

## A.2 fiche de TP

### A.2.1 Consignes

Un rapport doit être soigné. Quelques remarques utiles dans la rédaction du rapport sont également proposées :

- Le rapport doit obligatoirement contenir une *introduction*, une *conclusion*, et différentes sections (un sommaire pourrait aider le lecteur).
- Une section *analyse* décrit la modélisation (les structures de données) et les principaux algorithmes développés. Proposer une analyse claire de votre algorithme (le code doit obligatoirement se trouver en annexe).
- Une section *Validation* indiquera les résultats numériques obtenus. Proposer par exemple des résultats numériques selon différents critères (temps de calcul, nombre de noeuds générés, etc), critères préalablement expliqués et justifiés. Il est possible de comparer les différentes stratégies IA les unes contre les autres, une stratégie IA contre un joueur humain.
- Une section discussion dont l'objet sera de discuter des résultats obtenus ; et de souligner l'efficace des stratégies employées (où se positionne le joueur humain parmi ces stratégies). Dans cette section, il est utile de souligner les principales difficultés et limites de votre travail (pensez à des perspectives pour répondre à ces limites).
- Les figures sont numérotées et référencées dans le texte
- Une bibliographie (des livres, des articles) référencée dans le document aidant votre argumentation.
- les pages internet ne sont pas réellement considérées dans une bibliographie (si cela s'avère nécessaire, les mettre en pied de page).

### A.2.2 Sujet

L'objectif consiste à développer les principaux algorithmes des jeux à deux joueurs sur un jeu de plateau. Nous prendrons le jeu d'*Othello*, qui a l'avantage de la simplicité des règles tout en proposant très rapidement une explosion combinatoire (voir cours). Plus précisément, à la fin du TP, le joueur IA devra être capable de jouer à *Othello* contre un joueur humain, ou contre un autre joueur IA.

**Présentation :** *Othello* est un jeu de réflexion qui se joue sur un damier de 64 cases, avec des pions de deux couleurs. Le but d'une partie au final est d'obtenir le plus grand nombre de pions de sa couleur posés sur ce damier. La position de début de partie est fixée (voir cours) : les pions blancs débutent la partie.

**Algorithme :** La conception d'un algorithme permettant à l'ordinateur de jouer constitue le principal défi. La première étape consiste ainsi à élaborer l'algorithme min-max pour ce jeu. Le système IA regarde toutes les possibilités du jeu, évalue la qualité des réponses de son adversaire. Ce dernier essaie de

maximiser la qualité de la position sachant que son adversaire fera tout pour minimiser. Il est également nécessaire d'élaborer une fonction d'évaluation permettant de mesurer la solution. Notons que la version proposée dans le cours doit être adaptée en fonction de vos besoins<sup>77</sup>. Au vu de cette explosion combinatoire, il apparaît nécessaire de définir :

- profondeur maximale de recherche : par habitude, il est classique de considérer un demi-coup comme l'action d'un joueur. Une profondeur de 1 signifie que l'on étudie l'arbre de recherche avec les actions possibles du joueur ainsi que celles de son adversaire
- évaluation par un *time-out* (pour que chaque joueur puisse jouer dans des conditions contraignantes), etc.
- un moyen permettant de déterminer les états déjà traités
- la possibilité de reprendre l'analyse déjà construite au tour précédent par le même joueur. Si vous choisissez de faire jouer deux systèmes IA, les deux arbres de raisonnement doivent être distincts.

**Améliorations de l'algorithme :** Plusieurs améliorations de l'algorithme min-max (très coûteux en fonction de la profondeur de l'arbre) existent dans la littérature, dont le plus connu est  $\alpha-\beta$ , qui permet des gains en rapidité de recherche. La version *NegaMax* possède un code plus compact que min-max. Pensez à évaluer les différentes versions de votre algorithme avec les mêmes critères.

**Stratégie de recherche :** Il existe plusieurs stratégies de recherche modifiant la fonction d'évaluation. Nous en proposons quatre (vous pouvez proposer d'autres stratégies) :

- *positionnel* : prise en compte des poids statiques du tableau (Cf. cours). L'évaluation est la différence entre les poids associés des deux joueurs.
- *absolu* : prise en compte de la différence du nombre de pions.
- *mobilité* : maximise le nombre de coups possibles et minimise les coups de l'adversaire, tout en essayant de prendre les coins.
- *mixte* : le jeu est divisé en trois phases où les stratégies peuvent différer : (i) en début de partie (20 à 25 premiers coups), le joueur IA choisit une stratégie de type '*positionnel*', au milieu de partie, sélectionne une stratégie de '*mobilité*' ; et enfin en fin de partie (10 à 16 derniers coups), sélectionne la stratégie '*absolu*'.

Dans ce contexte, pensez à évaluer les différentes stratégies en les faisant jouer les unes contre les autres (sur quelques matches). De même, vous pouvez envisager de faire jouer votre système IA contre un joueur humain. Vous pouvez également proposer d'autres critères d'évaluation ou extensions de votre algorithme, par exemple :

---

77. Cette adaptation devra être justifiée dans votre rapport.

- faire en sorte que l'ordinateur s'améliore au cours des parties en prenant en compte les erreurs qu'il a pu faire ou en apprenant à jouer contre lui-même (apprentissage par renforcement)
- mémoriser les coups pour pouvoir re-jouer la partie
- indiquer le score de la fonction d'évaluation dans un fichier .txt

**Note :** l'interface graphique n'est pas essentielle dans le cadre de ce TP.