

SAE Serpiuto



2024/2025

Bicat Abdulkerim, Gezault Matéo, Kerrien Bastian

Groupe 2A (Le_SEL)



- 1) La stratégie de notre IA
- 2) La complexité des algorithmes
- 3) Ce qui à été réalisé et ce qui ne marche pas

La stratégie de notre IA

La stratégie de notre IA est de récupérer les points de façon optimisée. Afin d'effectuer sa tâche, notre serpent possède son propre calque qui se déplace avec lui. De cette manière, le serpent peut effectuer ses tâches de manière optimisée, sans se préoccuper des cases en dehors de sa zone. La stratégie repose sur plusieurs points. Premièrement, d'atteindre une certaine taille rapidement, notamment en récupérant toutes les boîtes de valeur 2. Certaines conditions ont été rajoutées pour s'adapter en fonction de la situation du serpent, comme récupérer une protection si sa taille dépasse un certain nombre ou encore si sa tête a un grand nombre de points, essayer d'aller au maximum sur les boîtes de multiplication ou encore d'addition. Nous avons par conséquent fait une analyse des différents bonus et sommes venus à la conclusion que les bonus de mange-mur et de surpuissance n'étaient pas utiles à notre stratégie, en particulier le mange-mur qui n'a aucune propriété offensive ou défensive. Notre IA aurait aussi la possibilité de ne pas rester proche d'un autre serpent, surtout si celui-ci possède une tête plus grosse que la nôtre. On peut facilement avoir cette information, car en plus d'avoir un calque autour du serpent, il y a aussi un calque de l'arène, qui permet à notre programme d'avoir des informations sur la partie sans forcément impacter les décisions du serpent.



La complexité des algorithmes

La complexité des algorithmes :

Fonctions :	Complexité :
directions_possibles	$O(1)$
objets_voisinage	$O(N) \times O(K) \times O(P)$ ($O(N)$ pour la boucle while, $O(K)$ pour le range et $O(P)$ pour le nombre de case)
calque_arene	$O(N^2)$
calque_serpent	$O(N^2)$
valeur_calque_arene	$O(N^2)$
valeur_calque_serpent	$O(N)$
jamais_arret	$O(1)$
get_temps_restant	$O(1)$
coordonnées boussole	$O(1)$
get_serpent	$O(N)$
directions_possibles_depuis_coords	$O(1)$
meilleure_route	$O(N^2)$
inondation	$O(N^3)$
genre_combinaison	$O(N^2)$
determine_route	$O(N^2)$
mon_IA2	$O(N^3)$

Ce qui à été réalisé et ce qui ne marche pas



Pour commencer, nous avons tout d'abord complété et testé toutes les fonctions des fichiers suivants : serpent, arène, case, partie, client. Ensuite, nous sommes passés à l'affichage du jeu avec la fonction Random qui était fournie.

Afin de créer notre propre IA comme demandé dans le projet, nous avons fait de multiples fonctions (tableau ci-dessus).

Cependant, avec notre programme, il y a quelques problèmes. Par exemple, notre serpent basant ses déplacements sur les positions des boîtes, il est incapable de se déplacer tout seul lors des premiers tours. C'est pourquoi pour les deux premiers tours, notre serpent se déplace grâce à la fonction Random. De plus, le code de notre IA n'étant pas terminé et comportant encore des bugs, la fonction Random est utilisée dans les situations dans lesquelles il y a des erreurs dans notre programme. De cette manière, on évite le fait que notre serpent se téléporte partout dans l'arène en perdant des points.

La finalité du projet n'est pas exactement ce qui était voulu à la base. Cependant, l'idée était prometteuse et aurait pu très bien fonctionner si certaines de nos fonctions avaient été corrigées à temps.