

Master Informatique
1^{ère} année

Projet Tutoré

**Implémentation d'un système de réalité
augmentée orienté localisation multi-agents**

Rapport Technique

Professeur Référent : Loïs VANHEE

Ajdin SUMIC & Maxime FRANÇOISE
(21608530 - 21302023)

2019 – 2020

Lien du projet : <https://github.com/ajdin14200/ASProject.git>

Ce projet s'intègre dans l'initiative "la confrérie du code", initiée par Loïs Vanhée. Cette initiative vise à permettre aux apprenants de développer, en plus des compétences techniques classiques, la panoplie des compétences impliquées par la mise en place de projets en pratique (communication, organisation, collaboration, etc) et de garder une vision réflexive sur leur propre développement. Les rapports complémentaire et pédagogique décrivent nos actions complémentaires à l'effort technique et l'évolution de notre développement.

TABLE DES MATIÈRES

Introduction.....	3
État de l’art.....	4
Outils existants.....	5
1) Unity.....	5
2) OpenStreetMap.....	5
3) Google Map Platform Premium Plan.....	6
Outils utilisés.....	7
1) Android Studio.....	7
2) Google Map Platform.....	7
3) Google Places.....	8
4) Amazon Web Services (AWS).....	8
Réalisation technique.....	9
1) Serveur.....	9
2) Les Services Google Map et Places.....	10
3) Les Sockets.....	12
Architecture.....	13
1) Architecture des packages.....	13
2) Implémentation des activités.....	13
3) Pattern Composite.....	14
Application.....	15
Validation.....	17
Conclusion.....	18
Remerciement.....	19

INTRODUCTION

L'informatique est un domaine très complet et s'est fortement développé au fil des années. À tel point qu'aujourd'hui, il est presque impossible de s'en passer. Il a permis de développer de nouvelles technologies, comme par exemple les smartphones ou encore la réalité augmentée. Deux technologies en plein essor. En effet, les smartphones font parti du quotidien de la majorité de la population mondiale. Quant à la réalité augmentée, elle se développe de plus en plus et de plus en plus vite. De plus, la réalité augmentée orientée localisation s'adapte parfaitement sur smartphone. C'est principalement pour ces deux raisons que nous avons choisi ce sujet, à savoir : « Implémentation d'un système de réalité augmentée orienté localisation multi-agents ».

Nous souhaitons avant tout apprendre à utiliser de nouvelles technologies en relation avec l'informatique et développer des applications sur un support autre qu'un ordinateur. La réalité augmentée a aussi été un élément décisif à cette décision. C'est une technologie encore nouvelle et peu développée qui présente pourtant un potentiel énorme. En effet, on peut définir la réalité augmentée comme étant un monde physique/réel sur lequel vont s'ajouter des éléments virtuels/immatériels. Il y a donc beaucoup de possibilités d'implémentation comme par exemple un jeu de capture de nœuds/camps à échelle réelle.

Dans ce projet, nous nous sommes plus particulièrement intéressés au côté localisation de la réalité augmentée pour une utilisation multi-utilisateurs. Ce côté localisation permet d'ajouter des éléments virtuels à un endroit bien précis et de développer des applications évoluant dans un environnement à taille réelle. Le côté multi-utilisateurs permet d'autoriser plusieurs personnes à observer les mêmes éléments virtuels dans un même environnement. Ces deux aspects combinés vont inciter les utilisateurs à se rencontrer afin de remplir leurs objectifs et ainsi fournir une raison supplémentaire aux utilisateurs de se sociabiliser d'avantage.

Il existe déjà plusieurs applications de ce genre, comme par exemple « *Pokemon Go* » ou encore « *Harry Potter : Wizards Unite* » pour les plus connues, mais très peu d'entre elles mettent l'accent sur ce côté multi-utilisateurs. En effet, en général ce type d'application n'autorise que peu, voir pas du tout, d'interaction entre les utilisateurs et encore moins de rencontre.

L'objectif de ce projet consiste à développer une application avec les caractéristiques suivantes :

- elle se déploie sur Android
- elle ajoute des éléments virtuels à des positions précises
- elle donne un moyen d'interaction avec ces objets
- elle récupère la position des utilisateurs et la partage avec d'autres

Pour cela, nous avons décidé d'implémenter notre application sous forme de jeu dont le but est de capturer des nœuds/camps. Ce jeu propose un affrontement entre deux équipes dont l'objectif est de capturer plus de nœuds que l'équipe adverse dans un temps imparti. Afin de gagner la partie, les joueurs devront communiquer entre eux et établir des stratégies. Ce jeu est développé pour être fonctionnel sur Android et utilise une connexion réseau et le capteur de position.

Le principal défi de ce projet se retrouve dans la gestion des serveurs permettant aux joueurs de jouer ensemble. En effet, afin de développer les mêmes éléments virtuels à chacun des joueurs, ils doivent être connectés entre eux afin de se partager la position des nœuds à capturer ou encore leur propre localisation. N'ayant que peu de connaissance dans ce domaine, ce défi est pour nous un réel challenge, d'autant plus que la recherche d'un hébergeur de serveur gratuit et possédant un DNS (Domain Name System) publique demande beaucoup de temps et de patience.

ÉTAT DE L'ART

Aujourd'hui, il existe peu d'applications implémentant un système de réalité augmentée orientée localisation multi-utilisateurs. Cela est particulièrement dû au fait que la plupart des personnes ne voient la réalité augmentée que comme étant orientée image. Il en existe cependant quelques unes.

