

Rapport de la SAE 2.01

I.] Présentation de l'application

Notre application consiste en deux parties :

- Une lecture de fichiers (des scénarii contenant chacun une liste de villes plus ou moins grande avec des noms de Pokémon dans chaque ville) qui doit être stockée dans une structure de données (peu importe laquelle). A partir de cela, faire le/les chemin(s) le/les plus court(s) afin de livrer toutes les personnes présentes dans le scénario choisi.
- Une interface graphique permettant à l'utilisateur de voir le scénario voulu, le fichier contenant les distances entre les différentes villes et les membres avec leurs villes.

II.] Qualité de développement

- Afin de se faciliter l'organisation du travail, nous avons décidé de le séparer en deux parties distinctes, l'un s'occuper du développement de l'application tandis que l'autre se charge des autres tâches comme la conception des diagrammes de classes, l'écriture du rapport final et du manuel utilisateur.
- Comme demandé dans les attendus, nous devons créer un dépôt centralisé via Git, pour notre part nous avons choisi GitHub Desktop car nous paraît plus simple d'utilisation.
- Pour une méthode de développement correcte, mon camarade a codé plusieurs méthodes intermédiaires pour permettre l'affichage du scénario et de son parcours. Il a aussi beaucoup utilisé de parcours de listes avec la boucle for et créé des variables en paramètres des méthodes importantes afin de ne pas avoir à chaque fois à rappeler les méthodes de lecture au sein de ces dernières.

III.] Diagramme de classes avec UML

Diagramme de classe de haut niveau :

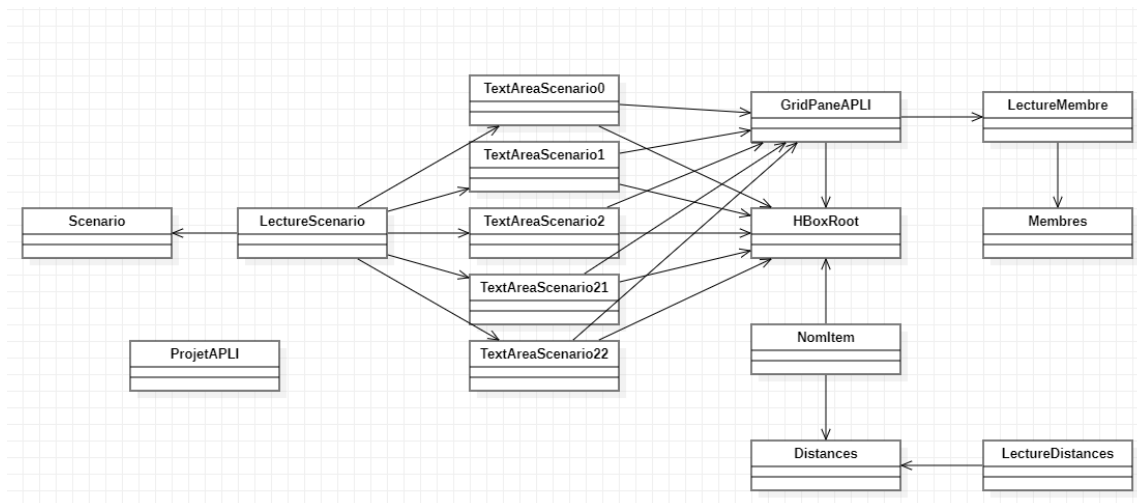
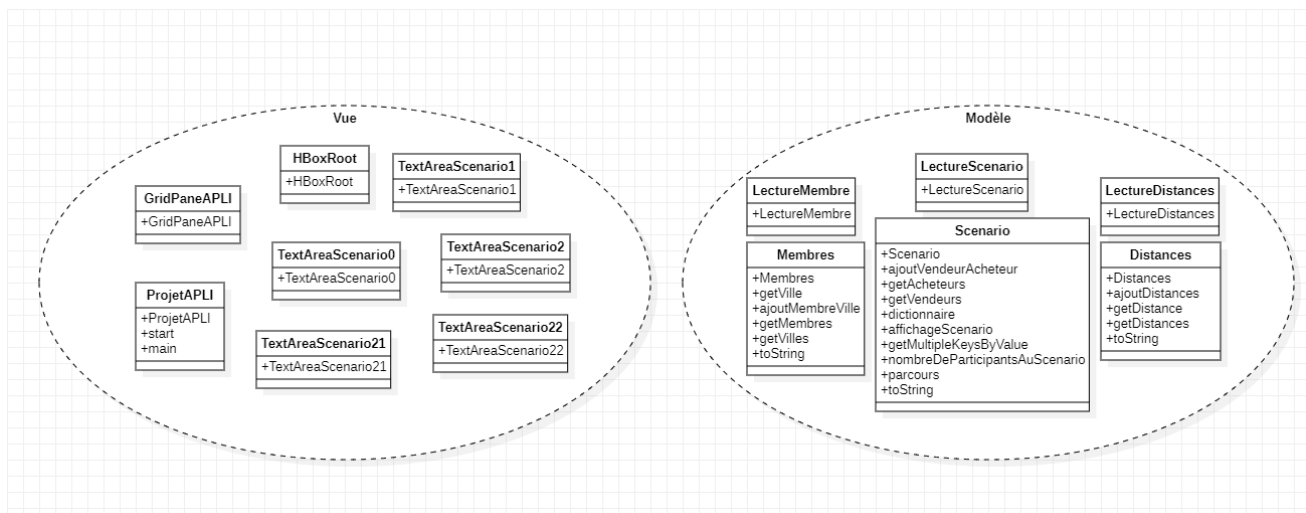


