

Échecs Réussis

William Auroux Pierre Ingelaere Oscar Plichon
Raphaël Rocard Yi Ye

<https://gitlab-cw1.centralesupelec.fr/oscar.plichon/chess-game>

24 Novembre 2023

- 1 Présentation du produit
- 2 Travail d'analyse et de conception
 - Choix du MVP
 - Découpage du Projet
 - Répartition du travail
- 3 Architecture du code
- 4 Fonctionnalités
 - Interface graphique
 - Quelques configurations
 - Règles spéciales
 - Tests
- 5 Améliorations possibles



Figure 1 – Une série sur les échecs (Netflix)

Notre produit

- 1 recréation d'un jeu d'échecs se jouant en local
- 2 pour les passionné.e.s du jeu souhaitant une interface simple d'utilisation
- 3 filtre les mouvements invalides, permet de revisionner les parties

- Interface graphique simpliste affichée dans le terminal
- Coup indiqué dans le terminal par l'utilisateur (coordonnées de départ et d'arrivée)
- Exclusion de certaines règles annexes comme les roques, les promotions et les prises en passant

Découpage du Projet

■ Sprint 1

- Fonctionnalité 1 :
Création de la grille initiale
- Fonctionnalité 2 :
Afficher la grille en version MVP

■ Sprint 2

- Fonctionnalité 3 :
Gestion des déplacements

■ Sprint 3

- Fonctionnalité 4 :
Gestion des échecs et de l'échecs et mat
- Fonctionnalité 5 :
Implémentation du pat

■ Sprint 4

- Fonctionnalité 6 :
Gestion des inputs du joueurs
- Fonctionnalité 7 :
Lancement du jeu

Après avoir complété ces fonctionnalités, le MVP est atteint.

■ Sprint 5

- Fonctionnalité 8 :
Affichage d'un échiquier avec l'interface graphique plus évoluée
- Fonctionnalité 9 :
Mise en place de la configuration du jeu via l'interface graphique

■ Sprint 6

- Fonctionnalité 10 :
Implémentation du roque

■ Sprint 7

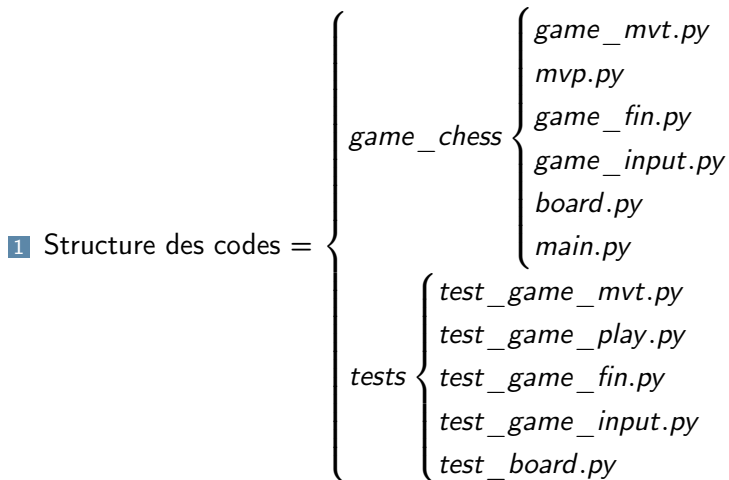
- Fonctionnalité 11 :
Implémentation de la promotion
- Fonctionnalité 12 :
Implémentation de la prise en passant

■ **Stade MVP :**

- Création et affichage du plateau (Oscar)
- Evolution des pièces (Yi, Pierre)
- Situations particulières (Raphaël)
- Gestion des inputs (William)
- Tests (Oscar, William)

■ **Jeu fini :**

- Refonte des fonctions de base (Pierre, Raphaël)
- Ajout des règles annexes (Pierre, Yi)
- Adaptation du MVP en boucle tkinter (Oscar)
- Débugage, perfectionnement (William)



