

# Développement de jeux vidéos

## Évaluation 1

### Projet intégrateur

#### Objectif

L'objectif de ce travail pratique est de :

- Réaliser un jeu de stratégie.
- Utiliser le langage de programmation Python et la bibliothèque Tkinter (Qt).
- Programmer des algorithmes d'intelligence artificielle.
- Rechercher de l'information technique sur Internet.

#### Travail à réaliser

Vous devez réaliser une application qui permet de simuler le Mancala. Un algorithme d'intelligence artificielle permettra au joueur d'affronter l'ordinateur. Trois (3) algorithmes d'IA devront être comparés.

Pour la réalisation du projet vous devez utiliser le langage de programmation Python et la bibliothèque Tkinter.

Vous serez évalué sur la qualité de la présentation, de l'ergonomie, la qualité de l'expérience de l'utilisateur (technique, émotionnelle, ergonomie...), le bon fonctionnement de l'application et la « propreté » de vos codes sources.

#### Remise du projet

- Date de remise : lundi le 25 septembre 2023 avant 11 h.
- Remettre via LÉA (Omnivox) un fichier ZIP contenant les fichiers sources du projet.
- Le 25 septembre vous devez présenter votre projet à votre enseignant.
- Pendant la présentation l'enseignant peut vous questionner sur le fonctionnement de votre application. **Un projet qui a été remis sur Omnivox mais qui n'a pas été présenté est considéré comme n'ayant pas été remis.**

## Règles du jeu

**Mancala** est l'appellation générique d'un ensemble de jeux de société traditionnels africains et asiatiques, aussi appelés jeux de semis. Il s'agit de jeux de stratégie combinatoire abstraits du type « compter et capturer » dans lesquels on distribue des cailloux, graines ou coquillages dans des rangées de coupelles ou de trous, parfois creusés à même le sol.

Il est possible de trouver les règles du jeu de Mancala sur ce site Web :

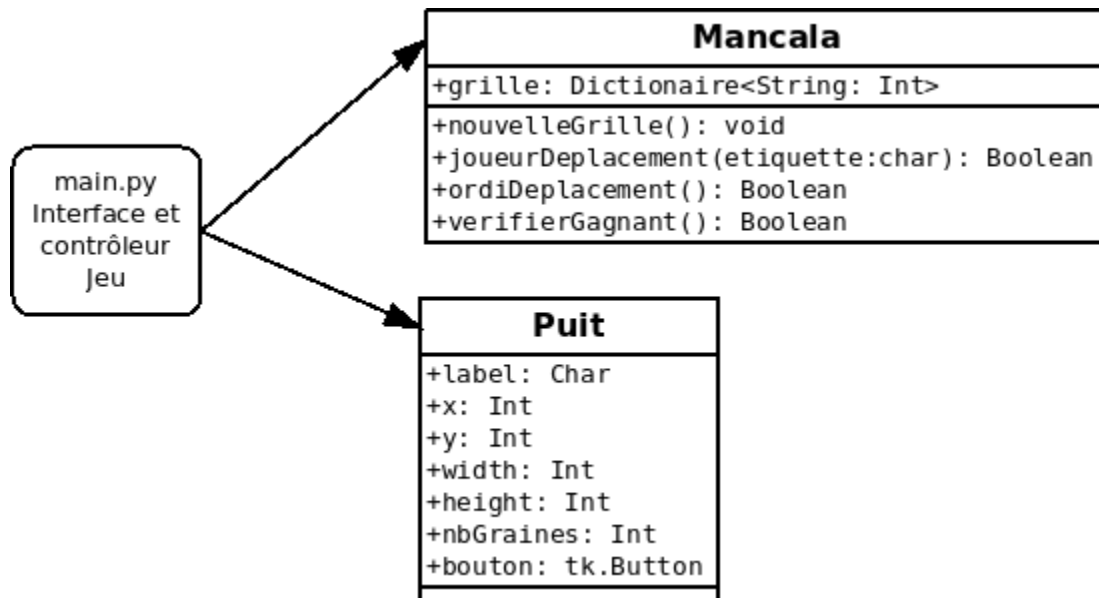
<https://brainking.com/fr/GameRules?tp=103>

## Interface utilisateur



L'interface est contenue dans le fichier **MancalaGui\_etudiant.py**.

## Diagramme de classe



### Description des méthodes

- **nouvelleGrille()** : remise à zéro du jeu.
- **joueurDeplacement()** : Gestion des choix du joueur.
- **ordiDeplacement()** : L'algorithme d'IA
- **verifierGagnant()** : Est-ce qu'il y a un gagnant

## Algorithmes d'IA

Les 3 algorithmes d'IA à implanter sont décrits dans l'article « **The Ancient Game and the AI** » : <https://towardsdatascience.com/the-ancient-game-and-the-ai-d7704bea280d>

- Random Agent;
- Max Agent;
- MinMax Agent.

## Pondération

### Fonctionnalités

|  |   |  |
|--|---|--|
| Algorithme 1 IA (Random Agent)                   | 1 |  |
| Algorithme 2 IA (Max Agent)                      | 2 |  |
| Algorithme 3 IA (MinMax Agent)                   | 2 |  |
| Gestion des choix du joueur                      | 1 |  |
| Vérifier le gagnant                              | 1 |  |
| Choisir qui commence (le joueur ou l'ordinateur) | 1 |  |
| Fonctionnement général                           | 2 |  |

### Respect des contraintes

|  |          |
|--|----------|
| Pendant la présentation l'enseignant peut vous questionner sur le fonctionnement de votre application. <b>Un projet qui a été remis sur Omnivox mais qui n'a pas été présenté est considéré comme n'ayant pas été remis.</b> | - 10 pts |
| Mauvaise lisibilité du code (la « propreté » de vos codes sources).  | - 2 pts  |
| Non respect des diagrammes de classe.  | - 4 pts  |
| Non utilisation de l'interface Tkinter fourni par votre enseignant.  | - 4 pts  |