La plupart des applications implémentant ce système sont des jeux vidéo, le plus connu étant « *Pokemon Go* ». C'est un jeu mobile massivement multijoueur développé par le studio Niantic. Le



Illustration 1: Image montrant la carte de Pokemon Go et ses éléments virtuels

jeu consiste à se déplacer sur une carte via la géolocalisation afin de capturer des Pokemon (petites créatures imaginaires). Le jeu utilise aussi bien l'aspect localisation, lors des déplacements des joueurs et de l'apparition des Pokemon sur la carte, que l'aspect image, lors de la capture des Pokemon. Sur la carte, le joueur pourra apercevoir des éléments virtuels, comme par exemple des Pokemon ou des arènes. En revanche, aucun autre joueur n'apparaît sur la carte. Ce qui retire le côté intéressant du massivement multijoueur. Les seules interactions disponibles entre joueurs sont des actions indirectes, comme lors de la capture d'une arène par exemple. De plus, il est impossible d'accéder aux fichiers du jeu gratuitement. C'est principalement pour ces deux dernières raisons que nous n'avons pas pu utiliser cette application pour implémenter la notre.

Il existe d'autres jeux vidéo comme par exemple « *Ingress* » ou encore « *Harry Potter : Wizards Unite* » dont le fonctionnement est très proche de celui de « *Pokemon Go* ». Ces sont tous les deux des jeux en réalité augmentée utilisant un système de géolocalisation et développés par le studio Niantic. Cependant, ces deux jeux rencontrent les mêmes problèmes que le précédent, à savoir :

- impossibilité d'accéder à la position des autres joueurs
- impossibilité d'accès aux fichiers du jeu gratuitement

Les applications autres que jeux vidéo qui utilisent un système de géolocalisation ne s'en servent que dans le but d'avoir l'orientation de l'appareil en leur possession afin d'utiliser l'image plus facilement. C'est le cas de nombreux projets dans le domaine de l'industrie.

Nous nous sommes donc retrouvés sans support et nous avons dû développer notre application en partant de zéro. Nous nous sommes cependant inspirés du fonctionnement principal de ces jeux vidéo et de leurs défauts afin de développer notre application.

OUTILS EXISTANTS

1) Unity



Unity est un moteur de jeu développé par Unity Technologies. Il fait parti des moteurs de jeu les plus utilisés notamment grâce à son côté multi-plateformes. En effet, Unity permet de créer des applications fonctionnant sur un grand nombre de plateformes, comme par exemple sur ordinateur sous Windows, Linux et MacOS, sur console de jeux vidéo, sur des sites web ou encore sur smartphone, sous Android et IOS. Il permet de développer des applications 2D et 3D et , depuis 2018, il est même possible de l'utiliser afin de réaliser de la modélisation 3D.

Une des grandes particularités d'Unity réside dans l'Asset Store, une sorte de boutique permettant d'accéder à du contenu créés par d'autres utilisateurs. Ce contenu peu être gratuit ou payant et permet de faciliter le développement d'applications en mettant à dispositions de nombreuses ressources, comme par exemple des éléments 3D.

Ce moteur de jeu est d'autant plus intéressant qu'il propose une licence gratuite dite « Personal ». Cette licence gratuite permet d'utiliser Unity dans un cadre pédagogique sans avoir à payer. Le seul inconvénient est que cette licence contient quelques limitations sur les plateformes disponibles. Elle n'autorise pas l'édition des applications sur les consoles de jeux vidéo. Cela ne constitue pas un problème dans notre cas car nous souhaitons implémenter notre application sur Android.

De nombreuses applications ont été développées avec ce moteur de jeu, comme par exemple « Pokemon Go » ou encore « Hearthstone : Heroes of Warcraft » et cela est d'autant plus vrai pour ce qui est des applications sur smartphone. Cette performance est dû notamment à sa palette d'API (Interface de Programmation Applicative) et à sa documentation très détaillée, sa rapidité d'implémentation grâce à la réduction du temps de programmation et grâce à ses outils de profilage permettant d'optimiser les performances.

En revanche, nous n'avons pas choisi cet outil car, pour des personnes n'ayant jamais touché à un moteur de jeu, Unity reste trop complet et difficile à prendre en main. Nous aurions perdu beaucoup trop de temps à comprendre son fonctionnement et cela n'était pas envisageable.

Implémentation	Apprentissage	Documentation	Prix
Très facile et rapide d'implémenter une application sur différentes plateformes	Très complet défavorisant la prise en main et l'apprentissage	Très vaste documentation avec beaucoup d'API et de contenu	0€ pour un développement sur mobile

Illustration 2: Table avantages/inconvénients pour Unity

2) OpenStreetMap



OpenStreetMap (OSM) est une plateforme collaborative de cartographie en ligne visant à constituer une base de données géographiques libre du monde en utilisant le système GPS et d'autres données libres. Elle a été mise en route en Juillet 2004 par Steve Coast à l'University

College de Londres. Cette plateforme est sous licence libre et implémentable sur ordinateur (Linux, Windows et MacOS) et smartphone (IOS et Android).

Elle a été conçue dans le but :

- d'alimenter la carte mondiale et en extraire certaines parties pour un usage personnel
- créer des cartes interactives ou statiques
- créer des cartes pour de nombreux terminaux GPS
- alimenter certains systèmes d'informations géographiques

Nous n'avons finalement pas choisi cet outil car il y a peu de documentation et la plupart des aides et/ou forum sur internet traite de Google Map et non OpenStreetMap.

Implémentation	Apprentissage	Documentation	Prix
Très facile et rapide d'implémenter dans une application sur différentes plateformes	Prise en main très facile et très rapide	Peu de documentation	0€

Illustration 3: Table avantages/inconvénients OpenStreetMap

3) Google Map Platform Premium Plan Google Cloud

Google Map Platform est une API mise à disposition par Google. Google Map Platform Premium Plan permet d'avoir accès à la totalité des API disponibles. Cela permet notamment à plusieurs joueurs de se partager des informations, comme par exemple leur position ou des lieux. Cela aurait été parfait pour notre cas d'application.

