RÉGION ACADÉMIQUE CORSE
Liberté
Égalité
Fraternité

Ce document est une notice pour le concours de programmation en langage Python. Le guide pour utiliser le kit de démarrage est à distribuer aux élèves.

Le concours se déroule du **lundi 2 décembre 2024 au lundi 13 janvier 2025.**

Comptes pour accéder à la plateforme

Des comptes ont été créés pour chaque élève inscrit : des couples « adresse mail / mot de passe » seront fournis le 2 décembre au plus tard. Quelques comptes supplémentaires, de secours, seront également fournis.

C'est à l'enseignant de garder la correspondance entre l'adresse mail fournie à l'élève participant et son identité.

Une fois connecté, l'élève ne doit changer ni son pseudo (de la forme ia corse 00X 0YZ), ni son école (IA Corse), ni son mot de passe.

Les élèves ne peuvent participer qu'en utilisant les comptes fournis par l'organisation du concours.

Plateforme de programmation

Pour accéder au jeu Olymbits sur la plateforme de programmation, utiliser le lien :

https://www.codingame.com/multiplayer/bot-programming/summer-challenge-2024-olymbits

puis cliquer sur « se connecter » et utiliser les comptes fournis. Lire les courtes instructions puis cliquer sur « rejoindre ».

Avant toute chose, bien lire les instructions de jeu sur la partie gauche de l'écran et bien lire le guide pour le kit de démarrage à la fin de ce document.

Le kit de démarrage, écrit en langage Python, est déjà copié dans la partie droite de l'écran, dans l'encart de programmation. Le fichier .py du kit a été également fourni par mail.

Classement

Le classement final sera établi en fonction tout d'abord de ligue atteinte (légende, or, argent, bronze, bois) puis du classement au sein de la ligue.

Le classement final sera arrêté le **lundi 13 janvier à 19h00.**

Attention ! Pour que le classement final d'un élève soit pris en compte :

- le langage Python doit être le seul langage de programmation utilisé ;

- le programme proposé doit être le résultat d'un travail personnel de l'élève ;

- le mot de passe du compte ne doit pas avoir été changé afin que celui-ci reste accessible pour la vérification

Remarques pratiques importantes

- Ne pas hésiter à passer en mode plein écran (bouton au-dessus du code à droite) et à adapter le niveau de zoom pour mieux voir.

- Le bouton *« Lancer mon code »* permet, comme son nom l'indique, de tester son code. Les paramètres de jeu sont modifiés à chaque fois, ce qui peut être intéressant quand le code est déjà assez efficace. C'est le seul moyen de sauvegarder le code sur la plateforme de jeu : ne pas oublier de cliquer sur ce bouton avant de quitter la plateforme.

- Le bouton *« Rejouer dans les mêmes conditions »* permet de tester son code en gardant les mêmes conditions de jeu (même graine aléatoire), ce qui est pratique pour visualiser les effets des modifications du code.

- Le bouton *« Tester dans l'arène »* permet de tester son code contre les autres joueurs et contre le bot leader de la ligue, de faire progresser son classement, et, éventuellement de passer dans la ligue de niveau supérieur. Il n'est pas pertinent de l'utiliser tant que le code n'est pas assez efficace.

- Penser à utiliser la fonction `log` .

Guide pour le kit de démarrage

A distribuer aux élèves

Ce kit de démarrage permet de faciliter la prise en main du jeu pour le joueur. Il est important de comprendre que le jeu tourne sur un ordinateur qui, quand c'est le tour de jeu du joueur, lui envoie les informations utiles (qu'on récupère avec la fonction `input`). Le but est d'utiliser ces données pour renvoyer les commandes qu'on veut effectuer pour ce tour de jeu (avec la fonction `print`). Le kit de démarrage est là pour rendre plus naturelle cette gestion des données qui sont envoyées (par exemple en les rangeant dans des listes). En voici une rapide description.

1) Les fonctions auxiliaires

Pour l'instant il n'y a qu'une fonction : la fonction `log` mais il sera possible d'ajouter dans cette section d'autres fonctions.

La fonction `log` ne sert à rien pour le jeu mais c'est de loin la plus utile de toutes car elle permet de repérer ses erreurs. En effet, elle permet d'afficher des variables comme un message d'erreur (donc en rouge) sans que cela soit pris comme une réponse et sans arrêter le jeu (comme si on affichait un commentaire).

Par exemple, si on veut connaître le contenu de la variable `liste_gpu`, à chaque tour de jeu, il suffit de rajouter `log(liste_gpu)` juste au-dessus de la ligne de commentaires *#Ma logique du jeu* dans le kit et le résultat s'affichera en rouge dans les sorties d'erreurs sans perturber le reste.

2) Les variables

Les variables proposées dans ce kit sont simplement des données et des listes dans lesquelles on a rangé les différents objets du jeu ; elles sont mises à jour à chaque tour de jeu.

- `mon_numero` : entier (0, 1 ou 2) qui permet de connaître le numéro de son joueur.
- `liste_scores` : liste qui contient les scores de chaque joueur.
- `liste_gpu` et `liste_reg` : listes contenant les données graphiques de chaque jeu et les registres (positions...), c'est-dire les données propres à chaque jeu.

3) La logique du jeu

C'est dans cette partie qu'il faudra principalement écrire le code. Pour l'instant, le kit propose un exemple simple où le joueur :

- saute s'il y a une haie sur la case suivante ;
- avance d'une case sinon.

Il faut modifier cela pour être plus efficace et passer en ligue supérieure.

Attention : à partir de la ligue suivante, il faudra jouer non plus à un jeu mais à quatre, cependant une seule réponse sera envoyée et devra permettre de jouer à ces quatre jeux en même temps.

À vous maintenant !