



Manuel du développeur

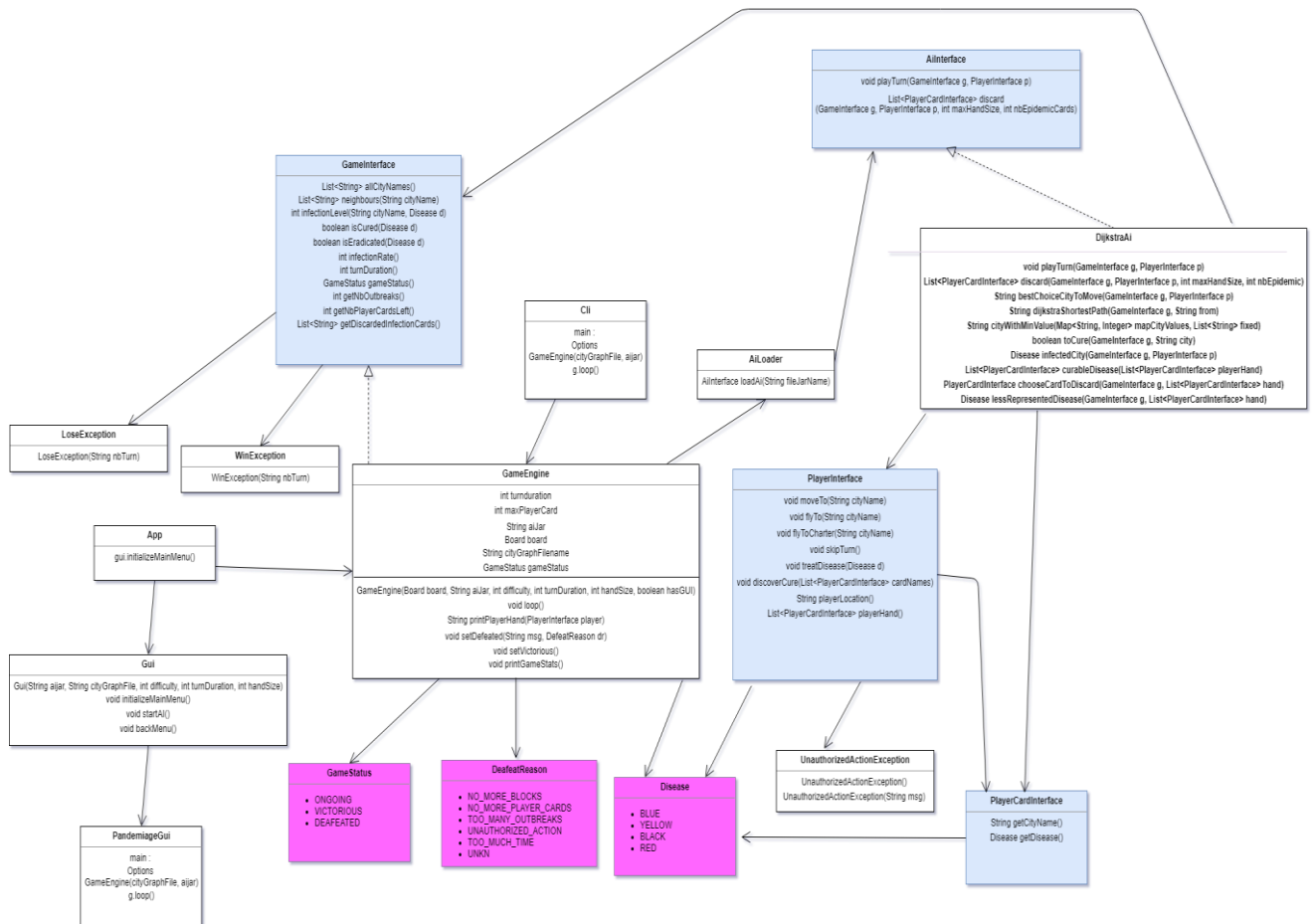
AVAST

Sommaire

1. Architecture choisie	3
2. Difficultés rencontrées.....	3
3. Répartitions du travail	4
4. Fonctionnalités apportées	4
Pistes résolues	4
Pistes abandonnées.....	4

1. Architecture choisie

Diagramme de classe



2. Difficultés rencontrées

Les premières difficultés rencontrées ont été de trouver les bonnes classes, où mettre les méthodes et les différents attributs. Nous avons aussi rencontré quelques difficultés pour comprendre la structure afin d'implémenter les interfaces. En effet, le fait de devoir respecter les interfaces fournies (AIInterface, GameInterface, PlayerInterface) nous a beaucoup contraints. A de nombreuses fois nous avons dû changer d'architecture car telle ou telle méthode/champ n'était pas accessible.