Malheureusement, comme indiqué c'est un service premium et donc payant. Les tarifs sont très variables et dépendent principalement du nombre de données utilisées par mois. C'est principalement pour cette raison que nous n'avons pas utilisé cet outil.

Implémentation	Apprentissage	Documentation	Prix
Très facile et rapide d'implémenter dans une application sur différentes plateformes	Prise en main très facile et très rapide	Vaste documentation	Très variable

Illustration 4: Table avantages/inconvénients Google Map Platform Premium Plan

1) Android Studio



Android Studio est un Environnement de Développement Intégré (IDE) développé par Google en 2013. Il a été créé dans le but de développer des applications mobiles Android. Il est basé sur l'IDE IntelliJ IDEA et utilise le moteur de production Gradle. Android Studio permet principalement d'éditer les fichiers Java et les fichiers de configuration XML d'une application Android. Il propose également des outils afin de gérer le développement d'applications multilingues et permet de visualiser rapidement la mise en page de l'application développée. Il intègre aussi un émulateur permettant de faire tourner système Android virtuel sur un ordinateur. Il dispose d'une licence entièrement gratuite dans le cadre universitaire.

Nous avons choisi cet outil pour développer notre application principalement car nous étions déjà familiarisés avec celui-ci. De plus, il existe beaucoup de documentations sur internet et il est plutôt simple à prendre en main car il crée toutes les dépendances et fichiers nécessaires au bon fonctionnement d'une application mobile. Le seul point faible se trouve dans le travail partagé via cet outil. En effet, les fichiers créés par cet IDE sont dépendant de l'environnement utilisé et cela posait quelques problèmes lors de l'échange des fichiers.

Implémentation	Apprentissage	Documentation	Prix
Très facile et rapide d'implémenter une application Permet une implémentation uniquement sur Android	Prise en main très facile et très rapide	Beaucoup de documentation surtout via des forums	0€

Illustration 5: Table avantages/inconvénients Android Studio

2) Google Map Platform



Google Map Platform est une API mise à disposition par Google. Elle utilise le service Google Maps qui est un service de cartographie en ligne. Cette API est gratuite pour une implémentation sur mobile. Dans sa version gratuite, elle est plus ou moins équivalente à l'API d'OpenStreetMap. Nous avons choisi cette API principalement car la documentation à propos de celle-ci est très développée, même si elle est souvent périmée.

Implémentation	Apprentissage	Documentation	Prix
Très facile et rapide d'implémenter une application	Prise en main très facile et très rapide	Beaucoup de documentation surtout via des forums	0€ pour une implémentation sur mobile

Illustration 6: Table avantages/inconvénients Google Map Platform

3) Google Places



Google Places est une API mise à disposition par Google. Elle permet d'avoir accès à des millions d'informations stockées sur une base de données. Elle est principalement utilisée pour associer une géolocalisation à une adresse ou un nom. Elle est gratuite pour des implémentations sur mobile.

Nous avons utilisés cette API principalement car elle est simple d'utilisation et qu'elle s'adapte parfaitement avec l'API Google Map Platform.

Implémentation	Apprentissage	Documentation	Prix
Très facile et rapide d'implémenter une application	Prise en main très facile et très rapide	Beaucoup de documentation surtout via des forums	0€ pour une implémentation sur mobile

Illustration 7: Table avantages/inconvénients Google Places

4) Amazon Web Services (AWS)



Amazon Web Services (AWS) a été lancé en 2006 par Andy Jassy et est une division du groupe américain de commerce électronique Amazon. Il fournit des services en lignes à d'autres sites internet ou application clientes. Il offre principalement des fonctionnalités aux développeurs à travers des API. Il existe plus de 90 services comprenant le calcul, le stockage, le réseau, la base de données, l'analyse de données, des services applicatifs, du déploiement, de la gestion de système, etc.. AWS met à disposition de ses clients un cluster (grappe de serveurs) virtuel de machines disponible à tout moment via internet. Ces machines virtuelles émulent des caractéristiques d'un ordinateur réel. Chaque machine virtuelle est entièrement programmable.

Nous l'utilisons principalement dans le but d'héberger nos serveurs permettant de connecter plusieurs joueurs entre eux. AWS est gratuit pour une application utilisant un transfert de données allant jusqu'à 5GB par mois. En revanche, la prise en main est très difficile notamment à cause du nombre important de possibilités et du fait que nous avons du programmer entièrement la machine virtuelle afin de pouvoir l'utiliser.

Implémentation	Apprentissage	Documentation	Prix
Une fois programmée la machine virtuelle est facile d'accès	Prise en main très difficile et configuration entière de la machine virtuelle	Beaucoup de documentation surtout via des forums	0€ jusqu'à une limite de transfert de données de 5GB/mois

Illustration 8: Table avantages/inconvénients Amazon Web Services

RÉALISATION TECHNIQUE

Dans cette partie, nous allons présenter les différentes étapes lors du développement de notre application et expliquer les raisons pour lesquelles nous avons fait ce choix dans certains cas.

1) Serveur

Tous d'abord, l'un des plus gros problèmes que nous avons rencontrés consistait à faire communiquer les appareils Android entre eux. En effet, chaque utilisateur doit faire savoir aux autres utilisateurs que sa position a changé pour qu'ils puissent mettre à jour sa position. C'est pour cette raison que nous avons décidé d'utiliser un serveur pour les connecter entre-eux. Le problème était que nous ne pouvions pas utiliser un serveur local car nous ne pouvions pas connecter tous les appareils à la même connexion. Nous avons donc cherché une plateforme qui nous permettrait d'héberger nos serveurs.

