BEAUDIOT Baptiste
EL KORCHI Youssouf

Othello Documentation :

Notre projet :

L'objectif de ce projet est la création du jeu Othello sous python.
L'othello est un jeu de plateau se jouant en un contre un sur un plateau de 8x8 cases. Les joueurs placent leur pions chacun leur tour pour capturer les pions adverses. L'objectif final du jeu est d'avoir le plus de pion de sa couleur à la fin de la partie lorsque les joueurs n'ont plus de coup disponible.

L'architecture du jeu :

Notre projet s'organise comme suit (voir annexe 1). Nous l'avons organisé en plusieurs répertoires et programmes différents, chacun responsable de leur propre partie du jeu.

```
main.py        # Point d'entrée de l'application, lance la boucle principale du jeu
src/           # Dossier des modules sources
tests/         # Dossier des tests unitaires
```

Description des modules principaux :

board.py

- **Responsabilité :**

Représente le plateau d'Othello, gère l'état des pions et les règles de placement.

Classe principale : Board

- **Attributs :**
  - board : liste 2D (8x8) représentant les cases du plateau
  - *0 = vide, 1 = pion blanc, -1 = pion noir*
- **Méthodes clés :**
  - __init__() : initialise le plateau avec les 4 pions centraux
  - place_piece(row, col, color) : place un pion si coup valide et retourne les pions capturés
  - get_valid_moves(color) : retourne la liste des coups valides pour un joueur donné
  - has_valid_move(color) : indique s'il existe au moins un coup valide
  - Méthodes privées pour vérifier les captures dans 8 directions (horizontal, vertical, diagonal)

Fonctionnement :

- Lorsqu'un pion est placé, la méthode cherche à retourner les pions adverses encadrés sur au moins une direction.
- Si aucune capture possible, le coup est invalide.

Le plateau est une matrice manipulée par indices [row][col].

game.py

- **Responsabilité :**

Gère la partie, le joueur courant, la progression, la détection de fin et la gestion des interactions utilisateur.

Classe principale : Game

- **Attributs :**
  - board : instance de Board
  - current_player : int (-1 pour noir, 1 pour blanc)
  - game_over : booléen indiquant si la partie est terminée
  - end_message : message affiché à la fin du jeu
- **Méthodes clés :**
  - handle_click(position, window_size) : gère un clic utilisateur, convertit coordonnées en case, joue le coup si possible
  - switch_player() : change le joueur courant (noir -> blanc, blanc -> noir)
  - end_game() : calcule le score, déclare le gagnant, met fin à la partie

Fonctionnement :

- Après chaque coup valide, le joueur est changé.
- Si aucun joueur ne peut jouer, la partie se termine.
- La méthode handle_click est appelée depuis l'interface graphique avec la position du clic.

display.py

- **Responsabilité :**

Gère tout l'affichage graphique du jeu (plateau, pions, messages)

- **Fonctions clés :**
  - Initialisation de la fenêtre graphique avec Pygame
  - Dessin du plateau (grille + pions)
  - Affichage des messages (joueur actuel, fin de partie)
- **Fonctionnement :**
  - Découpe la fenêtre en 8x8 cases
  - Actualise l'affichage après chaque coup ou événement

Gestion des tests :

Les tests sont organisés dans le dossier **tests** et peuvent être exécutés depuis la racine du projet avec la commande suivante :
python -m unittest discover tests

test_board.py

- Teste l'initialisation du plateau, le placement de pions valides et invalides, et la détection des coups valides.
- Vérifie que les captures retournent bien les pions adverses.
- Garantit que le plateau démarre avec la configuration standard.

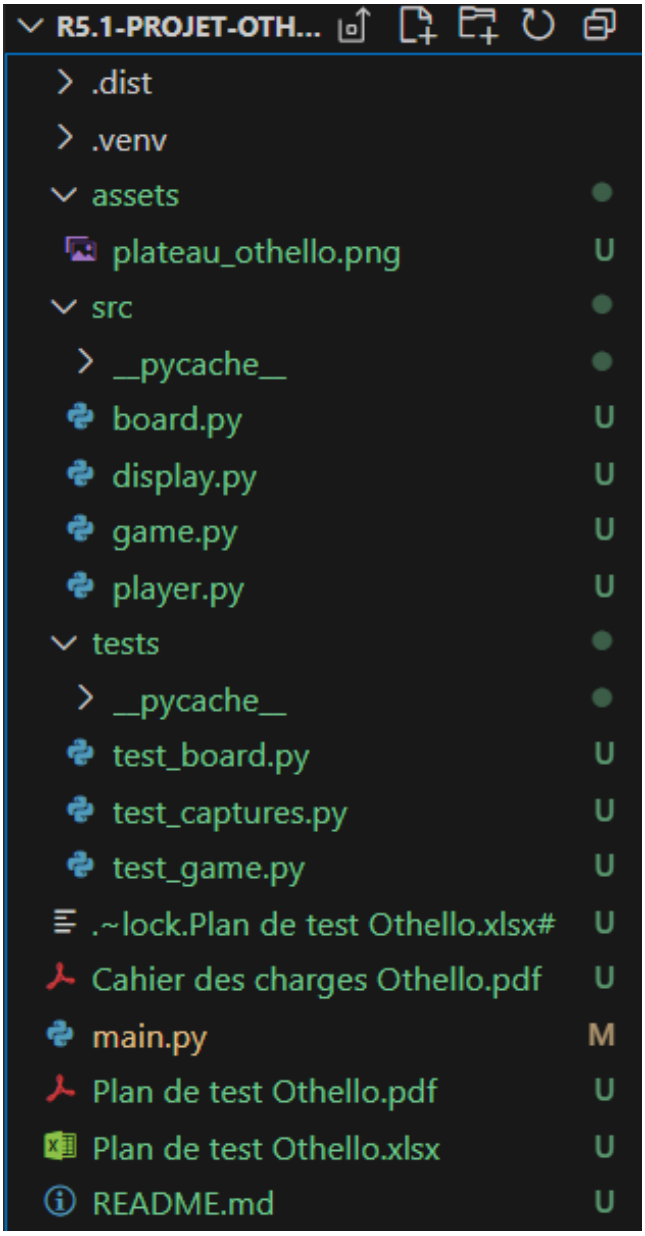
test_captures.py

- Vérifie que les captures fonctionnent correctement dans toutes les directions :
  - Horizontal
  - Vertical
  - Diagonal
- Initialise un plateau vide et place des pions spécifiques pour tester précisément les captures.

test_game.py

- Teste la logique du jeu :
  - Le changement de joueur après un coup valide
  - La gestion de passage de tour lorsqu'un joueur n'a aucun coup valide
  - La détection de fin de partie quand aucun joueur ne peut jouer
- Simule des clics souris et vérifie l'état du jeu.

Annexes :



Annexe 1