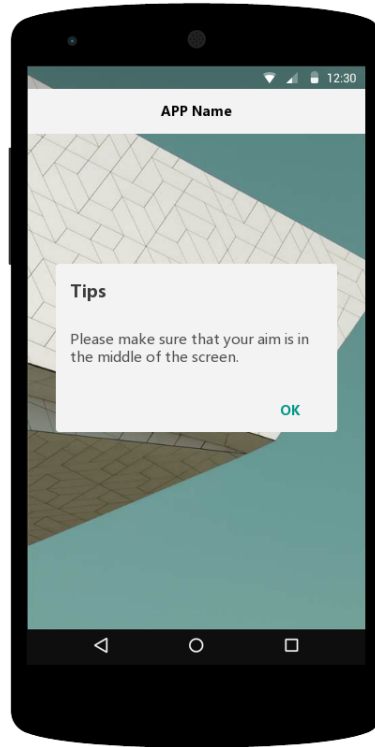


Figure 2 – Warning

Lors de l'utilisation, nous veillons à ajouter des invites et des avertissements aux utilisateurs au bon endroit pour les aider à comprendre comment utiliser et résoudre rapidement les problèmes rencontrés.

Type de navigation :

On préfère utiliser un Drawer pour la navigation (comme indiqué dans l'image ci-dessous) :

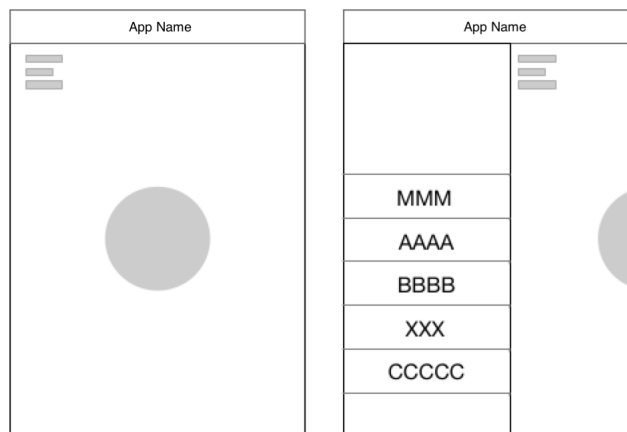


Figure 3 – Type de navigation : Drawer

Avantages de ce type :

La chose la plus importante dans les applications de réalité augmentée est l'expérience immersive. Pour cette raison nous essayons donc d'utiliser le plein écran, donc Drawer est le meilleur choix. Il peut économiser l'espace d'affichage de la page, le côté du "masqué" garantissant la mise en surbrillance du contenu de la page principale. Et une expérience immersive.

Interaction d'image :

Afin d'éviter que les informations superposées ne bouchent l'image, nous affichons un point d'intérêt sur lequel vous pouvez cliquer sur l'image et l'utilisateur peut cliquer sur le point d'intérêt pour ouvrir la page de détail qu'il souhaite connaître.

2.2 Niveau de interface :

Cette application ne comporte pas beaucoup de modules et l'interface d'interaction homme-machine n'est pas compliquée. Notre objectif est de fournir l'interaction la plus simple possible afin que les utilisateurs puissent se concentrer sur la fonction de réalité augmentée pour obtenir une meilleure expérience plutôt que de passer beaucoup de temps dessus. Donc une interface simple, une réponse rapide et une interaction simple sont donc au centre de la conception.

3 Interfaces logiciel/logiciel

Cette application collecte que les données de la géolocalisation d'utilisateur mais pas des données personnelles.

La transmission des données entre le périphérique mobile d'un client et le serveur est effectuée à travers le réseau internet. Les données sont insérées dans une base de données MariaDB qui est un « fork » du SGBD MySQL. Afin que le serveur contrôlé par Flagtown puisse contrôler les base de données pour des opération comme : insertion, modification, suppression ou consultation. Et l'utilisateur puisse faire un requête avec sa propre position à distance et puis il peut télécharger la base de donnée correspondante. Les données retournées par le service seront au format JSON.

4 Activité d'utilisateur

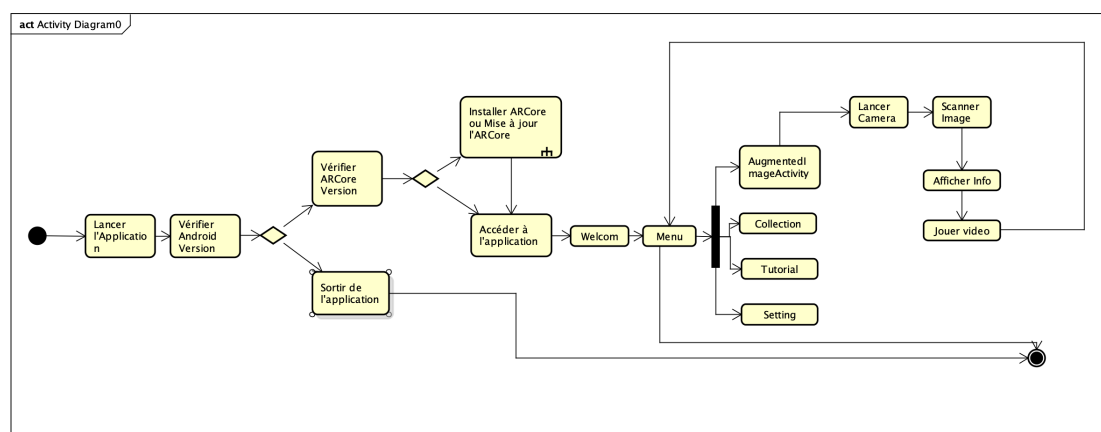


Figure 4 – Activité d'utilisateur

Une fois que l'utilisateur a installé l'application, lancez-la.

Tout d'abord, vérifiez si la version Android est bonne. Si ce n'est pas le cas, quittez l'application et demandez à l'utilisateur que "la version n'est pas prise en charge"; Si possible, effectuez une détection de version ARCore. Si la version est trop basse ou si ARCore n'est pas installé, l'utilisateur est invité à se mettre à jour ou à s'installer. Si la version est disponible, il entre dans l'interface principale.

L'interface principale passe au menu principal après 3 secondes, où le premier élément du menu est notre activité principale, cliquez pour entrer dans la reconnaissance d'image.

Maintenant, l'utilisateur peut scanner l'image et faire l'augmentation sur l'image comme superposer des informations, des images, la vidéo etc. Après, il peut retourner au menu principal.

8

Diagramme de classes

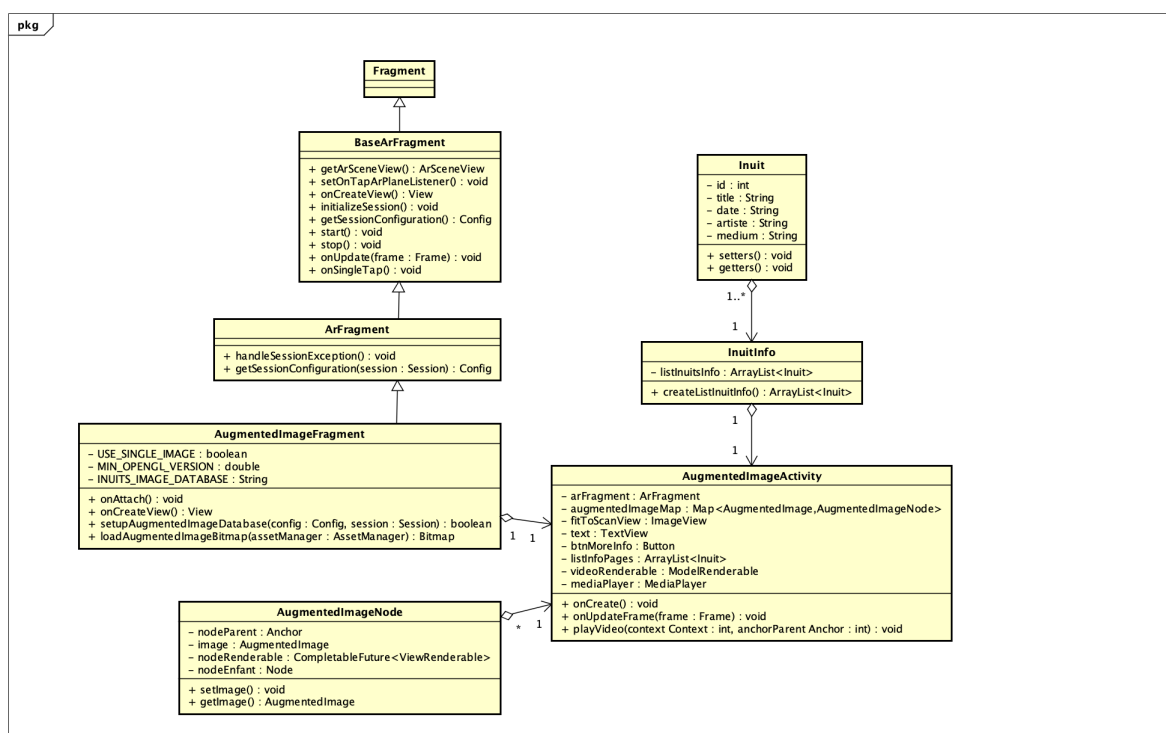


Figure 1 – Diagramme de classe

9

Conception de base de donnée

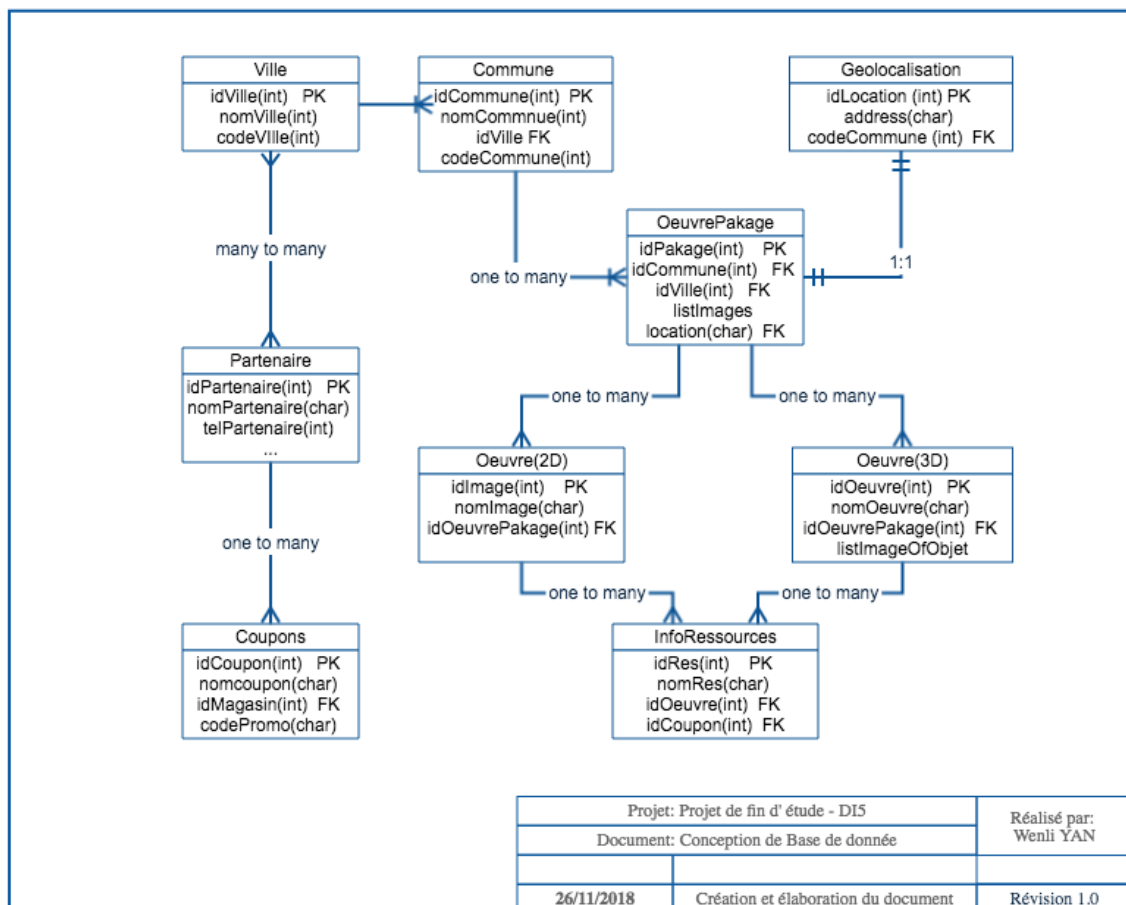


Figure 1 – Conception de BD

La conception de la base de données de l'ensemble du projet est une tâche plus complexe impliquant la localisation géographique, les travaux et les commerçants. On pense que la chose la plus importante dans la conception est que la taille de la base de données devrait être appropriée, car chaque base de données d'images ARCore peut stocker jusqu'à 1 000 images. Il est donc crucial de diviser et de regrouper les images de manière raisonnable.

