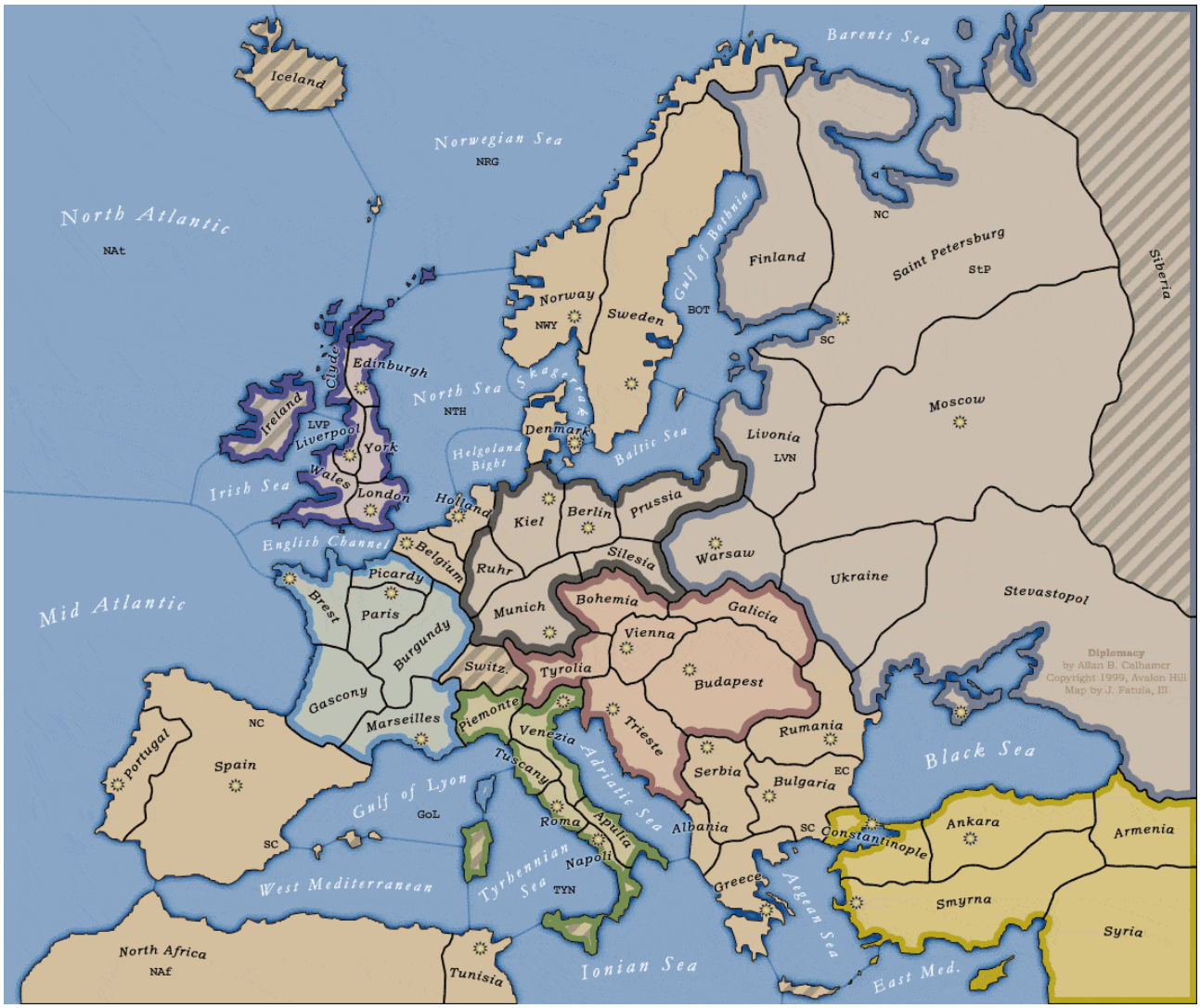
Rapport Projet Objet

Bot Diplomacy



POLYTECH NANTES

Benjamin Villain

Antoine Précigout

Remerciement

Nous tenons a remercier Mr Raschia pour ses conseils tout au long du projet ainsi que les membres du projet DAIDE pour nous avoir fourni toutes les ressources nécessaires au développement du notre robot.

