

# Intelligence artificielle

## Master 1 Informatique

### Projet – Awélé

#### Indications générales

Date de rendu : 3 mars 2026 avant 8 h 00

Travail en groupe de deux étudiants

Les fichiers seront déposés sur ARCHE dans une archive `nom.du.bot.zip`

#### Le jeu de l’Awélé

L’Awélé est un jeu traditionnel africain. Le jeu connaît de nombreuses appellations et de nombreuses variantes. C’est un jeu de stratégie combinatoire abstrait qui voit s’affronter deux joueurs jouant à tour de rôle, sans information cachée et sans hasard. Ce projet considère la variante la plus utilisée du jeu.

Un plateau de jeu comportant douze trous (deux lignes de six trous, une pour chaque joueur) est placé entre les joueurs. Dans chaque trou, on place quatre graines. L’objectif du jeu est de récolter plus de graines que son adversaire.

À tour de rôle, chaque joueur va prendre toutes les graines contenues dans un trou (non vide) de son côté du plateau et les égrener une par une dans les trous suivants de sa ligne, puis de la ligne de son adversaire en suivant le sens de rotation (généralement anti-horaire).

Si la dernière graine tombe dans un trou du côté de l’adversaire et que ce trou contient maintenant deux ou trois graines, le joueur récupère ces graines qui s’ajoutent à son score. De plus, si le trou précédent est également du côté de l’adversaire et que celui-ci comporte également deux ou trois graines, le joueur les récupère également. On continue à regarder le trou précédent jusqu’à arriver au camp du joueur actif ou jusqu’à ce qu’il y ait un nombre différent de deux ou trois graines.

Si l’on égraine 12 graines ou plus, le trou d’où proviennent ces graines est « sauté » durant l’égrenage. Certains coups ne sont pas valides, il ne faut pas « affamer » son adversaire : si l’adversaire n’a plus de graine de son côté, il est obligatoire de jouer un coup qui place au moins une graine de son côté. De plus, si un coup permet de récupérer toutes les graines du côté de l’adversaire, le coup est valide, mais aucune graine n’est récupérée.

La partie s’arrête quand il reste moins de six graines au total sur le plateau ou lorsqu’un joueur ne peut plus jouer de coup valide (auquel cas, il récupère toutes les graines encore de son côté).

#### Le projet

On cherche à construire un moteur d’intelligence artificielle permettant de jouer à l’Awélé (un *bot*), en utilisant un ou plusieurs algorithmes.

L’objectif est de construire un modèle de prise de décision permettant de choisir le plus efficacement possible le meilleur coup à jouer en fonction d’une situation donnée.

La solution proposée utilisera un algorithme d’intelligence artificielle, en évaluant la situation de jeu et en utilisant des connaissances extraites à partir d’une base de données de coups joués. Les techniques d’intelligence artificielle employées ne sont pas limitées à celles étudiées en cours, il est même vivement recommandé de faire une courte recherche bibliographique pour trouver des algorithmes potentiellement plus efficaces. La solution mise en œuvre pourra utiliser diverses méthodes d’analyse de données, de recherche opérationnelle, d’optimisation, etc. Les prises de décision ne devront en revanche pas être directement écrites dans le code.

#### Les données

On dispose d’un ensemble de données correspondant à 303 situations de jeu observées au cours de plusieurs parties entre un joueur expérimenté et un joueur novice. Le joueur expérimenté a remporté toutes les parties.

Chaque observation est décrite par 14 variables. Les 12 premières correspondent à la situation du plateau de jeu (nombre de graines dans les trous, de gauche à droite pour le joueur actif et de droite à gauche pour l’adversaire). La 13<sup>e</sup> variable est la décision prise par le joueur actif. La dernière variable indique si le joueur a gagné ou perdu la partie.

A6	A5	A4	A3	A2	A1
J1	J2	J3	J4	J5	J6

## Le code existant

Le projet devra être développé en langage Java en se basant sur les classes fournies. Tout le code développé se trouvera dans un **unique package** nommé `awele.bot.competitor.nomdubot` qui devra être créé. Ce *package* devra contenir **une seule classe étendant** `CompetitorBot`. L'**unique** constructeur de la classe ne devra comporter **aucun** paramètre et exécuter les méthodes `addAuthor` (pour chaque auteur) et `setBotName` afin d'identifier chaque étudiant du groupe et pour donner un nom au *bot*, et ainsi faciliter l'évaluation du projet. Il est autorisé d'utiliser des bibliothèques Java existantes sur les algorithmes d'intelligence artificielle afin de pouvoir appliquer des méthodes d'apprentissage complexes. D'autres classes peuvent bien entendu être créées dans le *package*.

