

M4105 - Intelligence Artificielle

À la pêche aux moules !

Deadline - Cette information vous sera fournie plus tard.

1 Contexte et travail à réaliser

Vous vous réveillez sur une plage paradisiaque : sable fin, soleil, douce odeur iodée. Vous pensez reconnaître Calais. Vous vous levez et commencez à parcourir la plage entourée de dunes. Au détour d'un virage, vous trouvez une moule puis une bière et un cornet de frites. Plus aucun doute, vous êtes bien à Calais !

Le but de cette épreuve est de développer une intelligence artificielle (ou au moins un truc qui pourrait passer pour une intelligence) pour diriger un personnage dans un espace clos (que nous appellerons "labyrinthe") rempli de sable, dunes, moules, frites et bières. Votre programme¹ devra être capable d'atteindre le but suivant : récupérer un maximum de points en ramassant des moules. Évidemment, votre personnage ne sera pas le seul dans cet espace. Il affrontera l'intelligence d'une ou plusieurs autres équipes en tour à tour. Afin de vous aider dans votre mission, divers bonus seront disséminés tout au long du parcours.

L'interface du jeu contient deux parties : le labyrinthe dans la partie gauche et les scores à droite. Le labyrinthe peut se présenter sous diverses formes et complexités :

- La figure 1 montre un labyrinthe **parfait**.
- La figure 2 montre un labyrinthe assez ouvert comprenant quelques dunes.



FIGURE 1 – Labyrinthe parfait.

Comme vous pouvez le voir sur les figures 1 et 2, les labyrinthes sont composés de blocs différents :

- du sable sur lequel vous pouvez marcher,
- les blocs marron foncé représentent des dunes quasi infranchissables donc des blocs sur lesquels vous ne pouvez pas vous positionner,
- des frites (bloc bonus) qui vous donnent la patate et vous permettront d'avancer de deux cases dans une direction, et ce, même s'il y a une dune qui vous sépare du point d'arrivée (vous sautez donc par-dessus la dune avec ce bonus). Lorsque vous utiliserez ce bonus, vous devrez bien faire attention à ce que le point

1. dont le langage est libre mais on vous fourni un client fonctionnel en java

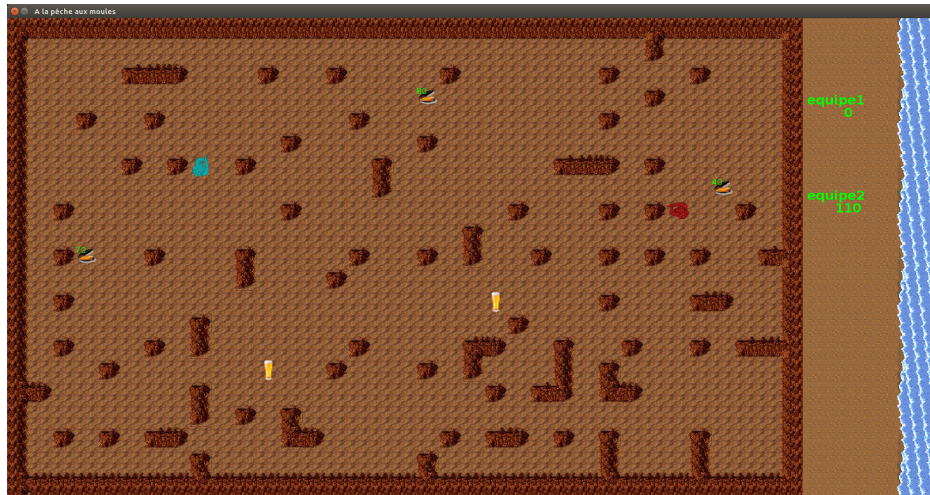


FIGURE 2 – Labyrinthe présentant une surface assez ouverte avec quelques dunes.

d'arrivée soit une case sur laquelle vous pouvez marcher². Dans le cas contraire, le personnage avancera d'une case dans la direction donnée si cela est possible. Évidemment, les bonus et/ou moules sur le passage seront ramassés.

