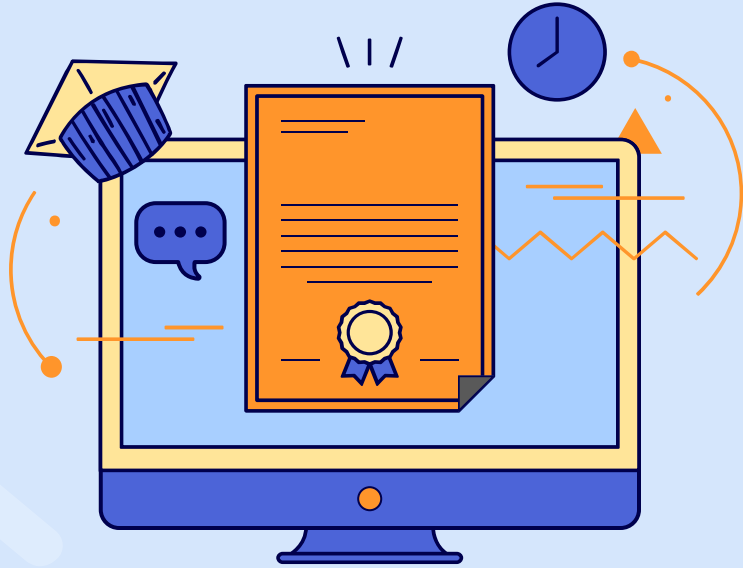


Présentation du projet Assisted Board Games

Membres de l'équipe :

Dahuiss AMOUSSOU
Marina ATANGANA
James BURTON
Aaron DEMAY
Yasmine HAMNACHE
Kerim HUDAYBERDIYEV
Anam SHAIKH YUNUS / NASAR





INTRODUCTION

Expliquer en quoi consiste le projet ABG -> Dire son objectif et annoncer le plan de présentation.

Plan de présentation

01

Présentation du projet existant

Dans cette partie nous allons expliquer toutes les fonctionnalités existantes dans le projet Assisted Board Games

02

Présentation de notre travail

1. Rendus en UML : Diagrammes approuvés
2. Rendus en JAVA : Fonctionnalités implémentées

03

Démonstrations

1. Interface
2. Horloge
3. Menu du jeu

04

Difficultés et axes d'améliorations

1. Difficultés rencontrées pendant le projet
2. Axes d'améliorations



01

Présentation du projet existant

De quoi sommes-nous parti ?

Qu'est ce que c'est un jeu de Plateau ?



ETAT DES LIEUX





02

Présentation du travail effectué

Qu'avons-nous réalisé cette année ?



Rendus approuvés en UML

Diagramme de séquence du PlayerState

- Capture d'écran du diagramme de séquence
- Expliquer le diagramme et dire pourquoi il est utile

Diagramme de classe du jeu d'échec

- Capture d'écran illustrant le diagramme de classe
- Expliquer et montrer son utilité

Diagramme de cas d'utilisation

- Illustration avec la capture d'écran du diagramme
- Expliquer ce que fait le diagramme ainsi que son utilité