Diagramme de classes détaillé :



Pour les structures de données retenues, il y a les constantes, les variables, les listes et les maps.

Pour les stratégies algorithmiques utilisées se trouvent les parcours de listes et les maps contenant des boucles for et while.

IV.] Conclusion

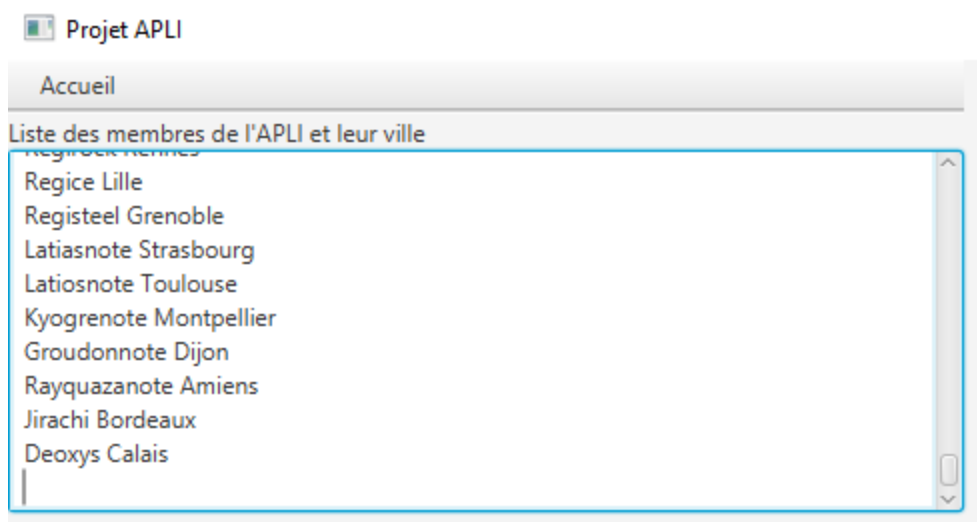
Comme tâches non réalisées, il y a le développement d'un contrôleur (**par manque de temps ?**), une partie CSS afin de donner à l'application un meilleur aspect utilisateur ainsi que les tests via JUnit. Comme perspective d'évolution, il peut y avoir la possibilité à l'utilisateur de créer son propre scénario ou bien de modifier un scénario déjà existant dans l'application.

