

CAHIER DES CHARGES

Projet développement Smartphone - Dépôt 3

1. Préambule
1.1 Présentation de l'équipe
1.2 Objectifs
1.3 IDE
1.4 Versionning
1.5 Organisation
-- dépôt 2 --
2. AdaptationV1
3. Déroulement du projet
3.1 Vue 1
-- dépôt 3 --
4. AdaptationV2
5. Introduction
6. Déroulement du projetV2
6.1 Vue 2
6.2 Vue 3
7. Conclusion final

1.1 - Présentation de l'équipe.

FRANCOIS Nicolas : Chef de projet, Développeur.

VASSEUR Sylvain : Scrum Master, Développeur.

AIRAUT Jérémy : Lead Développeur.

JUDAIS MORGAN : Développeur.

1.2 - Objectifs

Le projet consiste à développé une application mobile Android, qui nous permet de contrôler un drone marin, tout en respectant un délai de rendu sur plusieurs dates différentes. L'objectif de ce projet, grâce à notre application est de pouvoir représenter la trajectoire du drone à partir de données NMEA (simulateur drone), piloter manuellement le drone via notre application (boutons : accéléromètres, urgence, home...) et pour finir, pouvoir sauvegarder la trajectoire du drone.

1.3 - IDE

Nous avons décidé de concevoir ce projet sous Androïd. Au vu des événements (Covid-19), nous nous sommes posés la question de ce qui était le mieux. Pour le télétravail, android nous semble le plus adapté pour coder (pas besoin de VM MacOS, pas de PC Mac). Nous allons donc utiliser pour développer sur Android : *Android Studio*

1.4 - Versioning

Afin de travailler et développer ensemble, nous allons utiliser *GitHub*. Ce dernier nous permettra plus facilement de s'échanger les codes, ressources ...
Pour interagir ensemble, nous utilisons *Discord*. L'avantage est de pouvoir discuter et créer des groupes afin de séparer les différents sujets à traiter.

1.5 - Organisation

Le projet va utiliser la méthode agile *Scrum*. Cette méthode permet de mieux comprendre chaque aspect du projet, de mieux s'orienter et de voir l'avancée du projet.
Elle permet d'avoir une bonne communication dans l'équipe grâce à la réalisation de sprint où chacun pourra présenter son travail et interagir avec le reste de l'équipe sur certaines difficultés ou différents changements dans le projet.
Pour cette organisation, nous allons utiliser *Trello* (visualisation et répartition des tâches).

— — début du dépôt 2 — —

2. AdaptationV1

Suite au retour sur notre premier rendu, nous avons bien pris en compte vos conseils et modifiés quelques éléments.

Premièrement, le changement se trouve sur le *Trello*. Nous avons enlevé les tâches générales pour les remplacer par des tâches plus simples réalisables du coup plus rapidement.

Concernant la colonne « à Vérifier », nous justifions la présence de celle-ci car pour nous, la transition de « En développement » à « Terminer » était trop importante. De plus, l'ajout de cette colonne intermédiaire correspond bien avec les retours qui nous sont fait.

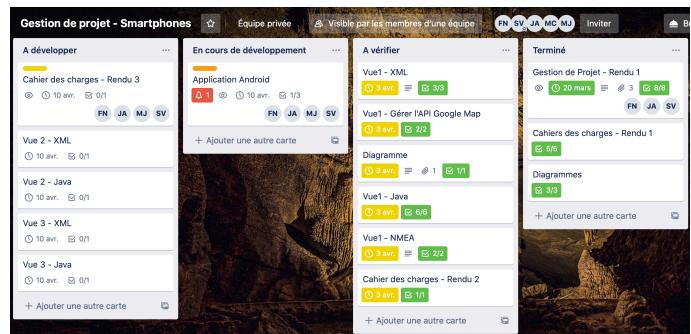
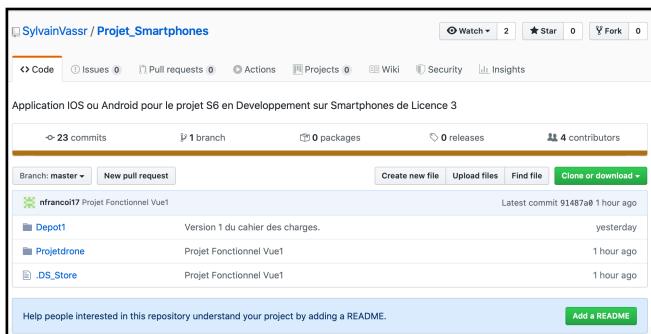
Concernant le GANTT, les tâches ont été assignées avec les codes couleurs, elles-mêmes associés avec un des membres du groupe dans la légende.