Nous n'allons pas expliquer le fonctionnement d'AWS car celui-ci possède de très nombreuses possibilités d'utilisation et là n'est pas le but. Pour plus d'informations sur AWS : <https://aws.amazon.com/fr>. En revanche, grâce à AWS nous avons appris à :

- créer une instance Linux à laquelle nous pouvons accéder en utilisant la technologie de connexion à distance SSH (Secure Shell) et qui possède une adresse publique et privée
- gérer une instance Linux de AWS avec les groupes de sécurité à définir ainsi que la bonne mise en place de l'instance
- utiliser l'instance dans le but de créer un serveur, en utilisant les technologies Node.js et LAMP, et configurer celui-ci afin que nous puissions nous y connecter via l'adresse IP publique de l'instance.

Les instances AWS ne sont pas faciles à prendre en main et nous avons perdu beaucoup de temps à les maîtriser suffisamment pour poursuivre le projet. Cependant, cela nous a permis d'en apprendre beaucoup sur la gestion d'un serveur dans son intégralité.

```
ajdin@ajdin-Legion-Y540-17IRH:~/Bureau$ ssh -i "ajdin.pem" ec2-user@ec2-3-10-56-65.eu-west-2.compute.amazonaws.com
Last login: Fri Mar 27 10:39:11 2020 from 91-166-117-243.subs.proxad.net

  _ | _ | _ )
 _ | ( _ | /  Amazon Linux AMI
 _ | \ _ | _ |

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
20 package(s) needed for security, out of 20 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-27-132 ~]$
```

Illustration 9: Image montrant la connexion à distance d'une instance AWS

2) Les Services Google Map et Places

Afin d'implémenter le côté localisation de notre système, nous avons utilisé les API Google Map et Google Places. Ce sont deux services de Google qui nous ont permis d'afficher une carte et de convertir des adresses et noms en coordonnées.

Afin d'utiliser les services Google, nous avons créé une licence Google. Cela nous a donné accès à tout une panoplie de services. En revanche, lors de l'utilisation de ces services, nous avons dû faire attention à ne pas dépasser un seuil de connexion journalier et/ou mensuel afin d'éviter des frais car, au même titre que AWS, les services Google sont sous licence payante avec une option découverte qui est limitée.

Premium plan web service requests

Premium plan web service requests par jour ▼




jeu. 27	sam. 29	mars 01	mar. 03	jeu. 05	sam. 07	lun. 09	mer. 11	
Nom du quota							Limite	Modifier
Premium plan web service requests per day							Illimité	
Premium plan web service requests per 100 seconds							10000	
Premium plan web service requests per 100 seconds per user							Illimité	

Illustration 10: Exemple de limite de connexion pour une licence Premium

Afin d'utiliser les services Google sous Android, nous les avons implémentés dans le Gradle de l'application. Ensuite nous avons créé un fichier XML contenant la clé de l'API dans le but de pouvoir utiliser les services Google.

Ensuite, nous avons intégré les fragments des API dans le fichier XML permettant la gestion de l'interface de l'application.

```
<fragment android:id="@+id/autocomplete_fragment"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:name="com.google.android.libraries.places.widget.AutoCompleteSupport
    Fragment"
/>
```

Puis dans l'activité, qui correspond aux codes brutes des différentes pages représentant l'interface graphique, nous avons d'implémenté le fragment comme ceci :

- pour Google Map

```
SupportMapFragment mapFragment =  
(SupportMapFragment).getSupportFragmentManager().findFragmentById(R.id.map);  
mapFragment.getMapAsync(this);
```

- pour Google Places

```
placesClient = Places.createClient(this);  
List<Place.Field> fields = Arrays.asList(Place.Field.ID, Place.Field.NAME,  
    Place.Field.LAT_LNG);  
searchPlace = new  
    Autocomplete.IntentBuilder(AutocompleteActivityMode.FULLSCREEN,  
    fields).setTypeFilter(TypeFilter.ESTABLISHMENT).build(this);  
startActivityForResult(searchPlace, AUTOCOMPLETE_REQUEST_CODE);
```

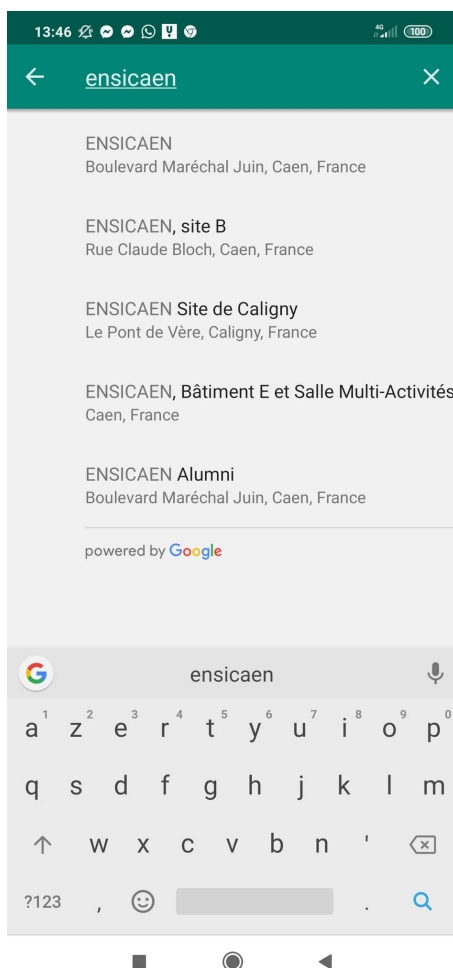


Illustration 11: Exemple d'utilisation de Google Places

Pour Google Map, une fois que le fragment « initializer » a effectué sa tâche, nous avons un callback sur la méthode « onMapReady » qui crée la map. C'est cette fonction que nous modifions afin d'afficher les éléments dont nous avons besoin, tel que les marqueurs avec l'objet « Marker », ou encore les nœuds avec des cercles de type « Circle ».

