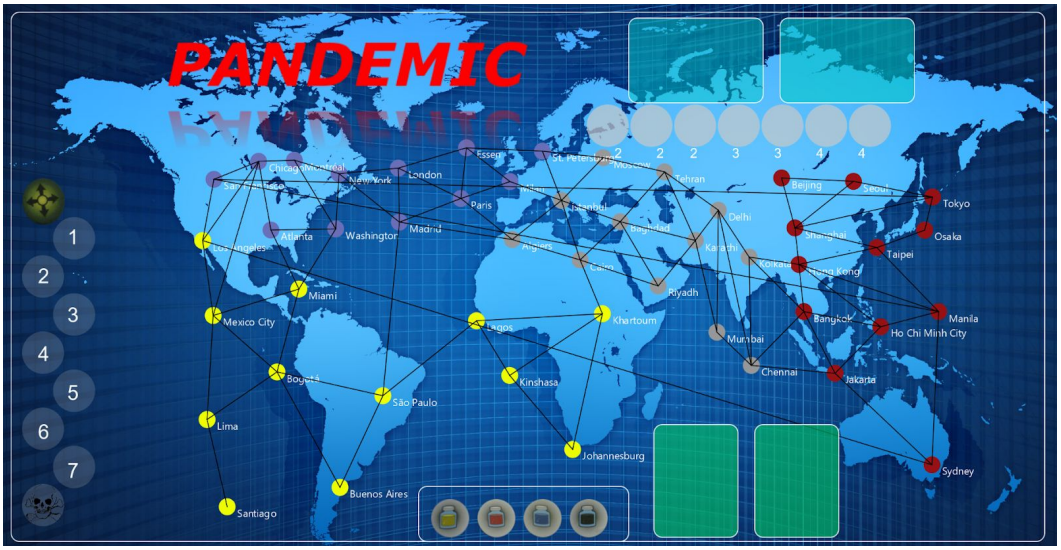
DAUPHINE
UNIVERSITÉ PARIS

JAVA :
documentation pour le développeur



M1 Informatique des Organisations - Année 2018/2019

Version 1.
Encadré par : Monsieur B. NEGREVERGNE
Date : 01 février 2019.

Paul VONG, Dyhia TOUAHRI, Shahram MAHDAVI

I. Diagramme de classe

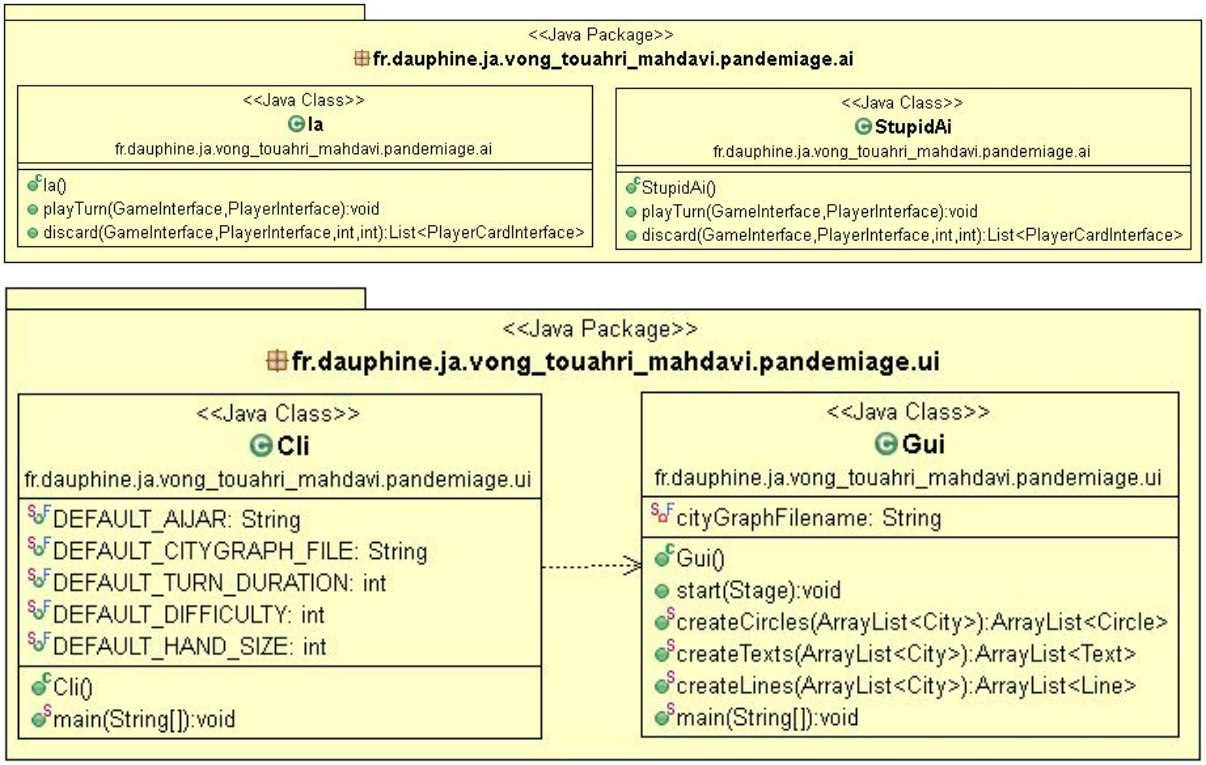
Choix de l'architecture :