Pour le Github, il nous manque juste votre identifiant afin de vous ajouter. J'ai essayé, en vain, de retrouver là où vous l'avez écrit lors de la première visioconférence.

« Nous vous remercions pour les retours positifs, mais aussi négatif qui permettent l'avancement du projet. »

Lien GitHub : https://github.com/SylvainVassr/Projet_Smartphones

Lien Trello : <https://trello.com/b/1GeUVHaP/gestion-de-projet-smartphones>



3. Déroulement du projet

Pour ce deuxième rendu, nous nous sommes focalisés sur la première vue, qui est pour nous la plus complexe car celle-ci est associée avec le simulateur NMEA. De plus celle-ci nous permettait de gérer la map est qui logiquement présentes sur nos autres vues.

3.1 Vue 1

Dans un premier temps, nous avons créé le fichier XML de la première vue. Pour cela nous sommes partis d'un nouveau projet Google Map activity. De ce fait la gestion de la map était donc facilitée.

Ensuite, nous avons pu parfaitement intégrer les différents boutons, et TextView dans notre première vue.

Les boutons permettant de changer de vue ne sont pour l'instant pas implémentés (*les boutons Waypoints et Pilotage*). Nous pouvons donc rester que sur la première vue pour ce dépôt.

Nous avons essayé de respecter au mieux le design de notre vue que nous vous avions fournis dans le dépôt 1. (Voir le schéma du téléphone vue 1 dans le dépôt 1).



(voici les données récupérés du NMEA)



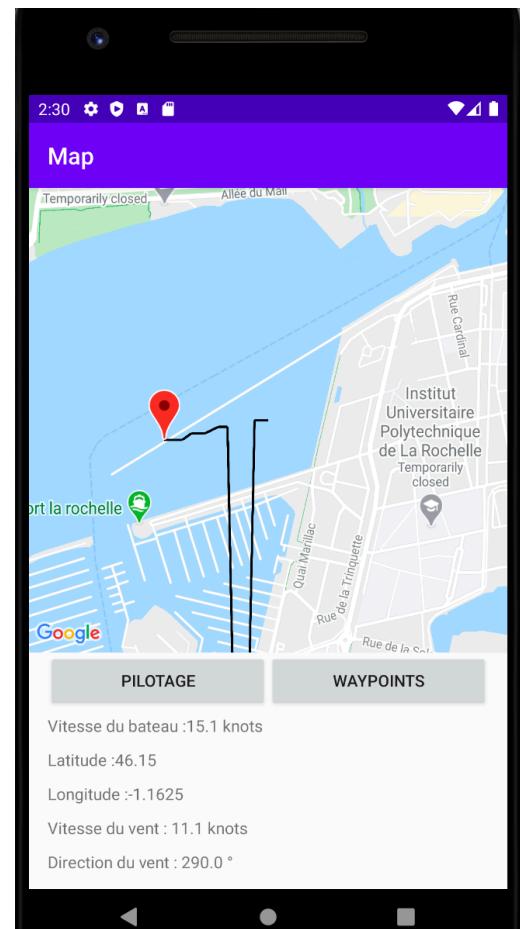
(voici les boutons pour les vues 2 et 3)

Nous nous sommes ensuite penchés sur la connexion avec le simulateur. Pour ce faire, nous avons créé un *Socket* et un *BufferedReader* pour lire les lignes qui résulte de la connexion entre le simulateur et le NMEA.

