



ÉCOLE CENTRALE LYON

APPLICATIONS WEB
RAPPORT

Projet - Jeu d'échecs multijoueurs

Élèves :

Anass OIRAR
Oussama LAFDIL

Enseignant :

Daniel MULLER
René CHALON

29 mars 2023

Table des matières

1	Introduction	2
2	Cahier des charges fonctionnel	2
2.1	Fonctionnalités principales	2
2.2	Conception technique	2
3	Technologies utilisés	2
3.1	Création du jeu d'échecs	2
3.2	Création d'un jeu multijoueur avec Socket.io	3
4	Fonctionnalités du jeu	4
4.1	Personnalisation de la partie	5
4.2	Fonctionnalités internes :	5

1 Introduction

Dans ce rapport, nous présentons un jeu d'échecs multijoueurs implémenté en JavaScript, HTML et CSS. Le jeu est conçu pour permettre à deux joueurs de s'affronter en temps réel sur une interface web. Dans ce rapport, nous décrirons les fonctionnalités de notre jeu d'échecs en ligne. Nous expliquerons également le mode d'emploi du jeu et les technologies que nous avons utilisées pour construire notre projet.

2 Cahier des charges fonctionnel

2.1 Fonctionnalités principales

- Un plateau de jeu d'échecs en ligne, jouable en temps réel contre un adversaire.
- Les joueurs peuvent personnaliser leur partie.
- La possibilité de jouer contre un adversaire aléatoire ou de rechercher un adversaire spécifique.
- Afficher les mouvements des pièces de chaque joueur sur le plateau de jeu en temps réel.
- Afficher les captures de pièces sur le plateau de jeu en temps réel.
- Gérer les règles du jeu, y compris la vérification des échecs et des mats.
- Permettre aux joueurs de proposer un match nul ou de se rendre.

2.2 Conception technique

Pour développer ce jeu, nous avons utilisé plusieurs frameworks et bibliothèques :

- Bootstrap : C'est un framework CSS, HTML et JavaScript, qui fournit des outils pour créer des interfaces web adaptatives rapidement et facilement.
- Node.js : C'est un environnement d'exécution JavaScript qui permet de gérer le serveur.
- Socket.io est une bibliothèque JavaScript qui permet une communication en temps réel entre les joueurs et le serveur.

3 Technologies utilisés

3.1 Création du jeu d'échecs

Lors de la création d'un jeu d'échecs en JavaScript, il est courant d'utiliser des fichiers `game.js` et `graphics.js` qui contiennent les règles du jeu, y compris le déplacement des pièces et les conditions de victoire, ainsi que la représentation graphique de l'échiquier et des pièces. Cependant, ces fichiers peuvent nécessiter des modifications pour s'adapter à la version finale du jeu. Dans ce projet nous avons utilisé `game.js` et `graphics.js`. Nous les avons importés dans notre projet et les avons modifiés en fonction de nos besoins spécifiques. Par exemple, nous pouvons ajouter des fonctions supplémentaires pour gérer la logique multijoueur en temps réel ou des effets sonores pour améliorer l'expérience de jeu.

3.2 Création d'un jeu multijoueur avec Socket.io

Pour créer un jeu multijoueur, nous pouvons utiliser Socket.io, une bibliothèque JavaScript qui permet la communication en temps réel entre les navigateurs web et les serveurs. Socket.io permet aux joueurs de se connecter au serveur et de communiquer en temps réel, permettant ainsi une expérience de jeu fluide et interactive.

Pour cela, nous devons configurer un serveur Node.js qui utilise Socket.io pour permettre la communication entre les joueurs. Lorsqu'un joueur effectue un mouvement, nous pouvons envoyer les données de mouvement au serveur, qui les transmet à l'autre joueur connecté. De cette façon, chaque joueur peut voir le mouvement effectué par l'autre joueur en temps réel. Voici le flux d'information entre deux joueurs et le serveur en utilisant Socket.IO :

1. Les joueurs se connectent au serveur en utilisant la méthode `socket.emit('connect')`.
2. Le serveur reçoit la demande de connexion et envoie un événement `connection` à chaque joueur en utilisant la méthode `socket.emit()`.
3. Les joueurs envoient des données au serveur en utilisant la méthode `socket.emit()`.
4. Le serveur reçoit les données et les traite en fonction de la logique du jeu.
5. Le serveur envoie des données à chaque joueur en utilisant la méthode `socket.emit()`.
6. Les joueurs reçoivent les données et les utilisent pour mettre à jour leur interface utilisateur.
7. Le flux d'informations se poursuit jusqu'à la fin du jeu ou jusqu'à ce que les joueurs se déconnectent du serveur en utilisant la méthode `socket.emit('disconnect')`.