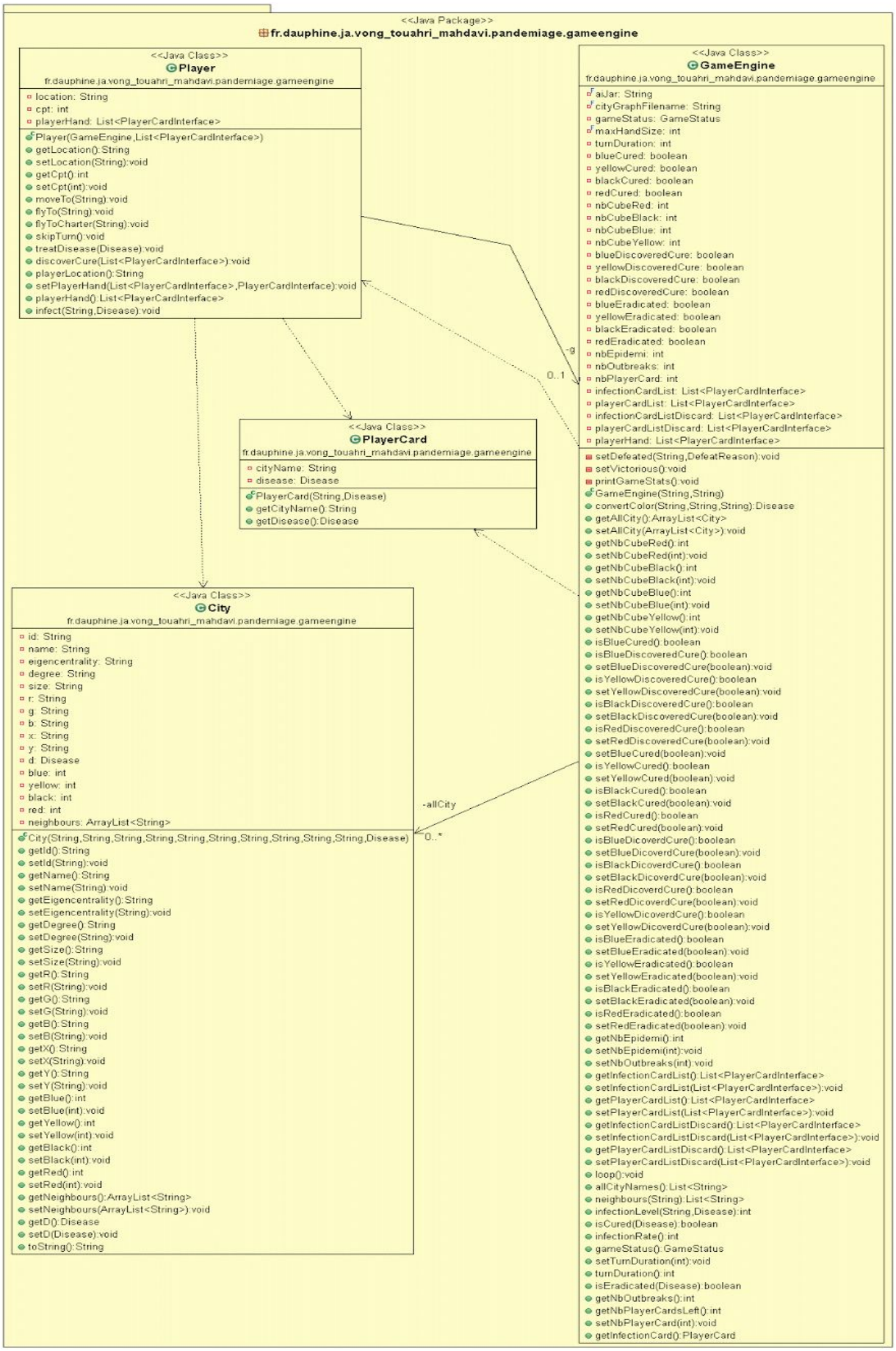
Pour faciliter la manipulation des données nous avons ajouté une class ***City*** où nous récupérons tout ce qui concerne une ville.