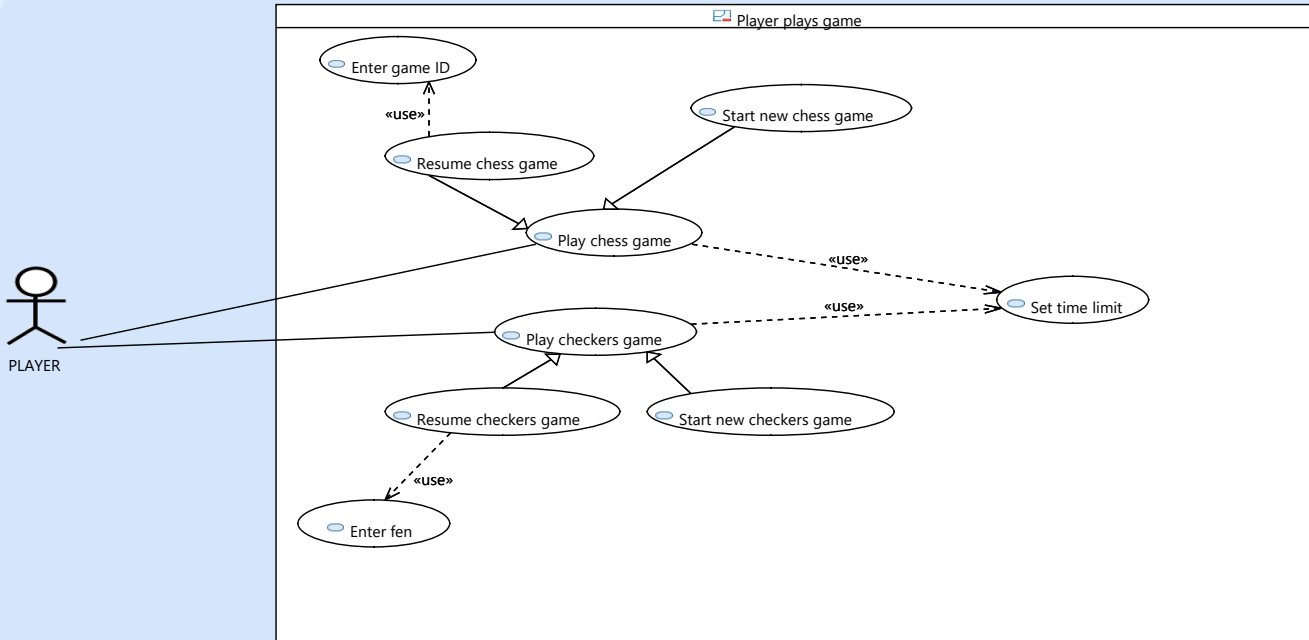
Diagramme de séquence du CheckerBoard

Montrer à l'aide de la capture d'écran du diagramme de séquence son utilité et expliquer ce qu'il fait.