L'idée de départ est de diviser le plus grand nombre des utilisateurs, à savoir : ville, combinaison,

ensemble de données de travail, base de données d'images 2D, base de données de travail 3D, bibliothèque d'informations supplémentaire, coupon et collaborateurs. Parmi ceux-ci, la partie de PRD est principalement constituée de la base de données d'images, et de la bibliothèque d'informations à superposer, notre objectif étant de pouvoir télécharger la base de données correspondante correctement et rapidement.

Cinquième partie

Mise en oeuvre

10

Outils et librairies utilisées

Dans ce projet, nous avons utilisé les dernières versions pour toutes les composantes, voici une liste de tout ce que nous avons implémenté dans ce projet :

- Java 8
- AndroidStudio 3.3.0
- Sdk 27
- ARCore (Sceneforme) 1.5.0
- JUnit4
- Espresso : 3.0.2
- Robolectric : 3.8

Plus de détails, veuillez voir la Pièce jointe : Doc d'installation et déploiement.

11

Déploiement et implémentation

Dans cette partie, je vais vous présenter le déploiement du projet et la réalisation des fonctionnalités, je vais choisir les points plus importants à présenter, pour plus de détails, vous pouvez télécharger mon projet sur git et lire mes codes.

Lien Github : <https://github.com/ywlqjl/PRD2.git>

1 Fonctionnalités et déviation

Le but de ce projet est de réaliser l'image augmentée et puis supposer des informations correspondantes dans l'écran de l'utilisateur. Puisqu'il s'agit d'une nouvelle technologie, toutes les parties de ce projet progressent dans l'exploration. Donc, il produit plusieurs versions de l'itération.

1.1 Préparation du projet

1.1.1 Choix de images

Pour le projet AR, la qualité d'image est importante. Il faut conformer des règles (comme indiqué dans l'article précédent) : format, taille et l'évaluation de chaque image.

Donc j'ai effectué un échantillonnage d'œuvres d'artistes canadiens sur le thème des Inuits depuis l'article fourni par l'enseignant :



PRD_Oeuvres_parours_casino_MTL_libres_de_droit_inuit.pdf

Figure 1 – Article : Collection Loto-Québec

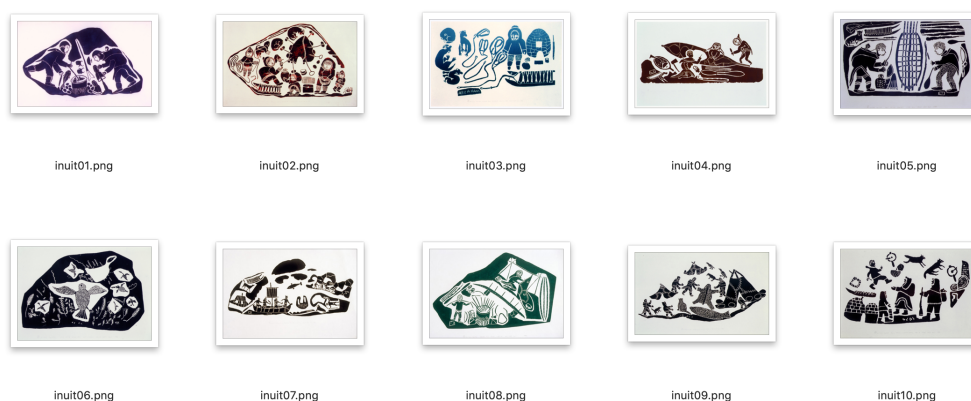


Figure 2 – Échantillonnage d'œuvres inuits

Ensuite, j'ai fait l'évaluation de qualité des images avec arcoring :

```
YanwenlideMBP-3:macos yanwenli$ ./arcoring eval-img --input_image_path=./images/inuit01.png
Unsupported image format.
YanwenlideMBP-3:macos yanwenli$ ./arcoring eval-img --input_image_path=./images/inuit01.png
100
YanwenlideMBP-3:macos yanwenli$ ./arcoring eval-img --input_image_path=./images/inuit02.png
65
YanwenlideMBP-3:macos yanwenli$ ./arcoring eval-img --input_image_path=./images/inuit03.png
100
YanwenlideMBP-3:macos yanwenli$ ./arcoring eval-img --input_image_path=./images/inuit04.png
65
YanwenlideMBP-3:macos yanwenli$ ./arcoring eval-img --input_image_path=./images/inuit05.png
100
YanwenlideMBP-3:macos yanwenli$ ./arcoring eval-img --input_image_path=./images/inuit06.png
100
YanwenlideMBP-3:macos yanwenli$ ./arcoring eval-img --input_image_path=./images/inuit07.png
60
YanwenlideMBP-3:macos yanwenli$ ./arcoring eval-img --input_image_path=./images/inuit08.png
100
YanwenlideMBP-3:macos yanwenli$ ./arcoring eval-img --input_image_path=./images/inuit09.png
80
YanwenlideMBP-3:macos yanwenli$ ./arcoring eval-img --input_image_path=./images/inuit10.png
75
```

Figure 3 – Évaluation de qualité des images inuits

Les scores de qualité des 10 images inuits varient de 60 à 100, ce qui signifie que la plupart d'entre eux sont de bonne qualité pour ce projet.

1.1.2 Configuration de projet

Créer d'abord un nouveau projet, et puis modifier la configuration.

Cette configuration contient beaucoup de parties : On parle justement la configuration d'ARCore, mais il y a aussi la configuration de test, version de Android, et d'autres dépendances. *Voir Doc instructions de programmation*

Dans le fichier **build.gradle(Module :app)** on ajoute des librairies d'ARCore :

```
// ARCore library
implementation 'com.google.ar:core:1.5.0'
implementation 'com.google.ar.sceneform:core:1.5.0'
implementation 'com.google.ar.sceneform.ux:sceneform-ux:1.5.0'
```

Figure 4 – ARCore librairies

1.1.3 Configuration de portable

Comme indiqué dans le chapitre précédent, il faut un portable qui support ARCore. Et assurez-vous l'installation d'ARCore dans ce portable, sinon il faut accéder à Google App Store pour télécharger la dernière version de ARCore. *Voir Doc installation et instructions d'utilisation*

1.1.4 Création de base de donnée

Pour ce projet, avec ARCore, on a deux façons de créer la base de données :

1. Charger une base de données d'images augmentée préexistante créée par l'outil `arcoring`

Il faut créer d'abord une base de donnée qui contient toutes les images qu'on a besoin :

Avec le chemin absolu:

```
YanwenliMacBook-Pro-3:macos yanwenli$ ./arcoring build-db --input_images_directory=
/Users/yanwenli/Documents/informatique/PRD/arcoring-android-sdk-1.2.0/tools/arcoring/macos/images --output_db_path=myimages.imgdb
```

Avec le chemin relatif:

```
YanwenliMacBook-Pro-3:macos yanwenli$ ./arcoring build-db --input_images_directory=
./images --output_db_path=myimages2.imgdb
```

Attention:

Assurez-vous d'écrire le chemin relatif correctement.

Voici le dossier d'images (input) et la base de donnée bien générée:

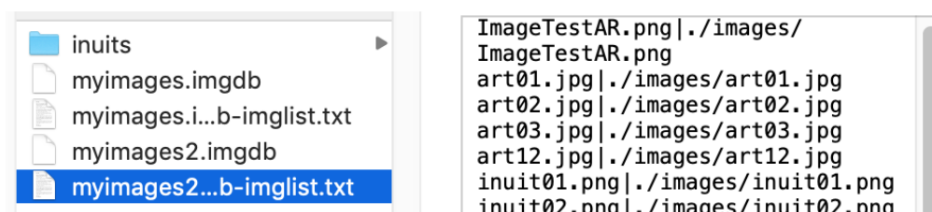


Figure 5 – Création de base de donnée de multi-images

Ensuite, implémenter cette base de donnée dans `ArFragment` de ce projet :

```
// This is a pre-created database containing the sample image.
private static final String INUITS_IMAGE_DATABASE = "inuits.imgdb";
```

Figure 6 – Implémentation dans projet

```

if (USE_SINGLE_IMAGE) {
    Bitmap augmentedImageBitmap = loadAugmentedImageBitmap(assetManager);
    if (augmentedImageBitmap == null) {
        return false;
    }

    augmentedImageDatabase = new AugmentedImageDatabase(session);
    augmentedImageDatabase.addImage(IMAGE_NAME, augmentedImageBitmap);
    // If the physical size of the image is known, you can instead use:
    // augmentedImageDatabase.addImage("image_name", augmentedImageBitmap, widthInMeters);
    // This will improve the initial detection speed. ARCore will still actively estimate the
    // physical size of the image as it is viewed from multiple viewpoints.
}

```

Figure 7 – Charger la base de donnée préexistante

2. Créer la base de donnée pendant l'exécution

```

} else {
    // This is an alternative way to initialize an AugmentedImageDatabase instance,
    // load a pre-existing augmented image database.
    try (InputStream is = getContext().getAssets().open(INUITS_IMAGE_DATABASE)) {
        augmentedImageDatabase = AugmentedImageDatabase.deserialize(session, is);
    } catch (IOException e) {
        Log.e(TAG, "msg: \"IO exception loading augmented image database.\", e);
        return false;
    }
}

```

Figure 8 – Créer la base de donnée pendant l'exécution

1.2 Version 1

Version 1 réalise d'abord le point le plus important : image augmentée. Dans la première version, nous avons implémenté la reconnaissance et l'amélioration d'une image unique.

Il y a trois classes importantes pour un arcore projet :

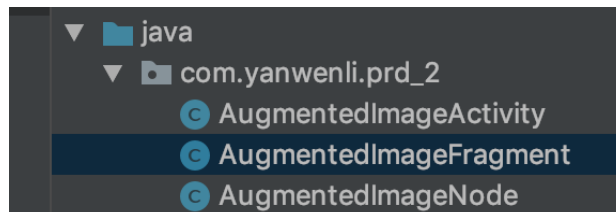


Figure 9 – Classes importantes

AugmentedImageNode

AugmentedImageNode définit des attributs, la position, des actions d'un point et des relations entre des points.

```

private static final String TAG = "AugmentedImageNode"; //Node label
private AugmentedImage image; //image to augment

//Three interest points
private static CompletableFuture<ViewRenderable> point;
private static CompletableFuture<ViewRenderable> point_left;
private static CompletableFuture<ViewRenderable> point_right;
private Node cornerNode;
private Node cornerNode_left;
private Node cornerNode_right;
public Context nodeContext;

private Anchor anchorParent;

```

Figure 10 – Définition des noeuds d'intérêts

Ici nous définissons des noeuds qu'on veut ajouter dans le scène. Tous les noeuds enfants sont liés au noeud parent qui est le centre de l'image.

```
/**
 * Build renderable images with layout.
 * @param context
 */
@RequiresApi(api = Build.VERSION_CODES.N)
public AugmentedImageNode(Context context) {
    this.nodeContext = context;
    if (point == null) {
        point = ViewRenderable.builder()
            .setView(context, R.layout.video_play_layout_renderable)
            .build();
        point_left = ViewRenderable.builder()
            .setView(context, R.layout.intrest_point_layout_renderable)
            .build();
        point_right = ViewRenderable.builder()
            .setView(context, R.layout.intrest_point_layout_renderable2)
            .build();
    }
}
```

Figure 11 – Lier les layouts rendreables pour les trois points d'intérêt

Pour chaque noeud, nous pouvons définir son interface et action. En général, c'est un objet 3D rendreable, mais pour une image 2D c'est pas possible. Mais ici j'ai utilisé une bonne façon spéciale d'ARCore : créer un layout rendreable pour supposer une image 2D dans le scène réel.