Enfin nous avons développé les fonctions manquantes nécessaires pour le bon fonctionnement de la première vue.

Nous avons donc implémenté le parseur, qui traite la ligne reçue (seulement si elle est en GPRMC) et récupère les valeurs.

Cependant, certaines valeurs avaient besoin d'une conversion, nous avons donc du implémenter deux autres fonctions, Longitude/Latitude au format décimal.



(voici la trajectoire du drone en noir)

Une fois fais, nous avons donc dû faire la fonction `printLine` qui affiche le déplacement du drone selon les coordonnées récupérés.

La réalisation de cette étape n'a pas été non sans mal, puisque nous avons été confrontés à pas mal de problème :

- Tout d'abord, l'IDE utilisés Android studio est très gourmand est à des difficultés à fonctionner chez deux membres du groupe sur quatre.
- Nous avons aussi rencontré des problèmes avec GitHub, aucun des membres de notre groupe n'étant familier avec ce logiciel. (*conversation audio avec les professeurs sur Discord pour résoudre et comprendre nos problèmes*).
- Un des problème a été pour nous la map. En effet nous avons eu énormément de soucis avec l'API nous empêchant de voir la carte et donc d'avancer sur plusieurs aspect du code en parallèles.

Nous sommes assez satisfait du rendu de la vue 1. Nous pouvons voir quelle est fonctionnelle, nous arrivons à récupérer toute les données, ainsi qu'un tracé en temps réel du drone.

Néanmoins un problème persiste. Le point de départ du drone est très superflu, c'est à dire qu'il se positionne que ce soit dans la mer ou bien, sur la terre. Nous arrivons pas à fixer sa position.

Une information assez utile si vous voulez tester le projet, nous devons configurer le NMEA avec des sorties toute les 5 secondes. Toute les 1 secondes le simulateur ainsi que notre application à du mal à tourner.

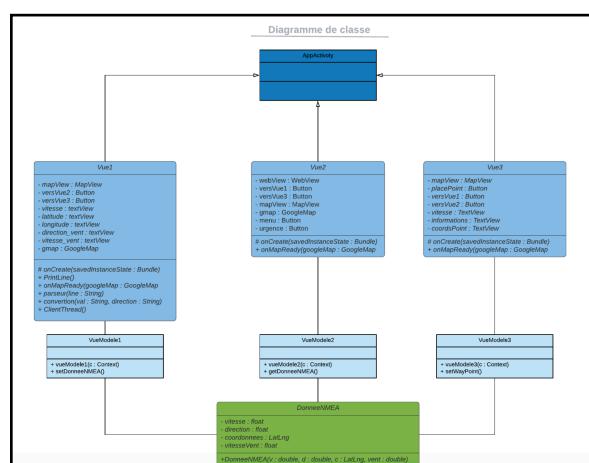


— — début du dépôt 3 — —

4. AdaptationV2

Lors de ce dernier rendu, nous avons pris encore une fois en compte vos remarques. Nous avons modifié en haut de ce rendu (page 3), les images par rapport à notre vue1. En effet comme vous l'avez demandé nous avons réalisé plus de screen de notre application pour que vous ayez une meilleure vision de notre projet.

Pour finir, pour le diagramme nous avions eu remarque comme quoi il était étrange d'avoir un modèle par vue. Nous allons essayé de vous répondre succinctement à cela : En effet, il aurait été possible de faire une vue commune. En revanche pour nous, nous avions une idée précise pour chaque vue. C'est pour cela que nous avons préféré développer chaque vue.