Cahier des charges

Présentation du sujet :

Il s'agit pendant ce projet de développer une intelligence artificielle (bot) permettant de jouer au jeu Diplomacy. Nous avons utilisé le framework fournit par le projet DAIDE (Diplomacy AI Development Environment), il contient un serveur de jeu complet sur lequel des joueurs humains peuvent affronter d'autres joueurs ou des bots (ou alors des bots contre d'autres bots), il gère toutes les phases de communication (des plus simples aux plus complexes), ainsi qu'une interface de programmation permettant de communiquer facilement avec le serveur.

Nous allons utiliser la version Java de l'API car nous nous sentons plus à l'aise avec ce langage. Le bot « The Diplomator », écrit en Java par des étudiants dans le cadre d'un de leur projet, prend en charge toute la partie gestion d'une carte et communication « haut niveau » avec le serveur, c'est pourquoi nous l'utiliserons comme base de notre projet (il permet aussi de jouer en niveau de communication 10 mais nous n'utiliserons pas cette partie).

Objectif du projet :

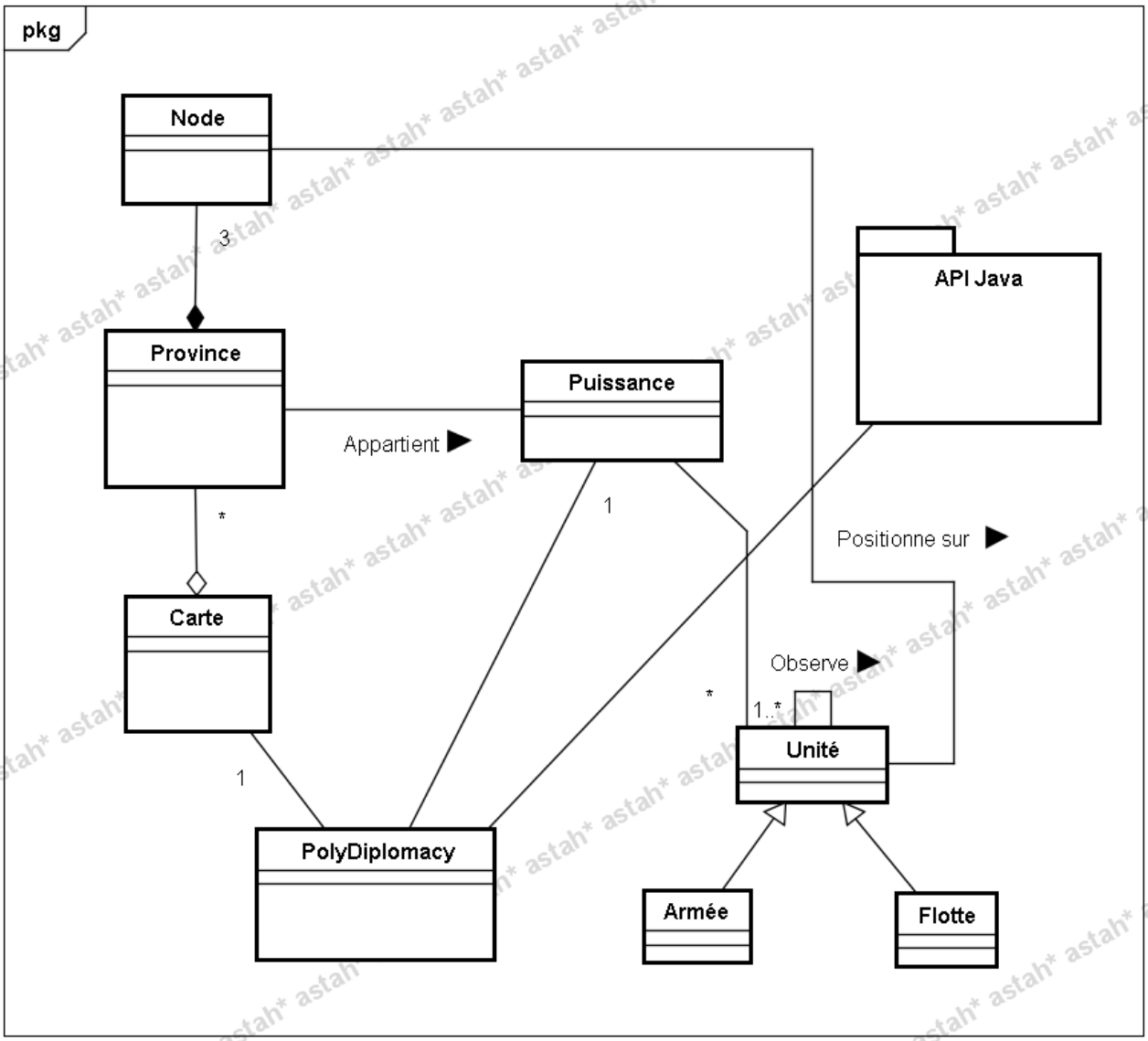
- Développer un bot fonctionnel
- Dans un premier temps sans communication avec les autres participants, si nous avons le temps nous essaierons d'ajouter la gestion des alliances (press 1).

