

2019-2020

# PS5\_PROJET

● *Citadelles*

## ÉQUIPE JTHS :

Julien N'DIAYE  
Titouan LE MAO  
Hamza AYOUB  
Sana DIBE

# SYNTHÈSE DU PROJET

Dans ce projet, nous avons implémenté citadelles, un jeu de société, en java pour avoir une version informatisée de citadelles.

Le jeu n'est pas jouable avec des joueurs, seulement des bots s'affrontent entre eux reprenant soit des concepts aléatoires ou bien des concepts humains par l'intermédiaire de décisions réfléchies pour les bots les plus perfectionnés. Notre projet citadelles respecte toutes les règles de citadelles, étant conforme au jeu mais aussi à la réalité elle y glisse quelques bots tricheurs par ci par là. Ils bien évidemment retirable mais ils sont beaucoup plus fort que les autres types de bots !

Citadelles est un jeu non résolu, c'est-à-dire qu'il n'existe aucune façon de prédire à partir de n'importe quel moment de la partie l'issue de cette dernière si les 2 joueurs jouent à la perfection. Cela nous a conduit à faire des Test bench pour en déduire quelles stratégies étaient les meilleures notamment pour la gestion des cartes citadelles et du choix du personnage. En effet notre projet nous a permis d'optimiser certaines stratégies, c'est pour cela que nous avons confiance en notre projet tout en gardant un esprit critique sur celui-ci.

En ce qui concerne le jeu, chaque bot dirige une ville en essayant de l'améliorer au cours de la partie. Chaque bot possède un nombre de pièce qu'il se fait attribuer au début par une banque et peut en réclamer à chaque début de tour. Il peut aussi choisir un personnage parmi une liste de personnage disponible à chaque début de tour, tout comme choisir une carte citadelles s'il ne prend pas de pièce d'or. Une carte citadelle représente potentiellement un futur quartier qu'un joueur pourra construire dans sa ville. Lorsque 8 quartiers distincts sont construit dans une ville, le tour en cours se termine et à l'issue de celui-ci est déterminé le gagnant.

Sur ce projet nous avons volontairement décidé de ne pas mettre en place une banque. En effet le jeu citadelles est un jeu de construction, nous n'avons pas trouvé utile après de multiple Test bench de laisser les stratégies d'économisation aux bots. Ainsi chacun construit son meilleur quartier dès qu'il peut. Du haut de ses 5 bots implémentable maximum, jamais la somme de toutes leurs économies ne dépasse 30 pièces ce qui respecte bien les règles du jeu.



## CRITIQUE DU CODE

Lorsque nous avons commencé le projet, nous étions relativement inexpérimentés et nous sommes donc partis sur de mauvaises bases. S'en est suivi de multiples réorganisations du code. Donc un point négatif est que notre code a subi beaucoup de refactoring ou réorganisation du code.

La création de bot étant un axe principal du projet, nous nous sommes concentrés dessus. Leurs codes sont donc de bonne qualité. À contrario, étant pris par le temps, l'implémentation des cartes violettes s'est faite relativement rapidement et nous sommes convaincus que nous aurions pu optimiser le code d'une meilleure façon.

Le moteur de jeu quant à lui est très bien implémenté se découpant même en 2 classes. Les personnages, les cartes citadelles et les pioches étant relativement simple à implémenter (appart le personnage Condottière), nous sommes confiants de leurs qualités.