5. Introduction

Pour ce troisième et dernier rendu, nous nous sommes focalisés sur la deuxième et troisième vue, mais aussi sur la finalisation d'ensemble du projet.
Nous allons voir comment nous l'avons conçu, ainsi que nos différents avis sur ce projet, ce que cela nous a apportés, notre vision ...

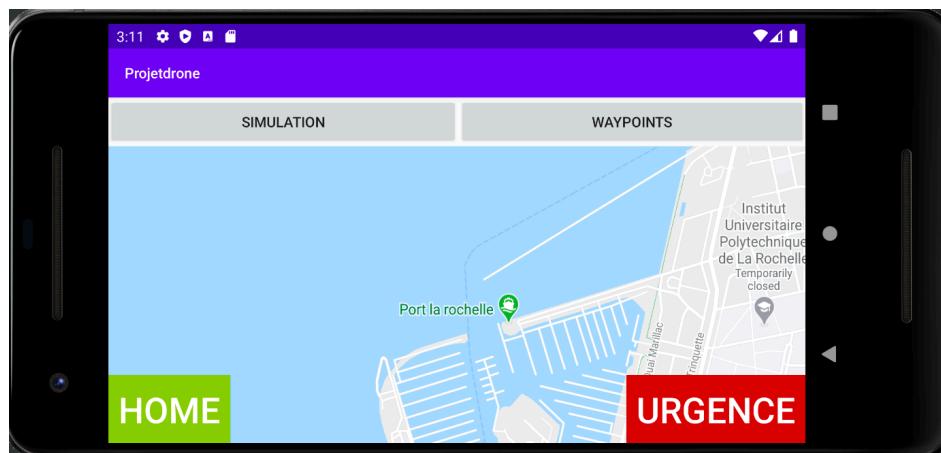
6. Déroulement du projetV2

6.1 Vue 2

Pour cette deuxième vue, nous avons du nous regrouper pour pas mal échanger. En effet cette vue nous a posé problème. Nous avions pas une vision claire, et nous savions pas totalement quel était le résultat attendu.

Nous avons du passé pas mal de temps à chercher une solution pour que cette vue soit fonctionnelle.

Une fois la réflexion faite, nous sommes passés tous ensemble à la réalisation. D'une part nous avons réalisé le XML, pour que la vue ressemble à notre schéma de notre premier rapport (voir dépôt 1), mais nous avons du aussi implémenter le bouton de la vue 1 pour justement accéder à cette vue.



(voici un screen sur l'affichage XML de la vue)

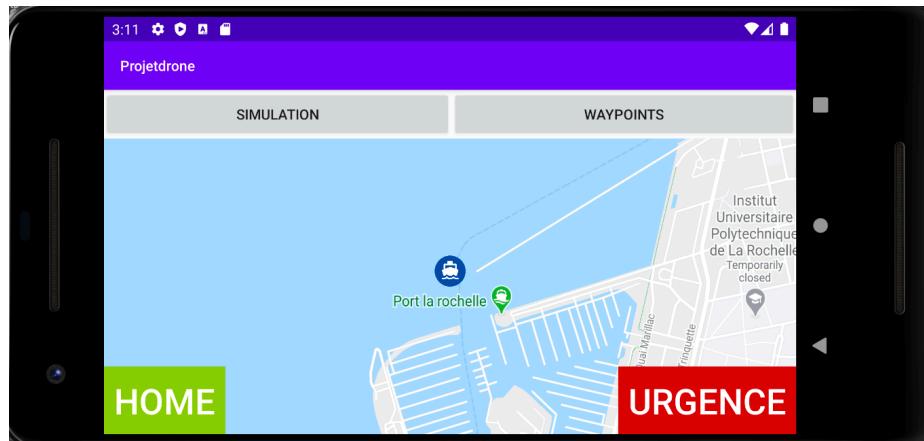
Ensuite, une fois le XML et l'implémentation faites, nous nous sommes penché sur l'aspect pilotage. Cette vue contient un bouton HOME qui permettra au drone de revenir à la position initiale ou nous l'avons lancé.

