MERCIER-TALLET Yanis, PETIT Juliette

Rapport de la phase 2 du projet Lowatem

Présentation des IA :

Pour rappel, au niveau 6, on calcule les points de vie des unités apres une attaque de la manière suivante :
Les points de vie de l'unité menant l'attaque deviennent :
oldPvAttaquant - 2 - (int)((oldPvAttaqué-5)/2);

Les points de vie de l'unité attaquée deviennent :
oldPvAttaqué - 4 – (int)((oldPvAttaquant-5)/2);

Première IA, l'IA gentleman :

Pour notre première IA, nous avons déterminé trois différents cas qui surviennent lors du choix de l'action jouée :
- le cas où une attaque est possible
- le cas où aucune attaque n'est possible
- le cas où nous avons moins de 4 unités sur le plateau

Pour le premier cas, nous avons étudié chaque attaque possible en fonction des dégâts de l'attaque et de la riposte, ce qui donne le tableau suivant :

Points de vie attaquant	**Dégâts infligés**	**Points de vie attaqué**	**Dégâts de la riposte**
1 pv	2	1 pv	0
2 pv 3 pv	3	2 pv 3 pv	1
4 pv 5 pv 6 pv	4	4 pv 5 pv 6 pv	2
7 pv 8 pv	5	7 pv 8 pv	3
9 pv	6	9 pv	4

Ici, nous voyons qu'il y a 5 « catégories » selon les points de vie de l'unité. La plus puissante des catégories correspond à la dernière ligne du tableau (quand une unité a 9 pv) et donc par conséquent la moins puissante correspond à la première ligne du tableau.

Nous avons donc essayé de sélectionner les attaques qui nous semblaient les plus judicieuses en établissant un ordre de priorité.
Notre IA cherche en priorité les attaques pouvant tuer en un coup une des unités adverses. Elles mêmes sont ordonnée par le plus grand écart gagné. Par exemple, nous avons retenu en tout premier coup, une unité de 9 points de vie attaque une unité de 6 points de vie (écart de 4), puis en deuxième une unité de 7 points de vie attaque une unité de 5 points de vie (écart de 3).

Ensuite, parmi les attaques restantes, nous avons regardé les meilleurs moyens de faire perdre en catégorie les unités adverses. Par exemple nous utilisons nos unités à un point de vie comme une unité de 1 point de vie attaque une unité de 8 points de vie, le 8 passe à 6 points de vie et perd donc de la puissance d'attaque.
Nous avons terminé par écrire les attaques restantes en les ordonnant selon l'écart créé et la perte de catégorie engendrée.

Pour le deuxième cas, nous avons fais en sorte que l'IA choisisse un déplacement qui ne met pas en danger une de nos unités. Cependant dans notre code, nous choisissons toujours un déplacement sécurisé pour la première unité apparaissant dans le tableau des actions possibles.

Pour le dernier cas, lorsque nous avons moins de 4 unités sur le plateau, nous choisissons d'attaquer systématiquement. Lorsque c'est impossible nous réalisons un déplacement sécurisé comme nous l'avons expliqué dans le deuxième cas.

Deuxième IA, l'IA « plus grand gap »:

Pour cette seconde IA, nous avons séparé les cas avec et sans attaque comme pour pour la première IA. La gestion des déplacements est identique cependant la sélection des attaques est gérée différemment.
Nous avons choisi d'appliquer la méthode du « plus grand gap » c'est à dire que nous choisissons systématiquement une attaque lorsqu'elle est possible et cela sera celle qui créera le plus grand écart de points de vie entre nous et l'adversaire (elle ne prend pas en compte la mort de notre unité ou de l'unité adverse par exemple).
Nous déterminons le plus grand écart avec le calcul suivant :

|(pv de l'attaquant avant l'attaque – pv de l'attaquant après l'attaque) - (pv de l'attaqué avant l'attaque – pv de l'attaqué après l'attaque)|

Par exemple, dans le cas où une unité de 9 pv attaque une unité de 6 pv, le calcul donnera :

écart = |(9-7)-(6-0)|
écart = |2-6|
écart = |-4|
écart = 4

Ce qui résulte à un écart de 4pv.

