**Projet en programmation java**

**Conception et développement d’une application informatique.**

Rapport Synthèse :

Comment s’est passé le projet :

Le projet s’est globalement bien déroulé, avec une bonne répartition des tâches et des problèmes. Nous avons mis du temps à démarrer après avoir fait les diagrammes UML car nous avions des problèmes sur le code source, le fichier pom.xml n’était pas bon. Ensuite, nous avons passé la plupart de notre temps sur la partie interface graphique. La partie sur OpenStreetMap et le calcul d’itinéraire de l’IGN nous à posé de petits problèmes mais nous n’y sommes jamais resté bloqué plus d’une journée. Quand problème il y avait, comme un manque de connaissances sur une méthode ou sur une formule, nous avons trouvé des solutions sur internet, notamment sur stackoverflow ou des blogs prévus à l’effet.

Synthèse :

Partie sur OpenStreetMap :

Le principal problème de cette partie à été au début de refaire une classe avec une méthode qui ressortait la liste des noms des toponymes et leur coordonnées car elle était de base dans le main de la classe MainExempleOpenStreetMap. Ne comptant pas réinventer la roue, cette partie est accompagnée de la classe Repère qui contient un calcul de distance entre deux coordonnées en utilisant la formule d’haversine directement trouvé sur stackoverflow : <https://stackoverflow.com/questions/14560999/using-the-haversine-formula-in-javascript>

L’accessibilité aux instructions, et ainsi à la recherche de toponymes aux alentours a grandement impacter la vitesse d’exécution du programme. Nous avons donc dû faire le choix de passer cette sélection pour certaines étapes peu importantes telles que les “Continuez tout droit”. Nous avons donc dû faire un choix au profit de l’efficacité du programme.

Nous avons aussi voulu utiliser le thème bati d’OSM pour trier les toponymes. Après beaucoup de recherches sur le langage xml et en se documentant avec les liens fournis par le sujet. Nous avons décidé de ne pas trop nous attarder dessus pour ne pas perdre trop de temps.

Concernant l’itinéraire / le service IGN

En comparaison avec notre analyse, on se retrouve finalement avec des classes plutôt bien prédites mais avec beaucoup plus de méthodes, de détails à gérer. Après la primordiale et longue étude du JSon, nous avons plutôt bien réussi à obtenir tout ce que nous voulions au bout d’un certain moment selon le problème. Nous nous sommes ensuite par exemple heurtés au problème de donner la bonne instruction à l’utilisateur : droite, devant ou gauche accompagnée des précisions avant ou après selon la position du repère choisi. Il fallait donc toujours rester cohérent et bien garder en tête ce que nous avions, ce que nous voulions et ce que nous recevions de chaque classe, de chaque méthode ou de chaque formule que l’on créait ou utilisait.

Pour le portail / MapPanel

On retrouve dans cette partie probablement une des grandes difficultés de la mise en bon fonctionnement de l’application : tracer sur le MapPanel à notre guise. Après une longue analyse du MapPanel, cela a été réalisable même si comprendre la méthode PaintComponent et faire le lien entre les coordonnées du MapPanel et les coordonnées de la fenêtre n’a pas été simple. La particularité du PaintComponent à être appelé à chaque “Event”, nous a posé des difficultés, notamment le fait de devoir alléger au maximum cette méthode pourtant essentielle.

On s’est retrouvé face à une belle liste de problèmes ou de souhaits et donc de solutions :

Problème : pas de recherche gps

Solution : passage des étapes "à la main” à l’aide de boutons.

Souhait : afficher la liste des étapes ainsi que des instructions correspondantes dans un scrolling.

Problème : difficulté et lenteur d’exécution afin de retrouver toutes les instructions. (abandon du truc)

Souhait : pouvoir rentrer les coordonnées de départ et d'arrivée sans difficulté.

Solution 1 : afficher les coordonnées longitude et latitude de la souris sur le mapPanel

Solution 2 : ajouter à cela, la possibilité de rentrer automatiquement les coordonnées sans avoir à les taper dans les espaces dédiés.

Souhait : pouvoir tracer l’itinéraire afin d’avoir une représentation graphique.

Souhait : pouvoir se repérer sur la carte (quelle étape on est) même sans gps donc indication graphique sur le MapPanel de l’étape actuelle.

Souhait : interface graphique sobre et facile d’utilisation

A côté de cela, on a dû apprendre à utiliser Swing par exemple pour la création d’une interface graphique sans Builder car il était impossible d’en télécharger un. De plus, il a été très intéressant de finalement imaginer l’interface graphique. On se retrouve alors avec beaucoup de détails qui facilitent et sont utiles pour la bonne utilisation de l’utilisateur.

Comment s’utilise l’application :

Pour faire fonctionner l’application, il suffit de run la classe “main” dans le package “example”. L’application nous indique donc qu’il faut entrer des coordonnées de départ et d’arrivée, pour cela un bouton “modifier” est situé en haut à droite, celui-ci nous ouvre un panneau de configuration où l’on peut directement modifier les coordonnées en les rentrant dans les cases dédiées. Les coordonnées en longitude et latitude de votre souris sont affichées sur la droite de l’application. Ou alors une méthode bien plus simple et rapide, celle du remplissage automatiquement. Il vous suffira simplement de choisir un point sur la carte (une croix rouge apparaîtra) puis vous pourrez choisir de prendre ce point comme point de départ ou d’arrivée. Il ne vous restera plus qu’à valider vos coordonnées et l’itinéraire s’affichera automatiquement après quelques secondes d'attente.

Vous y verrez apparaître un nouveau panneau permettant de gérer les étapes du parcours. Vous pourrez passer à l’étape suivante ou revenir à celle d’avant. Des instructions vous indiqueront où vous diriger face aux intersections. Vous pourrez à tout moment changer les coordonnées de départ de d’arrivée grâce au même bouton “modifier” évoqué précédemment.