2 Modules utilisés : tkinter, pytest, pygame


```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

 8 | E |  |  |  |  |  |  |  |  |
 7 | Δ | Δ | Δ | Δ | Δ | Δ | Δ | Δ |
 6 |  |  |  |  |  |  |  |  |
 5 |  |  |  |  |  |  |  |  |
 4 |  |  |  |  |  |  |  |  |
 3 |  |  |  |  |  |  |  |  |
 2 | ▲ | ▲ | ▲ | ▲ | ▲ | ▲ | ▲ | ▲ |
 1 | X |  |  |  |  |  |  |  | X |
    a b c d e f g h

Aux blancs de jouer.
Entrez la coordonnée de la case de départ de votre coup : e2
Entrez la coordonnée de la case d'arrivée de votre coup : e4

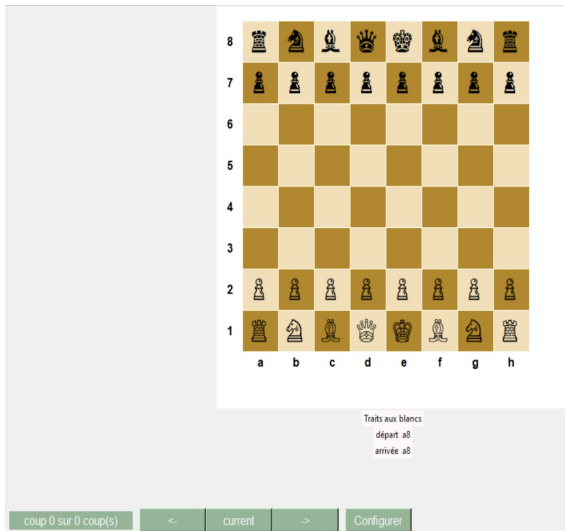
 8 | E |  |  |  |  |  |  |  |  |
 7 | Δ | Δ | Δ | Δ | Δ | Δ | Δ | Δ |
 6 |  |  |  |  |  |  |  |  |
 5 |  |  |  |  |  |  |  |  |
 4 |  |  |  | ▲ |  |  |  |  |
 3 |  |  |  |  |  |  |  |  |
 2 | ▲ | ▲ | ▲ | ▲ |  | ▲ | ▲ | ▲ |
 1 | X |  |  |  |  |  |  |  | X |
    a b c d e f g h

Aux noirs de jouer.
```

Échiquier

- 1 Interface graphique dans le terminal
- 2 Position de la pièce indiquée par 1 lettre et 1 numéro
- 3 Coordonnées de la case de départ et d'arrivée demandées

Interface graphique améliorée



Échiquier

- 1 Interface graphique : tkinter
- 2 Position de la pièce indiquée par 1 lettre et 1 numéro
- 3 En bas : les éléments de l'interface graphique

Mouvement des pièces

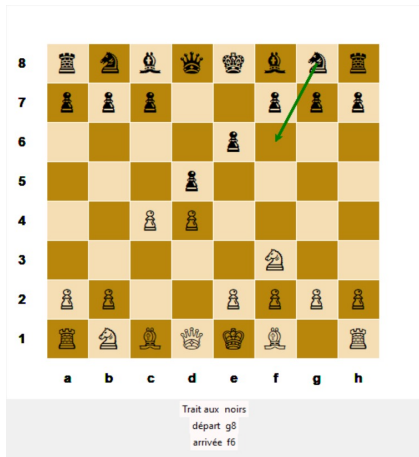


Figure 2 – Cavalier g8 à f6

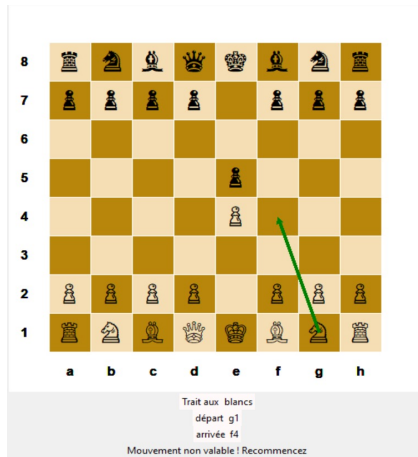
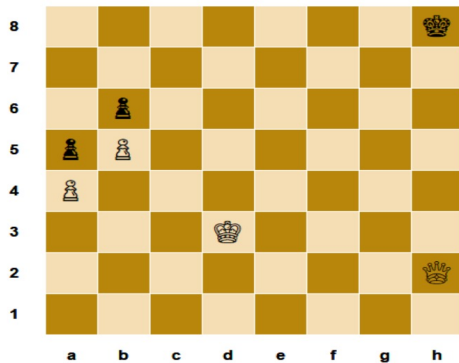


Figure 3 – Si le mouvement n'est pas valable, l'échiquier ne sera pas modifié

Échec



Trait aux noirs
départ g1
arrivée h2

Les noirs sont en échec.

coup 1 sur 1 coup(s)



current



Configurer

Figure 4 – Échec

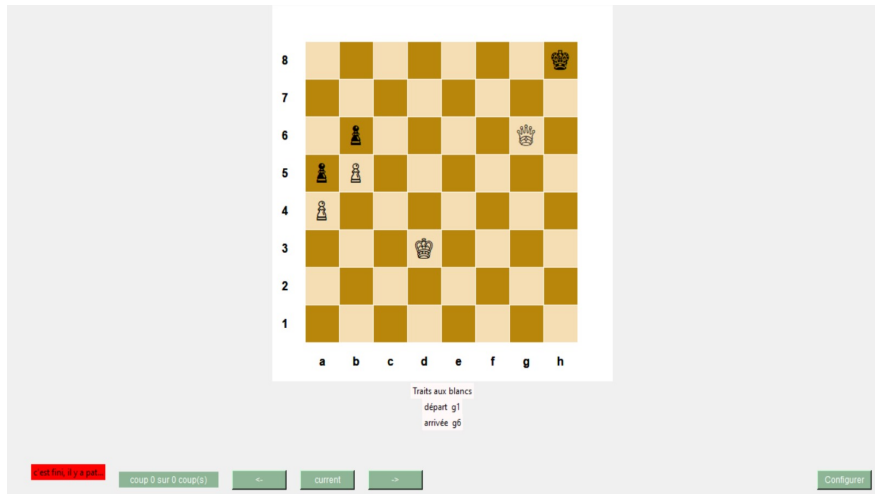


Figure 5 – Pat

1 Implémentation de l'échec

- On parcourt tout le plateau pour voir si le roi risque d'être pris par une pièce adverse

