



Projet Tutoré : La baille de café

Semestre 4

Les Arabicas

MAIRESSE Guillaume

FER Hugo

PIERRE Andréas

RAHMANI Idris



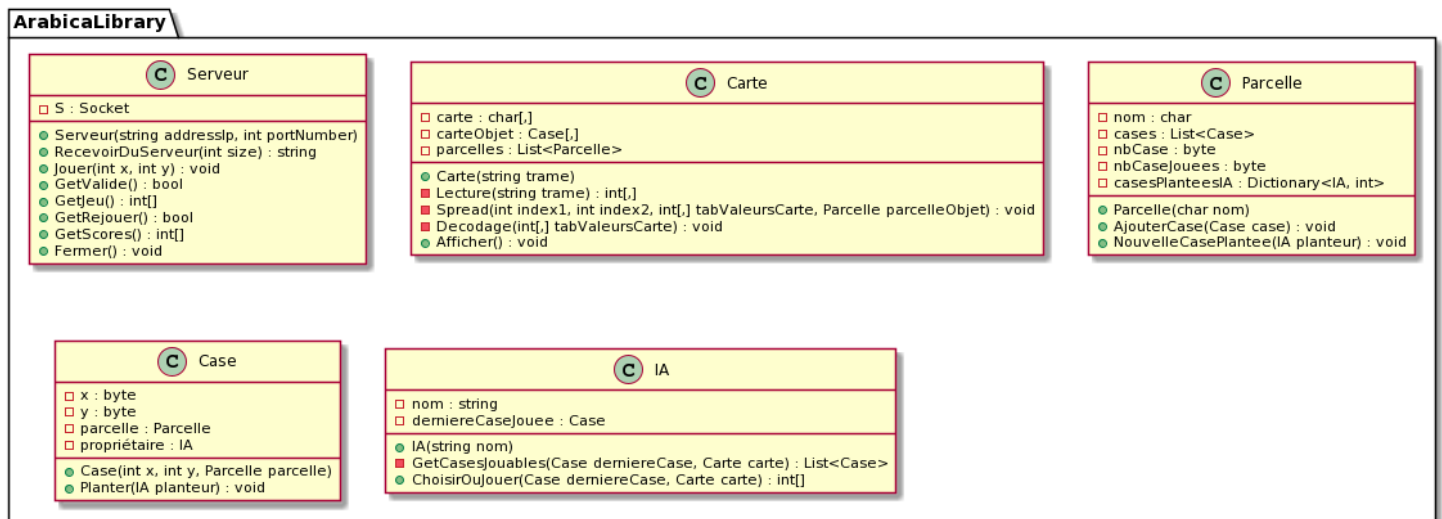
DÉPÔT GIT :

https://gitlab.com/Andrp99/bataille_cafe_arabica.git

Introduction

Il nous a été demandé de réaliser une IA capable de battre une autre IA à un jeu de société nommé « bataille du café ». Ce fut un projet très intéressant pour ce qu'il nous a amené à réaliser. Dans ce document vous trouverez la modélisation objet de notre projet, l'explication de la stratégie employée par notre IA ainsi qu'une conclusion regroupant les apports de ce projet, les difficultés rencontrées et les éventuels axes d'amélioration du programme.

Modélisation objet



Une version plus lisible est disponible sur le dépôt git (Bataille_cafe_arabica\s4\Diagramme de classes UML.png).

Stratégie

Tout d'abord, l'IA joue la première case aléatoirement parmi toutes les cases jouables du plateau. Cela permet de rendre le combat plus intéressant en évitant d'avoir toujours la même issue (les deux IAs se comportant toujours de la même façon selon la carte). Sans cela, pour gagner il suffirait de choisir une case (par carte) où l'on sait que la victoire est assurée et de faire en sorte que l'IA la joue toujours au premier tour. De même, une partie étant très rapide, une équipe peu scrupuleuse pourrait faire plusieurs parties d'affilé et choisir de refaire la partie leur offrant le plus de points (en gardant les résultats en mémoire), le programme affichant uniquement le score de cette dernière.

Ensuite, nous avons mis en place un système de priorité qui permet de classer les cases que notre IA peut jouer par importance, celle ayant la plus haute priorité sera jouée. La priorité pour une case est définie selon ce tableau :

Condition	Priorité
Une case voisine est plantée par l'IA Client (pour le bonus)	+1 par case voisine
Les points de la parcelle sont bloqués par le serveur	-10
Il est possible de bloquer les points de la parcelle en jouant cette case de sorte à ce qu'aucunes des 2 IA ne gagne les points	+4
Les points de la parcelle ne sont pas bloqués	+2
L'IA client a déjà bloqué les points de la parcelle	-4

(Les points d'une parcelle sont considérés comme bloqués quand plus de la moitié des cases sont jouées par une IA)

Conclusion

Mettre en place une connexion avec un serveur, décoder une trame, réaliser des classes permettant de mettre en place une carte dynamique et créer une IA capable de traiter les informations nécessaires à la stratégie d'un jeu de société; cela a été très enrichissant pour nous et nous a permis de nous améliorer en programmation objet.

Étant donné que nous étions plutôt organisés, nous avons rencontré que très peu de difficultés. Une chose qui a été difficile à réaliser est la valorisation de certaines conditions pour déterminer l'importance de jouer une case plutôt qu'une autre. Il nous a été nécessaire de jouer beaucoup au jeu en amont du projet pour trouver ce qui est le plus important à prioriser pour une stratégie viable.

Pour améliorer notre projet, nous avons songé à ajouter un élément à notre stratégie. Le but de cet élément aurait été de bloquer l'IA adverse pour qu'elle ne puisse plus jouer et mettre fin à la partie, cela uniquement quand l'IA client aurait fait suffisamment de points pour s'assurer la victoire. Ce point est difficile à mettre en place : il aurait été nécessaire de calculer le score à chaque tour et de faire beaucoup de calculs pour chaque coup pour bloquer l'adversaire.

De plus, nos valeurs de priorités ne sont sûrement pas assez équilibrées car, comme nous l'avons déjà dit, déterminer l'importance de jouer une case plutôt qu'une autre reste toujours très complexe.