Comme nous le savons, dans Android, nous pouvons créer une variété de fichiers layout.xml de mise en forme dont nous avons besoin pour dessiner l'interface, ce qui est très utile. Et ARCore a conçu un tel moyen de réaliser un rendu d'image 2D pour Android, ce qui a parfaitement résolu notre problème.

```
/**
 * Load the ViewRenderable points.
 * Define their local positions, angle and the world position.
 * Bind the points with nodes
 * Connect to the parent node.
 * @param image
 */
public void setImage(AugmentedImage image) {
    this.image = image;

    if (!point.isDone() || !point_left.isDone() || !point_right.isDone()) {
        CompletableFuture.allOf(point, point_left, point_right)
            .thenAccept((Void aVoid) -> setImage(image))
            .exceptionally(
                throwable -> {
                    Log.e(TAG, msg: "Exception loading", throwable);
                    return null;
                }
            );
    }

    // Set the anchor based on the center of the image.
    anchorParent = image.createAnchor(image.getCenterPose());
    setAnchor(anchorParent);

    // Create the local position Vector for every point
    Vector3 localPosition = new Vector3();
    Vector3 localPosition2 = new Vector3();
    Vector3 localPosition3 = new Vector3();

    // Centre position: localPosition.set(-0.0f, 0.0f, 0.0f);
    localPosition.set(v: -0.0f * image.getExtentX(), v1: -0.0f, v2: 0.0f * image.getExtentZ());
    localPosition2.set(v: -0.5f * image.getExtentX(), v1: -0.0f, v2: 0.7f * image.getExtentZ());
    localPosition3.set(v: 0.5f * image.getExtentX(), v1: -0.0f, v2: -0.2f * image.getExtentZ());

    // Create world position with quaternion for changing local position to world position
    Quaternion quaternion = new Quaternion(v: 0.0f, v1: 0.0f * image.getExtentX(), v2: 0.0f, v3: 0.0f * image.getExtentZ());
}
```

Figure 12 – Chargement et configuration des points

Dans cette méthode, il faut d'abord finir le rendu des layouts des noeuds. Après définir la position et angle pour chaque noeud, ensuite, lier tous les noeuds enfants au noeud parent.

AugmentedImageFragment

AugmentedImageFragment crée la scène avec les coordonnées mondiales et locales, implémenter la base de donnée.

```
private static final String TAG = "AugmentedImageFragment";
private static final String IMAGE_NAME = "inuit01.png";

// This is a pre-created database containing the sample image.
private static final String INUITS_IMAGE_DATABASE = "inuits.imgdb";

// Augmented image configuration and rendering.
private static final boolean USE_SINGLE_IMAGE = false;
private static final double MIN_OPENGL_VERSION = 3.0;
```

Figure 13 – Variables nécessaires

Ici, j'utilise une variable "USE SINGLE IMAGE" pour consigner le mode actuel qu'on est en train d'utiliser.

```
/**
 * Check versions and devices supports
 *
 * @param context
 */
@Override
public void onAttach(Context context) {
    super.onAttach(context);
    // Check for Sceneform being supported on this device. This check will be integrated into
    // Sceneform eventually.
    if (Build.VERSION.SDK_INT < Build.VERSION_CODES.N) {
        Log.e(TAG, msg: "Sceneform requires Android N or later");
        Toast.makeText(context, "Sceneform requires Android N or later", Toast.LENGTH_SHORT).show();
    }
    String openGLVersionString = ((ActivityManager) context.getSystemService(Context.ACTIVITY_SERVICE)).getDeviceConfigurationInfo().getOpenGlVersionString();

    if (Double.parseDouble(openGLVersionString) < MIN_OPENGL_VERSION) {
        Log.e(TAG, msg: "Sceneform requires OpenGL ES 3.0 or later");
        Toast.makeText(context, "Sceneform requires OpenGL ES 3.0 or later", Toast.LENGTH_SHORT).show();
    }
}
```

Figure 14 – Version check

Ce projet nécessite une version d'OPENGL dans Sceneform d'ARCore au moins 3.0. Et une version d'Android SDK au moins 24. Il faut vérifier tous les deux chaque fois quand l'utilisateur lance cette application.

```
/**
 * Create view in the scene. We don't use Plane for 3D so we close the plane detection when we create the view.
 *
 * @param inflater
 * @param container
 * @param savedInstanceState
 * @return view that we created
 */
@Override
public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
    View view = super.onCreateView(inflater, container, savedInstanceState);

    // Turn off the plane discovery since we're only looking for images
    getPlaneDiscoveryController().hide();
    getPlaneDiscoveryController().setInstructionView(null);
    getArSceneView().getPlaneRenderer().setEnabled(false);
    return view;
}
```

Figure 15 – Créer AR scène

ARCore fournit une fonctionnalité *Détection de plan*, pour notre projet, nous n'avons pas besoin de cette partie donc nous devons l'annuler en avance.

```

/**
 * Load and set configuration for our database.
 *
 * @param config
 * @param session
 * @return
 */
private boolean setupAugmentedImageDatabase(Config config, Session session) {
    AugmentedImageDatabase augmentedImageDatabase;

    AssetManager assetManager = getContext() != null ? getContext().getAssets() : null;
    if (assetManager == null) {
        Log.e(TAG, msg: "Context is null, cannot initialize image database.");
        Toast.makeText(getContext(), text: "Context is null, cannot initialize image database.", Toast.LENGTH_SHORT).show();
        return false;
    }

    if (USE_SINGLE_IMAGE) {
        Bitmap augmentedImageBitmap = loadAugmentedImageBitmap(assetManager);
        if (augmentedImageBitmap == null) {
            return false;
        }

        augmentedImageDatabase = new AugmentedImageDatabase(session);
        augmentedImageDatabase.addImage(IMAGE_NAME, augmentedImageBitmap);
        // If the physical size of the image is known, you can instead use:
        // augmentedImageDatabase.addImage("image_name", augmentedImageBitmap, widthInMeters);
        // This will improve the initial detection speed. ARCore will still actively estimate the
        // physical size of the image as it is viewed from multiple viewpoints.
    } else {
        // This is an alternative way to initialize an AugmentedImageDatabase instance,
        // load a pre-existing augmented image database.
        try (InputStream is = getContext().getAssets().open(INUITS_IMAGE_DATABASE)) {
            augmentedImageDatabase = AugmentedImageDatabase.deserialize(session, is);
        } catch (IOException e) {
            Log.e(TAG, msg: "IO exception loading augmented image database.", e);
            return false;
        }
    }
}

```

Figure 16 – Charger la base de donnée

Il faut d'abord obtenir le contexte et après charger la bonne base de donnée : Single image ou Multi image, et avec le bon nom pour le mode Multi image.

AugmentedImageActivity

AugmentedImageActivity initialise l'interface principale, le processus de reconnaissance d'image et les actions supposées qu'on veut faire.

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_augmented_image2);

    arFragment = (ArFragment) getSupportFragmentManager().findFragmentById(R.id.ux_fragment);
    arFragment.getArSceneView().getScene().addOnUpdateListener(this::onUpdateFrame);
    fitToScanView = findViewById(R.id.image_view_fit_to_scan2);
    fitToScanView.setVisibility(View.VISIBLE);
    introduction_layout = findViewById(R.id.image_intro_layout2);
    introduction_layout.setVisibility(View.GONE);

    txtTitle = findViewById(R.id.txt_title_image);
    txtArtist = findViewById(R.id.txt_artist_image);
    txtDate = findViewById(R.id.txt_date_image);
    txtMedium = findViewById(R.id.txt_medium_image);

    btnMoreInfo = findViewById(R.id.btn_more_info);

    LoadInuitInfo loadInuitInfo = new LoadInuitInfo();
    listInuitPages = loadInuitInfo.createListInuitInfo();

    btnMoreInfo.setOnClickListener(v -> {
        Intent intent = new Intent( packageContext: AugmentedImageActivity.this, MoreInfoActivity.class);
        startActivity(intent);
    });
}

```

Figure 17 – Initialisation de l'interface

Dans la méthode `onCreate()`, c'est l'initialisation des variables avec quelques listeners des composants. Tout sur la réalité augmentée, définie dans le fragment : `arFragment`.

```

/**
 * Processus de reconnaissance d'image et liaison de noeud selon besoin
 * @param frameTime
 */
private void onUpdateFrame(FrameTime frameTime) {
    Frame frame = arFragment.getArSceneView().getArFrame();

    // If there is no frame or ARCore is not tracking yet, just return.
    if (frame == null || frame.getCamera().getTrackingState() != TrackingState.TRACKING) {
        return;
    }

    Collection<AugmentedImage> updatedAugmentedImages = frame.getUpdatedTrackables(AugmentedImage.class);

    for (AugmentedImage augmentedImage : updatedAugmentedImages) {
        switch (augmentedImage.getTrackingState()) {
            case PAUSED:
                // When an image is in PAUSED state, but the camera is not PAUSED, it has been detected,
                // but not yet tracked.
                int index_of_current_image = augmentedImage.getIndex();
                String text = "Detected Image" + index_of_current_image;
                txtInfo.setText(listInfoPages.get(index_of_current_image).getInfo());
                txtTitle.setText(listInfoPages.get(index_of_current_image).getTitle());
                txtArtist.setText(listInfoPages.get(index_of_current_image).getArtist());
                txtDate.setText(listInfoPages.get(index_of_current_image).getDate());
                txtMedium.setText(listInfoPages.get(index_of_current_image).getMedium());

                Toast.makeText(context: this, text, Toast.LENGTH_SHORT).show();

                break;