Spécifications du bot :

- Jouer une partie en niveau no-press sur un serveur DAIDE
- Sauvegarder/charger une partie
- Revenir d'une déconnexion

Modélisation UML

Diagramme de classe :



Originalité de la solution

Lors de nos recherches sur les bots existants, il est ressorti que les performances du bot « DumbBot » étaient très intéressantes concernant le rapport temps de calcul - qualité de décision. La stratégie principale mise en place par celui-ci est d'affecter un poids à chaque province de la carte, ce poids est calculé en fonction de plusieurs paramètres et va correspondre à l'attirance de la case. Après avoir affecté un poids à chaque province, il parcourt la liste de ses unités de manière aléatoire et va pour chacune d'elle, choisir une des provinces adjacentes ayant un poids fort. Il en ressort que même si ce bot ne fait aucune recherche d'optimisation (recherche de plus court chemin ou prédiction des mouvements de ses adversaires) il se déplace de manière intelligente sur la carte occupant les points les plus stratégiques. De plus des tests ont montré que ce bot gagnait presque à tous les coups face à des joueurs débutants.

Partant de cette constatation, nous avons choisi de continuer dans cette voie là, ainsi nous avons gardé la pondération des provinces. Cependant nous allons améliorer le comportement de nos unités. Au lieu que chacune d'elle soit complètement autonome et ne tienne pas compte des déplacements des autres, nous allons créer un réseau entre nos unités, chaque unité est maitresse d'elle-même mais travaille pour le groupe. En reliant chaque unité avec toutes les autres (grâce au patron Observable-Observer par exemple) nous allons pouvoir amorcer une dynamique de groupe. Lorsqu'une unité va se sentir menacée, elle pourra prévenir toutes les autres du danger, ensuite chaque unité notifiée pourra choisir ou non de lui venir en aide. De même lorsqu'une unité voudra attaquer une province, elle pourra demander de l'aide etc… Il faut veiller particulièrement à ne pas créer d'appels infinis aux observateurs, par exemple une unité qui choisi d'aider une autre va notifier toutes les autres ce que peut changer les ordres d'une autre qui va à son tour prévenir les autres etc…

Nous pensons également que pour certaines actions, la notification aux observateurs ne devra pas engendrer de changement d'action de leur part (i.g « je supporte telle unité » ne doit pas entrainer d'action des autres unités mais elles seront au moins informée de cette action).

En donnant cette dynamique de groupe à nos unités, nous espérons pouvoir améliorer grandement les performances obtenues par les précédents bots, néanmoins nous ne pouvons être sûr que cela sera vraiment payant en terme d'efficacité.

Difficultés rencontrées

Nous avons rencontré certaines difficultés à assimiler toutes les règles du jeu Diplomacy en effet celles-ci sont très complexes malgré notre première approche de ce jeu durant le projet C du premier semestre.
Cependant, en effectuant des parties sur le client fourni par l'API (DAIDE Mapper) nous avons pu apprendre plus facilement les règles de déplacement (transport), d'attaque, de contre lors d'une attaque etc… ce qui nous a permis de mieux appréhender la création du bot.

En plus des difficultés concernant les règles, nous avons manqué de temps pour pouvoir développer de manière plus approfondie notre IA.

Conclusion

La réalisation de ce projet nous a permis d'appliquer nos connaissances en programmation acquises au cours de cette année, ainsi que durant le projet de programmation C du premier semestre. Ce fut un projet très intéressant tant par sa dimension pédagogique (réalisation d'une intelligence artificielle) que par sa dimension ludique car nous avons pu jouer face à notre robot.
De plus nous avons du faire face à certaines contraintes de temps et avons du nous organiser pour mieux gérer le travail en groupe, ce qui nous a permis de mieux comprendre les mécanismes de gestion de projet.