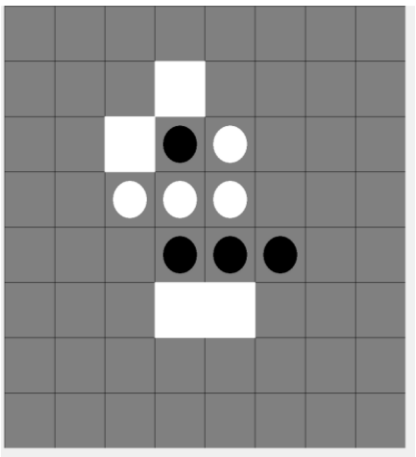
Mohamed Alla – Sarah Nguyen **Gustave Eiffel**
L1 TD A TP 1 **2021**

Reversi

à

deux joueurs



SOMMAIRE

INTRODUCTION

Objectifs :

- Programmer le jeu Reversi sous python incluant les fonctionnalités de base :

    - Dessiner le plateau de jeu (avec les quatre pions disposés aux centres en début de partie)
    - Pouvoir jouer un pion de sa couleur à tour de rôle par un clic de souris sur une des cases jouables.
    - Retourner les pions (adverses encadrés par le pion) après que chaque joueur ait placé un pion
    - Détecter la fin de la partie quand elle survient, ainsi qu'une situation où un joueur est obligé de passer

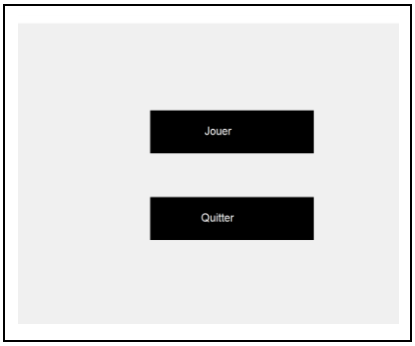
Amélioration :

- Un menu simplifiant l'accès au jeu a été créer comme amélioration au programme.

PROGRAMME

Documentation utilisateur :

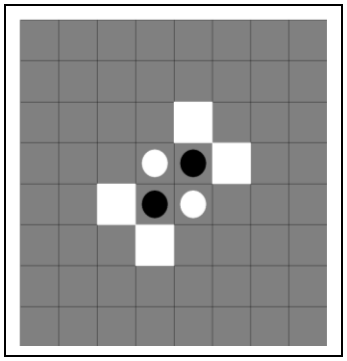
Jeu : En début de partie le jeu commence avec un plateau de 8x8 sur lequel 4 pions dont 2 pions blanc et 2 pions noirs sont placés au centre du plateau. Les pions blancs commencent, le jeu se joue en tour par tour et l'objectif est d'encadrer de deux côtés les pions adversaires avec ses pions. La partie s'arrête lorsque tous les pions ont été joués ou lorsqu'aucun des deux joueurs ne peut plus jouer. Le joueur ayant le plus de pions de sa couleur gagne alors la partie.



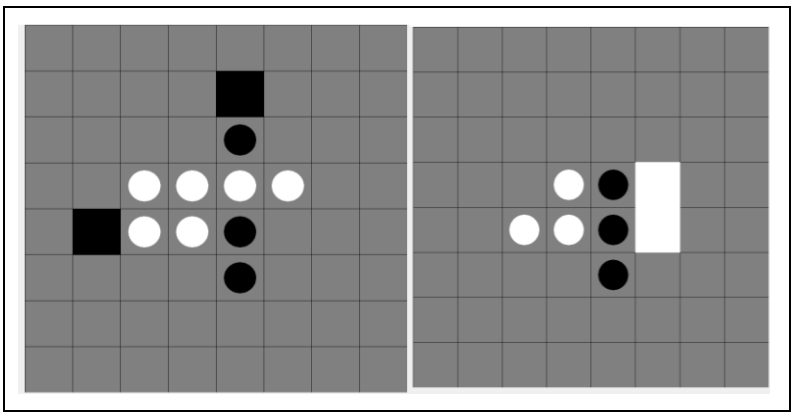
1) Lancer le jeu en cliquant sur « Jouer » ou fermer la fenêtre en cliquant sur « Quitter »

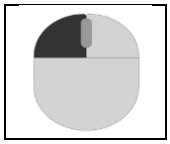
2) La partie commence sous cette configuration c'est-à-dire

les 4 pions au centre et le joueur avec les pions blancs commence.