Pour Google Places plusieurs options sont disponibles pour l'affichage. Ici, nous avons opté pour que l'API se lance dans une autre activité et renvoie, après le choix du lieu effectué, le nom et les coordonnées de celui-ci.

Google Place n'est pas parfait car ce n'est pas un GPS, il connaît juste la position de lieu connu par Google Map, comme par exemple ici où nous pouvons voir la localisation de l'ENSICAEN ou du campus2 mais il ne connaît pas par exemple les bâtiments Sciences 1,2 et 3.

3) Les Sockets

Afin de faire communiquer les différents utilisateurs entre eux, il nous a fallu mettre en place un serveur mais également tout un système de socket entre les utilisateurs et le serveur afin de récupérer les coordonnées de chaque utilisateur et de faciliter la gestion entre les différentes parties de jeu. Dans un premier temps, nous avons souhaité utiliser la librairie OkHttp d'Android afin d'établir une connexion au serveur via une URL http. Malheureusement, nous avons rencontré un certain nombre de problèmes avec cette technologie pour finalement nous rendre compte que nous ne pouvions pas l'utiliser.

Nous nous sommes donc intéressé aux outils de Socket mis à disposition par Android Studio. Dans un premier temps, nous avons essayé d'utiliser cet outil en brut sans grand résultat, principalement dû au manque de documentation de cet outil. Nous avons fini par trouver une librairie implémentant le système de Socket sur Android permettant d'établir une connexion avec un serveur distant (lien de la librairie <https://github.com/nkzawa/socket.io-android-chat>).

Grâce à cette librairie, nous avons réussi à nous connecter à notre serveur sur AWS, mais le gros défaut de cette librairie est son manque de documentation et donc cela a été difficile de comprendre le fonctionnement de ses socket surtout la fonctionnalité de certaine méthode présente dans cette librairie que nous avons pas pu utiliser.

Pour se connecter à notre serveur nous avons donc initialisé la socket :

```
String url1= "http://ec2-3-10-56-65.eu-west-2.compute.amazonaws.com:8080";  
socket= IO.socket(url1);
```

Puis nous avons établi la connexion :

```
socket.connect();
```

Pour envoyer un message au serveur, on utilise socket.emit(cle, message) :

```
while (!socket.connected()){  
socket.emit("firstuser",user.getUsername());
```

Pour se déconnecter il suffit de faire :

```
socket.close();
```

Pour recevoir un message du serveur nous utilisons « server.on », dans l'exemple ci-dessous, l'utilisateur collecte les coordonnées et les met à jour dans le cas où il est pas celui qui a effectué l'envoi :

```
socket.on("coord", new Emitter.Listener() {  
    @Override  
    public void call(Object... args) {  
        //JSONObject data= (JSONObject)args[0];  
        String res= String.valueOf(args[0]);  
        System.out.println("res");  
        SplitResponse splitResponse = new SplitResponse();  
        ArrayList<String>data = splitResponse.splitSingleListData(res);  
        User userToSet = new User(data.get(0),new  
            ObjectCoordinate(Double.parseDouble(data.get(1),  
            Double.parseDouble(data.get(2))));  
        boolean b = user.equals(userToSet);  
        if (!user.equals(userToSet)){  
            ((MapsActivity)context).setUserOnMap(map, userToSet);  
        }  
    }  
}
```

ARCHITECTURE

1) Architecture des packages

Dans cette partie nous allons vous montrer la manière dont nous avons organisée notre application. Tout d'abord voici un diagramme qui montre les différents packages de l'application.

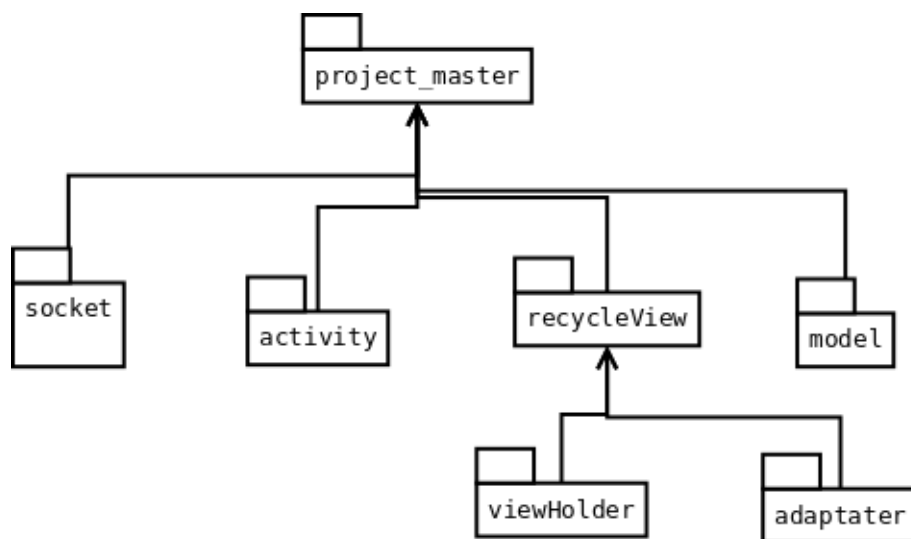


Illustration 12: Diagramme des différents packages de l'application

2) Implémentation des activités

Ensuite nous allons vous présenter les activités qui constituent notre application :

- La première activité permet à un utilisateur de créer un pseudo pour que les autres utilisateurs puissent le reconnaître. Grâce au système de socket, il envoie son pseudo au serveur, puis l'application va le stocker dans une liste et si l'utilisateur est le premier à se connecter, alors il devient le maître de jeu et le serveur va lui envoyer l'envoyer vers l'activité suivante tandis que les autres vont attendre que le maître de jeu leur envoie la notification indiquant que la partie peut être lancée.
- La deuxième activité est destinée au maître de jeu. Il doit y choisir les nœuds de la partie. Ces nœuds sont affichés via une « RecyclerView ». Lorsque tous les nœuds ont été choisis, le maître de jeu les envoie au serveur et lance la partie en envoyant la notification aux autres utilisateurs qui attendent et tout le monde se retrouve sur la dernière activité.