Un autre problème a été de trouver les bonnes fonctionnalités à rajouter à l'IA et les algorithmes à appliquer afin de la rendre plus performante. En effet, il a été assez simple au début d'améliorer l'IA en lui ajoutant à chaque fois des nouvelles règles de réflexion. Mais il fut compliqué à la fin lorsque l'IA implémente de nombreuses règles linéaires et qu'il n'est plus vraiment possible de l'améliorer en ajoutant de nouvelles règles. Il aurait fallu passer à une IA plus poussée implémentant des méthodes telles que Minimax etc.

L'ultime difficulté a été l'implémentation du GUI pour le rendre réactif et mettre à jour la carte à chaque événement. Il a été également difficile d'organiser le GUI selon un modèle MVC car il a fallu savoir ce que chacun jouait comme rôle.

3. Répartitions du travail

Au début, nous travaillions ensemble sur un seul pour trouver la structure de départ. Ensuite nous avons défini toutes les méthodes dont nous avons besoin pour la mise en place du moteur de jeu. Nous avions pour habitude de nous rejoindre à la bibliothèque en fin de journée après le travail (3 ou 4 fois par semaine pour 3 ou 4 heures). A l'aide d'un outil appelé Trello, à la fin de chaque séance, nous listions les tâches à faire avant notre prochaine séance. Ensuite, nous sommes passés à l'implémentation de l'IA, son amélioration ; puis l'implémentation de certains rôles. Enfin, on s'est reparti les tâches entre l'implémentation du GUI et la rédaction des différents rapports demandés.

4. Fonctionnalités apportées

Pistes résolues

- Les règles basiques demandées pour le projet.
- Une IA performante donnant 40% de victoires sur le benchmark « alecoute/m1miage » .
- Ajout d'une interface graphique en Swing permettant de voir le déroulement du jeu. graphiquement avec une carte et les éléments du jeu.
- Ajout d'un mode de jeu permettant à l'utilisateur de jouer.
- Ajout d'un menu du jeu permettant de choisir les paramètres et le mode de jeu.

Pistes abandonnées

- Après avoir implémenté les règles basiques du jeu, nous avons voulu ajouter des nouvelles fonctionnalités supplémentaires. En regardant les possibilités d'amélioration nous avons listé :
 - Le GUI
 - Le multi joueur
 - Utiliser un algorithme poussé de l'IA
 - Ajouter les rôles
 - Ajouter les cartes « Événement spécial »
 - Nous avons déjà abandonné l'idée de mettre les stations de recherche car :
 - Pour cela il aurait fallu implémenter les rôles (chercheur),
 - L'ajout de cette nouvelle règle aurait eu un impact sur l'IA déjà implémentée
 - Sauf que la modification de l'IA respectant ce nouveau règle l'aurait rendue incompatible avec le moteur du jeu du Benchmark.
 - Cela aurait rendu plus complexe l'implémentation du GUI.

On a voulu faire plus simple et nous avons implémenté les rôles et les cartes « Événement spécial ». Sauf que nous nous sommes rendus compte que la majorité de ces fonctionnalités demandait soit d'avoir plusieurs joueurs soit d'avoir des centres de recherche. Et donc nous tournions en boucle. Et n'ayant plus beaucoup de temps, nous avons dû faire des choix.

Nous nous sommes dit :

- en premier lieu que le GUI était indispensable pour voir le déroulement du jeu.
 - En second lieu qu'il serait dommage de ne pas jouer à notre propre jeu, d'où l'ajout du mode Humain.
- Nous n'avons pas implémenté la durée maximale d'un tour car, ayant codé le GUI, le joueur doit cliquer sur le bouton « CONTINUER » à chaque action de l'IA. Nous ne pouvons donc pas limiter le temps de réflexion de l'IA.