- des bières (bloc bonus) qui vous permettent de jouer trois coups à la suite (mais ne permet pas de franchir les dunes). Évidemment, les bonus et/ou moules sur le passage seront ramassés.
- des moules qui vous permettent d'enregistrer des points. Les moules ont un nombre de points aléatoires décidés en début de partie.
- plusieurs personnages dont un sera déplacé grâce à votre programme.

Les blocs frites, bières et moules disparaissent lorsqu'un joueur passe dessus. Ils sont automatiquement transformés en sable. Les frites et bières sont collectées et conservées pour pouvoir être utilisées lorsque le joueur le souhaite.

2 Fonctionnement et dialogue avec le serveur

Le programme client qui vous est fourni est une non intelligence artificielle qui va toujours vers l'Est. Il fonctionne de la manière qui suit. Dans un premier temps, un socket est créé qui ouvre une connexion avec le serveur³. Une fois cette connexion ouverte, le client envoie le nom d'équipe et passe en écoute pour récupérer son numéro de joueur. Lorsque tous les joueurs sont connectés et que c'est le tour du client, les informations complètes du labyrinthe lui sont envoyées. Il décide ensuite quel coup jouer et l'envoie au travers du socket (ici la décision est toujours d'aller vers l'Est).

Le programme suit cette architecture :

```
DEBUT DE PROGRAMME
  Création du socket
  Envoi du nom d'équipe
  Réception du numéro de joueur
  Boucle
    Réception des informations du labyrinthe
    Calcul du prochain coup à jouer (votre IA - pour le moment, toujours vers l'Est)
    Envoi du coup
  Fin De Boucle
FIN DE PROGRAMME
```

Le jeu s'arrête lorsque le serveur envoie le mot FIN à la place des informations sur le labyrinthe. Cela signifie que toutes les moules ont été ramassées, que le nombre de tours maximum a été dépassé ou que tous les joueurs ont passé leur tour.

2. du sable, un bonus ou une moule

3. Par défaut, le port utilisé est le 1337

2.1 Réception des données

Par exemple :

A la pêche aux moules

equipe1
0

equipe2
0

equipe3
0

La taille du plateau correspond à la largeur et à la hauteur séparées par le caractère 'x'.

- La lettre "D" pour une dune.
- La lettre "S" pour du sable.
- La lettre "F" pour un bonus frite.
- La lettre "B" pour un bonus bière.
- Un nombre pour une moule. Le nombre représente le nombre de points que vous gagnerez en ramassant la moule.

Les informations concernant les joueurs contiennent : Le nombre de joueurs dans le labyrinthe suivi d'autant de couples X,Y séparés par des tirets donnant leur position.

3

L'exemple précédent représente un labyrinthe de 13 cases horizontalement par 11 cases verticalement. Les 143 (13×11) codes suivants représentent la structure du labyrinthe. Finalement, il est indiqué qu'il y a 3 joueurs. Le joueur 0 se trouve en (1,1), le joueur 1 se trouve en (11,9) et le joueur 2 se trouve en (11,1). Ce labyrinthe est illustré en figure 3.

Remarque : Les indices commencent à 0. La case se trouvant en (0,0) sera toujours une dune.

Attention ! En fonction des langages, il vous est parfois demandé de donner le nombre de caractères à lire sur le socket. Le labyrinthe pouvant être assez grand, assurez vous bien de recevoir une chaîne de caractères suffisamment longue.

2.2 Envoi des données

Les coups que vous allez devoir envoyer seront représentés par une lettre simple pour les mouvements simples et un ensemble de plusieurs lettres séparées par des "-" lorsque vous utiliserez des bonus.

Les messages simples sont les suivants :

- "N" pour aller au Nord (en haut)
- "S" pour aller au Sud (en bas)
- "E" pour aller à l'Est (à droite)
- "O" pour aller à l'Ouest (à gauche)
- "C" pour rester là où vous êtes (Centre). A utiliser si vous voulez passer votre tour.