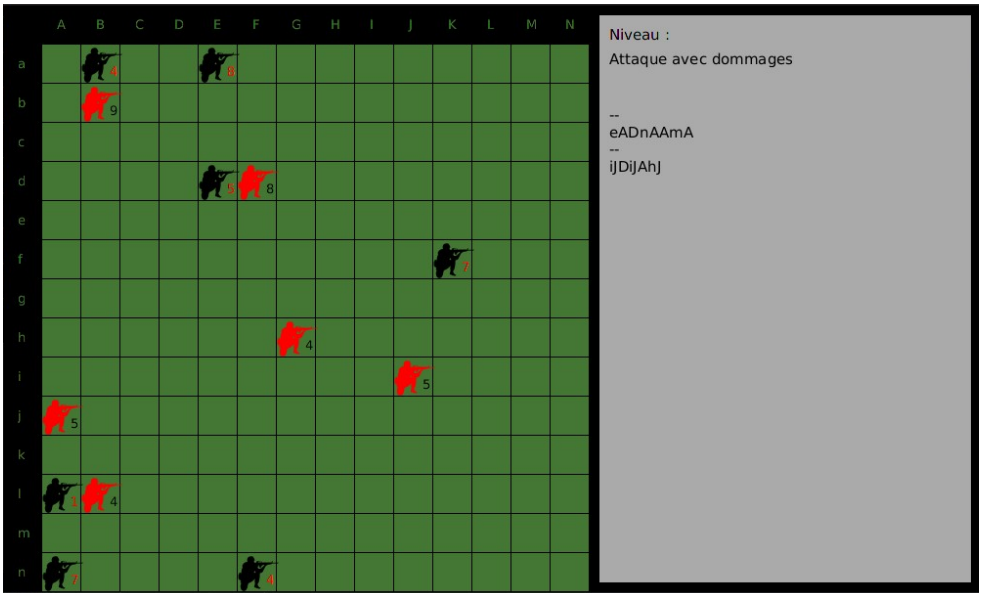
Choix de l’IA :

Nous avons choisi la seconde IA car elle est plus précise et plus structurée. Le défaut de notre première IA est que nous avons nous même choisi les bonne combinaisons d’attaques et que nous pouvons en oublier. De plus nous ne savons pas concrètement si l’ordre que nous avons choisi est le meilleur.
Les résultats de nos deux IA contre les IA du Tâcheron et du Guerrier sont similaires. Elles gagnent systématiquement contre le Tâcheron et gagnent environ à 95 % contre le Guerrier lorsque nous sommes le joueur Noir. Quand nous commençons (joueur Rouge) les deux IA ont un taux de victoire de 100 %. Cependant lors des simulations, nous avons remarqué que l’IA Plus grand gap (au début : top 15, sur les deux dernières simulations : top 3) avait tendance (généralement) à se retrouver plus haut dans le classement que l’IA Gentleman (au début : top 25, sur les trois dernières simulations : top 15). Nous n’avons pas choisi notre IA par rapport aux test contre les 2 IA aléatoires car le taux de victoire est le même comme nous l’avons montré et que l’écart de points de vie à l’issue des parties est globalement similaire.
Si nous avions eu plus de temps et plus de test nous aurions peut être choisi la première IA, car le défaut de la seconde est qu’elle est plus difficilement améliorable, en terme de choix d’attaque du moins.

Portfolio :

MERCIER-TALLET Yanis :

Le projet Lowatem avait deux objectifs : nous faire programmer des règles d'un jeu de plateau les unes après les autres et de développer en binôme une IA afin de gagner des parties sur ce jeu. Le projet m'a permis d'approfondir mes connaissances en java avec l'utilisation des tableaux en deux dimensions, et d'exploiter l'importance des tests unitaires. La phase d'IA était fort intéressante de par la réflexion qui était nécessaire. Les deux phases se déroulaient sur un temps très court ce qui m'a poussé dans mes retranchements.



PETIT Juliette :

Le projet Lowatem consistait en deux phases : la première où nous devions coder les règles d'un jeu en possédant déjà le code du jeu et la deuxième qui consistait à coder une IA par binome qui jouait à ce jeu. Grâce à la première étape, j'ai appris à coder de manière plus structurée, en utilisant correctement les test unitaires et en faisant attention au découpage en fonctions. On a du réaliser ce code en peu de temps et cela nous a permi de juger un peu nos limites et faiblesses. Les deux phases ont entraîné beaucoup de reflexion.