Diagramme de cas d'utilisation

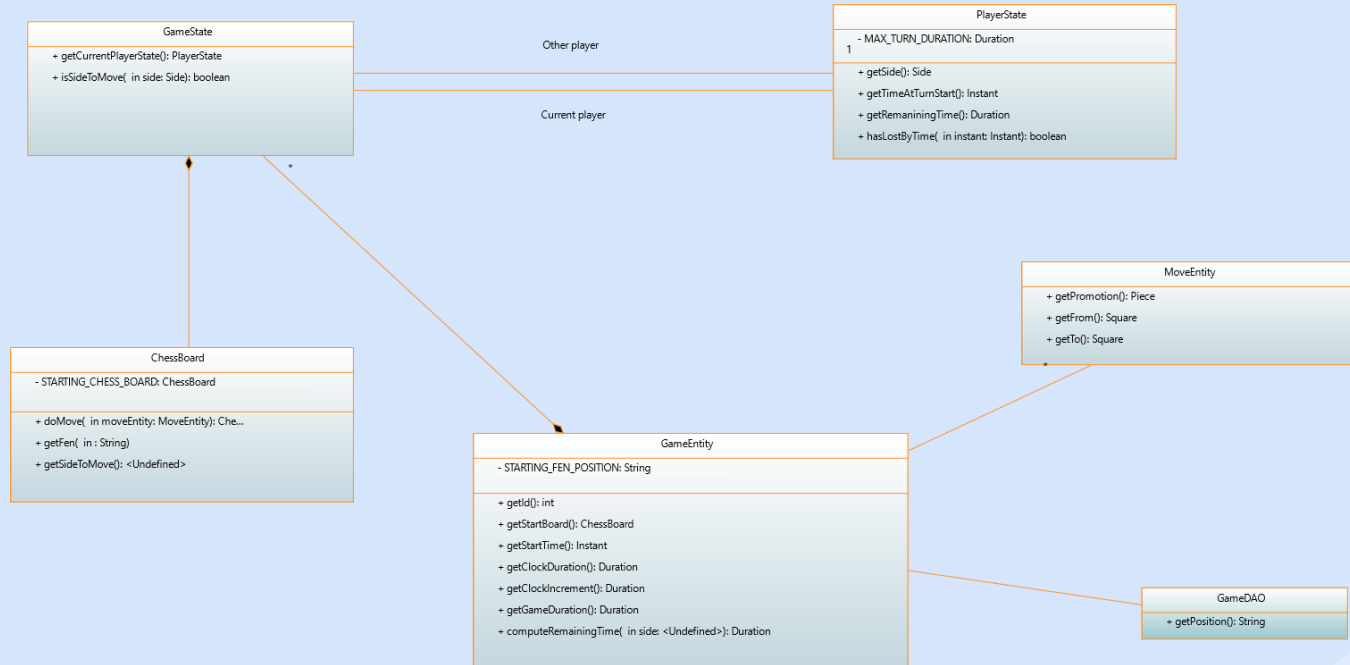
PlayerPlaysGame



<https://github.com/oliviercailloux-org/projet-assisted-board-games-1/blob/Presentation/Doc/Images/playerPlaysGame.svg>

Diagramme de classe du jeu d'échec

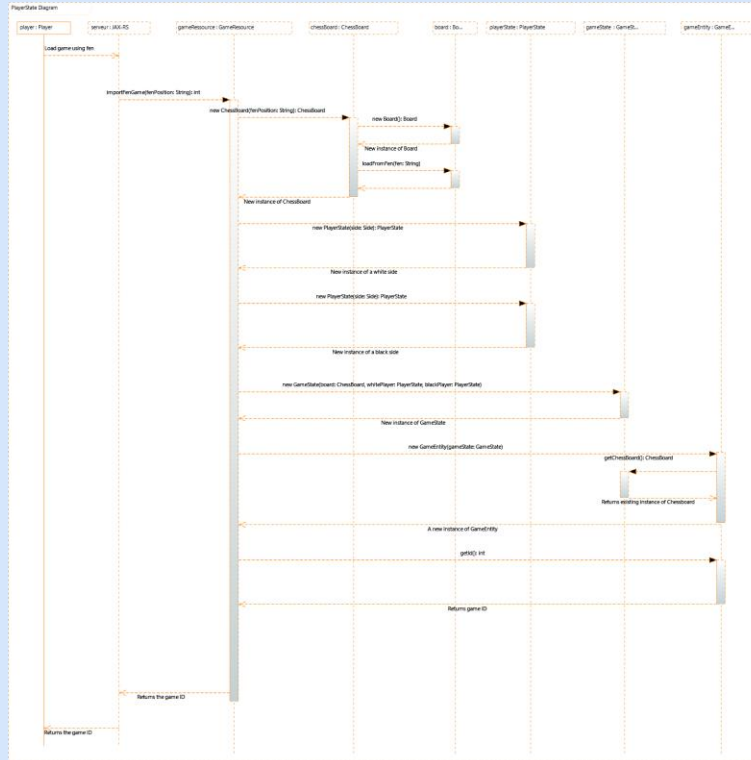
Game_And_Player_Management



[https://github.com/oliviercailloux-org/projet-assisted-board-games-1/blob/Presentation/Doc/Images/Game And Player Management.svg](https://github.com/oliviercailloux-org/projet-assisted-board-games-1/blob/Presentation/Doc/Images/Game%20And%20Player%20Management.svg)