- La dernière activité, elle, va afficher la map et, grâce au système de socket, va récupérer toutes les informations de la partie (nœuds , joueurs , teams créés par le serveur, etc...). Une fois toutes les informations récupérées l'application les affiche sur la map. Grâce au système de socket et à un listener de Location d'Android, nous pouvons mettre à jours la localisation de l'utilisateur toutes les 500ms et, dans le même temps, demander au serveur de notifier les autres utilisateurs.

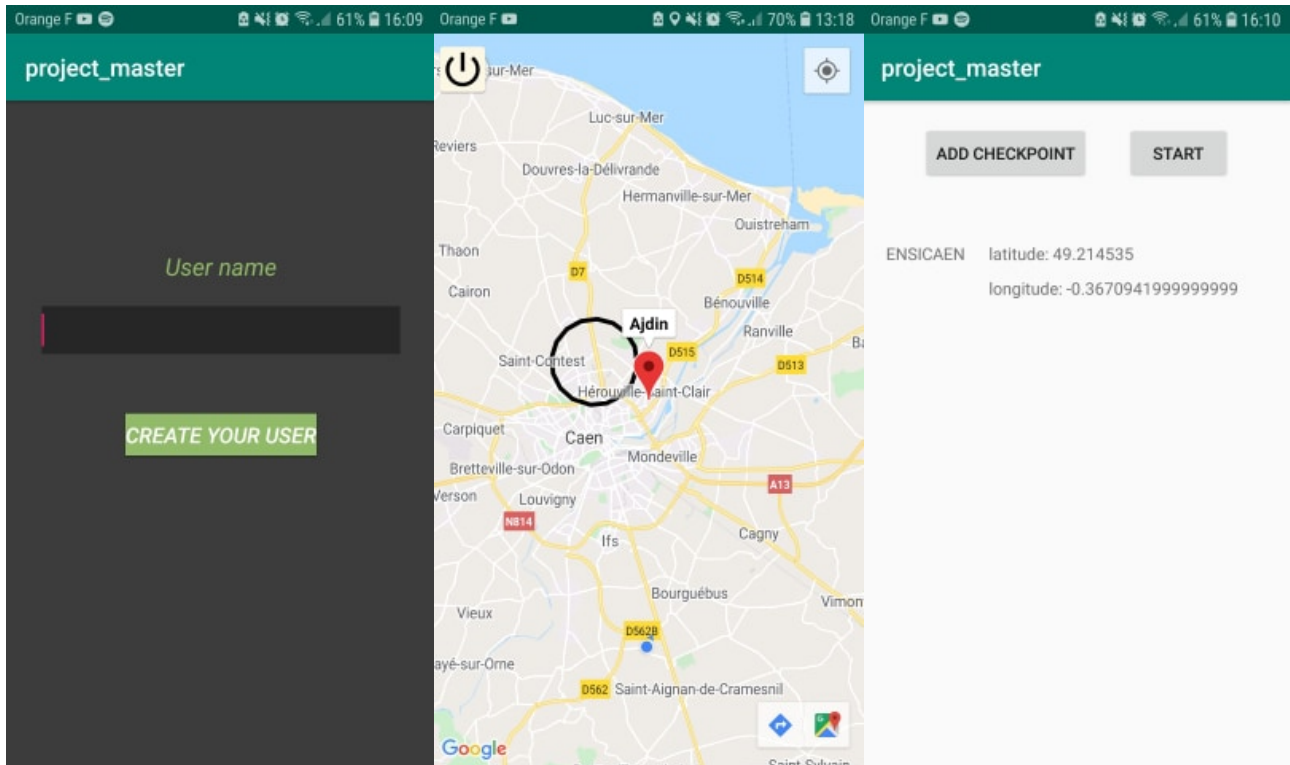


Illustration 13: Première activité

Illustration 14: Deuxième activité

Illustration 15: Dernière activité

3) Pattern Composite

Pour finir la présentation de l'architecture de notre application nous allons vous présenter le pattern composite que nous avons mis en place.

Nous avons notre classe Socket qui contient l'utilisateur, le context de l'activité pour pouvoir switcher entre les activités, l'URL et le port de notre serveur afin d'initialiser la connexion. Elle contient aussi 4 méthodes (initSocket, emitSocket, receiveSocket(), destroySocket()) qui sont réutilisées ou modifiées par toutes les autres classe qui héritent de celle-ci.