            case TRACKING:
                // Create a new anchor for newly found images.
                if (!augmentedImageMap.containsKey(augmentedImage)) {
                    //augmentedImageMap.clear();
                }
        }
    }
}

```

Figure 18 – le processus image tracking

J'ai fait une capture pour une partie de code de cette méthode. Cette méthode est très importante : en cas de "TRACKING", nous pouvons créer des noeuds avec la cordonnée mondiale ou locale et ajouter des interactions pour des noeuds ; en cas de "PAUSED", cela signifie que l'image est reconnue, donc nous pouvons ajouter des interactions pour l'image reconnue, par exemple, ici nous allons afficher des informations correspondantes comme auteur, date etc.

Résultat

Premier but est arrivé : augmentation d'une seule image, et supposer un bouton cliquable pour lancer une nouvelle page d'information. En plus, j'ai ajouté un cadre transparent pour rappeler à l'utilisateur de capturer l'image à la bonne distance. Lorsque l'image est détectée, le cadre disparaîtra.

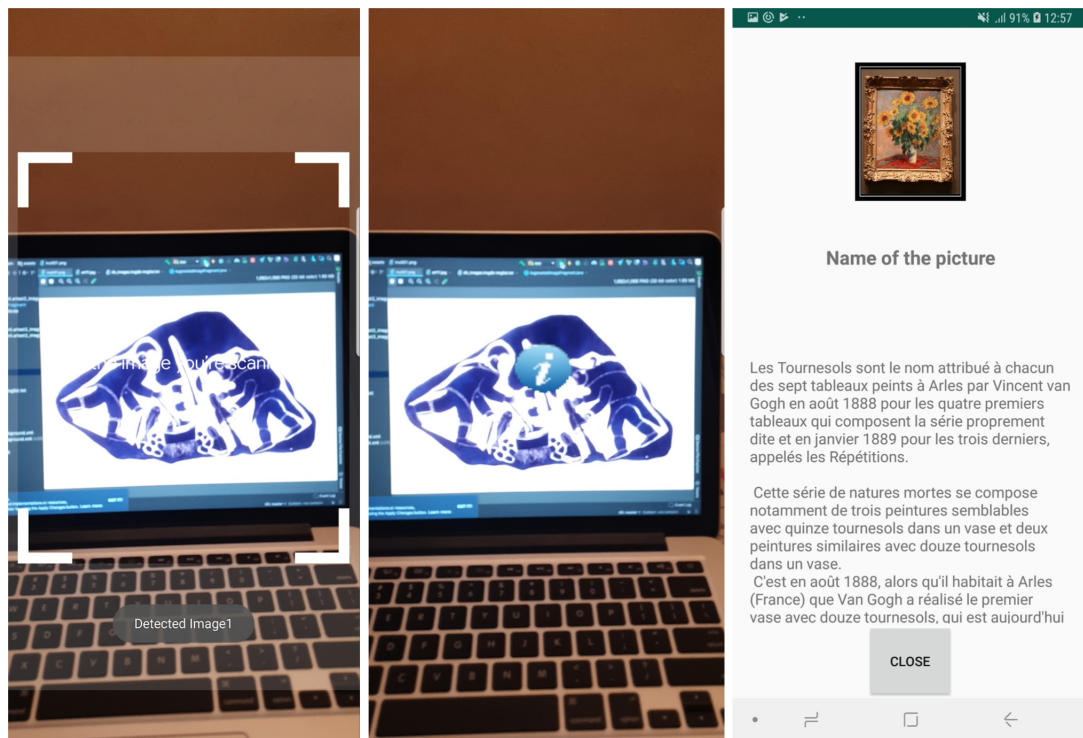


Figure 19 – Résultat version 1 : single-image augmentée

Après, j'ai fait une image de masque en orange selon la forme de la personne et j'ai remplacé le bouton d'info par cette forme pour mieux indiquer ce que nous voulons vous présenter.

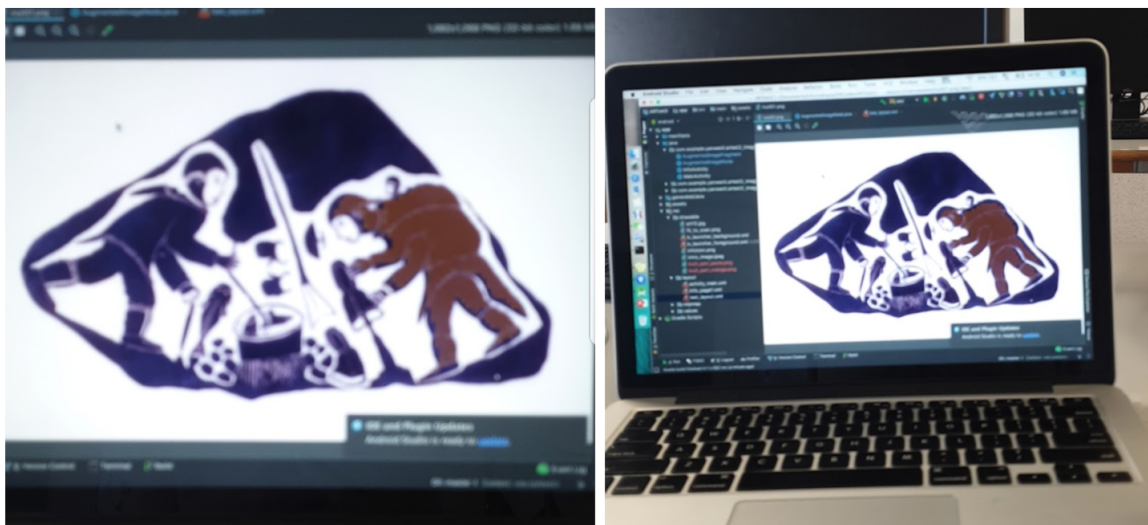


Figure 20 – Version 1 avec une forme personnage

1.3 Version 2

Dans la version 2, nous avons étendu l'augmentation à plusieurs images. Plusieurs images peuvent être reconnues et augmentées en même temps.

Premièrement, j'ai créé une base de données de plusieurs images en utilisant les méthodes mentionnées précédemment et l'ai chargé dans ce projet. Ensuite, j'ai modifié des paramètres comme suivant :


```
// This is a pre-created database containing the sample image.
private static final String INUITS_IMAGE_DATABASE = "inuits.imgdb";

// Augmented image configuration and rendering.
private static final boolean USE_SINGLE_IMAGE = false;
```

Figure 21 – Mode pour base de donnée

Si on utilise single-image, il faut mettre cette valeur comme True, sinon False.

Résultats

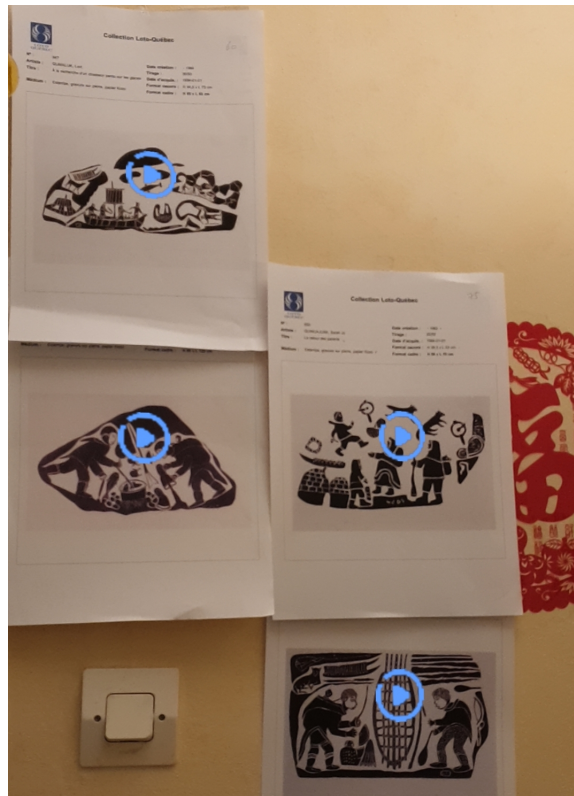


Figure 22 – Plusieurs images sont reconnues en même temps

1.4 Version 3

Dans la version 3 (version actuelle), j'ai ajouté le menu du tiroir, l'interface de bienvenue, et directement superposé l'introduction d'image dans l'interface de reconnaissance AR, et le bouton "Plus d'informations"; dans le même temps, j'ai ajouté un bouton cliquable pour lire une vidéo correspondante dans la scène réelle, créant une interaction entre virtuel et réel.

Résultats

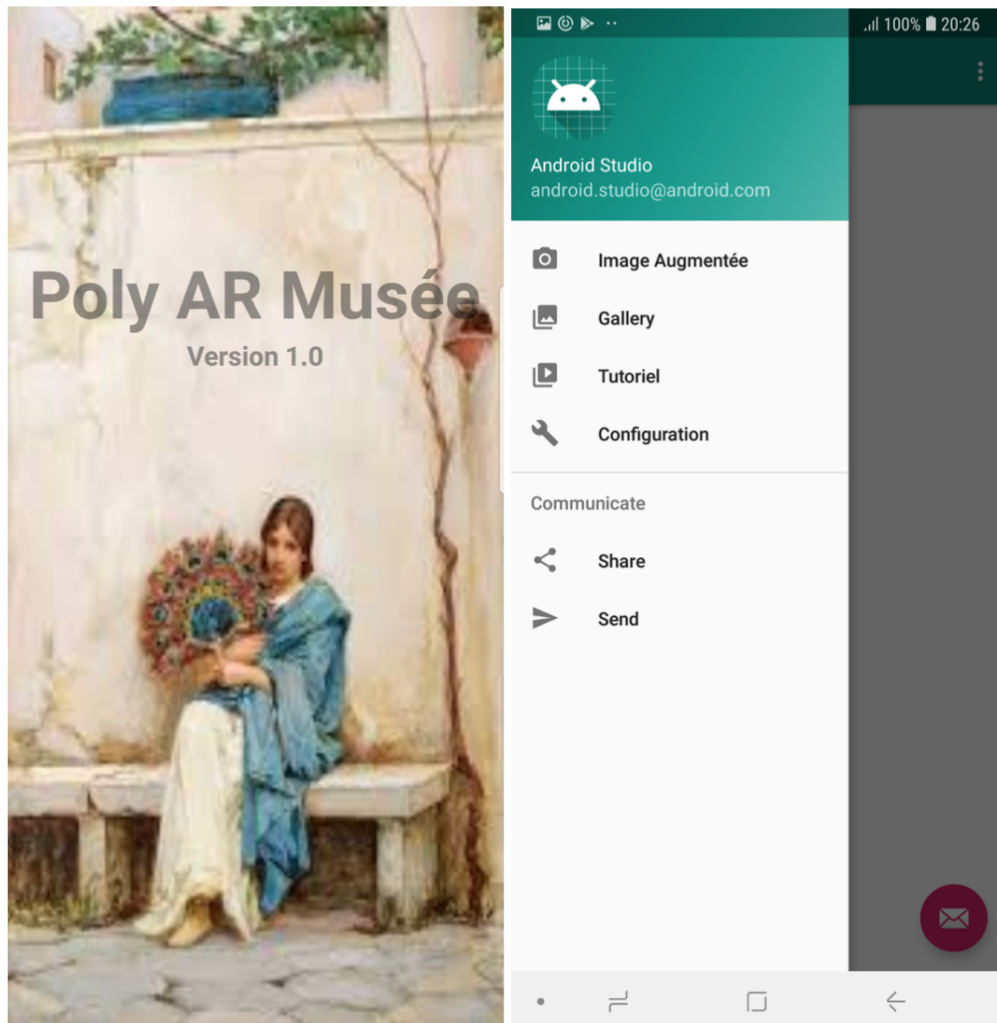


Figure 23 – Activité Bienvenue et Menu drawer

L'interface de bienvenue permet d'informer intuitivement l'utilisateur le but de la conception de l'application, y compris du nom et des informations de version de celle-ci. L'interface de bienvenue passe automatiquement à l'interface du menu après 3 secondes.

L'utilisation de menu du tiroir et l'enlèvement des barres de navigation de la tête et du bas est pour but d'offrir une meilleure immersion aux applications de réalité augmentée. Les utilisateurs peuvent avoir une interface plus grande et plus complète pour interagir avec la réalité.

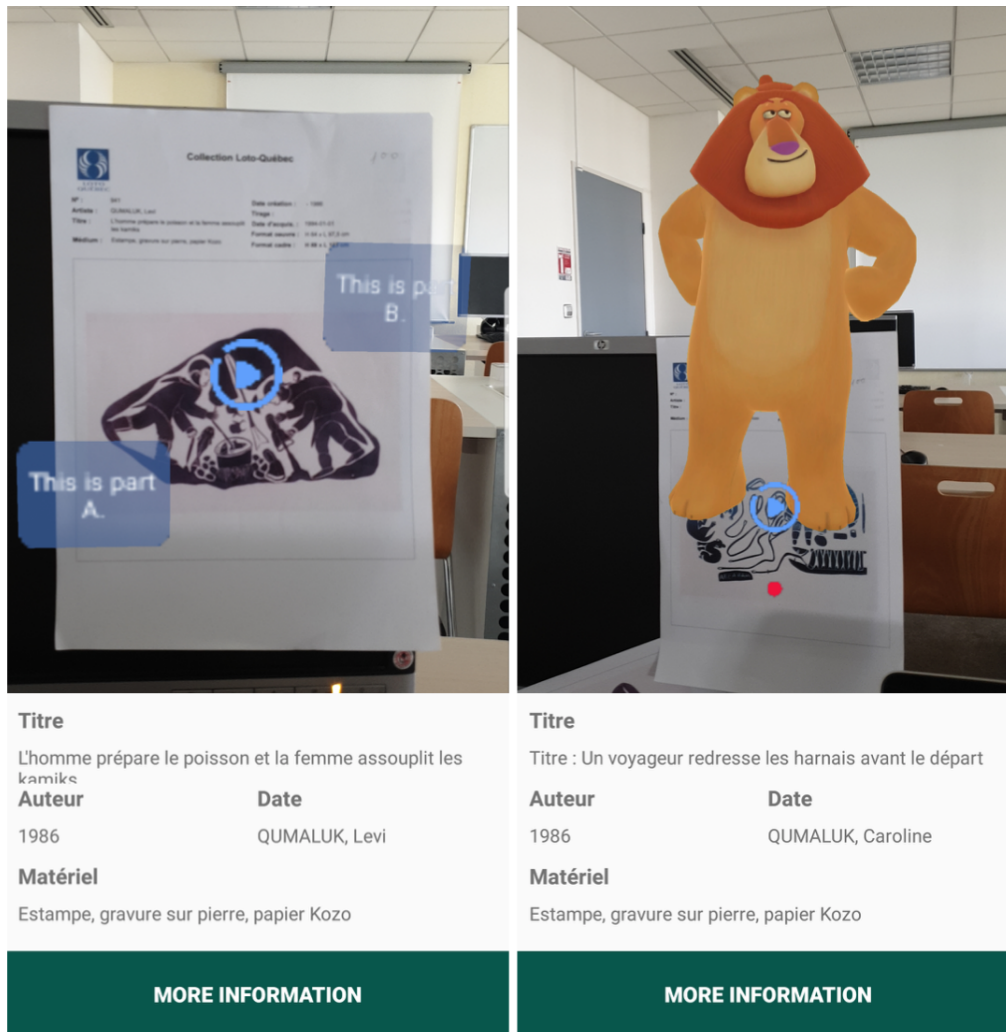


Figure 24 – Superposition d'informations des points d'intérêt et Interaction : introduction animation video

Afin d'afficher les informations pertinentes de l'image plus intuitivement, nous avons supprimé le clic sur le bouton pour accéder à la page d'informations, mais nous avons ajouté directement le bouton "More Information" et le tableau des informations en bas qui contient le titre, l'auteur, la data de création et le matériel d'oeuvre.

Ce qui est plus intéressant, c'est que nous avons pleinement exploité les avantages d'Arcore et que nous avons réalisé la possibilité de superposer une vidéo jouable de manière dynamique sur l'image dans le scène réel. Avec cette vidéo, nous pouvons nous aider à mieux afficher une «peinture» dynamique pour offrir une meilleure expérience au public.

1.5 Version 4

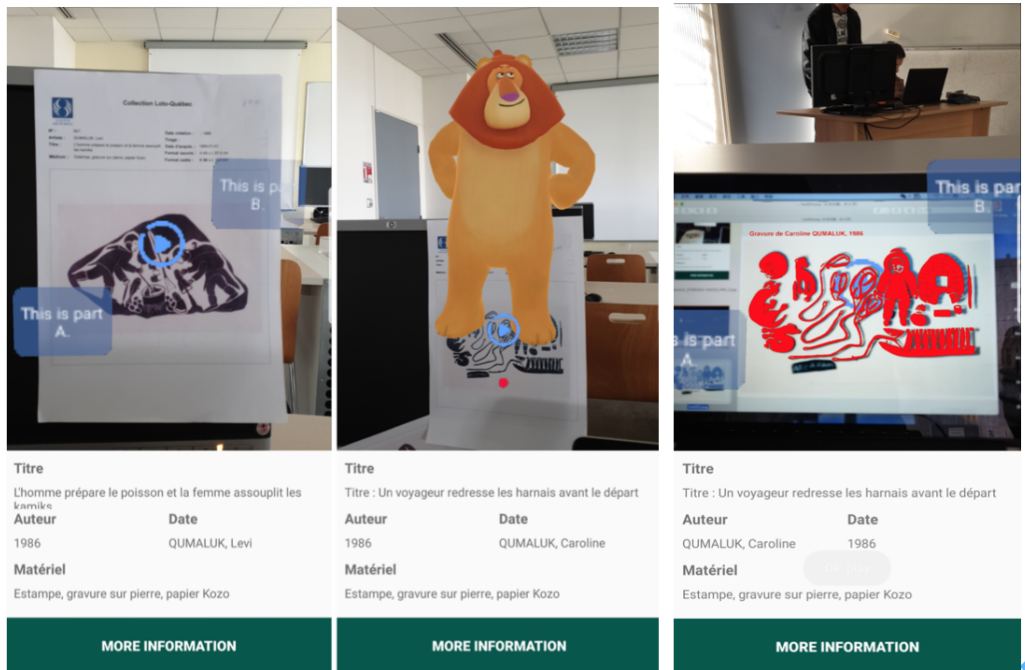


Figure 25 – Resultat final avec quelques améliorations

Amélioration sur le système de coordonnées

Dans la version 4, j'ai fixé le système de coordonnées : Avant : Rotation world

Problème : se limite à détecter des images dans un plan vertical et les images superposées de toutes les images sont orientées dans la même direction.

Résolution : Utilisez la rotation locale à la place.