V.] Annexes

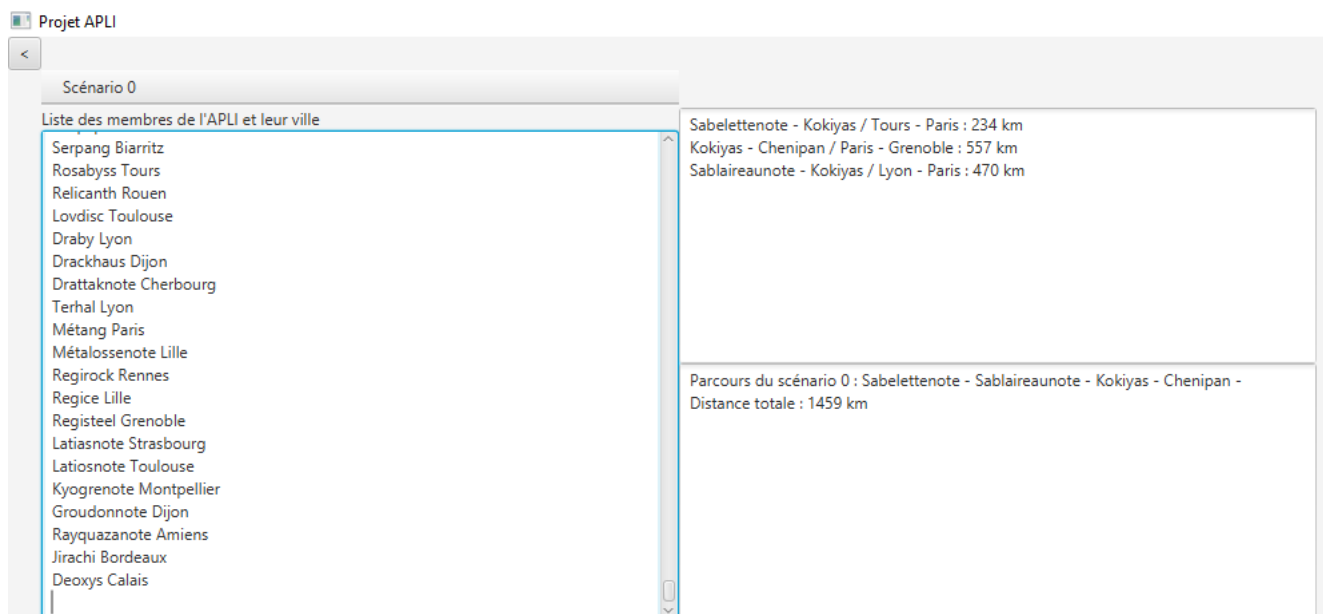
Lien Git : https://github.com/Kagzea0/sae_java

Manuel utilisateur :