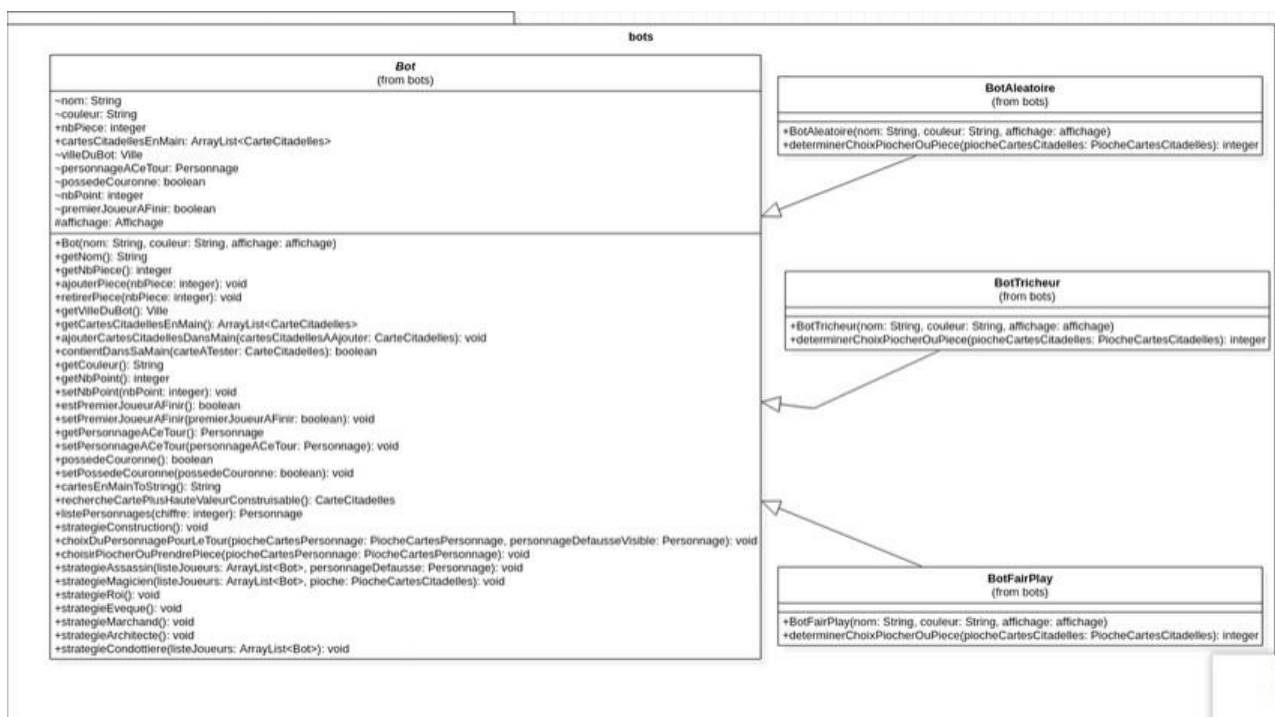
Tous les tests des fonctions ont été réalisés (sauf sur les fonctions bidons ex : la fonction get). Mais cela s'est compliqué au niveau des tests. En effet le jeu citadelle repose en grande partie sur l'aléatoire et cela fut dur pour nous de faire tous les tests adéquats prenant en compte le caractère aléatoire de citadelles.

Pour tout ce qui se ramène aux concepts objets cela a été bien réalisé.

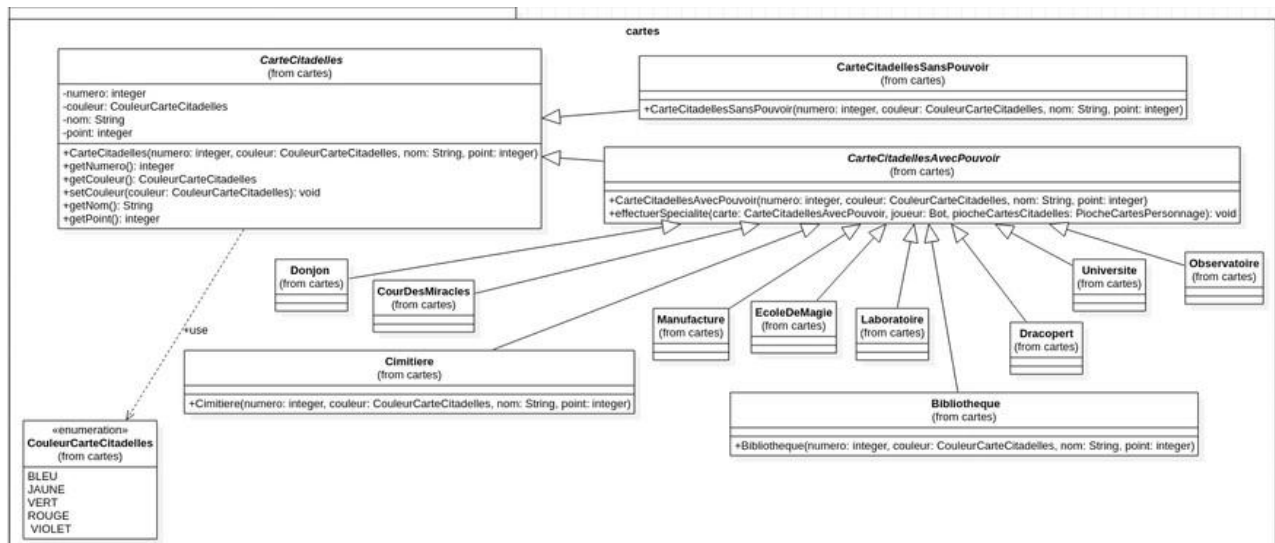
Enfin nous avons eu une répartition plutôt équitable des responsabilités.