```
cornerNode = new Node();
cornerNode.setParent(this);
cornerNode.setLocalPosition(localPosition);
cornerNode.setLocalRotation(Quaternion.local);
// cornerNode.setWorldRotation(Quaternion);
cornerNode.setRenderable(point.getNow( valueIfAbsent: null));
```

Figure 26 – Changement de système de coordonnées

Contrôle de vidéo : La vidéo disparaît quand terminée

J'ai ajouté le CompletionListener pour faire disparaître la vidéo une fois la lecture terminée.

```
private Boolean isFinished = true;
```

Figure 27 – Variable pour contrôler l'état de vidéo

```
mediaPlayer.setOnCompletionListener(new MediaPlayer.OnCompletionListener() {
    @Override
    public void onCompletion(MediaPlayer mp) {
        mediaPlayer.release();
        anchorNode.removeChild(videoNode);
        isFinished = true;
    }
});
```

Figure 28 – *CompletionListener*

Contrôle de bouton pour jouer une seule vidéo dans le scène

Au début, c'est possible de jouer plusieurs vidéo dans le scène, mais du point de vue de l'utilisateur, en fait, concentrer sur une seule vidéo à la fois, l'expérience est meilleure. Je l'ai donc modifié pour ne permettre qu'une seule vidéo, et l'utilisateur peut la sélectionner à nouveau après la lecture est terminée .

