Votre premier projet (morpion, puissance 4, mots mêlés ou démineur) vous a permis de découvrir l'ampleur de la tâche à accomplir pour réaliser un premier jeu en programmation. Il faut, en particulier :

- être organisé ;
- se répartir le travail ;
- bien documenter ses fonctions avec notamment le docstring qui récapitule :
  - les préconditions (quels sont les paramètres : leur type, leur contenu, ...)
  - les postconditions (qu'obtient-on en sortie, sous quel type, ...)
  - une aide pour l'utilisateur de la fonction.
- tenir un planning qui permet de rendre un projet fini ;
- ...

Aujourd'hui, nous vous proposons de réaliser un projet identique pour chaque groupe. Mais, contrairement au projet précédent, vous êtes plus libres pour modéliser, concevoir, coder le projet (en Python évidemment).
Par contre toutes les exigences listées ci-dessus (organisation, répartition, documentation, planning, ...) restent valables et seront évaluées. Il y aura également un cahier des charges à respecter, explicité ci-après.
Il est bien sûr possible, mais ce n'est pas obligatoire, de réaliser une interface graphique avec `Pygame` (voir Bonus).

1 L'Awalé : Présentation et mise en place

Source ici.
L'awalé est un jeu de stratégie ancien. Son origine serait l'ancienne Égypte.
C'est un jeu très populaire en Afrique où il symbolise l'importance de l'agriculture.

But du jeu : s'emparer d'un maximum de graines. Le joueur qui a le plus de graines à la fin de la partie l'emporte.

Nombre de joueurs : 2

Préparation : le plateau est placé horizontalement entre les deux joueurs. Chaque joueur place 4 graines dans chacun des 6 trous devant lui.

2 L'Awalé : Déroulement d'une partie, règle du jeu

Plateau : la partie se joue sur un plateau de 2×6 trous, avec éventuellement un septième trou appelé grenier.

Greniers : si votre awalé contient deux greniers (trous plus grands, un par joueur) alors ils ne comptent pas comme des trous, ils ne servent qu'à ranger les graines des joueurs.

Jouer un coup : le 1er joueur prend toutes les graines de l'un des 6 trous se trouvant de son côté et en dépose une dans chaque trou suivant celui qu'il a vidé.

Sens de la partie : antihoraire.

Capture : si la dernière graine est déposée dans un trou de l'adversaire comportant déjà 1 ou 2 graines, le joueur capture les 2 ou 3 graines résultantes. Les graines capturées sont alors sorties du jeu (grenier) et le trou est laissé vide.

Rafle : lorsqu'un joueur s'empare de 2 ou 3 graines, si la case précédente contient également 2 ou 3 graines, elles sont capturées aussi et ainsi de suite.

Boucle : si le nombre de graines prises dans un trou de départ est supérieur à 11, cela implique que l'on boucle un tour, auquel cas, à chaque passage, la case de départ est sautée et donc toujours laissée vide.

Fin du jeu : le jeu se termine lorsqu'un joueur :
- a déjà récolté plus de la moitié des graines en jeu ;
- n'a plus de graines dans son camp (et ne peut donc plus jouer). Dans ce cas l'adversaire capture alors les graines restantes.

3 Cahier des charges :

- la grille de jeu sera stockée dans un dictionnaire (ou tuple nommé voir TP D05) avec 4 clés auxquelles on associera les valeurs ainsi :
  - 2 clés '`trous_J1`' et '`trous_J2`' dont les valeurs associées seront des listes de taille 6 (qui seront actualisées après chaque coup joué par un joueur) contenant le nombre de graines dans chaque trous pour chaque joueur.
  - 2 clés '`grenier_J1`' et '`grenier_J2`' dont les valeurs associées seront des entiers indiquant le nombre de graines récoltés par chaque joueur.
- à chaque tour, le programme demande une case à jouer et le joueur qui joue propose un nombre de 1 à 6, le programme :
  - redemande une case à jouer si la case ne contient pas de graine, le tour recommence ;
  - met à jour le contenu des grilles de chaque joueur et du grenier du joueur qui a joué si celui-ci fait une capture (ou une rafle) ;
  - tant que personne n'a gagné, le tour suivant est proposé ;
- à la fin de la partie, un message indique quel joueur a gagné.

