

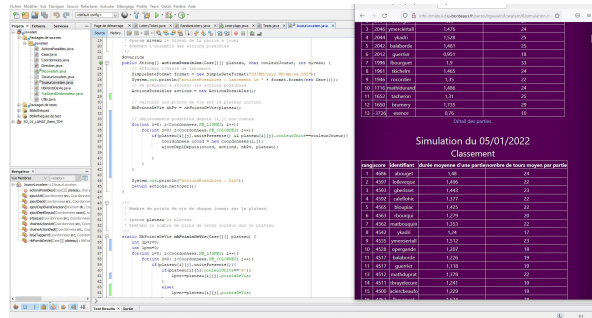
Retour sur expérience : JPO

Cahier des charges :



- Coder selon les règles du jeu fournies
- Réaliser un maximum de tests
- Correction d'une IA
- Création d'une IA
- Propreté du code
- Découpage en sous fonctions
- Optimisation de l'IA
- Rendre un rapport résumant rapidement l'ensemble du projet

Livrables :



La phase 2 du projet lowatem concerné la création d'une IA qui serait capable de jouer les meilleurs coups possible de ce jeu inventé par Mr Gowin. Dans cette optique, nous avons décidé de commencer par réfléchir à une stratégie à appliquer pour pouvoir gagner à chaque fois. Très vite nous nous sommes aperçus que la meilleure solution consiste à attaquer dès que possible. Nous avons donc décidé de baser notre stratégie sur ce constat.

La première IA (celle que l'on a pas retenue) a une stratégie plutôt simple mais néanmoins efficace. La stratégie employée consiste à attaquer le plus possible : en effet, à chaque tour de jeu l'IA parcourt l'ensemble des attaques possibles mais ne retiendra qu'une seule d'entre elles. Pour chaque action comportant une attaque, on va calculer le gain que nous apporterait ce choix. On conserve la valeur et l'action choisie jusqu'à avoir terminé le parcours des actions possibles ou si nous trouvons une attaque avec un gain supérieur. Dans le cas où aucune attaque n'est possible, l'IA se contentera de faire un déplacement d'une unité sur lui-même. Suite à cette stratégie, nous sommes certains de ne pas permettre à l'adversaire de nous attaquer puisque si nous n'avons aucune possibilité, alors lui n'en a pas non plus.

Il s'agit donc d'une IA dont le fonctionnement est simple mais il s'agit néanmoins d'un fonctionnement ne comportant pas beaucoup de défauts. Les problèmes que peut rencontrer cette IA serait d'être dans une situation où on est en possession de moins de points que l'adversaire. Si aucune attaque n'apparaît on va se contenter d'attendre hors si l'adversaire fait de même, la victoire devient alors impossible.

La deuxième IA, celle retenue pour la simulation finale est une amélioration de la stratégie brutale que l'on pouvait trouver dans la première. On parcourt chaque action disponible dans notre tableau de possibilités jusqu'à trouver une attaque. Comme pour notre première stratégie, on calcule le gain apporté par cette attaque. On en conserve la valeur s'il s'agit de la première ou si elle est supérieure ou égale à la valeur conservée précédemment. C'est à partir de ce moment que nos 2 IA se différencient puisque celle-ci ne s'arrête pas là. On vérifie ensuite si à partir de la même case source, on peut attaquer la même case de destination. On utilise ensuite une fonction qui nous permet de savoir si l'une des cases est préférable à l'autre. Pour cela on va vérifier si une des deux possibilités permet de réaliser l'attaque en se plaçant sur une case qui n'est pas exposée à l'attaque d'un ennemi lors du tour suivant.

Si on vient à valider la sécurité de notre troupe après l'attaque, la condition change et la valeur du gain stockée ainsi que l'action associée ne seront alors pris que si le gain est strictement supérieur au gain actuel.

Les problèmes que peut rencontrer cette IA serait d'être dans une situation où on est en possession de moins de points que l'adversaire. Si aucune attaque n'apparaît on va se contenter d'attendre hors si l'adversaire fait de même, la victoire devient alors impossible.

Technologie et Outils :

Pour réaliser ce projet **Netbeans** a été utilisé afin de coder. Un logiciel de simulation créé pour l'occasion a été utilisé pour tester les IA. Le document a été rédigé sur un drive pour permettre une avancée plus rapide et organisée.



Compétences développées :

Ce projet a permis de confirmer les connaissances de Netbeans en obligeant à reprendre toutes les fonctionnalités connues. La capacité à **travailler en groupe** et à se répartir le travail a été de mise pour rendre le travail sur l'IA dans les délais. Ce projet m'a aussi permis d'apprendre à **rédigier** de nombreux tests pour certifier le résultat obtenu.