```
if(isFinished == true){
    playVideo( context: this, augmentedImage);
    Toast.makeText( context: AugmentedImageActivity.this, text: "Ok: play", Toast.LENGTH_SHORT).show();
}
else {
    Toast.makeText( context: AugmentedImageActivity.this, text: "Sorry: a video is playing now, you can't open another!",
    )
}
```

Figure 29 – *Jouer qu'une vidéo*

2 Limites

Même si ce projet a été achevé à temps et à l'heure comme prévu, il reste encore certaines limites de stabilité, d'intégrité et de précision :

Stabilité

Étant donné que les adaptations d'écran de plusieurs tailles ne sont pas conçues, des blocages d'application se produisent parfois lorsque l'écran est pivoté plusieurs fois.

Intégrité

Je ne peux pas réaliser un projet très complet en raison de contraintes de temps. Par exemple, la reconnaissance d'objets 3D.

Par conséquent, j'ai pré-conçu les autres sous-menus du menu drawer et les ai laissés au successeur pour continuer.

Précision

Parfois, le manque de lumière, la vitesse du système d'exploitation mobile et d'autres problèmes affecteront légèrement l'exactitude de la reconnaissance et la précision de la position de la superposition d'image, mais ceci est principalement déterminé par la bibliothèque ARCore que nous utilisons.

Nous devons justement assurer que la position de l'image que nous superposons à partir du point central est cohérente avec la position du point dans l'image réelle.

3 Choix techniques

Comme indiqué dans les chapitres précédents, il y a beaucoup d'outils comme Vuforia, wikitude, EasyAR SDK etc. Mais on a choisi ARCore à la fin en raison de plusieurs facteurs majeurs :

- Gratuit
- Stable
- Reconnaissance des ombres et des lumières
- Détection de plan rapide.

12

Performance

1 Stratégie

Selon des caractéristiques de mon application, j'ai décidé de l'évaluer à partir de trois aspects : Temps d'exécution, Taux de reconnaissances, Précision d'interaction.

Afin d'obtenir les meilleurs résultats de reconnaissance, nous avons évalué la vitesse de reconnaissance et l'effet de reconnaissance d'images de qualité différente. Leurs scores sont répartis entre 20 et 100 :

Scores	Temps de reconnaissance	Stabilité de reconnaissance	Stabilité d'augmentation
20-40	4-5s	+	+
40-60	2-3s	+++	+
60-80	1-2s	++++	++++
80-100	1-2s	+++++	+++++

Figure 1 – Évaluation de la performance des images de différentes qualités

Nous pouvons voir que les images avec des scores supérieurs à 60 n'affectent fondamentalement pas l'expérience.

Et les images des œuvres inuites que nous venons d'échantillonner étaient toutes supérieures à 60. Nous pouvons donc utiliser cet ensemble d'images.

En outre, pendant le déroulement du projet. J'ai également observé plusieurs indices de performance importants, qui sont évalués comme suit :

1.1 Temps

Pendant le processus de programmation, j'ai mesuré le temps de démarrage de l'application, le temps à laquelle l'image a été détectée et le temps à laquelle l'image a été augmentée.

Il a été observé que toutes ces opérations sont inférieures ou égales à 2s.

C'est un moment qui n'est pas facilement perceptible par l'utilisateur et qui est acceptable pour utilisateurs.

1.2 Taux de reconnaissances

Selon la mesure, le taux de reconnaissance de dix images d'œuvres inuits est de 100 pourcent, et la stabilité est bonne, surtout pour des images qui a un score 100.

1.3 Précision d'interaction

À cet égard, le score de qualité de l'image est plus important. La précision de l'augmentation des images avec des scores élevés est très bonne. Certaines images avec des scores plus bas peuvent parfois apparaître légèrement décalées.

Sixième partie

Plan de développement

13

Gestion de projet

Le projet PRD dure environs 6 mois, qui est composé de deux parties principales : une composante de gestion de projet et une composante de mise en œuvre technique.

Parmi ceux-ci, la gestion de projet comprend de l'analyse de sujet, des recherches préliminaires et la rédaction de documents ; La section de mise en œuvre technique comprend : le codage, les tests, le retour d'information des clients, ainsi que les améliorations.

1 Analyse de projet

Une fois le sujet est identifié, contactez immédiatement le tuteur, déterminer le premier RDV, pour obtenir plus d'informations sur le sujet, comme les objectifs du sujet, les besoins du client et penser l'idée de départ.

Pour mon PRD, le sujet est une Application pratique de la vie sur le mobile Android qui support la découverte de la ville avec la réalité augmentée. Il y des éléments très importants :

- Maîtriser le développement Android et Android native, utiliser le langage Java et C++ de manière efficace.
- Avoir une certaine compréhension de la technologie de réalité augmentée.
- Comprendre les caractéristiques de chaque plateforme.
- Comprendre les principes de mise en œuvre.
- Avoir la bonne machine de débogage
- Afin d'optimiser la normalisation et la portabilité du code, il faut faire attention aux habitudes de codage standard, et en utilisant le style unix, par exemple, combien d'espaces pour l'indentation de ligne.

2 Découpage du projets en tâches

Selon la durée totale, les événements importantes (telle que le début de projet, la première soutenance en décembre, la soutenance finale), et des tâches principales et ses complexités, indiquez la liste des tâches et leur temps requises. Et puis définir un diagramme de Gantt.

3 Planning

Après de nombreuses fois de communication avec les clients, nous pensons que ce projet est un projet relativement volumineux et que la plupart des idées sont originales. Il existe donc certaines fonctions idéales, toujours au stade du concept. Notre projet se concentrera sur la partie la plus importante et la plus fondamentale de la technologie de réalité augmentée : la reconnaissance d'image et la superposition d'informations. De ce point de vue, nous avons mené des recherches sérieuses et avons élaboré la liste de tâches et le tableau de cycle de tâches comme indiqué dans la figure suivante.

3.1 Liste de tâches

On a édité une liste de tâches pour tout ce projet, mais notre projet ne complète que la partie principale : la partie réalité augmentée.

(* : à discuter. ** : Dans le futur)

Conduite

1. Recherches de l'État de l'art
2. CDC
3. CDS
4. Découpage de projet
5. Conception
6. Pilotage du projet

Fonctionnalités

1. Recherches de l'État de l'art
2. Détection de bornes (*)
3. Géolocalisation (*)
4. Communication avec le serveur (*)
5. Téléchargement des bases de données (*)
6. Image augmentée (2D)
7. Afficher des informations supplémentaires
8. Reconnaissance d'oeuvre (3D) (*)
9. Ajouter des objets dans la scène réelle
10. Ajouter des jeux
11. Stocker des coupons dans l'espace locale. (*)

Tests

1. Modèle/base de données
2. Tests de connexion avec des bornes (*)
3. Tests de la géolocalisation (*)
4. Tests de Téléchargements de BD (*)
5. Tests de reconnaissance de l'Image Augmentée (2D)
6. Tests afficher des informations supplémentaires.
7. Tests de reconnaissance d'oeuvre (3D) (*)
8. Tests ajouter des objets dans la scène réelle
9. Tests des jeux (*)
10. Tests le stockage de coupons. (**)

Finalisation

- 1. Test utilisateurs
- 2. Mise en prod
- 3. Tests intégration

3.2 Gantt

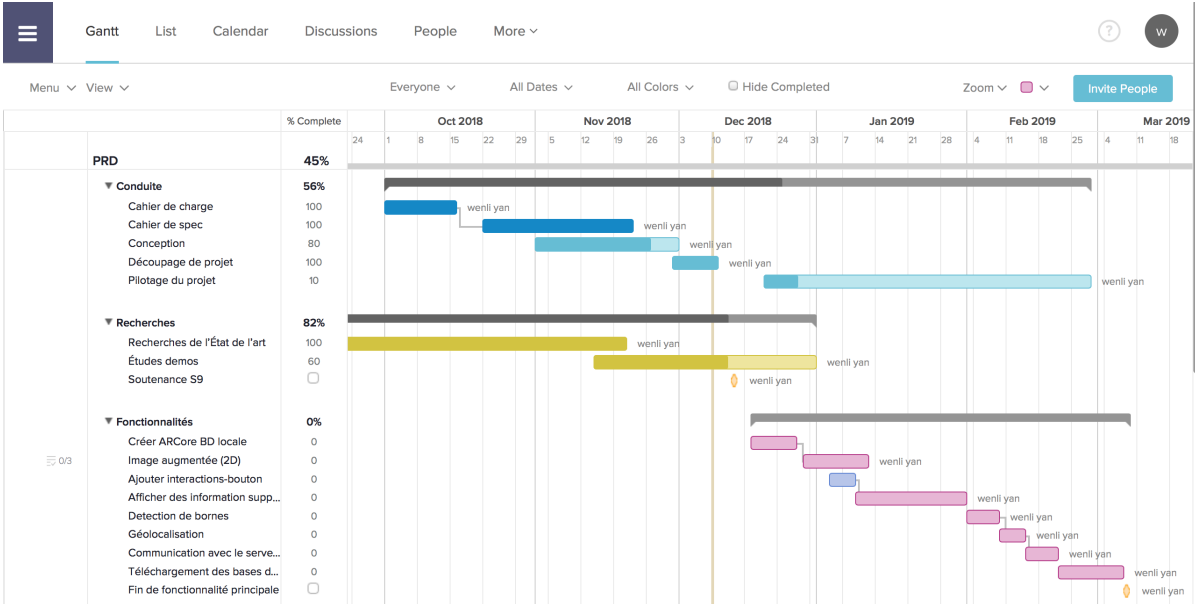


Figure 1 – Période du projet-Gantt0

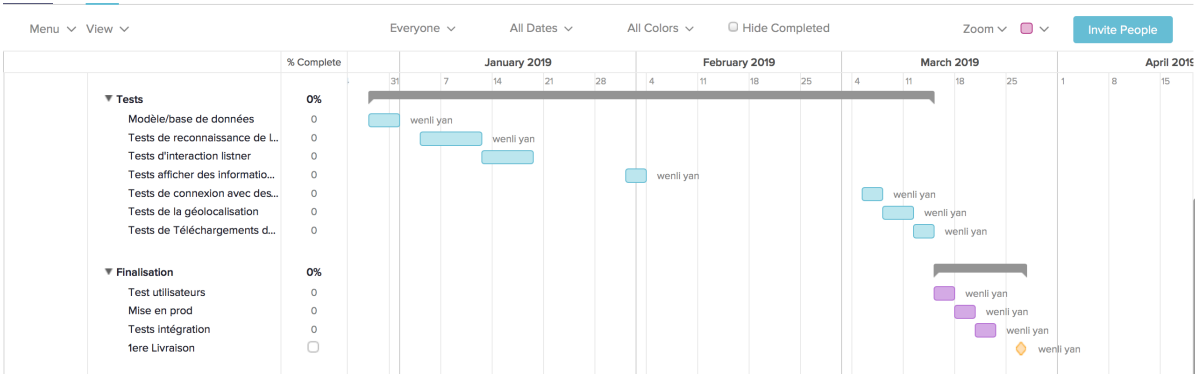


Figure 2 – Période du projet-Gantt1

Septième partie

Bilan et conclusion

C'est un sujet très intéressant qui est populaire de nos jours et un sujet que j'aime bien. Depuis avoir obtenu le sujet, j'ai fait beaucoup d'efforts et j'ai approfondi mes connaissances : j'ai fait beaucoup de recherches, non seulement sur AR et ARCore, mais aussi sur d'autres plateformes connexes ainsi que sur d'autres technologies. J'ai lu beaucoup de littérature, lu de nombreux articles et des nouvelles en chinois, en français et en anglais et, enfin, j'ai une profonde compréhension de ce domaine. Dans le même temps, j'ai également effectué de nombreux tests sur la qualité et la reconnaissance des images et fourni une aide aux autres étudiants du même groupe. En outre, j'ai étudié les divers exemples fournis par le site officiel d'ARCore, compris sa mise en œuvre, puis essayé le positionnement GPS et la détection de la direction du gyroscope (mais, dans ce dernier cas, il n'est peut-être pas utilisé). Ces enquêtes et l'expérience de pratiques m'ont également apporté plus d'idées pour résoudre le problème.

Toutes les semaines, je rencontre mes deux mentors pour discuter de solutions et parfois, nous contactons les techniciens de Flagtown à distance, qui nous ont fourni des conseils. Mais principalement nous le faisons nous-mêmes à partir de zéro. Nous avons avancé diverses idées, qui se sont progressivement clarifiées au fil du processus d'analyse et de démonstration, et les objectifs se sont également progressivement clarifiés. Notre première étape consiste à superposer des informations sur des images bidimensionnelles. C'est aussi ma priorité absolue dans les semaines à venir.

L'achèvement du PRD sera systématiquement géré selon les méthodes que nous avons apprises au cours des trois dernières années, ce qui est très significatif. C'est notre compréhension et la planification du projet est devenue serrée et organisée. Cela apportera une aide précieuse à notre carrière. Ceci est ma note hebdomadaire : <https://docs.google.com/document/d/1xussnh316Ybw0r8odg7JSLzi0Svbm0UwLtYyba7L3z4/edit?usp=sharing>

Enfin, je remercierai les enseignants pour leur soutien et leurs encouragements. Je serai mieux à atteindre les objectifs qui suivent.

Huitième partie

Annexes

14

Spécifications fonctionnelles

1 Géolocalisation de l'utilisateur

La fonction « géolocalisation » est la première fonctionnalité que l'utilisateur rencontre à l'ouverture de l'application afin de lui permettre de sélectionner une bonne base de donnée correspondante pour des oeuvres dans un certain lieu.

L'emplacement de l'utilisateur sera déterminé par les trois bornes à proximité. L'emplacement de ces sites est déterministe grâce à l'échange d'informations entre App et des bornes, l'APP peut déterminer avec précision l'emplacement de l'utilisateur selon un algorithme de calcul.

Ici, nous n'utilisons pas le GPS car les performances du positionnement par GPS à l'intérieur ne sont pas satisfaisantes et nos applications se déroulent souvent à l'intérieur.

Précondition :

- Il y a des bornes autour de l'utilisateur.
- Il y a des oeuvres autour de l'utilisateur.
- L'utilisateur doit posséder l'application et démarrer celle-ci.

Post-condition :

- Se localiser et afficher la géolocalisation de l'utilisateur.
- Envoie les informations de localisation au serveur.

2 Connexion et communication avec serveur

Il faut télécharger des données depuis le serveur donc l'utilisateur doit faire la connexion avec le serveur avec le protocole socket.

Lorsque l'utilisateur lance l'application, celle-ci envoie une demande de connexion au serveur et envoie simultanément son propre emplacement géographique au serveur. Le serveur renvoie alors la liste des données des oeuvres proches de l'emplacement géographique à l'utilisateur.

Précondition :

- Autorisation de connexion.
- Autorisation de l'utilisation du réseau.
- L'utilisateur a réussi de se localiser.

Post-condition :

- Connexion avec le serveur directement.
- Envoie les informations de localisation au serveur et recevoir une liste des données.
- L'utilisateur sélectionne et télécharge les données correspondantes.

