

PROJET DEV APPLI OBJET

JEU OTHELLO

Par Anna TYLKOWSKA, Matis GALVIN, Antonin SIMON

INTRODUCTION	1
FONCTIONNALITÉS	2
DIFFICULTÉS RENCONTRÉES	4
Check_autour	4
Retournement	4
Cas particulier	4
Gestion du positionnement des pions sur Tkinter	4
Écran de fin de partie	5
AMÉLIORATIONS POSSIBLES	5

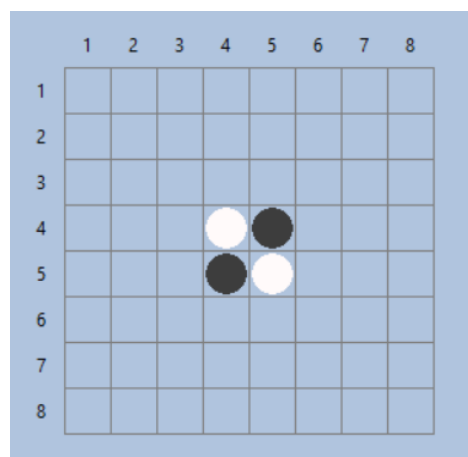
INTRODUCTION

Nous avons recréé, à partir des règles originales, le jeu de société Othello (pour 2 personnes).

Le but du jeu est d'avoir en fin de partie, le plus de pions de sa couleur.

Pour cela, l'un après l'autre, chaque joueur va devoir encadrer un ou plusieurs pions adverses par deux pions de sa propre couleur. Le joueur attaquant va pouvoir retourner les pions qu'il encadre pour les transformer en pions de sa couleur. Toutes les directions sont prises en compte : la verticale, la diagonale et l'horizontale.

Les pions sont positionnés dès le début de la partie comme sur la photo ci-dessous.



La partie peut commencer ! Les Blancs commencent !

⚠ Attention !

Un pion peut être posé uniquement s'il permet le retournement d'un ou plusieurs pions adverses.

Si, à votre tour de jeu, vous ne pouvez pas poser un pion en retournant un pion adverse suivant les règles, vous devez passer votre tour et c'est à votre adversaire de jouer. Si les deux joueurs passent leur tour, la partie prend fin.

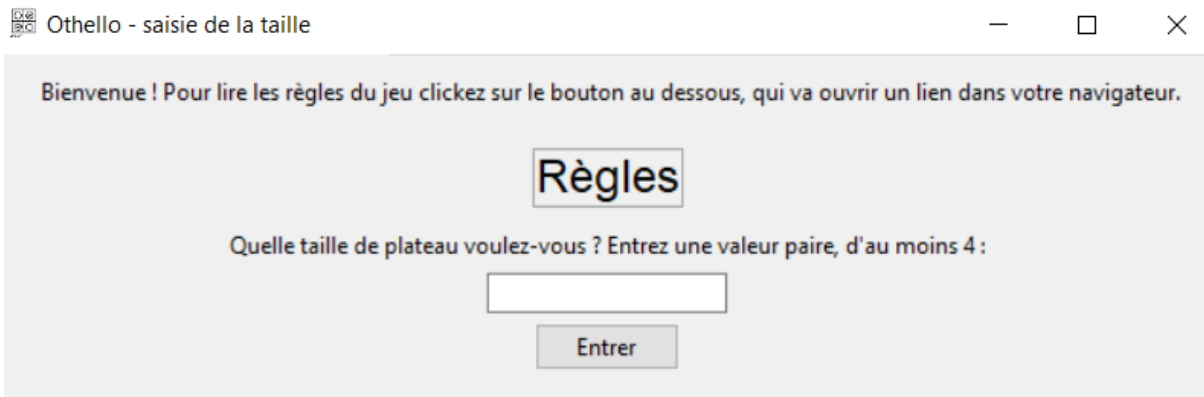
Pas de retournement en chaîne.

FONCTIONNALITÉS

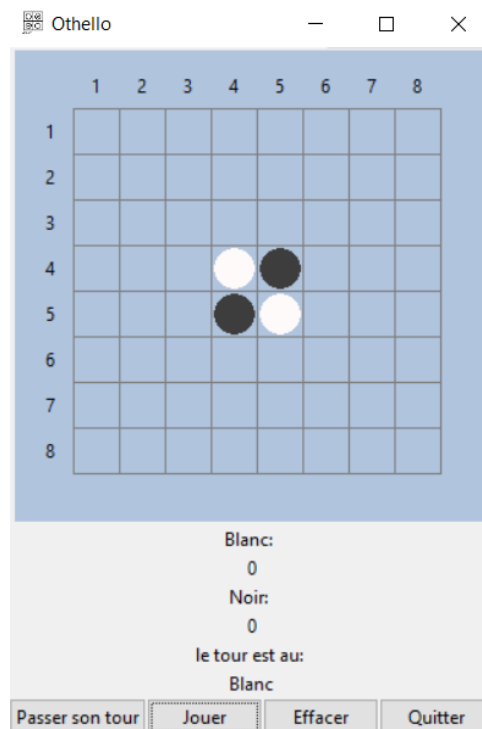
Nous avons essayé de faire une application relativement épurée.

Au lancement, une pop-up s'affiche avec un bouton permettant d'être redirigé vers un site proposant les règles du jeu.

Sur le même écran, vous pouvez saisir la taille de la grille, ce qui offre une durée de jeu variable suivant l'envie des utilisateurs.