De plus, vous devrez avoir deux modes de jeu, partie à 2 ou mode solo.
Pour le mode solo, vous écrirez une fonction qui permettra à l'ordinateur de jouer un coup aléatoire (au moins, voir bonus) lorsque c'est son tour.

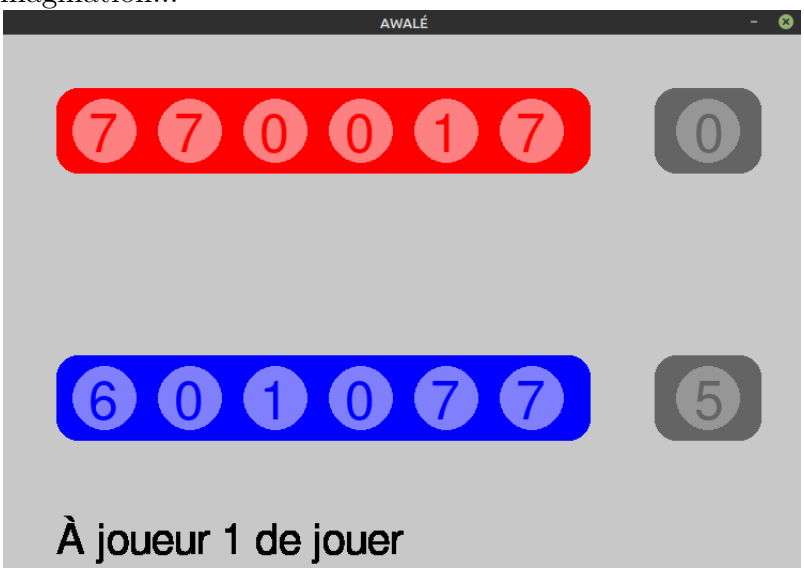
Exemple d'affichage :

```
J1 | 6| 5| 4| 3| 2| 1|   G
     6  6  6  6  1  0    0
-----------------------
     4  4  0  5  5  5    0
J2 | 1| 2| 3| 4| 5| 6|   G


C'est au tour de joueur  2  de jouer
colonne ? 2
J1 | 6| 5| 4| 3| 2| 1|   G
     6  6  6  6  1  0    0
-----------------------
     4  0  1  6  6  6    0
J2 | 1| 2| 3| 4| 5| 6|   G


C'est au tour de joueur  1  de jouer
colonne ?
```

4 Bonus

Pour améliorer votre jeu, vous pourrez :

- Ajouter une fonction `coup_intelligent_v1` qui fera jouer à l'ordinateur, lorsque cela est possible, un coup lui permettant de capturer des graines lorsque cela est possible et un coup aléatoire sinon.
- Ajouter une fonction `coup_intelligent_v2` qui fera jouer à l'ordinateur, lorsque cela est possible, un coup lui permettant de capturer des graines lorsque cela est possible ou un coup qui empêchera le joueur de faire une capture.
- ...
- Ajouter une version graphique, permettant d'afficher le jeu graphiquement avec `Pygame`. Voici un exemple d'affichage graphique, mais vous êtes libre de faire autrement... Laissez place à votre imagination...



5 Évaluation

Pour ce projet, vous serez évalués selon les mêmes critères que précédemment :

- réalisation d'un programme fonctionnel ;
- respect des consignes ;
- bonne répartition des tâches dans le groupe ;
- présentation du code clair et soigné (avec noms et prénoms de tous les membres du groupe notamment)
- code documenté, fonctions avec docstrings. . . ;
- code commenté intelligemment(« par bloc »)
- optimisation du code ;
- un carnet de bord rempli à chaque séance (et à chaque fois que vous travaillez le projet), au format numérique, qui indique :
  - ce qui a été fait ;
  - vos choix (de modélisation notamment) ;
  - les difficultés/problèmes rencontrés ;
  - ce qui a été fait pour résoudre ces difficultés/problèmes ;
  - . . ....

Ce projet est à conserver . . .