3 Reconnaissance et amélioration d'images 2D

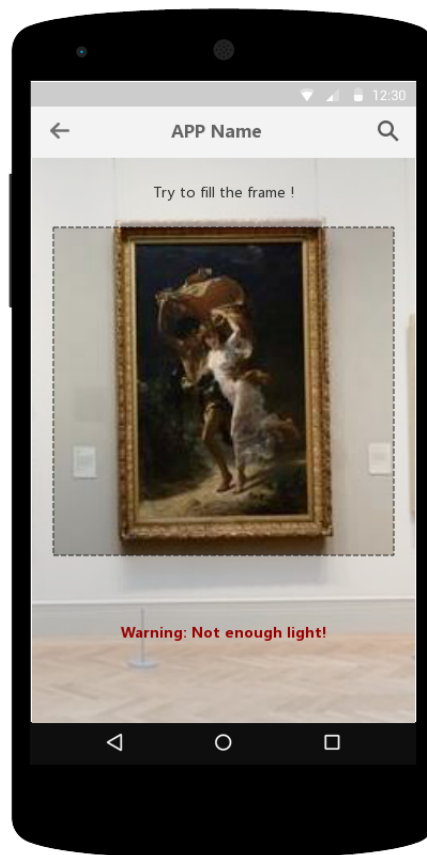


Figure 1 – Reconnaissance d'images 2D

C'est la partie de reconnaissance d'image. Après la base de donnée est prête, est l'utilisation de caméra est autorisée, l'utilisateur peut ouvrir l'interface principale pour scanner une oeuvre en 2D devant lui. La détection sera vite et la reconnaissance prendra un peu de temps. Une fois l'image est testée et reconnue, un bouton apparaît au centre de l'image. Lorsque l'utilisateur clique sur le bouton, une nouvelle petite page (par exemple popwindow) apparaît pour afficher des informations sur l'image, par exemple l'introduction de l'auteur, l'arrière-plan créatif, les caractéristiques et certaines informations masquées.

Précondition :

- Des données sont téléchargées.

- Autorisation de la caméra.
- Environnement approprié pour la reconnaissance(lumière, plate, etc.)
- L'utilisateur est à une distance appropriée de l'oeuvre(Pas très loin, pas très proche).
- L'image est avec assez bonne qualité.

Post-condition :

- L'image est rapidement reconnue.
- Dans le processus d'identification, il affiche des avertissements et informations d'aide.
- Une fois l'image reconnue, les points d'intérêt disponibles pour l'interaction apparaissent à l'écran.

4 Reconnaissance et amélioration partielle de l'image

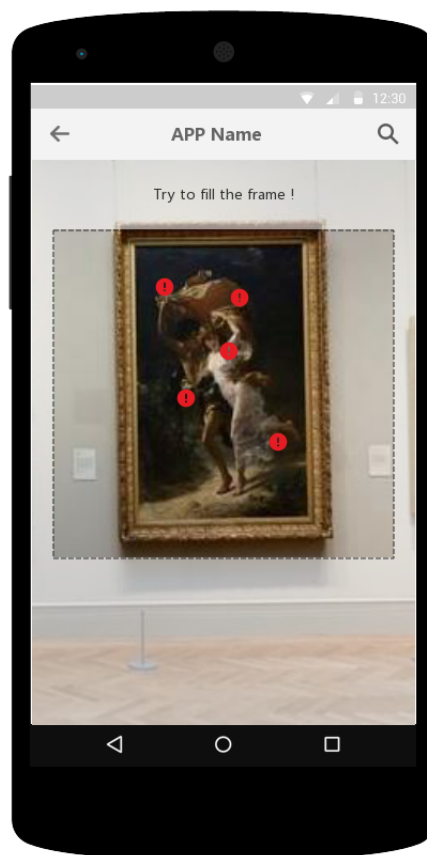


Figure 2 – Reconnaissance des points d'intérêt

Plusieurs points d'intérêt peuvent être trouvés dans une peinture, donc cette application permet de faire une reconnaissance sur des points d'intérêt dans une image. C'est à dire une reconnaissance sur quelque partie d'une image avec une autre base de donnée supplémentaire.

Précondition :

- L'image entière est reconnue.
- Des points d'intérêt sont activés.

Post-condition :

- Des informations sont affichés par des interactions des points d'intérêt.

5 Reconnaissance et amélioration d'objets 3D (optionnel)



Figure 3 – Reconnaissance d'oeuvre 3D

En raison de la technologie limitée, cette partie est la plus difficile à mettre en œuvre. Un objet en 3D peut être vu comme une collection d'innombrables plans et lorsqu'il reçoit l'influence de l'environnement, la reconnaissance change aussi. L'effet idéal est que l'utilisateur puisse reconnaître en douceur l'objet 3D qui se trouve devant et y superposer différentes informations graphiques comme sur des images 2D. Même les informations peuvent être mises à jour avec les changements de position et d'angle de l'utilisateur, tels que différents informations des différents aspects de l'objet.

La pré-condition est la post-condition est similaire avec celui en 2D.

6 Ajouts des actions interactives

En plus, afin d'accroître le plaisir, nous espérons ajouter des interactions avec l'oeuvre.

Interactions sur des points d'intérêt

Que ce soit en 2D ou en 3D, nous configurons les interactions pour chaque point d'intérêt, on peut cliquer sur les points d'intérêt affichés à l'écran et un popupWindow apparaît pour afficher des informations sur l'œuvre.



Figure 4 – Informations supplémentaire sur l'oeuvre

Interactions pour images 2D

Des images 2D sont dans plates. Pour ça on propose des idées comme : une palette peut apparaître, permettant à l'utilisateur de teinter et colorier comme le peintre; ou un casse-tête(puzzle) permettant à l'utilisateur de s'amuser et en même temps approfondir sa compréhension et sa mémoire de l'oeuvre réalisé dans ce processus.

Même, on peut ajouter un objet en 3D lié à cette oeuvre. Cela ressemble à un objet plat qui saute du papier, tout comme l'utilisateur entre dans le monde de l'image..

Interactions pour oeuvres 3D (optionnel)

Des oeuvres 3D sont un peu différentes que oeuvres 2D, car c'est dans un espace spatial. Bien que cela soit plus compliqué, cela offre plus de possibilités aux utilisateurs de jouer à des jeux dans l'espace. Par exemple, nous créerons un élément 3D comme la sculpture pour parler à l'utilisateur, raconter l'histoire et jouer dans la scène, comme une sculpture vivante.

7 Gestion de coupons

Ces coupons de réduction peuvent être gagnés en remportant des jeux interactifs, et des jeux de difficulté différente gagneront différents niveaux de réduction. Ces coupons de réduction obtenus seront stockés dans l'espace de stockage de l'application locale et l'utilisateur pourra accéder au "Gestion de coupons" via le menu de la barre de navigation.

Attention

Dans un court laps de temps, nous ne considérons pas que l'utilisateur fournisse l'enregistrement du compte pour le moment, de sorte que ces coupons de réduction locaux stockés peuvent être partagés avec eux ; si l'application est supprimée, le coupon de réduction sera effacé....

15

Spécifications non fonctionnelles

1 Contraintes de développement, de conception et de maintenance

1.1 Matériels

matériels : Ce projet nécessite des portables des nouveaux modèles qui supportes l'ARCore comme indiqué dans l'état de l'art.

1.2 Langages de programmation imposés ou adoptés

j'ai adopté Java pour le développement Android. Mais aussi native pour augmenter l'efficacité.

Android Native Development Kit (NDK) : cet ensemble d'outils vous permet d'utiliser le code C et C ++ pour Android et fournit un certain nombre de bibliothèques de plates-formes vous permettant de gérer des activités natives et d'accéder à des composants de périphériques physiques tels que des capteurs et des entrées tactiles.

Actuellement, nous ne développons que sur Android. Mais cela sera fait aussi pour la plate-forme ios dans le futur.

1.3 Logiciels et bibliothèques à utiliser pour le développement

ARCore est la technique principale utilisé pour ce projet avec son dernier SDK.

Ce projet vise à atteindre tous les objectifs avec ARCore au lieu d'autres plate-formes comme le Vuforia, wiktitude et EasyAR SDK, etc.

1.4 Environnements nécessaires : simulateurs, outils logiciels;

Le développement de ce projet est sur le système Android d'un mobile qui supporte l'ARCore. Il faut le caméra pour voir la scène réelle. Donc le simulateur ne marche pas pour quelques fonctionnalités de ce projet.

Environnement spécifique d'utilisation : Avant d'utiliser cette application, l'utilisateur doit d'abord télécharger l'ARCore depuis le Google Store.

Environnement spécifique de développement : Logiciel de développement : Android Studio



Figure 1 – *Android Studio*

- Android Studio Version 3.1 ou supérieur.
- Android SDK Platform 7.0 (API de niveau 24) ou supérieur.

2 Contraintes de fonctionnement et d'exploitation

2.1 Performances

2.1.1 Du point de vue de l'utilisateur

Temps de réponse souhaité :

- L'application démarrera dans un délai d'inférieur à 0,2 seconde ;
- Temps de positionnement dans un délai d'inférieur à 0,5 seconde ;
- Temps de téléchargement de la base de données dans un délai d'inférieur à 0,5 seconde ;
- Détecter le temps de l'image dans un délai d'inférieur à 0,2 seconde ;
- Temps de reconnaissance des images 2D dans un délai d'inférieur à 1 seconde ;
- Temps d'ouvrir le popupwindow avec la superposition d'informations superposées est dans un délai d'inférieur à 0.3 seconde ;
- La fréquence de trame du jeu doit correspondre à la fluidité du jeu, au moins 30 trames par seconde.

En fait, pour ce type d'application de réalité augmentée, en raison de l'interaction avec le monde réel, et le monde réel comporte de nombreuses incertitudes, les temps mentionnés ci-dessus sont idéaux et il peut y avoir des retards mineurs, dus notamment à une faible lumière ou au fait que l'utilisateur est trop loin de l'œuvre. Le temps menant à la reconnaissance d'image devient plus long. Afin d'éviter qu'un tel phénomène n'entraîne une mauvaise expérience pour l'utilisateur, il est nécessaire d'ajouter des informations rapides sur l'état actuel, l'information sur l'avancement et le guide d'exploitation afin d'aider l'utilisateur à ignorer le léger retard et à obtenir une meilleure expérience.

2.1.2 Du point de vue de l'environnement

Comme mentionné ci-dessus, pour les applications de réalité augmentée, l'environnement a un impact significatif sur l'expérience de l'application. Par exemple : lumière, distance, reconnaissance de l'objet identifié et statut du réseau (lors de la mise à jour et du téléchargement de la base de données).

L'environnement idéal est :

- 1. La lumière est suffisante,
- 2. L'image reconnue est "basse répétitive"
- 3. L'objet identifié se trouve au milieu de l'écran de l'utilisateur et occupe plus des 2/3 de l'écran.
- 4. Le réseau est bon.

2.2 Capacités

C'est pour préciser les limites des problèmes traitables par le système et les limites des éventuelles extensions comme indiqué ci-dessous :

- Nombre max de terminaux : Ce nombre dépend de la capacité du serveur à supporter et il ne fait aucun doute que la société fournira une capacité suffisante à un grand nombre d'utilisateurs pour le téléchargement d'activités se déroulant comme prévu.

Une fois la base de données téléchargée, l'utilisation de l'utilisateur ne fera pas pression sur le serveur.

- Nombre max de connexions : Ce nombre dépend de la capacité du borne.
- Capacité max de stockage : Pour chaque base de donnée d'ARCore dans ce projet, il permet de stocker 1000 images au maximum.

2.3 Sécurité

Pour protéger la sécurité des données des utilisateurs, l'application ne collecte pas des données des utilisateurs. Tout le monde peut l'utiliser directement sans l'accès de compte ni mot de passe. L'application a besoin de juste les données de la géolocalisation de l'utilisateur pour télécharger les bonnes bases de données.

2.4 Intégrité

Une fois que la base de données correspondante est chargée, le réseau ne sera plus nécessaire. Par conséquent, même si le réseau ne parvient pas à se déconnecter, l'application ne sera pas affectée, elle peut aussi être utilisée normalement.

2.5 Conformité aux standards

L'application mobile devra respecter les standards de dépôt imposé par Google ou Apple (dans le future) afin de pouvoir être déposé sur le PlayStore (Android) ou AppStore (Apple).

Bibliographie

ARCore

- <https://developers.google.com/ar/develop/java/quickstar>
- <https://www.coursera.org/learn/ar>
- <http://www.chinaar.com/ARbaike/5592.html>
- <https://www.leiphone.com/news/201802/L7aIJD1wQKQ94HPb.html>

AR

- https://www.sohu.com/a/236055671_465915
- https://blog.csdn.net/qj_27489007/article/details/78765810
- <https://blog.csdn.net/cf2SudS8x8F0v/article/details/80731195>
- http://www.sohu.com/a/167176445_472880
- https://en.wikipedia.org/wiki/Augmented_reality
- <https://www.realitytechnologies.com/augmented-reality/>
- <https://www.jianshu.com/p/a85a460924de>
- <https://www.forbes.com/sites/bernardmarr/2018/07/30/9-powerful-real-world-applications-of-augmented-reality-ar-today/#575778502fe9>
- <http://www.cqvip.com/qk/98109x/201706/672478511.html>

Base de donnée

- <https://blog.csdn.net/Firewall15788/article/details/78112516>
-

Tests

- <https://www.jianshu.com/p/f6a156d1c4a4>
- <https://blog.csdn.net/yjp19871013/article/details/71597469>
- <https://blog.csdn.net/yjp19871013/article/details/70570760>
- <https://oneflare.io/android-how-to-test-ui-fragments-with-espresso>
- https://blog.csdn.net/qj_17766199/article/details/78710177
- <https://www.jianshu.com/p/a7e6f39aad60>
- <https://juejin.im/post/5b43817ce51d45198e720f40>
- <https://www.jianshu.com/p/f6a156d1c4a4>



GLOSSAIRE

Accelerometer

Accéléromètre - mesure de l'accélération ; pour les smartphones compatibles avec ARCore, il permet d'activer le suivi du mouvement.