2 Implémentation du pat

- Vérifier si chacun de nos mouvements possible met notre roi en échec

Promotion

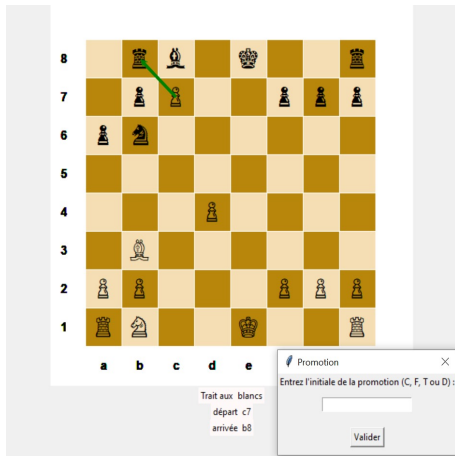


Figure 6 – Promotion d'un pion blanc

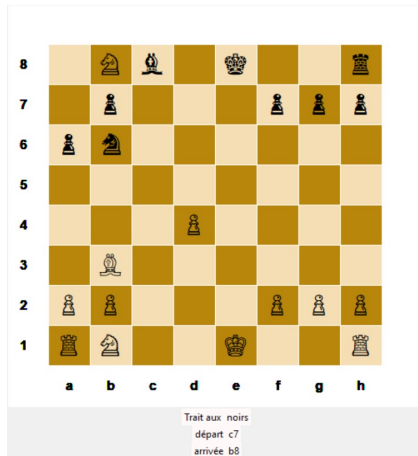


Figure 7 – Promu en cavalier

Règles à mémoire

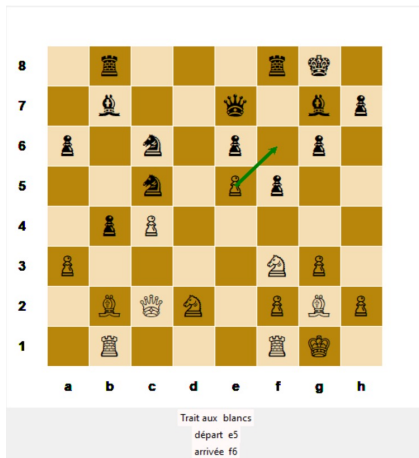


Figure 8 – Prise en passant e5 à f6

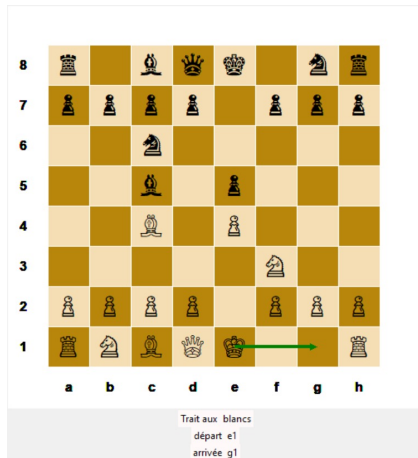


Figure 9 – Petit roque blanc

- Implémentation de la prise en passant
 - 1 Vérifier que **le dernier coup** est l'avancement de 2 cases du pion adverse
 - 2 Rendre le mouvement en diagonale possible en prenant le pion adverse
- Implémentation du roque
 - 1 Vérifier si la tour et le roi concernés ont déjà bougé **depuis le début de la partie**
 - 2 Vérifier si les cases entre le roi et la tour sont vides
 - 3 Vérifier si le roi n'est pas en échec en se déplaçant

Coverage report: 71%

coverage.py v7.3.2, created at 2023-11-23 14:33 +0100

<i>Module</i>	<i>statements</i>	<i>missing</i>	<i>excluded</i>	<i>coverage</i>
game_chess__init__.py	0	0	0	100%
game_chess\board.py	19	0	0	100%
game_chess\game_fin.py	75	0	0	100%
game_chess\game_input.py	23	0	0	100%
game_chess\game_mvt.py	199	55	0	72%
game_chess\game_play.py	54	54	0	0%
Total	370	109	0	71%

- Ajout d'une horloge
- Retournement de l'échiquier pour le noir
- Implémenter un IA qui peut jouer contre le joueur
- Partage de l'interface
- Affichage des mouvements possibles

Merci pour votre attention !