Le bouton urgence lui, permettra de tout arrêter justement en cas de problème.

Nous avons pris pas mal de temps sur la gestion des capteurs, (l'accéléromètre), n'ayant pas une idée précise, cela a été compliqué de réalisé cette vue.

Nous avons du créer plusieurs fonctions dont une onSensorChanged, cette fonction est appelée lors d'un changement détecté sur les capteurs. Une autre fonction qu'il a fallu implémenté, updateLocation qui est appelé à la fin du traitement du capteur afin de modifier la position du drone sur la carte en fonction du nouvel attribut this.location.

Voici sur la page suivante le screen de la position de départ du drone :



6.2 Vue 3

Pour la troisième vue (affilier au bouton WAYPOINTS) , le but était que l'utilisateur puisse tracer le chemin que suivrait le drone. Pour cela il fallait dans un premier temps récupérer les points par lesquels devaient passer le drone, puis les transformer en trame GPRMC. Pour ce faire, nous avons réalisé 2 fonctions importante, drawLines et transformIntoNMEA.

Aucun souci concernant le tracé, il fallait juste faire attention à ce qu'on ait 2 points pour afficher le chemin que devait suivre le drone. Pour cela nous avons stocké tous les points entrées par l'utilisateur dans une liste.

Le plus gros du travail a concerné la seconde fonction. En effet, dès qu'un point était rentré, nous voulions avoir sa trame GPRMC correspondante afin que ce même chemin puisse être lu et compris par une autre NMEA. Il nous a fallu dans un premier temps comprendre à quoi correspondait chaque valeur d'une ligne du format GPRMC.

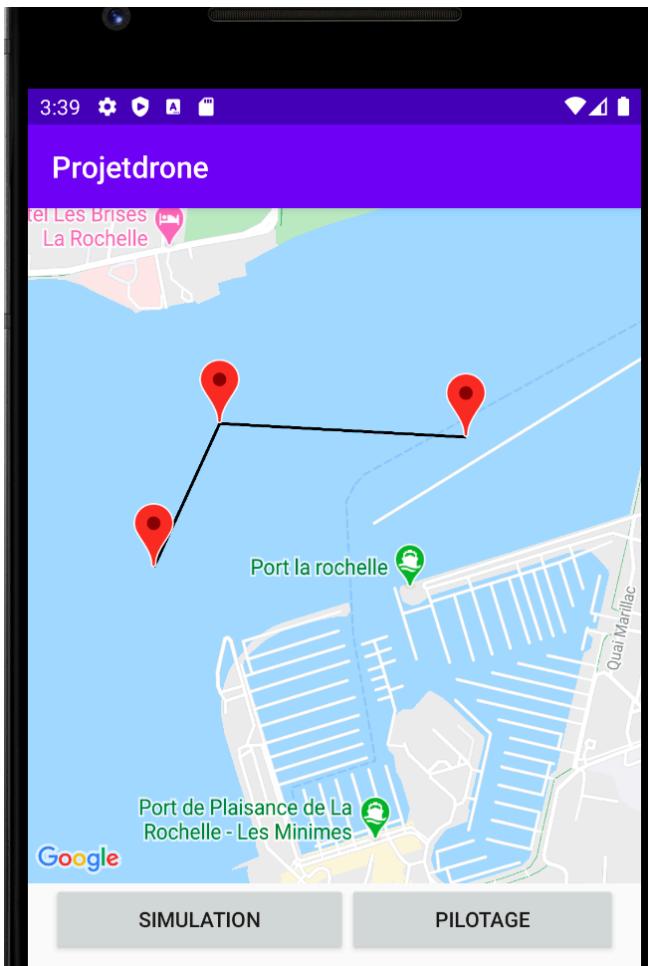
Après un peu de recherche nous avions ce qu'il nous fallait :

eg2. \$GPRMC,225446,A,4916.45,N,12311.12,W,000.5,054.7,191194,020.3,E*68

225446	Time of fix 22:54:46 UTC
A	Navigation receiver warning A = OK, V = warning
4916.45,N	Latitude 49 deg. 16.45 min North
12311.12,W	Longitude 123 deg. 11.12 min West
000.5	Speed over ground, Knots
054.7	Course Made Good, True
191194	Date of fix 19 November 1994
020.3,E	Magnetic variation 20.3 deg East
*68	mandatory checksum