Anchors

Ancre - points d'intérêt définis par l'utilisateur sur lesquels des objets RA sont placés. Les ancrages sont créés et mis à jour en fonction de la géométrie (plans, points, etc.). ARCore détecte l'environnement

Asset

Fait référence à un modèle 3D.

Augmented Reality (AR)

Réalité Augmentée - Vision en directe ou indirecte d'un environnement physique réel dont les éléments sont "augmentés" par des informations perceptuelles générées par ordinateur.

Computer Vision

Vision par ordinateur - un mélange d'intelligence artificielle et de science informatique visant à permettre aux ordinateurs (comme les smartphones) de comprendre visuellement le monde qui l'entoure comme le fait la vision humaine.

Concurrent Odometry and Mapping (COM)

Odométrie et cartographie simultanées (COM) - processus de suivi des mouvements pour ARCore et suit l'emplacement du smartphone par rapport à son environnement.

Design Document

Document de conception - Un guide pour votre expérience AR qui contient tous les actifs 3D, sons et autres idées de conception que votre équipe peut mettre en œuvre.

Drift

Dérive - fait référence à l'accumulation d'erreur de suivi de mouvement potentielle. Si vous parcourez trop rapidement les ressources numériques, il se peut que la pose de l'appareil ne reflète pas votre position actuelle. ARCore tente de corriger la dérive dans le temps et met à jour Anchors pour conserver les objets numériques correctement placés par rapport au monde réel.

Edit-time

Heure d'édition - lorsque des modifications sont apportées en mode non-gameplay / en mode édition et avant le déploiement de votre application ou de votre jeu.

Runtime

Durée d'exécution - lorsque des modifications / modifications sont apportées en mode de jeu actif ou pendant l'exécution de votre application. Par exemple, vous pouvez télécharger des ressources Poly pendant que votre application est en mode de jeu ou en cours d'exécution.

Environmental understanding

Compréhension de l'environnement - comprendre le monde réel en détectant des points caractéristiques et des plans et en les utilisant comme points de référence pour cartographier l'environnement. Également appelé «connaissance du contexte».

Feature Points

Points de fonctionnalité - caractéristiques visuellement distinctes de votre environnement, telles que le bord d'une chaise, un interrupteur d'éclairage sur un mur, le coin d'un tapis ou tout autre élément susceptible de rester visible et placé de manière cohérente dans votre environnement. ARCore utilise des points de caractéristique dans l'image capturée de la caméra pour calculer les changements d'emplacement, améliorer la compréhension de l'environnement et placer les plans dans une application AR.

Framing

Cadrage : en ce qui concerne la conception d'AR mobile, il s'agit du placement stratégique d'objets 3D dans l'environnement afin d'éviter la rupture d'immersion.

Google Poly

Google Poly - un référentiel gratuit d'actifs 3D pouvant être rapidement téléchargés et utilisés dans votre expérience ARCore.

GPS

GPS - système mondial de navigation par satellite fournissant des informations sur la géolocalisation et l'heure ; pour les smartphones compatibles avec ARCore, il permet d'activer des applications AR basées sur la localisation.

Phone Camera

Phone Camera - fournit une image en direct du monde réel environnant sur lequel le contenu AR est superposé lors de l'utilisation de l'AR mobile .

Gyroscope

Gyroscope - mesure l'orientation et la vitesse angulaire ; pour les smartphones compatibles avec ARCore, il permet d'activer le suivi du mouvement.

Magnetometer

Magnétomètre - mesure la direction cardinale et permet aux smartphones compatibles ARCore d'auto-faire pivoter automatiquement les cartes numériques en fonction de leur orientation physique, ce qui permet d'activer les applications de RA basées sur la localisation.

Hit-testing

Hit-testing - utilisé pour prendre une coordonnée (x, y) correspondant à l'écran du téléphone (fournie par un robinet ou toute autre interaction que votre application doit prendre en charge) et projeter un rayon dans la vue du monde de l'appareil photo. Cela renvoie tous les plans ou points de repère intersectés par le rayon, ainsi que la pose de cette intersection dans l'espace mondial. Cela permet aux utilisateurs de sélectionner ou d'interagir avec les objets de l'environnement.

HMD

HMD - Afficheur monté sur la tête.

Immersion

Immersion - le sentiment que les objets numériques appartiennent au monde réel. Rompre l'immersion signifie que le réalisme a été brisé ; Dans la RA, il s'agit généralement d'un objet dont le comportement ne correspond pas à nos attentes.

Inside-Out Tracking

Suivi Inside-Out - lorsque l'appareil dispose de caméras et de capteurs internes pour détecter les mouvements et suivre le positionnement.

Outside-In Tracking

Suivi Out-Inside - Lorsque l'appareil utilise des caméras ou des capteurs externes pour détecter les mouvements et suivre le positionnement.

Light estimation

Estimation de la lumière - permet au téléphone d'estimer les conditions d'éclairage actuelles de l'environnement.

Motion Tracking

Suivi des mouvements - dans le sens fondamental, cela signifie suivre le mouvement d'un objet dans l'espace. ARCore utilise la caméra, le gyroscope interne et l'accéléromètre de votre téléphone pour estimer sa pose dans l'espace 3D en temps réel.

Multi-plane detection

Détection multi-plans - La capacité d'ARCore à détecter diverses surfaces à différentes hauteurs et profondeurs.

Occlusion

Occlusion - lorsqu'un objet 3D bloque un autre objet 3D. Actuellement, cela ne peut se produire qu'avec des objets numériques. Les objets ARCore ne peuvent pas être obstrués par un objet du monde réel.

Placing

Placement - lorsque le suivi d'un objet numérique est fixé, ou ancré, à un certain point du monde réel.

Plane Finding

Recherche de plan - Processus spécifique au smartphone par lequel ARCore détermine l'emplacement des surfaces horizontales et verticales dans votre environnement et utilise ces surfaces pour placer et orienter des objets numériques.

Pose

Pose - la position et l'orientation uniques de tout objet par rapport au monde qui l'entoure, de votre appareil mobile à la ressource 3D augmentée que vous voyez sur votre écran.

Raycasting

Raycasting - projection d'un rayon pour aider à estimer l'endroit où l'objet RA doit être placé pour apparaître de manière crédible sur la surface du monde réel ; utilisé pendant les tests.

Scaling

Lorsqu'un objet AR placé change de taille et / ou de dimension par rapport à la position du périphérique AR ; activé par la compréhension de l'environnement.

SLAM

SLAM - processus de suivi de mouvement qui suit l'appareil par rapport au monde qui l'entoure.

Spatial mapping

Cartographie spatiale - possibilité de créer une carte 3D de l'environnement et d'établir où les actifs doivent être posés.

Standalone headset

Casque autonome - Casque VR ou AR ne nécessitant ni processeur externe ni mémoire, ni alimentation.

Surface detection

Détection de surface : permet à ARCore de placer des objets numériques sur différentes hauteurs de surface, de restituer différents objets à différentes tailles et positions et de créer des expériences de RA plus réalistes en général.

Unity

Unity - moteur de jeu multi-plateforme et environnement de développement pour les applications interactives 3D et 2D.

User Flow

Flux d'utilisateurs - le parcours des utilisateurs de votre application et la manière dont une personne s'engage, étape par étape, avec votre expérience de RA.

User Experience (UX)

Expérience utilisateur (UX) - le processus et le cadre sous-jacent d'amélioration du flux utilisateur pour créer des produits offrant une grande facilité d'utilisation et une grande accessibilité pour les utilisateurs finaux.

User Interface (UI)

Interface utilisateur - les éléments visuels de votre application et de tout ce avec quoi un utilisateur interagit.

User interface metaphor

Métaphore de l'interface utilisateur - fournit à l'utilisateur une connaissance instantanée de la manière d'interagir avec l'interface utilisateur, comme un clavier QWERTY ou une souris d'ordinateur.

Virtual Reality (VR)

Réalité virtuelle (VR) - utilisation de la technologie informatique pour créer un environnement simulé, plaçant l'utilisateur dans une expérience.

Wenli Yan

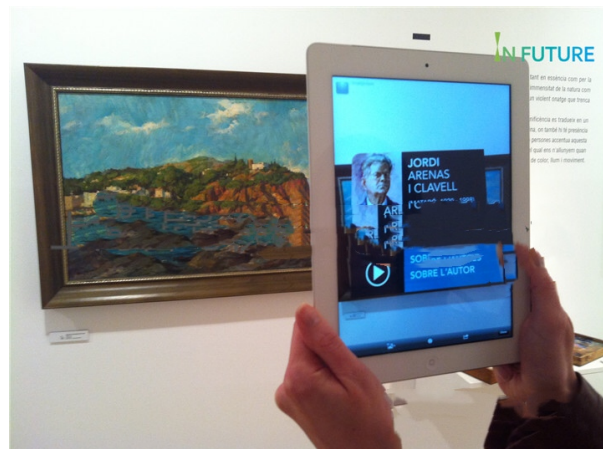
Encadrement : Gilles Venturini et Barthelemy Serres



En collaboration avec Flagtown

Objectifs

Ce projet vise à mettre en œuvre une application mobile utilisant la dernière technologie d'AR: ARCore, pour permettre aux utilisateurs d'en apprendre davantage sur les œuvres artistiques grâce à une expérience d'interaction virtuelle: augmenter les œuvres d'art par supposer des éléments sur l'image.



Objectif du projet

Mise en œuvre

Avec ARCore, nous avons réalisé l'idée de cette application: Détecter l'image, augmenter l'oeuvre par supposer des **image(2D)**, **vidéo**, **objet(3D)** et des **interactions** qui peut fournir plus d'information sur lui.



ARCore

Librairie d'AR

Résultats attendus

- Détection d'une image ou plusieurs image en même temps
- **Supposer Image 2D** transparent layout rendu sur l'oeuvre
- **Jouer une vidéo** dans la scène réelle
- Afficher des informations complémentaires
- Interactions utilisateurs

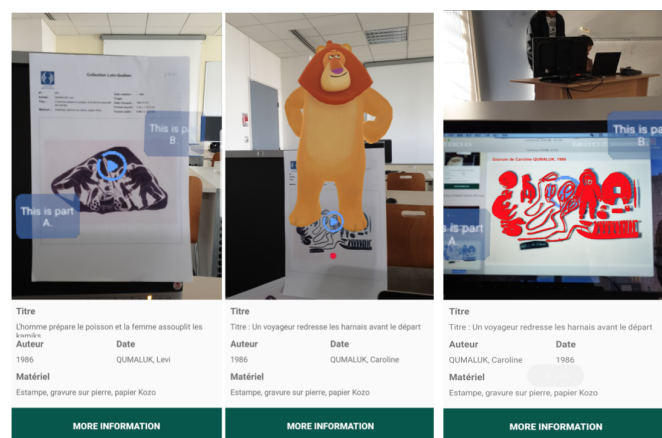


Image de Résultats



Objectifs

Ce projet vise à mettre en œuvre une application mobile utilisant la dernière technologie d'AR: ARCore, pour permettre aux utilisateurs d'en apprendre davantage sur les œuvres artistiques grâce à une expérience d'interaction virtuelle: augmenter les œuvres d'art par supposer des éléments sur l'image.



Objectif du projet

Mise en œuvre

Avec ARCore, nous avons réalisé l'idée de cette application: Détecter l'image, augmenter l'œuvre par supposer des **image(2D)**, **vidéo**, **objet(3D)** et des **interactions** qui peut fournir plus d'information sur lui.

Résultats attendus

- Détection d'une image ou plusieurs image en même temps
- **Supposer Image 2D** transparent layout rendu sur l'oeuvre
- **Jouer une vidéo** dans la scène réelle
- Afficher des informations complémentaires
- Interactions utilisateurs



ARCore

Librairie d'AR

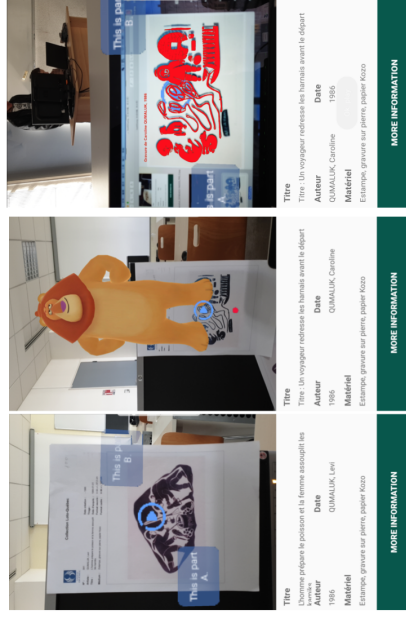


Image de Résultats

Mobile App en Réalité Augmentée polytech

Résumé

Ce projet vise à mettre en œuvre une application basée sur Android utilisant la dernière technologie de la réalité augmentée: ARCore, pour permettre aux utilisateurs d'en apprendre davantage sur les œuvres artistiques et culturelles d'une ville grâce à une expérience d'interaction virtuelle: augmenter les œuvres d'art par supposer des éléments sur l'image comme: images, vidéo, musique, animation, 3D objet etc.

Mots-clés

Réalité augmentée, ARCore, Android, Culture

Abstract

This project aims to implement an Android-based application using the latest Augmented Reality technology: ARCore, to enable users to learn more about a city's artistic and cultural works through a virtual interaction experience. : to increase the works of art by supposing elements on the image, like images, video, music, animation etc.

Keywords

Augmented Reality, ARCore, Android, Culture

Entreprise

Flagtown



Tuteurs entreprise

Stéphane GRANDJEAN (CEO Flagtown)

Xavier GOUPIL

Étudiant

Wenli YAN (DI5)

Tuteurs académiques

Gilles VENTURINI

Barthelemy SERRES