Nous avons ajouté aussi deux autres classes ***Player*** et ***PlayerCard***.

Player implémente ***PlayerInterface***, dans cette classe nous pouvons trouver toutes les actions que le joueur peut faire, notamment de jouer, se déplacer, découvrir un remède, passer son tour…

Quant à ***PlayerCard*** il s'agit de cartes que le joueur utilise, sur chaque carte se trouve le nom d'une ville et d'un remède, elles servent à se déplacer et à découvrir un remède, pour découvrir un remède il faudrait avoir 5 cartes du même remède.





Paul VONG, Dyhia TOUAHRI, Shahram MAHDAVI

II. Choix d'environnement et d'implémentation

Dans cette partie du rapport nous expliquons nos choix d'implémentations, les plateformes de travail et la façon dont nous avons abordé l'implémentation.

- **Outils utilisés et plateforme de travail :**

Principalement, nous avons travaillé sur Eclipse, Système d'exploitation : windows, linux et mac. Durant la réalisation du projet vu que nous nous sommes réparti les tâches, nous avions une plateforme sur GitHub où nous communiquions nos travaux.
Pour l'interface graphique nous avons travaillé avec JavaFX.

- **Choix d'ia :**

Pour le choix de l'IA, dans un premier temps nous avons implémenté une IA qui choisissait ses actions aléatoirements en fonctions de toutes les actions possibles, nous n'avons pas réussi à implémenter une IA avec des heuristiques.
Alors nous n'avons pu implémenter qu'une IA basique.

- **Implémentation du moteur de jeu :**

  Nous avons choisi d'implémenter que des cartes ***PlayerCard*** qui implémente l'interface ***PlayerCardInterface***, comme dans ***PlayerCard***, il y a déjà un nom de ville et de maladie, on s'est donc servi de ça pour faire les cartes d'infections.

  Cependant, au début nous avions pensé à faire une classe ***InfectionCard*** qui hérite de ***PlayerCard*** mais nous nous sommes rendu compte que ce qu'il y a avait dans ***InfectionCard*** était la même chose que ce qu'il y avait dans ***PlayerCard***.

  Au niveau de l'implémentation, nous avons implémenter une classe ***City*** qui regroupe toutes les informations qui concerne une ville depuis le fichier *pandemic.graphml*, Nous avons aussi une classe ***Player*** qui implémente ***PlayerInterface.*** Cette classe permet d'avoir toutes les actions que peut faire un joueur.