Les étudiants pourront s'inspirer des packages existants et des classes qui étendent la classe `Bot`.

Les classes fournies contiennent un moteur de jeu d'Awélé permettant de voir s'affronter deux *bots*. La classe `Board` contient les informations sur l'état actuel du plateau de jeu. La classe `Main` charge l'ensemble des *bots*, puis les fait s'affronter deux par deux. La classe `CompetitorBot` étend la classe `Bot`. Celle-ci contient trois méthodes abstraites. La méthode `learn` est exécutée une fois au chargement du *bot*, afin de réaliser l'apprentissage et la préparation du *bot*. La méthode `initialize` est exécutée une fois avant chaque partie. La méthode `getDecision` doit retourner un tableau de six `double` correspondant à l'efficacité supposée de chacun des six coups possibles. Le coup joué par le *bot* sera le coup valide dont la valeur est la plus forte. Par exemple, si la méthode retourne le tableau `{0.95; 0.23; 0.67; 0.42; 0.35; 0.12}` le *bot* jouera les graines du premier trou (valeur la plus élevée) sauf si ce coup est invalide auquel cas il jouera les graines du troisième trou, etc.

## Le travail demandé

Il est demandé de réaliser, en langage de programmation Java, **un unique** « *bot* » (joueur artificiel) permettant de jouer à l'Awélé. L'apprentissage est limité dans le temps. Il n'est pas autorisé de faire du calcul distribué, que ce soit en local (multi-threads) ou à distance (serveurs de calculs), de charger des informations autres que les données fournies (nouvelles données, modèle de classification, etc.), d'instancier les *bots* adverses. Il est autorisé de créer des *bots* qui étendent la classe `Bot`, capable de jouer à l'Awélé mais qui ne participent pas au championnat, pour entraîner l'IA du *bot*. Une fois le championnat commencé, il est interdit de modifier les paramètres du *bot*, ni de poursuivre l'apprentissage, la méthode `finish` ne peut être utilisée que pour réinitialiser l'instance.

Il conviendra d'utiliser l'encodage UTF-8 (si les chaînes de caractères comportent des caractères spéciaux, par exemple). Le code devra être rendu dans une archive `.zip`. **Seul** le dossier contenant le *package* `awele.bot.competitor.nomdubot` devra être envoyé. **Aucune** modification du code existant n'est tolérée. Si le projet nécessite l'utilisation d'une ou plusieurs bibliothèques, les fichiers `.jar` devront être rajoutés dans l'archive.

Un rapport de dix pages maximum est également demandé. Ce rapport devra être convenablement structuré (page de garde, introduction, conclusion, etc.). Il développera la recherche bibliographique réalisée et décrira le principe de la méthode d'apprentissage mise en œuvre. Les noms des auteurs et le nom du *bot* seront rappelés sur la page de garde du rapport et dans le nom de fichier. Le rapport devra être rajouté à l'archive envoyée.

## La notation

La notation sera décomposée en quatre parties :

- l'ingéniosité et la pertinence de la solution mise en œuvre (9 points) ;
- la qualité de la programmation en langage Java (3 points) ;
- la qualité de la rédaction du rapport (3 points) ;
- l'efficacité face aux solutions proposées par les autres groupes d'étudiants (5 points).

Les *bots* des différents groupes d'étudiants seront comparés entre eux sous la forme d'un championnat : chaque *bot* affrontera chaque autre *bot*, lors de deux parties (chaque *bot* commençant une partie), la somme des graines sur les deux parties déterminant le vainqueur. Une victoire sera récompensée par trois points, une égalité par un point. Chaque affrontement est alors répété 100 fois. Ces points serviront à réaliser un classement qui déterminera la note entre 1 (dernier du classement) et 5 (premier du classement). Une note de 0 sera attribuée aux groupes disqualifiés (*bots* non conformes, envoi de plusieurs *bots*, mauvaise organisation des fichiers, projet rendu en retard, groupes de trois étudiants ou plus, etc.)

## Remarques

Les questions concernant le projet sont à poser par e-mail à l'adresse : `alexandre.blansche@univ-lorraine.fr`.

Malgré toute l'attention qui a été apportée à la programmation du moteur de jeu, le programme peut contenir des bugs. Ne pas hésiter à le signaler rapidement si tel est le cas.