Voyez un exemple la structure des packages ci-dessous :

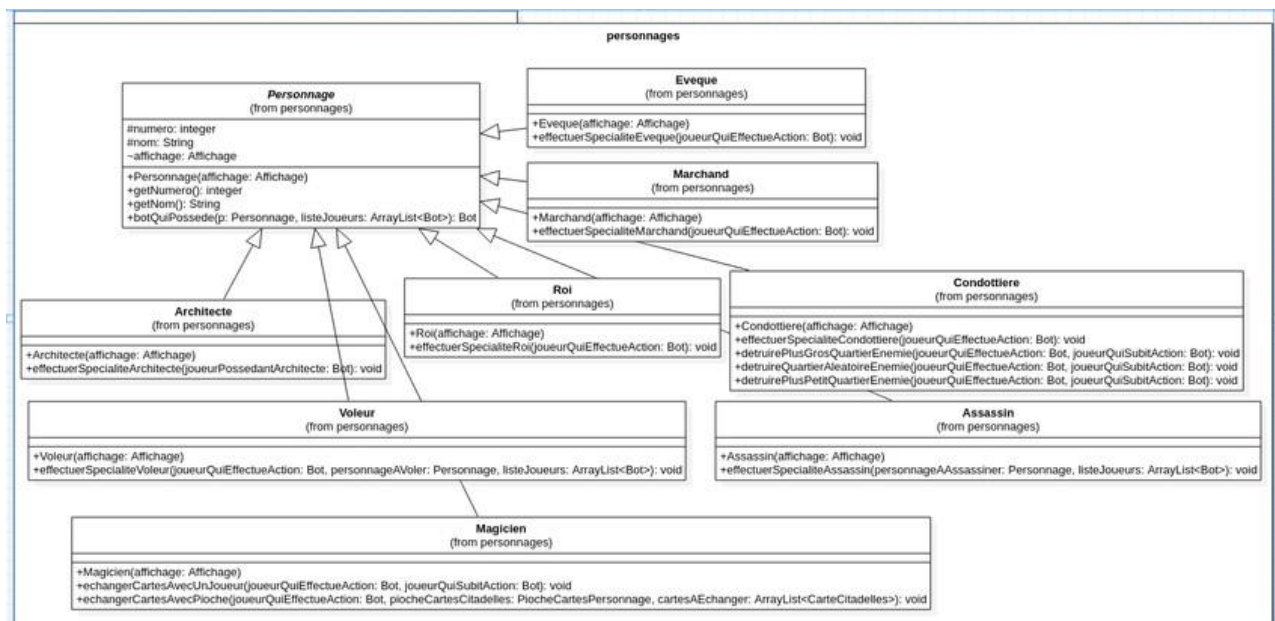
### Package bots :