Il ne nous restait plus qu'à récupérer chaque valeurs et les insérées dans ce qui serait notre futur trame. Cependant, un problème est survenu sur les coordonnés. Lorsque nous les rentrions sur un GPRMC converter, celles-ci n'indiquaient pas La Rochelle, mais une ville un peu plus à l'Est. L'erreur se situait donc au niveau des Latitude, Longitude. En effet nous aurions dû être plus attentif, car selon le signe, la direction n'est pas la même.

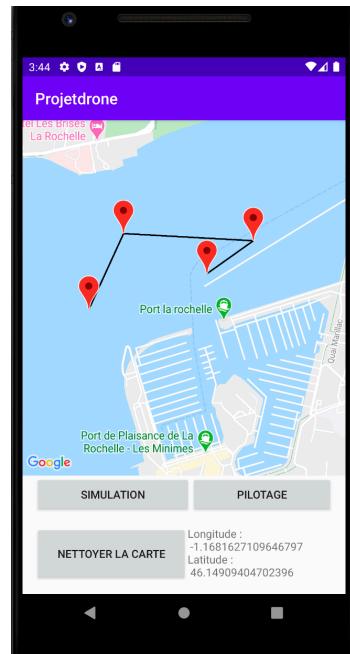
Voici des screens de cette vue 3, pour voir plus précisément le travail effectué.



(voici un screen de la map avec des points placés)



(voici un screen du bas de la vue pour nettoyer tout les points, ainsi que l'affichage des coordonnées du point que l'on place sur la map)



(voici un screen de la vue globale)

```
Accessing hidden method Lsun/misc/Unsafe;=>get0bject(Ljava/lang/Object;)
W/_lr.projetdrone: Accessing hidden method Lsun/misc/Unsafe;=>put0bject(Ljava/lang/Object;Ljava/lang/Object;)
W/_lr.projetdrone: Accessing hidden method Lsun/misc/Unsafe;=>put0bject(Ljava/lang/Object;Ljava/lang/Object;)
D/NMEA: $GPRMC,154003,,4608.9456,N,-0110.0897,E,30,,090420,,E*6E
```

(voici un screen où l'on voient la trame GPRMC qui se créer lorsque l'on place un point)

Nous sommes satisfaits du résultat obtenu pour cette vue 3, cependant nous aurions pu développer un bouton afin de supprimer le dernier point entrer par l'utilisateur pour gérer une éventuelle erreur, ce qui aurait rendu notre application plus portative. Excepté ce point-là, le résultat est conforme à nos attentes.

7. Conclusion final

Au début de ce projet, nous avions pas mal d'incertitude. Peur de mal faire, peur de ne pas réussir. Au fil du temps, nous avons réussi à former un groupe soudé, chaque membre du groupe s'attelait au travail qu'il devait faire. Aucun membre du groupe n'a lâché, au contraire, lorsque quelqu'un eu un problème, nous nous penchions tous dessus afin de le débloquer pour ensuite réussir à avancer tous ensemble.

Ce projet, dans l'ensemble, nous en sommes fiers, car nous jugeons notre projet à la hauteur de nos attentes. Merci de vos disponibilités qui nous on permis de nous aiguiller tout au long du projet.

FRANCOIS Nicolas, VASSEUR Sylvain, AIRAULT Jérémy, JUDAIS Morgan.