

## Projet Lowatem Partie 2

### Présentation des IA

#### Pour notre première IA :

Nous avons choisi une stratégie utilisant ce que nous avons fait à la phase 1. Le but de notre IA, c'est de voir s'il y a oui ou non des attaques possibles :

- Si oui, on attaque à chaque fois, car en choisissant en fonction de l'unité qui a le plus de point de vie et après celle qui fera le plus de dégâts à l'ennemi.

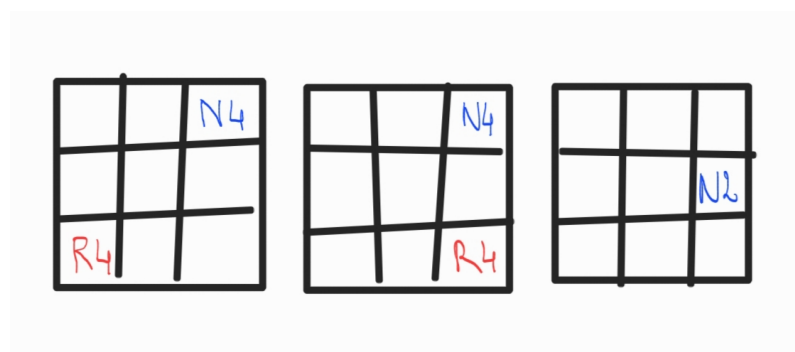
Nous attaquons dès que possible, car cela est plus avantageux pour nous les dégâts d'attaques étant calculer avec cette formule :  $-x$  (point de vie de l'autre unité  $-5$ )/2 ou  $x$  est égale à 4 quand on se fait attaquer ou 2 quand on attaque, le but donc ici est de protéger au maximum nos points de vies.

Nous choisissons les unités avec le plus de points, car en premier, c'est celle qui fera possiblement de plus de dégâts, par exemple une unité de 2 ne pourra faire un maximum de dégâts que de  $-4 - \text{int}((2 - 5)/2)$  soit  $-3$  alors qu'une unité de 9 points de vies pourra faire au maximum  $-4 - \text{int}((9 - 5)/2)$  soit  $-6$  du coup, il est plus préférable de prendre les grosses unités, car on fera le plus mal à l'ennemie et on aura plus de chance de tuer une unité ennemie ce qui est un grand avantage pour nous car cela enlèvera beaucoup de possibilité d'attaques à l'adversaire.

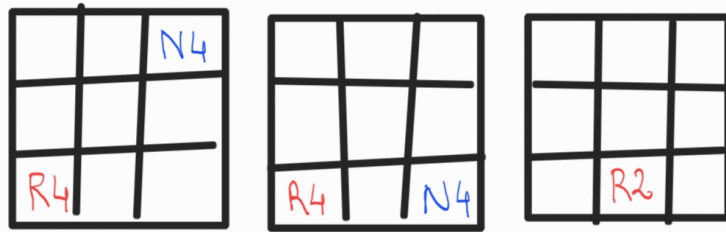
- Sinon, s'il n'y a pas d'attaque possible nous laissons toutes nos unités-là ou elle sont pour ne pas donner plus de possibilités d'attaques à l'adversaire.

Nous restons sur place pour ne pas donner des possibilités d'attaques en plus à l'adversaire.

Exemple nous somme les rouges et nous cherchons à attaquer :



Exemple 2 nous somme les rouge avec notre technique :



Notre méthode nous fait donc gagner.

**Pour notre deuxième IA, notre stratégie était :**

Les actions possibles générées sont séparées entre l'action en elle-même et les points de vie de chaque équipe dans une classe.

On définit un score  $S$ , associé à chaque action possible est égal à la différence de points de vie entre les points de vie de notre joueur et ceux du joueur adverse.

Pour chaque action, si une attaque est disponible, on sélectionne le meilleur coup ennemi, défini comme étant celui qui produit le plus grand résultat négatif sur la différence de points de vie entre les points de vie de notre adversaire, suite au meilleur coup ennemi, et ceux de notre soldat. Ce résultat est greffé à chaque action d'attaque. Une fois que toutes les actions ont été associées avec un score, on calcule l'action qui a le score le plus élevé possible. Cette action d'attaque est envoyée au Grand Ordonnateur.

Ce score est calculé dans le cas des attaques, car on recherche l'attaque qui causera le plus de dégâts à l'adversaire, et afin d'éviter des attaques qui pourraient être inutiles, voire dangereuses (car si on sélectionne seulement la meilleure attaque, sans regarder l'attaque que pourrait faire l'adversaire au prochain tour immédiat, le risque est que l'adversaire tue notre unité, ou cause plus de dégâts pour nous, à cause de l'attaque de nous avons fait, dans le cas d'un déplacement, que l'apport causé par notre attaque.).

Si plus aucune action d'attaque n'est disponible, on prend la première action sur-place possible, et on la renvoie au Grand Ordonnateur. En effet, étant donné qu'on ne calcule ici pas la présence ou non d'unités ennemies sur les autres cases du plateau, il est plus judicieux de rester sur place que de se déplacer vers une ligne ou une colonne avec potentiellement plus d'unités ennemies, ou des unités ennemies qui ont plus de points de vie que la nôtre, que de rester sur place et d'attendre de subir l'action ennemie potentielle.