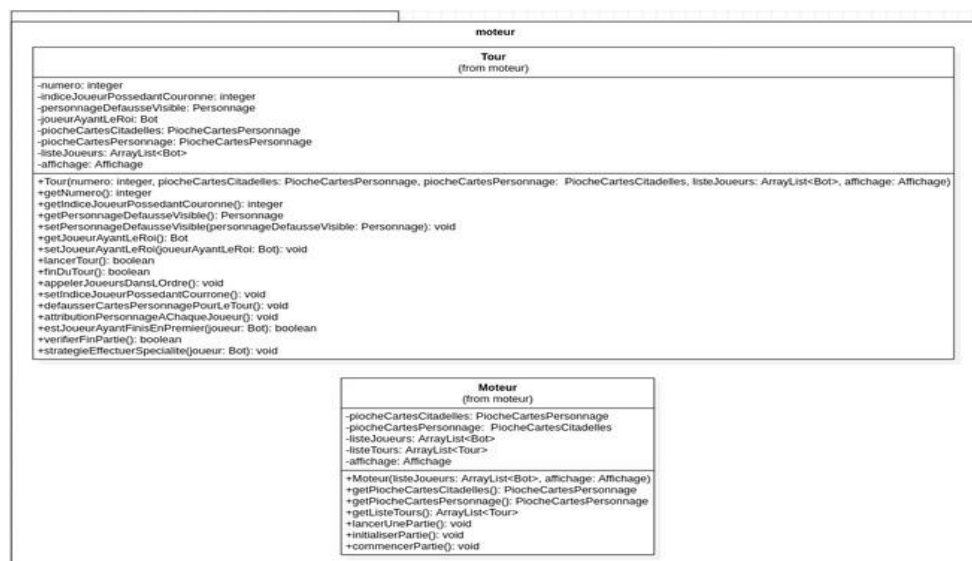
## Package cartes :



## Package personnages :



## Package moteur :





# Pourquoi notre projet est un bon / mauvais projet

## Pourquoi notre projet est un bon projet :

Notre projet est un bon projet car nous avons respecté toutes les règles du jeu. Nous avons même implémenté un type de bot censé représenter les joueurs tricheurs ce qui ne nous paraissait pas hors sujet. Bien que non demandé, il arrive régulièrement que des joueurs trichent. Il est à noter que les bots tricheurs reprennent des concepts humains, cela ne veut pas dire qu'à la fin de la partie, ils s'autoproclament vainqueurs en trafiquant leur nombre de point. Non. Tout comme dans la réalité ils essaient de tricher en regardant les cartes de leurs adversaires.

Nous avons créé 3 types de bots :

- Les Bots Tricheurs
- Les Bots FairPlays
- Les Bots Aléatoire

Remarque : les Bots Tricheurs ne sont en réalité que la copie des Bots Fairplay. La différence est qu'en plus ils trichent, c'est-à-dire, ils sont codés avec la même structure et ont le même code pour les aspects du jeu où il n'est pas utile de tricher (ex : si l'on veut construire 2 quartiers mais que l'on doit en choisir 1, il faut choisir de construire le quartier avec la plus grande valeur. La tricherie n'y change rien).

## Pourquoi notre projet est un mauvais projet :

Notre projet peut être considéré comme un mauvais projet car nous n'étions pas vraiment ponctuels au début. En effet, ne sachant pas comment appréhender ce langage de programmation, nous avons tardé à nous mettre au travail, ce qui n'est pas acceptable sachant que l'on doit livrer chaque semaine une version de plus en plus élaborée du projet. Il faut savoir répartir le travail sur la durée pour ne pas être dépassé ce qui nous a malheureusement conduit à réorganiser de maintes et maintes fois la structure du code.

En plus, nous ne nous sommes pas concertés dès la première semaine pour partir sur une bonne voie. Le dernier point que l'on aurait pu améliorer concerne la redondance du code.

# Rétrospective

Même si notre groupe ne pourra plus être reformé, nous envisageons pour la suite de nos projets une bonne répartition des tâches ce qui nous évitera beaucoup de réorganisation du code notamment lors de l'ajout de nouvelles fonctionnalités.

La création de toutes les classes ne se fait pas d'un coup, en effet, c'est en avançant pas à pas que nous découvrons d'autres aspects d'un projet. Il n'est donc pas nécessairement de tout planifier à l'avance car il arrivera en cours de route de revenir sur ce qui a été fait. Mais il faut que cela se fasse le plus tôt possible pour ne pas gêner les livraisons et par la même occasion générer du retard.

Nous sommes fiers de notre travail d'équipe.

Au début, apparaît une personne, ne nous savions pas faire grand-chose. Mais c'est en expliquant aux autres et en travaillant de son côté que l'on progresse et que l'on peut à son tour dépanner une personne lors d'une tâche un peu trop complexe.