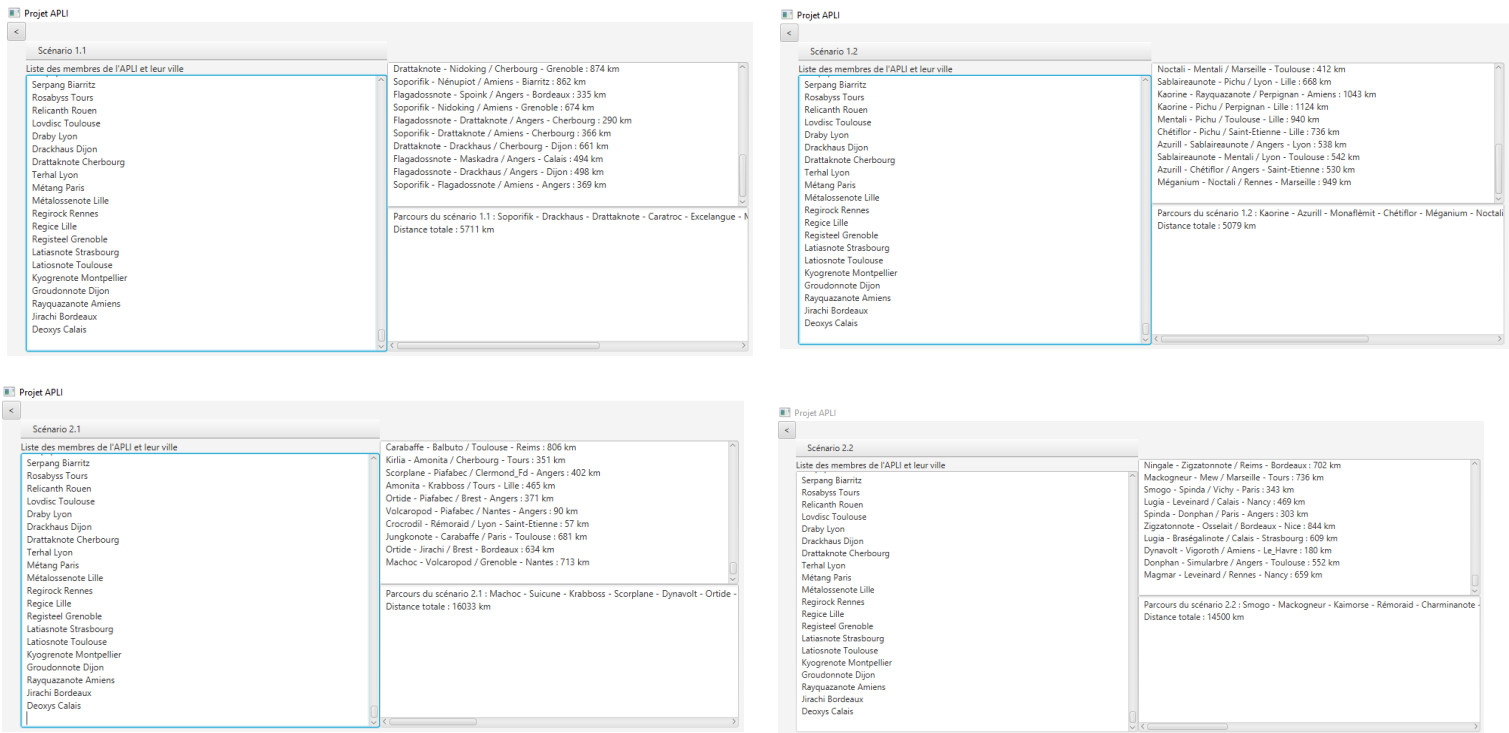
En entrant dans l'application, on se retrouve dans la page principale. En cliquant sur accueil, nous pouvons accéder aux différents scénarii proposés. Nous retrouvons aussi la liste des membres de l'APLI ainsi que leurs villes.



En cliquant sur le scénario 0, nous accédons à cette page. A gauche se trouve la même liste proposée à la page d'accueil. En haut à droite, il y a les villes par lesquels le président de l'association doit forcément passer avec la distance entre les différentes villes. Le parcours du scénario 0 est présent en bas à droite.



Pour toutes les pages contenant chacune un scénario, il y a la même disposition des éléments présents.



Maquette de l'application :

Les différents scénarios

Membres et villes :

- Bulbizarre Brest
- Herbizarre Perpignan
- Florizarrenote Brest
- Salamèche Lille
- Reptincel Biarritz
- Dracaufeunote Calais
- Carapuce Nice
- Carabaffe Toulouse
- Tortanknote Nancy
- Chenipan Grenoble
- Chrysacier Strasbourg
- Papilusion Clermond_Fd
- Aspicot Amiens
- Coconfort Cherbourg
- Dardagnannote Nantes
- Roucool Saint-Etienne
- Roucoups Nice

- 1- Sabelettenote - Kokiya / Tours - Paris : 234 km
- 2- Kokiya - Chenipan / Paris - Grenoble : 557 km
- 3- Sablaireunote - Kokiya / Lyon - Paris : 470 km