- **Motivation et choix de l'interface graphique :**

Dans le sujet du projet, c'était mentionné que nous avions le choix pour l'interface graphique, nous pouvions trés bien utilisé JavaFX, comme nous pouvions travailler avec Swing.
Nous avons choisi de travailler avec JavaFX, dans tous les cas la personne qui était en charge de travailler sur l'interface graphique, n'a auparavant jamais travailler avec JavaFX et non plus avec Swing. Sur wikipédia nous avions trouver : *"Avec l'apparition de Java 8 en mars 2014,* ***JavaFX devient la bibliothèque de création d'interface graphique officielle du langage Java,*** *pour toutes les sortes d'application (applications mobiles, applications sur poste de travail, applications Web), le développement de son prédécesseur Swing étant abandonné (sauf pour les corrections de bogues)."*
Cela nous a vivement encouragé à utiliser JavaFX, alors nous nous sommes plus penché vers cette voie là.
Suite à cela, nous avions trouver qu'il y avait énormément de tutoriels et forum de discussion à propos de JavaFX sur internet qui pourrait nous aider, où cas où nous rencontrions des problèmes.
Pour le design de notre plateau de jeu, nous avons pris les photos depuis internet.

Pour le choix des emplacements des gadgets besoin dans le jeu, nous nous sommes inspiré de plateau déjà existants, trouver un peu partout sur les moteurs de recherches, exemple comme dans “**Image 1.1**” et **“Image 1.2”**.

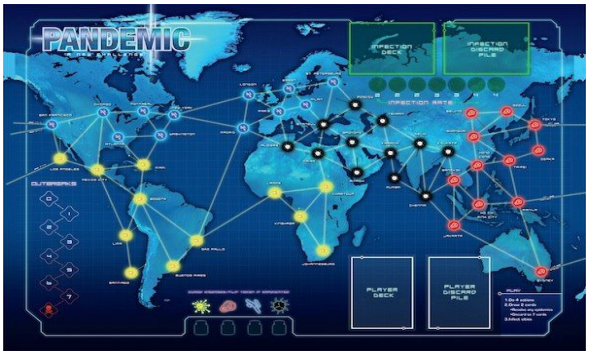


Image 1.1

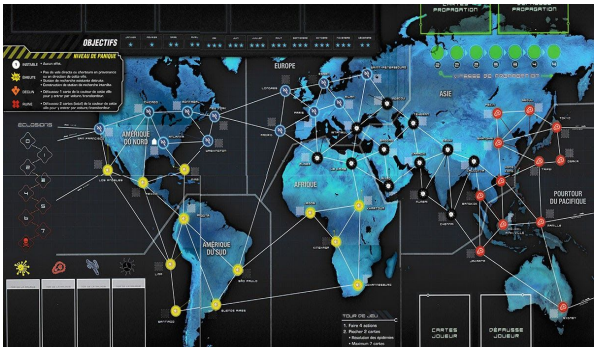


Image 1.2

III. Difficultées rencontrées

Ici nous décrivons les difficultés rencontrés, ainsi que les choses que nous avons faites, et pas faites parmis tout ce qui nous a été demandé.

- **Difficultés moteur de jeu :**
  Nous avions du mal à comprendre et à bien interpréter les règles du jeu, le manque d’informations dans l’interface y a beaucoup contribué. Par exemple la méthode isCured() n'était pas trop claire, nous devions savoir dans quel cas une maladie est soignée, si soigner voulait dire que cette maladie était soignée dans toutes les villes, ou bien une fois dans une seule ville, le second problème était comment on considère qu’une maladie était soigner, c’était d’enlever un cube ou tout les cubes, les informations n’étaient pas tout à fait claire, après nous avons rencontrer des problèmes du même type dans certaines méthodes.
  Avant d’implémenter les règles c’étaient surtout de comprendre ce que faisaient les méthodes.
- **Difficultés IA :**
  Préfinir le comportement que devait subir l’IA était assez compliqué, comme il y avait énormément de cas possibles. Nous avons essayé d’appliquer différentes stratégies pour l’IA mais nous n’avons pas atteint les résultats souhaités.