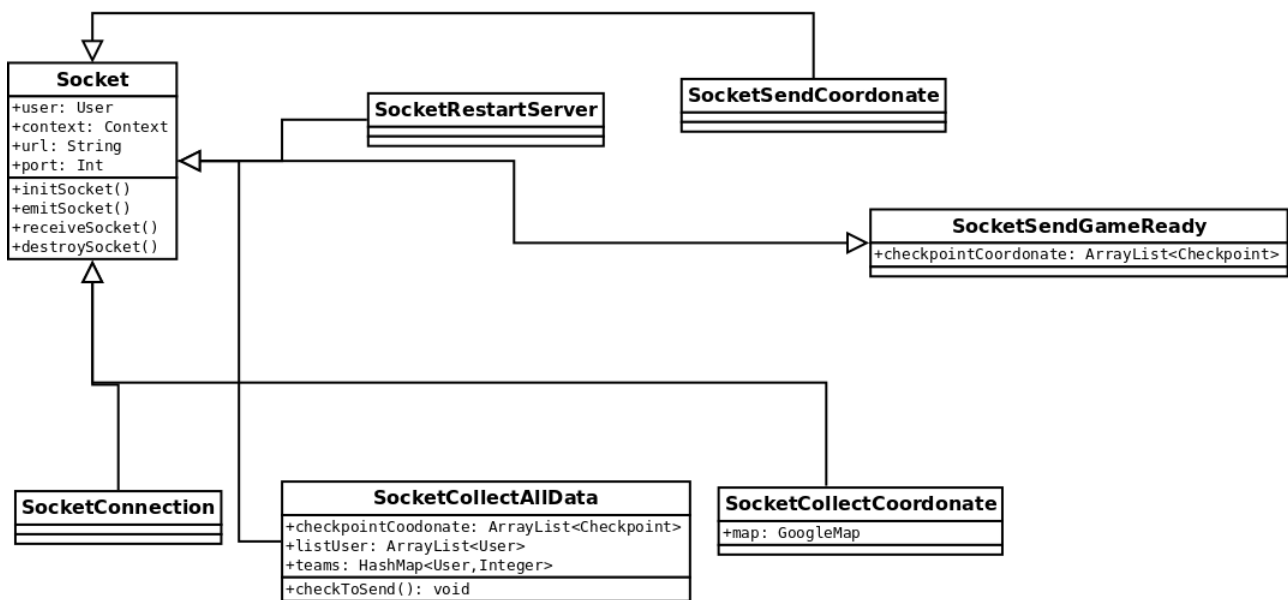


Illustration 16: Diagramme du pattern Composite

APPLICATION

Dans cette partie, nous allons présenter l'implémentation des différentes réalisations techniques à notre application. Celle-ci présente les caractéristiques suivantes : un utilisateur se déplace vers des nœuds pour les capturer, il peut voir les autres utilisateurs (allié ou ennemi), la capture de nœud d'une équipe se fait si le nombre de joueur d'une équipe est supérieure à celle de l'autre sur le nœud.

Ici pour afficher les utilisateurs on utilise des marqueurs, des objets de l'API Google Map de type « Marker ». Nous les créons en implémentant un MarkerOption() à qui nous donnons toutes les informations sur l'utilisateur (nom, coordonnée) :

```

MarkerOptions markerOptions= new MarkerOptions()
    .title(user.getUsername())
    .position(new LatLng(0,0));
Marker marker=map.addMarker(markerOptions);

```

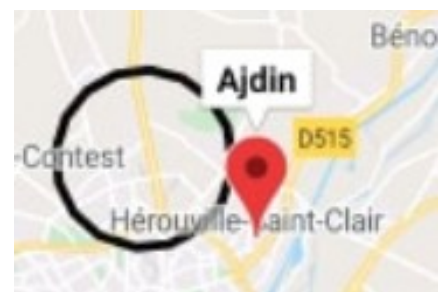


Illustration 17: Marqueur de position d'un autre joueur

Nous pouvons modifier la couleur du marqueur grâce à la méthode icon() :

```

markerOptions.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory
    .HUE_BLUE));

```


Pour les cercles qui définissent la zone d'un nœud, nous procédons de la même manière à l'exception que nous lui rajoutons un rayon :

```
CircleOptions circleOptions= new CircleOptions()  
    .center(newLatLng(checkpoint.getCoordonate().getLatitude(),checkpoint.getCoordonate().getLongitude()))  
    .radius(2000)  
    .strokeWidth(10)  
    .strokeColor(Color.BLACK)  
    .fillColor(Color.TRANSPARENT);
```

Par défaut, la couleur interne du nœud est transparent et devient bleu ou rouge selon l'équipe qui la capturé.



Illustration 18: Exemple de capture de nœud par l'équipe allié

Pour finir nous avons ajouté une petite icône en haut a gauche qui permet de fermer proprement l'application tout en réinitialisant le serveur. Malheureusement nous avons pas encore réglé la réinitialisation lorsque l'application se ferme subitement car les sockets n'ont pas le temps d'envoyer un message au serveur.



Illustration 19: Bouton de fermeture de l'application

VALIDATION

Au fil de l'avancement de notre projet, nous avons été amenés à effectuer diverses tests afin de vérifier le bon fonctionnement des éléments implémentés. Les premiers tests effectués ont servi à vérifier le bon fonctionnement du serveur et donc du côté multi-utilisateurs de l'application. Pour cela, nous nous sommes regroupés à plusieurs aux alentours de l'université du campus 2. Ces premiers tests furent une réussite. On pouvait observer la position de tous les utilisateurs sur chaque appareil.

Pour les tests suivants, nous avons eu quelques difficultés. En effet, suite à la fermeture des établissements scolaires décrétée le jeudi 12 Mars, nous avons eu plus de difficulté à nous retrouver afin d'effectuer d'autres tests et nous n'avons pas pu non plus participer à la journée de validation, initialement prévue la semaine du 16 Mars, qui devait être l'occasion pour nous d'effectuer des tests grandeur nature. De plus, suite au confinement total de la population Française dû au COVID-19, qui a pris effet le mardi 17 Mars, nous avons pris un peu de retard et nous n'avons pas été en mesure de réaliser d'avantages de tests correctement dû au fait principalement que notre application est en réalité augmentée orientée localisation.