Les messages composés (si vous utilisez des bonus que vous auriez récoltés) sont les suivants :

- pour utiliser une frite qui vous permet d'avancer de deux cases dans une direction même si une dune vous sépare de la position d'arrivée : envoyez le code "F" suivi d'un "-" et de la direction dans laquelle vous voulez aller. Par exemple, si vous envoyez le code "F-S", vous allez avancer de deux cases vers le sud même si une dune se trouve juste en dessous de vous. Rappel : ceci n'est possible que si la position d'arrivée est une case sur laquelle vous pouvez marcher.
- pour utiliser une bière qui vous permet de faire trois déplacements au lieu d'un seul : envoyez le code B suivi d'un "-" et de trois directions (et exactement trois) séparés également par des "-". Par exemple, si vous envoyez le code "B-E-E-S", vous allez effectuer les trois déplacements (Est, Est puis Sud) en un seul tour. Rappel : Seuls les coups valables seront effectués.

Attention à bien utiliser vos bonus lorsque c'est possible seulement sinon ils seront perdus.

A noter : Deux joueurs peuvent se trouver sur la même case au même moment.

A noter également qu'en fonction du langage que vous utiliserez, il faut envoyer ou non un retour chariot ('\n') pour que l'envoi sur le socket soit effectif.

3 Fichiers fournis

Sur Moodle, vous avez dû récupérer trois archives : une contenant la bibliothèque MG2D, une archive contenant le serveur et une contenant le client. Avant de compiler et lancer le serveur, soyez assuré d'avoir installé MG2D.

Le serveur doit être compilé de manière classique :

```
javac ServeurEpreuve1.java
```

Pour lancer le serveur, il vous suffit d'exécuter cette classe. Ajoutez l'option -h à la fin pour avoir les différentes options possibles.

```
java ServeurEpreuve1
```

Afin de vous aider et de vous avancer dans le développement, un client fonctionnel mais à compléter vous est fourni.

La compilation est classique :

```
javac Client.java
```

Pour lancer le client, vous devez fournir 3 arguments à passer sur la ligne de commande : l'adresse ip du serveur, le port et le nom de l'équipe. Si vous lancer le serveur et le client sur le même ordinateur (ce que vous devriez faire normalement), la commande sera de la forme suivante :

```
java Client 127.0.0.1 1337 monNomDEquipe
```

Pour ce client à compléter, la commande envoyée au serveur est toujours "E".

A noter : À chaque fois que vous lancer un serveur, un fichier de log est créé. Tous ces fichiers peuvent rapidement prendre beaucoup de place. Supprimez les régulièrement.

Pour tester votre IA, vous pouvez lancer le serveur en mode multijoueur et lancer plusieurs votre IA dans plusieurs terminaux différents.

4 Méthode d'évaluation

Un tournoi ou championnat sera organisé entre les IA que vous allez développer. Le temps de calcul est limité à 2 secondes. De plus, le nombre de tours par joueur sera limité à 500.

Pour chaque rencontre de 4 équipes (ou 3), il y aura 4 cartes x 4 orientations (ou 3) = 16 matchs (ou 12). Chaque match gagné donnera six points, une seconde place donnera quatre points, une troisième place donnera deux points et rien pour le dernier.

Suite au tournoi ou championnat, les deux meilleures IA se verront octroyer la note de 20. Les 3ème et 4ème seront évalués de manière classique et recevront un gros bonus Les 4 suivants seront évalués de manière classique et recevront un bonus. Finalement, tous les autres seront évalués de manière classique.

5 Dépôt de votre programme

Pour le dépôt de votre programme, vous déposerez simplement une archive sur Moodle contenant tous les fichiers nécessaires à la compilation de votre programme. L'archive doit contenir le Main.java fourni en exemple ou un README qui explique comment lancer votre IA. La deadline est fixée à **euuh.... plus tard**.