

Rapport – AI Game Programming

Bot Awalé (Protocole stdin/stdout)

MAZOUZ Mohamed El Amine

Table des matières

1. Introduction	3
2. Protocole de communication.....	3
2.1 Entrées du bot	3
2.2 Sorties du bot.....	3
3. Représentation du jeu	4
3.1 Plateau	4
3.2 Scores et fin de partie	4
4. Algorithme de décision.....	4
4.1 Génération des coups.....	4
4.2 Minimax avec élagage alpha-bêta	5
4.3 Fonction d'évaluation	5
5. Tests et validation	6
5.1 Tests locaux	6
5.2 Compatibilité	6
6. Conclusion.....	6

1. Introduction

Dans le cadre du module *AI Game Programming*, l'objectif de ce projet est de concevoir un bot capable de jouer automatiquement au jeu d'Awalé selon les règles fournies.

Le bot doit interagir avec un arbitre externe via une communication standard stdin/stdout, sans interface graphique, et produire uniquement les coups joués.

- L'accent est mis sur :
- la conformité au protocole,
- la robustesse du bot,
- et l'utilisation d'une stratégie d'intelligence artificielle pour la prise de décision.

2. Protocole de communication

La communication entre l'arbitre et le bot repose exclusivement sur l'entrée et la sortie standard.

2.1 Entrées du bot

Le bot reçoit une ligne à chaque tour :

- START : indique que le bot est joueur 1 et doit jouer le premier coup.
- Un coup adverse (ex : 3R, 12TB) : indique que le bot est joueur 2 ou que le tour suivant commence.
- END : indique la fin de la partie.

Aucun autre message n'est attendu.

2.2 Sorties du bot

À chaque tour, le bot doit afficher exactement un coup valide, sous la forme :

- nR, nB
- nTR, nTB

où $n \in [1, 16]$.

Le bot n'affiche rien d'autre sur stdout (pas de messages de debug).

Les sorties sont suivies d'un flush pour garantir la réception immédiate par l'arbitre.

3. Représentation du jeu

L'état du jeu est maintenu en interne par le bot.

3.1 Plateau

Le plateau est représenté par trois tableaux de taille 16 :

- $r[i]$: graines rouges du trou i
- $b[i]$: graines bleues du trou i
- $t[i]$: graines transparentes du trou i

Les joueurs possèdent alternativement les trous :

- Joueur A : indices pairs
- Joueur B : indices impairs

3.2 Scores et fin de partie

Le bot maintient également :

- les graines capturées par chaque joueur,
- le nombre de coups joués.

Les conditions de fin de partie sont :

- capture ≥ 49 graines,
- moins de 10 graines sur le plateau,
- ou limite de 400 coups.

4. Algorithme de décision

4.1 Génération des coups

À chaque tour, le bot génère l'ensemble des coups valides possibles à partir des trous qu'il contrôle :

- coups simples (R, B),
- coups avec graines transparentes (TR, TB).

Chaque coup est validé avant d'être considéré.

4.2 Minimax avec élagage alpha-bêta

Le bot utilise un algorithme Minimax avec élagage alpha-bêta afin d'explorer l'arbre des coups possibles.

- Profondeur maximale : 6
- Alternance joueur / adversaire à chaque niveau
- Élagage des branches non prometteuses pour réduire le temps de calcul

Cet algorithme permet d'anticiper les réponses adverses tout en respectant la contrainte de temps par tour (2 secondes).

4.3 Fonction d'évaluation

Lorsque la profondeur maximale est atteinte ou qu'une position terminale est détectée, une fonction d'évaluation heuristique est utilisée.

Elle prend en compte :

- la différence de graines capturées,
- le nombre de graines contrôlées sur le plateau,
- les risques et opportunités de capture (trous contenant 2 ou 3 graines),
- la pression de famine sur l'adversaire.

Cette évaluation permet de comparer efficacement les positions intermédiaires.

5. Tests et validation

5.1 Tests locaux

Le bot a été testé localement à l'aide d'un arbitre Java lançant deux processus concurrents et simulant une partie complète.

Exemple de test :

```
java Arbitre bot.exe bot.exe 3
```

Ces tests ont permis de vérifier :

- la conformité du protocole stdin/stdout,
- l'absence de blocage ou de timeout,
- la stabilité du bot sur des parties complètes.

5.2 Compatibilité

Le bot est compatible avec :

- des matchs entre bots étudiants,
- des arbitres externes respectant le protocole "dernier coup seulement".

6. Conclusion

Ce projet a permis de mettre en œuvre un bot d'Awalé autonome reposant sur une stratégie Minimax avec élagage alpha-bêta.

Le bot respecte strictement le protocole de communication imposé et est capable de jouer des parties complètes sans intervention humaine.

Des améliorations possibles incluraient :

- un ajustement dynamique de la profondeur selon le temps restant,
- des heuristiques plus avancées basées sur l'analyse statistique des coups.