Une fois la saisie de la taille effectuée, un nouveau pop-up s'affiche avec cette fois l'écran principal. C'est ici que tout se passera durant les minutes de jeux qui suivront !



Vous trouverez différents boutons permettant de lancer, effacer la grille (dans le cas où l'utilisateur souhaite rejouer), un bouton pour passer son tour en cas d'impossibilité de jouer, ou encore un bouton pour quitter.

A la fin de la partie, un pop-up s'affiche, indiquant la marche à suivre pour rejouer ainsi que le vainqueur de la manche !



Victoire



Congratulations Joueur 2 !

Pour rejouer clicker 'Effacer' et ensuite 'Jouer'

OK

DIFFICULTÉS RENCONTRÉES

Check_autour

Lors de la création, nous avons rencontré plusieurs problèmes.

Après avoir fait l'affichage de la grille en console, nous sommes passés à la pose des pions. Nous avons créé une fonction "check_autour" pour s'assurer de la présence d'au moins un pion autour de l'endroit souhaité pour la pose.

Après plusieurs tests, nous avons remarqué que les joueurs pouvaient poser leurs pions sur la dernière ligne ainsi que sur la dernière colonne de la grille.

Nous avons décidé d'entourer notre grille des symboles " _ _ _ " et " | | | ".

Cette "bordure" nous a permis d'appliquer la fonction "check_autour", et de résoudre le problème.

Retournement

C'est sur l'étape "Retourner les pions" que nous avons été bloqués par la suite.

L'algorithme pour retourner les pions fonctionnait sur la direction horizontale, verticale, mais pas pour les diagonales. Le principe de vérification pour le retournement fonctionne de la manière suivante :

A partir du pion posé, on vérifie dans toutes les directions (horizontale, verticale, diagonale) ce qu'il se trouve dans les cases.

A chaque case, on regarde :

- S'il y a un " . ", un " _ " ou un " | " on arrête de regarder dans cette direction
- S'il y a un pion adverse, on continue et on passe à la case suivante
- Si c'est un de nos pions alors on a atteint la limite, et nous pouvons retourner les pions parcourus jusque là

A chaque vérification d'un pion, on stock ses coordonnées dans une checklist pour savoir si nous devons procéder au retournement ou pas et nous n'avons plus qu'à regarder ce que contient cette liste.

Cas particulier

Nous avons rencontré un problème concernant une règle du jeu. Lorsqu'un joueur ne peut plus poser de pions, il doit passer la main à son adversaire. Nous avons contourné le problème en ajoutant un bouton "Passer son tour".

Gestion du positionnement des pions sur Tkinter

La principale difficulté lorsque l'on est passé sur Tkinter fut de gérer la position des pions comme on pouvait le faire sur le terminal. Nos pions n'étaient que des cercles de couleur, et non des caractères. Pour pouvoir cibler un pion en particulier, nous n'avions pas d'éléments qui nous permettaient de le retrouver. Finalement nous avons repris la même construction que dans notre terminal: une liste de listes qui classaient tous les pions comme ils apparaissent sur tkinter. De plus, chaque pion est défini par [(tuple de ses coordonnées x) , (tuple de ses coordonnées y) , un booléen qui indique si le pion existe ou non, sa couleur]

Écran de fin de partie

Lorsque le jeu se termine, nous voulons faire apparaître une nouvelle fenêtre Tkinter avec le score et un bouton pour rejouer. Nous avons essayé de le faire en utilisant un écouteur fournie par Tkinter qui s'appelle `trace_add()`. Elle permet de suivre l'état des variables pendant l'exécution du `mainloop`. Dans la classe `Damier` nous avons créé une variable booléenne pour stocker si le jeu est terminé ou non, et le `trace_add`, qui se trouvait dans le `main`, suivrait quand il serait changé et exécuterait un `callback`.

Notre `callback` aurait été une fonction qui afficherait un écran de fin.

Malheureusement, il n'a fonctionné qu'une seule fois, à l'initialisation de la variable. Quand on a fait le `callback` une deuxième fois pour terminer le jeu, nous obtenions une erreur: `'TypeError: 'nonetype' object is not callable'`.

Il doit y avoir un problème soit avec la variable d'état sur laquelle nous avons appelé `trace_add()`, soit avec les arguments passés dans le `callback`, du moins c'est ce que nous pensons. Avec un temps limité et une documentation de Tkinter qui manque d'exemples d'utilisation, nous avons décidé d'opter pour une solution plus simple. Lorsqu'un jeu est terminé, nous affichons un `pop-up` avec le gagnant et des informations sur les étapes à suivre pour rejouer.

AMÉLIORATIONS POSSIBLES

Dans une version 2, nous aimerions inclure un background musical.
Un bouton de pause serait présent pour stopper la musique, mais cette fonctionnalité peut apporter une véritable dynamique au jeu.

Un autre point important serait de modifier l'UI du jeu.
Celui proposé par Tkinter de base ne permet pas de créer une interface soignée.
Nous avons néanmoins trouvé un package à importer: ttk
Il a permis un design un peu plus agréable, mais il faudrait vraiment accentuer le travail sur cette partie.

Créer un bouton "REJOUER".
Nous n'avons pas réussi à l'implanter avant la fin du temps imparti. Nous vous proposons à la place, un pop-up affichant le vainqueur ainsi que la marche à suivre pour relancer une partie.

L'ajout d'un descriptif pour les règles.
Nous avons été confrontés à la même problématique, à savoir le manque de temps. Nous avons ajouté un lien vous permettant d'être redirigé vers un site pour pouvoir les lire.