Diagramme de séquence du PlayerState

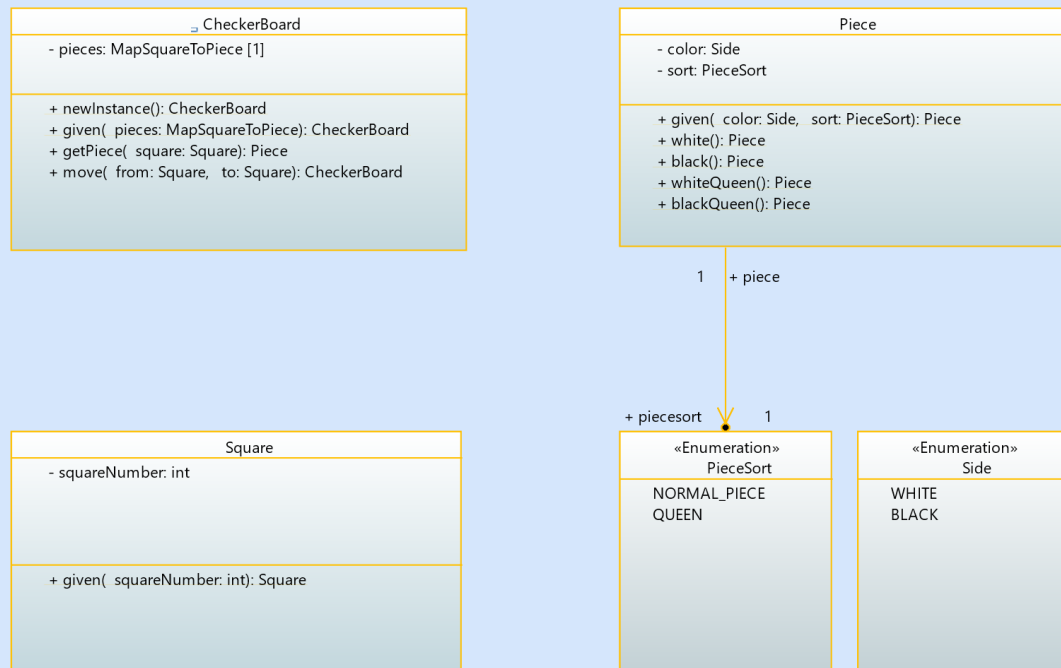
Illustration du PlayerState



<https://github.com/oliviercailloux-org/projet-assisted-board-games-1/blob/Presentation/Doc/Images/PlayerStateSequenceDiagram.svg>