- **Difficultés interface graphique :**
  En ce qui concerne l’interface graphique, c’était assez dur de commencer, nous n’avions pas forcément déja utilisé JavaFX auparavant.

Ce que nous avions fait, c’était de regarder des cours sur le internet, s’inspirer de site comme “openclassroom”, “o7planning.org”, de la documentation sur “oracle” cela nous a énormément aider.

Nous avons rencontré pas mal de problèmes, les plus importants sont les suivants :

- Quand nous devions placer nos villes sur le plateau, nous n’avions aucune ville qui était à sa place, nous ne voyons sur l’écran que 4 ou 5 villes d'affichée, le problème était dû au fait

que certaines valeurs X, Y étaient négatives tandis que d'autres sortaient totalement de la fenêtre que nous affichions parce qu'elles avaient des coordonnées X et Y trop grandes, pour résoudre ce problème nous avions trouver un compromis entre les valeurs. Après plusieurs essais de valeurs, nous avons appliquer des formules sur les valeurs de X et Y qui faisaient en sorte de placer, plus ou moins bien, les villes à leurs places sur la map.

- Une autre difficultés c'était d'avoir des erreurs et des exceptions sans pour autant savoir d'où elles venaient, la plupart du temps un oublie de signe "-", un "=" en trop ou bien tout simplement un oublie d'importation de bibliothèques nécessaires, ou bien importation des mauvaises bibliothèques, pour résoudre ce problème futile ça pouvait nous prendre des heures, mais nous avons développé une patience énorme pour aboutir à un résultat plus ou moins satisfaisant.

- Nous avons eu un problème au niveau de l'interface graphique vers la fin du projet, nous n'avons pas pu connecter correctement l'interface au jeu, notamment d'afficher en temps réel le déroulement du jeu. Nous n'avions pas terminer l'implémentation de l'interface, il nous manquait quelques icônes, notamment les cartes pour le jeu, nous avions voulu les personnaliser nous même, mais ça aurait pris plus de temps que prévu, nous aurions relier chaque noeud à une action qui se serait déroulé au même temps avec l'exécution de l'IA.

IV. Répartition des tâches

En ce qui concerne le moteur de jeu, et l'IA ***Paul*** et ***Shahram*** se sont occupé de cette partie là, ils se sont réparti l'implémentation des méthodes de manière équitable.
Dyhia s'est occuper d'implémenter l'interface graphique.
Bien évidemment, à chaque problème que nous rencontrions, nous essayons de nous réunir ensemble et de nous entraider pour résoudre le problème, cela était très important pour le bon déroulement du projet, nous avons travailler en équipe.
Quant au rapport nous l'avons réalisé en collaboration.

V. Point sur les fonctionnalitées

Dans l'ensemble nous avons implémenter tout ce qui était optionnel, nous n'avons fait aucune fonctionnalitée en plus. Cependant il y a une méthode que nous n'avons pas implémenter qui est *turnDuration()* où nous devions donner à l'IA une seconde pour choisir l'action à effectuer, bien évidemment nous avions essayer de l'implémenter, mais nous n'avons pas vraiment compris comment implémenter cela, le problème c'était dû au fait que nous ne savions pas où placer exactement le test.
Nous avons commencez à faire une interface graphique, qui représente le jeu, mais c'était une interface static, nous ne l'avons pas relié au jeu.

VI. Sources

Image 1.1 : https://www.le-passe-temps.com/aventure/3737-pandemic.html.

Image 1.2 :
http://pandemiclegacyresetkit.blogspot.com/2016/07/tutoriel-de-reinitialisation-de-boite.html.
Site web : https://fr.wikipedia.org/wiki/Pandémie_(jeu).
Site web : https://o7planning.org/fr/11009/javafx.
Site web : http://lardj.free.fr/docs/Regles/pandemie.pdf.
Site web : https://openclassrooms.com.
Site web : https://www.oracle.com.