Cependant, nous avons fini par trouver des moyens peu traditionnels dans le but de continuer l'avancement du projet et surveiller le bon fonctionnement de celui-ci. Nous avons d'abord commencé à nous envoyer les différentes versions du jeu afin de les tester chacun de notre côté et observer les différentes anomalies. Puis, lorsque nous avons voulu tester la fonctionnalité des équipes, nous avons eu recours à des sessions de tests avec des camarades de classe sur les réseaux sociaux (à savoir Discord) en envoyant le fichier .apk (fichier permettant l'installation de l'application sous Android) à tous. Bien que nous ne pouvions pas nous déplacer nous pouvions observer le fonctionnement de base des éléments ajoutés. Nous avons également sollicité nos familles en leur demandant leurs appareils ou en les faisant participer. C'est donc dans ces conditions que nous avons pu tester les fonctionnalités d'équipe et de prises de camp par équipe.

Malheureusement, nous n'avons pas pu valider le bon fonctionnement complet de l'application sachant que nous ne pouvions pas nous déplacer. Nous n'avons pas pu non plus effectuer une vidéo de démonstration de l'application car, en l'état, celle-ci aurait été peu pertinente. En revanche, les fonctionnalités de base ont bien été validées, même si les moyens d'y parvenir était un peu original.

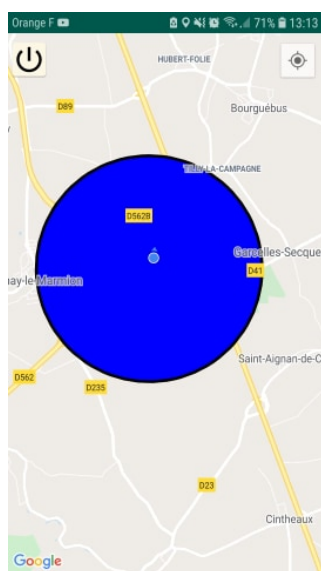


Illustration 20: Image de capture de camps



Illustration 21: Image montrant plusieurs utilisateurs et un camps non capturé

CONCLUSION

Dans ce projet, nous nous sommes fixés comme objectif d'implémenter un système de réalité augmentée orientée localisation multi-utilisateurs. Pour cela, nous avons développé une application avec les caractéristiques suivantes :

- elle se déploie sur Android
- elle ajoute des éléments virtuels à des positions précises
- elle donne un moyen d'interaction avec ces objets
- elle récupère la position des utilisateurs et la partage avec d'autres

Afin de permettre une application ayant la caractéristique multi-utilisateur, nous avons mis en place un serveur, ayant un DNS publique, qui va gérer les informations des différents utilisateurs afin de les envoyer aux autres. La communication entre les appareils des utilisateurs et le serveur s'effectue via un système de socket. La carte et la gestion des coordonnées des adresses, noms de lieu, etc.. s'effectue via les services Google Map et Places.

L'application développée est un jeu de capture de nœuds/camps. Ce jeu propose un affrontement entre deux équipes dont l'objectif est de capturer plus de nœuds que l'équipe adverse dans un temps imparti. Afin de gagner la partie, les joueurs devront communiquer entre eux et établir des stratégies. Cette application correspond bien aux attentes fixées par la problématique du projet, à savoir :

- de la réalité augmentée orientée localisation
- un système multi-utilisateurs

Elle y répond dans le cas où c'est bien une application multi-utilisateurs, elle ajoute des éléments virtuels (les nœuds) sur des emplacements précis et les déplacements des joueurs s'effectuent par géolocalisation.

Nous aurions aimé ajouter quelques petites fonctionnalités, comme par exemple un système de chat afin de permettre aux utilisateurs de communiquer plus facilement entre eux, mais nous n'en avons pas eu le temps principalement pour les raisons évoquées précédemment.

Cependant, nous avons remarqué qu'il était possible de combiner notre application avec celle d'un autre groupe dont la problématique se rapproche de la notre à la différence que leur problématique est orientée image. En effet, nous aurions pu intégrer leur application à la notre, comme par exemple pour augmenter les interactions entre joueurs lors de la capture d'un nœud en ajoutant un système de réalité augmentée orientée image.

À la différence des applications déjà disponibles, notre application fournit aux utilisateurs un intérêt social plus poussé. Ils doivent effectivement communiquer entre eux afin de battre l'équipe adverse et devront parfois même se regrouper à un endroit précis dans le but de capturer un nœud.

Ce projet nous a permis de développer tout un panel de compétences. Ces compétences ont avant tout été techniques, comme pour la création d'un serveur et sa gestion ou encore lors du développement de l'application Android, mais elles ont aussi été d'ordre social ou d'organisation. En effet, nous avons été amenés à travailler en équipe, que ce soit au sein de notre groupe ou avec d'autres groupes, à présenter nos projets devant diverses personnes, et ainsi développer des compétences en communication, ou encore lors de l'organisation du projet. Mais nous avons surtout appris un nouveau point de vue pour ce qui est de l'interprétation d'un projet et sa réalisation.

REMERCIEMENT

Nous tenons avant tout à remercier M. Vanhee pour nous avoir permis de réaliser ce projet et de nous avoir guidé tout au long de celui-ci. Il nous a apporté les pistes de réflexion nécessaires au moment où nous en avions le plus besoin. Il nous a également permis de percevoir différemment la manière de réaliser un projet.

Nous remercions également tous nos collègues et les membres de nos familles qui nous ont aidés lors de nos sessions de tests et qui nous ont apporté des retours objectifs sur le fonctionnement de notre application.

Nous souhaiterions également remercier Julien Martinez qui nous a permis de trouver une solution à nos problèmes concernant l'hébergement des serveurs et aussi pour avoir participé à nos sessions de tests.