Diagramme de classe du CheckerBoard

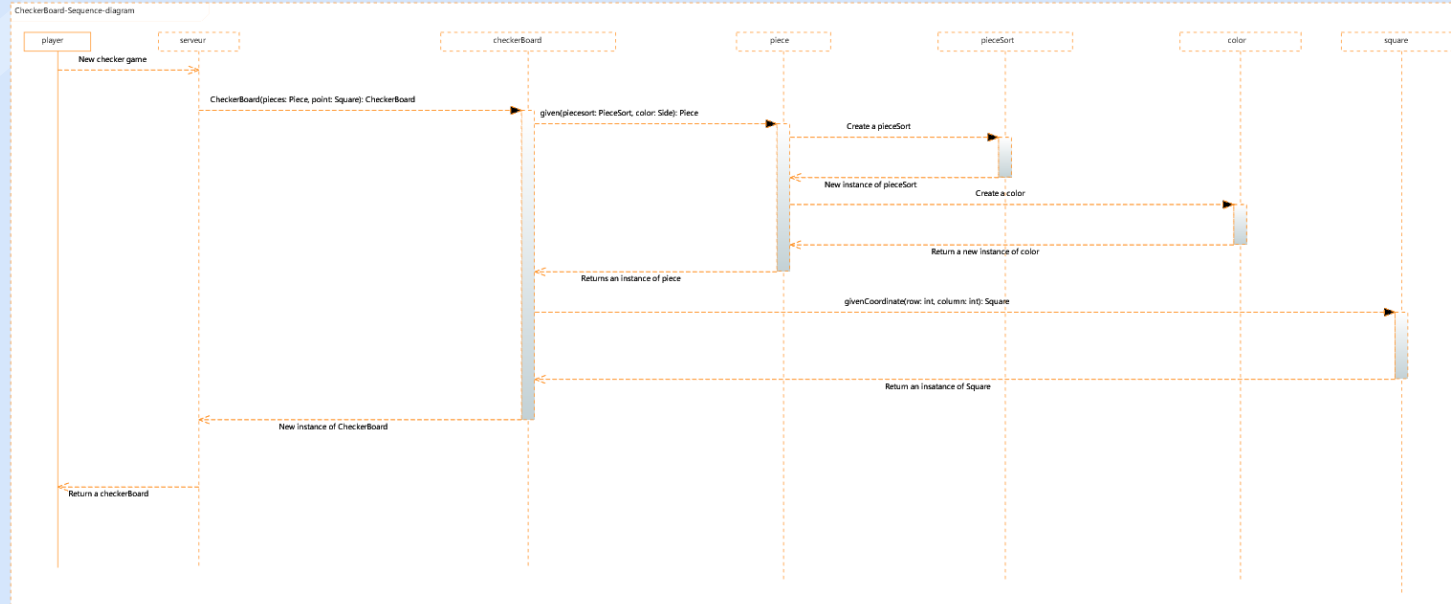
Interaction des classes



https://github.com/oliviercailloux-org/projet-assisted-board-games-1/blob/main/Doc/Images/Checkerboard_class_diagram.SVG

Diagramme de séquence du CheckerBoard

Illustration du CheckerBoard



<https://github.com/oliviercailloux-org/projet-assisted-board-games-1/blob/Presentation/Doc/Images/CheckerBoard-Sequence-diagram.svg>

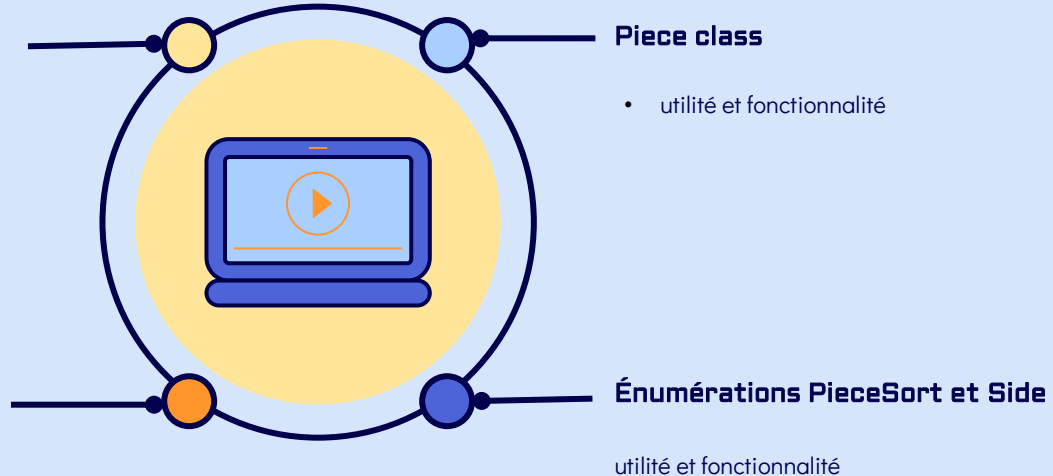
Rendus approuvés en Java

Square class

- utilité et fonctionnalité

CheckerBoard class

- utilité et fonctionnalité



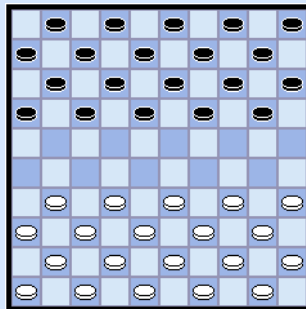
Les méthodes de classe Square

Square

Cette classe représente une case et toutes les fonctionnalités qui lui sont liées. La case est défini à l'aide d'un nombre respectif, par exemple : 1, 2 ... 50

`public static Square given(int squareNumber)` - pour créer une case en utilisant le nombre de case fourni

Le nombre de case est tout nombre compris entre 1 et 50. Seuls les cases noirs sont utilisés pour y placer des pièces (cf. : la notation Manoury).



	01		02		03		04		05
06		07		08		09		10	
11		12		13		14		15	
16		17		18		19		20	
21		22		23		24		25	
26		27		28		29		30	
31		32		33		34		35	
36		37		38		39		40	
41		42		43		44		45	
46		47		48		49		50	

Source : <http://www.ffjd.fr/Web/index.php?page=notation> - (FFJD - Fédération Française)

Les méthodes de classe Piece

Piece

Les pièces peuvent être blanches ou noires. En ce qui concerne leurs types, ils sont représentés par des pièces normales ou une dame.