3) A tour de rôle, sélectionner une case où jouer son pion afin d'avoir le plus de pions possibles de sa couleur sur le plateau (les joueurs jouent à l'aide du clic gauche de la souris.)





4) Une fois toutes les cases remplies ou qu'il n'est plus possible de jouer pour aucun des joueurs, la partie se termine. Appuyer à nouveau pour revenir au menu.

Documentation technique :

Les bibliothèques utilisées sont :

- « doctest » avec la fonction « testmod » qui permet de tester les fonctions créées.
- « upemtk » responsable de la création de la fenêtre, de l'affichage des pions et du clic de la souris. La fenêtre créée permet l'affichage du plateau à l'aide de cette fonction.

Les fonctions utilisées sont :

- « quadrillage() » qui est responsable de l'affichage du plateau. Elle créer un quadrillage en fonction du nombre de case et des mesures du plateau.
- « affiche_pion () » qui permet l'affichage des pions en fonction de la liste plateau. Ainsi, cette fonction permet de faire le lien entre la liste de liste ayant la position des pions (plateau) ainsi que le plateau graphique.
- « copy_list() » fonction responsable de la copie de la liste principale dans une autre liste afin de la garder en mémoire (enregistrée dans plateau2 pour être utilisé dans « verification() » pour la partie vertical).
- « remplir_case() » qui affiche des rectangles colorés permettant de voir les possibilités de jeu pour le joueur.
- « verif_souris () » qui vérifie que le clic de la souris s'effectue bien sur une case jouable et retourne un booléen.
- « verification() » elle permet de chercher les cases jouables parmi la liste plateau (pour horizontal et plateau2 pour vertical) dans un premier temps avec l'appel de la fonction remplir_case() puis renvoie True et est réutilisé dans un second temps pour une vérification du clique sur la case en appelant la fonction verif_souris().
- « comptage_pions() » qui compte la somme de pions de chaque joueur (elle lit dans liste de liste s'il y'a un -1 ou 1 représentant respectivement les pions blancs et les pions noirs).

Le programme principal :

- Création de la fenêtre menu avec les boutons Jouer et Quitter.
- Création de la fenêtre de jeu, initialisation de plateau (et plateau2) puis la partie peut commencer avec appel des fonctions avec la variable code_color qui alterne entre 1 et -1 à chaque tour (pour changer de joueur).
- A la fin de la partie un message s'affiche donnant le nombre de pions des 2 joueurs puis attend le clic de l'utilisateur pour fermer la fenêtre.

INFORMATIONS COMPLEMENTAIRES

Les problèmes rencontrés :

- Le principal problème rencontré fut le lien entre l'interface graphique du plateau et les données du plateau représenté par une liste de liste, le problème a été contourné avec la fonction affiche_pion () qui lit la liste de liste et observe s'il y'a -1 ou 1 pour afficher ses données visuellement.
- Le second problème rencontré fut pour les « combos » (quand on doit retourner les pions à la vertical et à l'horizontal). Il fallait pour contourner le problème créer une seconde liste dans laquelle la première était copier avant modification. Pour cela comme les commandes tels que 'copy' ne marchaient, une fonction fut créée : plateau2.

CONCLUSION

Ce projet était le premier jeu qui nous était donné de faire de manière complètement autonome sans aucun squelette nous donnant une piste pour le début. Le projet était complet et il y avait plein de façons d'arriver à ce résultat ce qui étais assez difficile dut au fait qu'il fallait partir d'un programme vide et devoir créer soit même toutes les fonctions et la documentation. Le jeu contre l'IA est un point à travailler car il s'agissait d'une amélioration qui peux apporter un renouveau à notre projet.