## Choix de l'IA

Nous avons conservé la première IA, car la seconde IA était initialement prévue comme étant une application au Lowatem de l'algorithme Minimax. Cependant, par manque de temps et de compétences, l'algorithme Minimax n'a pas pu aboutir à un résultat convaincant, que ce soit sur le guerrier ou le tâcheron. Un autre algorithme, le second ici présenté dans le rapport, a donc été codés, mais ne présente là non plus pas de résultats satisfaisants, ne battant le guerrier que 3 fois sur 10 environ, contre 8 fois sur 10 pour la première IA, et ayant certaines actions disqualifiantes à cause d'un bug qui n'a pas pu être résolu au moment de la remise du projet. Sur les simulations, la première IA se place environ 30e dans le classement, tandis que la seconde est 3ème en partant de la fin de la liste, à cause de disqualifications.

## Portfolio de Maud

Ici, on peut voir les résultats lors d'une simulation le 16/01/2022, notre IA est arriver 9e avec 11 744 points, environ elle fessait des matchs de 27 tours sur 40 et la moyenne d'une partie joué était de 1,724 secondes.

Par la suite, on peut voir une simulation contre le tâcheron avec le plateau de départ et d'arriver au nombre de points à la fin, c'est des simulation qui nous servait à vérifier le bon fonctionnement de notre IA pour voir si elle ne se fessait pas disqualifier.

Pour ce projet, nous devons créer des IA suite à la phase 1 du Lowatem qui était de coder les niveaux du jeu sur lesquels nos IA allaient évoluer.

Ce projet m'a appris à travailler dans un temps plus réduit et à rester calme face à des difficultés rencontrées comme le fait qu'une de nos IA n'a pas pu être aboutie. J'ai pu aussi travailler avec des IA pour la première fois et j'ai appris beaucoup de choses sur les stratégies d'attaques ce qui fut très intéressant et enrichissant pour moi.

9	11744	mgenetet	1,724	27
---	-------	----------	-------	----

Partie :  
Début : 12/01/2022 09:01:30.862  
Fin : 12/01/2022 09:01:32.506  
Durée : 1.644 seconde(s)  
Nombre de tours de jeu : 30

Plateau initial :

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
a				SN5		SN8								
b														
c								SN6						
d					SR6									
e				SR9										
f	SR5										SN7			
g											SN9			
h											SR7			
i	SR8													
j					SR7								SN9	
k	SN9			SR9				SR9						
l														
m														
n								SN7						

Plateau en fin de partie :

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
a														
b														
c												SN6		
d		SN4												
e														
f									SN3					
g								SN2						
h														
i				SN3										
j														
k							SN3							
l														
m					SN3									
n														

Les 2 joueurs :

Joueur n° 1 :

rouge  
« Intelligence artificielle » Tacheron  
Perdant  
Points de vie sur le plateau final : 0  
Nombre de points : 57

Joueur n° 2 :

noir  
mgenetet  
Gagnant  
Points de vie sur le plateau final : 24  
Nombre de points : 105

Historique : 1[1/2]fBDFH(7) 2[2/2]gLdGHAFH(34) 3[1/2]eEdEA(3) 4[2/2]jNDjGAjF(102) 5[1/2]iADiI(7)  
6[2/2]kADdAAeA(97) 7[1/2]jFDjM(27) 8[2/2]aFDeFAdF(93) 9[1/2]kEdEAaA(7) 10[2/2]fLDiLahl(92) 11[1/2]eADiA(6)  
12[2/2]gHDkHAKI(95) 13[1/2]iIDfI(5) 14[2/2]nIDjIAkI(92) 15[1/2]aEdME(6) 16[2/2]cHdcFAdF(93) 17[1/2]iADiJ(6)  
18[2/2]eFdMfAmE(97) 19[1/2]hLDhH(6) 20[2/2]jGDjJAiJ(98) 21[1/2]mEdMG(6) 22[2/2]jIDgIAfI(98) 23[1/2]jMDbM(6)  
24[2/2]cFdcMABM(98) 25[1/2]mGDmC(5) 26[2/2]jJdFJAfI(111) 27[1/2]hHDhE(5) 28[2/2]iLDiEAhE(95) 29[1/2]mDcEc(4)  
30[2/2]dADdCAeC(97)

Format d'un tour de jeu : <tour> [ <ordre joueur> ] <action> ( <durée en ms> ).

## Portfolio Valentin

Dans ce projet, j'ai réussi à coder un jeu avec des règles, jusqu'à certains niveaux. J'ai surtout appris l'humilité : je n'ai pas le niveau nécessaire pour réussir les algorithmes de théorie des jeux les plus compliqués. J'ai cependant réussi la partie 1 du Lowatem en faisant les  $\frac{3}{4}$  des niveaux. J'utiliserai désormais un cahier des charges pour faire mon travail en partant du plus simple.

70	440	vchaud	1,293	18
----	-----	--------	-------	----