- `public static Piece given(PieceSort pieceSort, Side color)` – pour créer une pièce en utilisant une case et une couleur donnés

Diverses static factory méthodes sont disponibles pour créer directement des pièces noires, blanches ou une pièce dame avec une couleur noire ou blanche.

Les méthodes de classe CheckerBoard

CheckerBoard

Le plateau est composé de 100 cases (damier de 10 x 10) et de pièces noires ou blanches.

Les cases du plateau sont représentées par un chiffre (1, 2, 3, ... 50).

`public static CheckerBoard newInstance()` - pour créer une représentation d'un CheckerBoard par défaut. Les pièces placées dans leur position par défaut au début de la partie

`public static CheckerBoard given(Map<Square, Piece> inputBoard)` - pour créer une représentation d'un CheckerBoard avec les pièces fournies et leur position sur le plateau

`public Optional<Piece> getPiece(Square square)` - pour obtenir la pièce sur la case donnée

`public CheckerBoard move(Square from, Square to)` - permet de déplacer une pièce d'une case à une autre



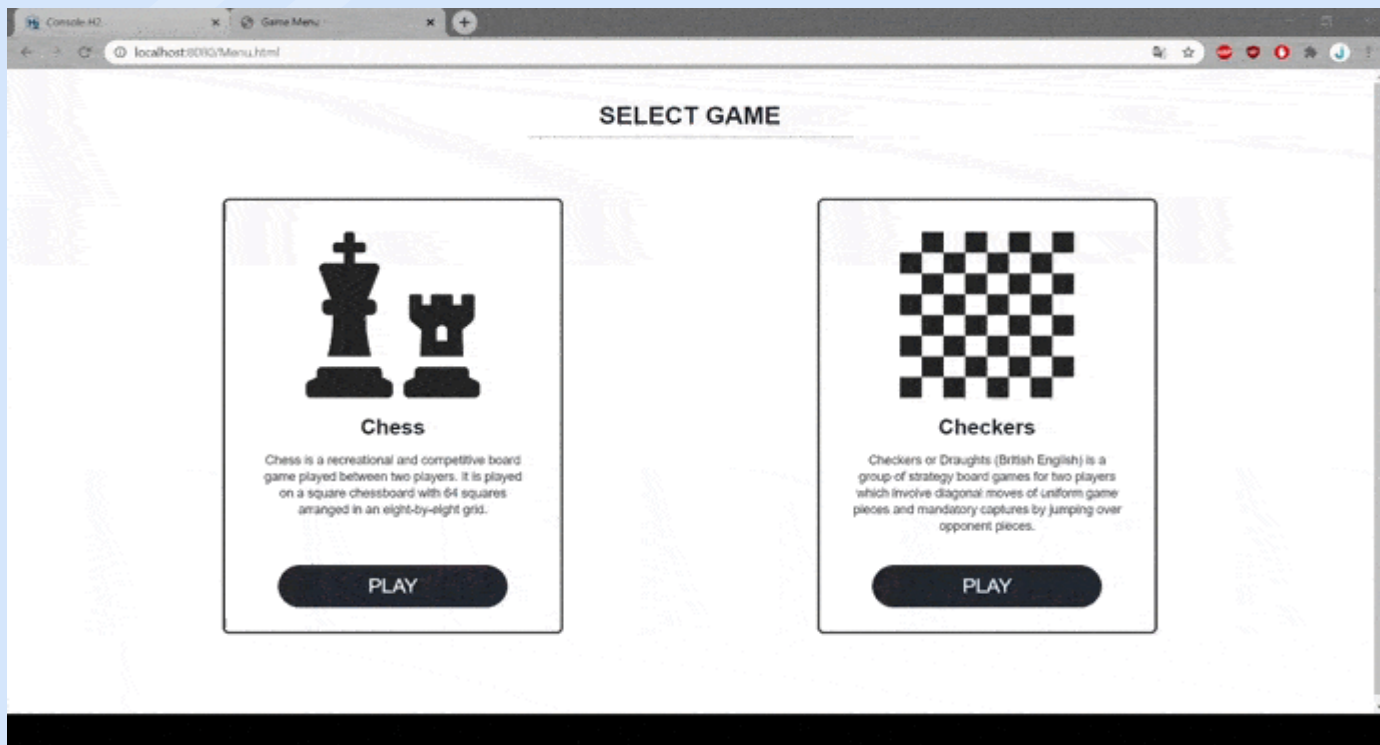
03

Démonstrations

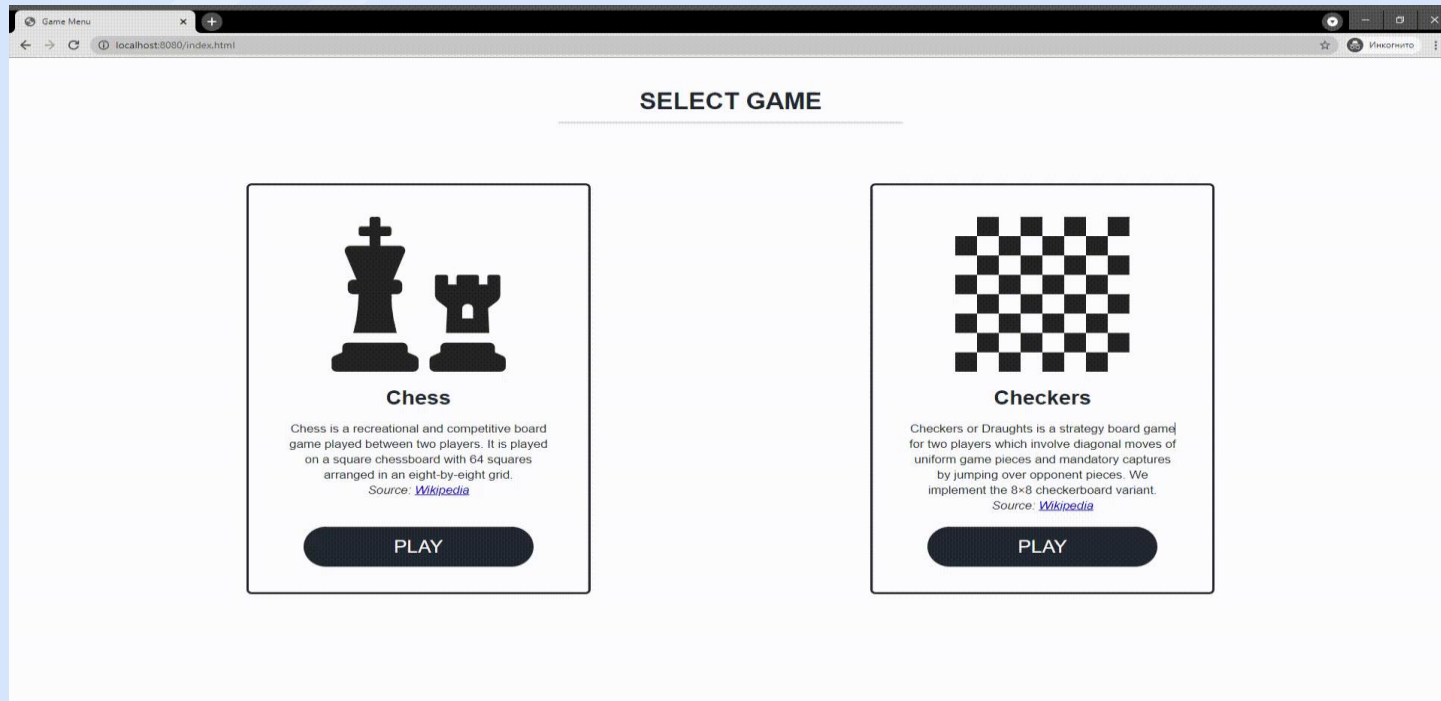
Voici notre participation à ce projet



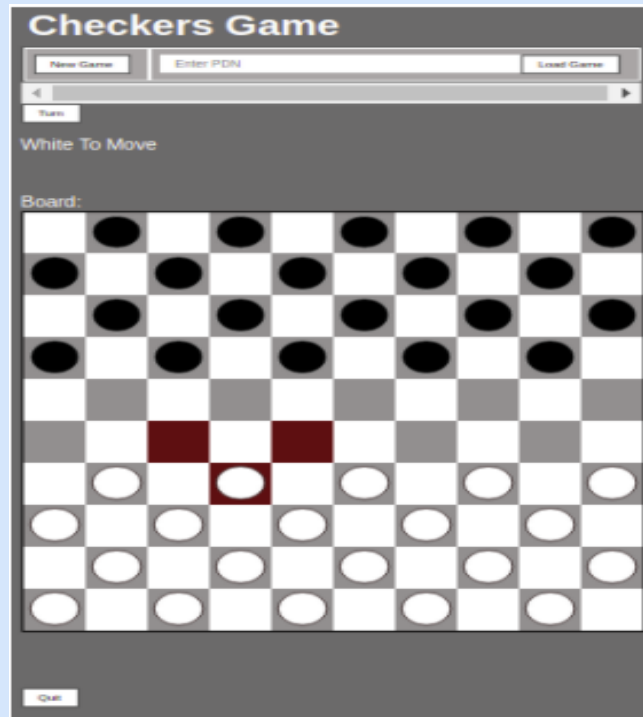
Le menu du jeu de plateau



L'horloge pour le jeu d'échecs



Interface du jeu de dames





04

Difficultés et axes d'améliorations

*Quels problèmes avons-nous eu ?
Que faudrait-il améliorer pour la suite ?*



Difficultés rencontrées et axes d'améliorations



Difficultés rencontrées



Gestion du temps

- Répartition entre charge de travail, charge professionnelle et charge projet
- Planning de réunion sur Teams



Difficultés d'orientation

- Mise en route
- Installation des logiciels
- Manque d'idées tâches à réaliser
- Changement de binômes à chaque itération
- Entraide entre les membres de l'équipe progressive
- Suggestions et conseils de Monsieur Cailloux



