Flavien GALBEZ
Alexandre GHAHARIAN
Corentin NEAUD
Jean-Baptiste MESNIL

Groupe 9
TD J

Rapport IA

Explication de l'optimisation :

Notre programme peut tourner en profondeur 4 assez aisément avec un interpréteur Python comme Pypy mais pour ça nous avons dû faire quelque choix (sans Pypy, il est conseillé d'utiliser une profondeur 3 => décommenter la première ligne du programme et commenter la seconde) :

- Nous permettons à l'IA de seulement jouer dans une case voisine à une case qui a déjà été joué
- Nous avons pris le soin de faire un élagage alpha-beta avec notre minmax
- Nous avons développé notre heuristique
- Nous avons essayé de limiter les fins de boucle inutiles avec des break ou d'autre méthode

Détails importants :

- Nous avons pris soin de faire respecter les règles que ce soit pour l'humain ou pour l'IA
- Le choix de commencer en premier est demandé en début de partie
- Les saisies sont sécurisées
- Quand les heuristiques sont identiques la case est choisie aléatoirement parmi celle possédant cette heuristique afin que l'IA soit moins prévisible tout en respectant la logique du minmax
- La case pour le deuxième tour de l'IA quand se dernier commence est choisit aléatoirement en faisant attention donc de respecter les règles ou de ne pas choisir la case choisit au premier tour par le joueur humain
- Notre IA va privilégier la défense mais attaque quand le moment est opportun (cf **heuristic**)
- On fait attention dans l'heuristique si le coup peut mener à une victoire (en effet un groupe en étaux de 4 cases ne pourra jamais donner de victoire, donc ça ne sert à rien de jouer à cet endroit pour l'IA)

Explications des fonctions :

actionsJoueur : Permet d'obtenir le tableau des actions disponibles pour le joueur humain

actions : Permet d'obtenir le tableau des actions disponibles pour l'IA. Permet de réduire le nombre de calcul en inspectant seulement les voisins des cases déjà remplies contrairement à **actionsJoueur**

TerminalTest : Permet de tester si le jeu est terminé (première valeur=vrai/faux) et donne une valeur selon le gagnant +1/-1 ou 0 si égalité ou non terminé (deuxième valeur=-1/0/+1). Les boucles de cette fonction ont été écrites de sorte à s'arrêter dès qu'on obtenait un résultat.

heuristic : Cette fonction permet de classer l'utilité de l'état. Plus le chiffre est élevé mieux c'est pour l'IA, plus il est bas mieux c'est pour le joueur humain. On y compte le nombre de cases voisine suivante et précédente (un compteur pour celle qui sont voisin directe et un autre pour les plus éloigné mais ayant un plus faible impacte sur le résultat final). On fait cela pour les lignes, les colonnes et les diagonales et on prend le soin de ne pas compter les suites inférieures à 5 ne pouvant pas aboutir sur une victoire car les cases sont occupées par le joueur adverse.

Flavien GALBEZ
Alexandre GHAHARIAN
Corentin NEAUD
Jean-Baptiste MESNIL

Groupe 9
TD J

Minmax : minmax classique avec élagage alpha beta avec la particularité de garder en mémoire tout les actions permettant d'aboutir au meilleur choix et de choisir au hasard parmi celle-ci.

MAX : Le même max que dans l'algorithme donnée en TD avec un élagage alpha beta.

MIN : Le même min que dans l'algorithme donnée en TD avec un élagage alpha beta.

AfficherGrille : Permet d'afficher la grille de Gomoku

tourJoueur : Permet au joueur de jouer son coup de manière sécurisé. Si on est au début dans le cas où le joueur humain a commencé (debut=True) alors il ne pourra pas saisir sa case dans un carré de 7 au tour du centre.

jeuGomoku : Programme du jeu. Il demande à l'humain s'il veut commencer ou non et fait respecter les règles de départ du Gomoku. Si l'IA joue en première, pour son second tour, on place aléatoirement sa case sans faire appel à minmax (de toute façon inutile ici) en faisant attention qu'elle ne se trouve pas dans le carré de 7 autour du centre ou sur la case déjà saisie par l'humain. Après les deux premiers tours la partie continue normalement jusqu'à la victoire de l'IA ou de l'humain en faisant appel à **MinMax** pour l'IA et à **tourJoueur** pour l'humain