Manque de communication

- Même tâche pour deux binômes
- Reviewer manquant
- SM/PO en contact permanent avec chaque binôme



Travail à distance

- Faire connaissance virtuellement
- Moins de concentration qu'en présentiel
- Planification réunion par binômes



Rendus des tâches

- Compréhension du code initial (documentation quasi inexistante...)
- FOCUS
- EVAL → apprendre de nos erreurs → Approved



- Généraliser autant que possible pour faciliter l'implémentation des nouveaux jeux.

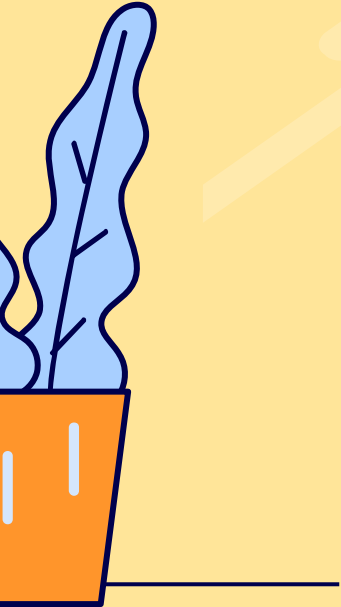
- Niveaux de difficultés d'une partie

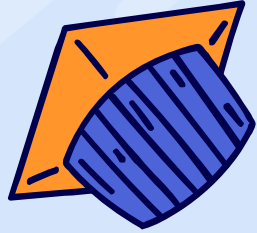
- Jouer un son à chaque déplacement

- Connecter le jeu de dames au serveur

- Historique des parties pour chaque joueur (Nb de victoires, Nb coups, temps moyen partie ...)

- Implémentation horloge jeu de dames





Conclusion

Dire ce que ce projet nous a apporté professionnellement et personnellement