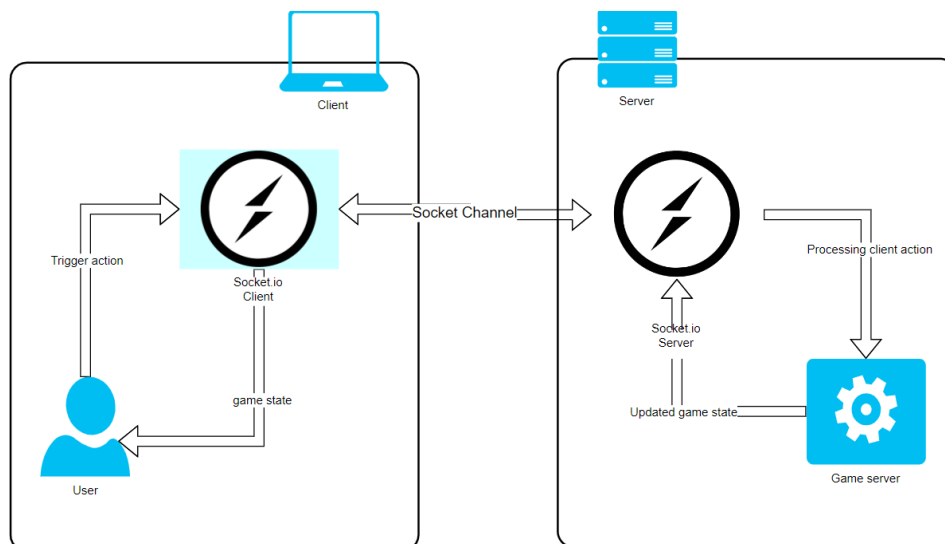


FIGURE 1 – Principe de Socket.io

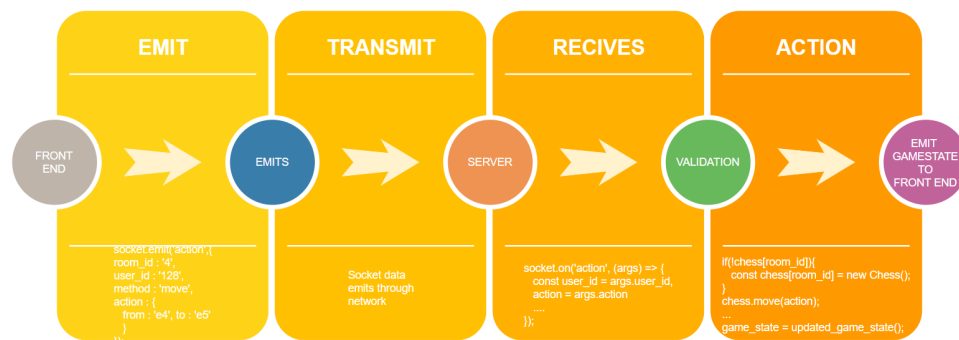


FIGURE 2 – Flux d'information

4 Fonctionnalités du jeu

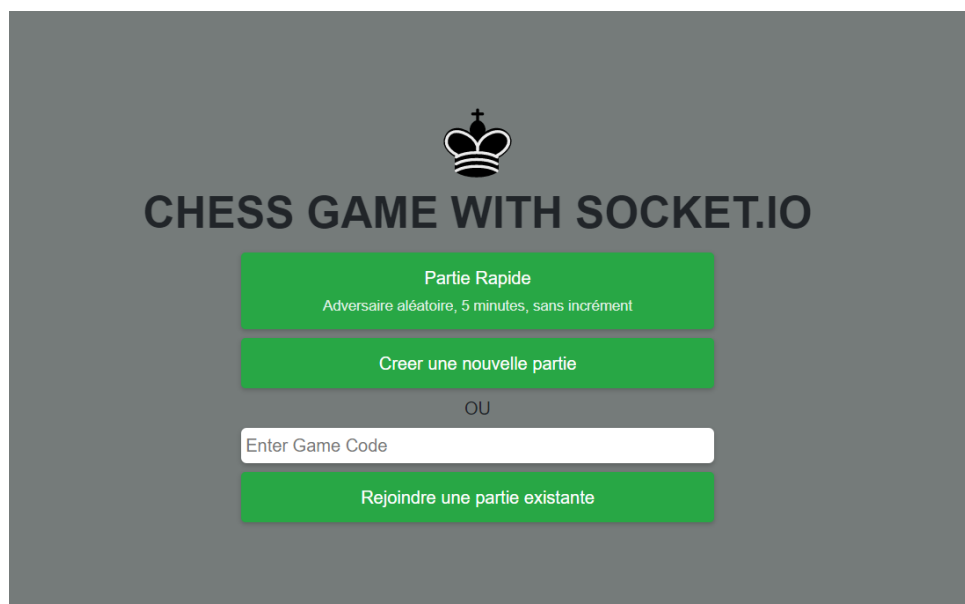


FIGURE 3 – Page d'accueil

4.1 Personnalisation de la partie

Partie rapide :

Permet de lancer une partie rapide de 5 minutes pour chaque joueur sans incrément.

Créer une nouvelle partie :

Permet de Créer une partie et de choisir le temps, l'incrément et la couleur.

Rejoindre une partie existante :

Permet de rejoindre une partie déjà créée en entrant le code de la partie.

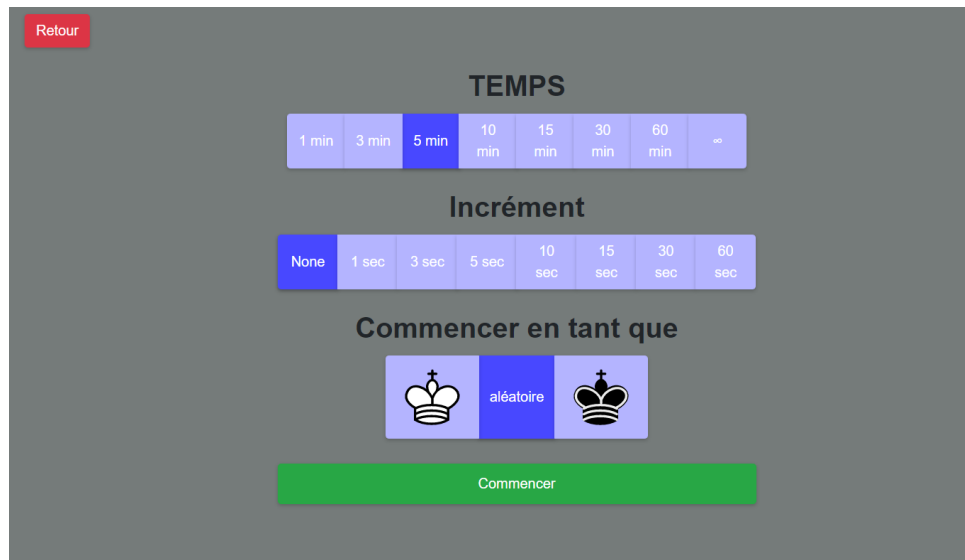


FIGURE 4 – Page d'accueil

4.2 Fonctionnalités internes :

Revenir en arrière :

Permet aux joueurs de demander de revenir en arrière dans la partie et de rejouer leur dernier coup.

Demande d'égalité :

Permet à un joueur de proposer un match nul à son adversaire.

Abandonner :

Permet à un joueur d'abandonner et quitter la partie et de reconnaître la victoire de son adversaire.

Rejouer :

Permet aux joueurs de jouer une nouvelle partie après la fin de la partie actuelle.

Messagerie instantanée (chat) :

Permet aux joueurs de communiquer entre eux et envoyer des messages en temps réel pendant la partie.

Gestion des scores :

Permet aux joueurs d'enregistrer les scores et suivre leur progression et leur performance au fil du temps.

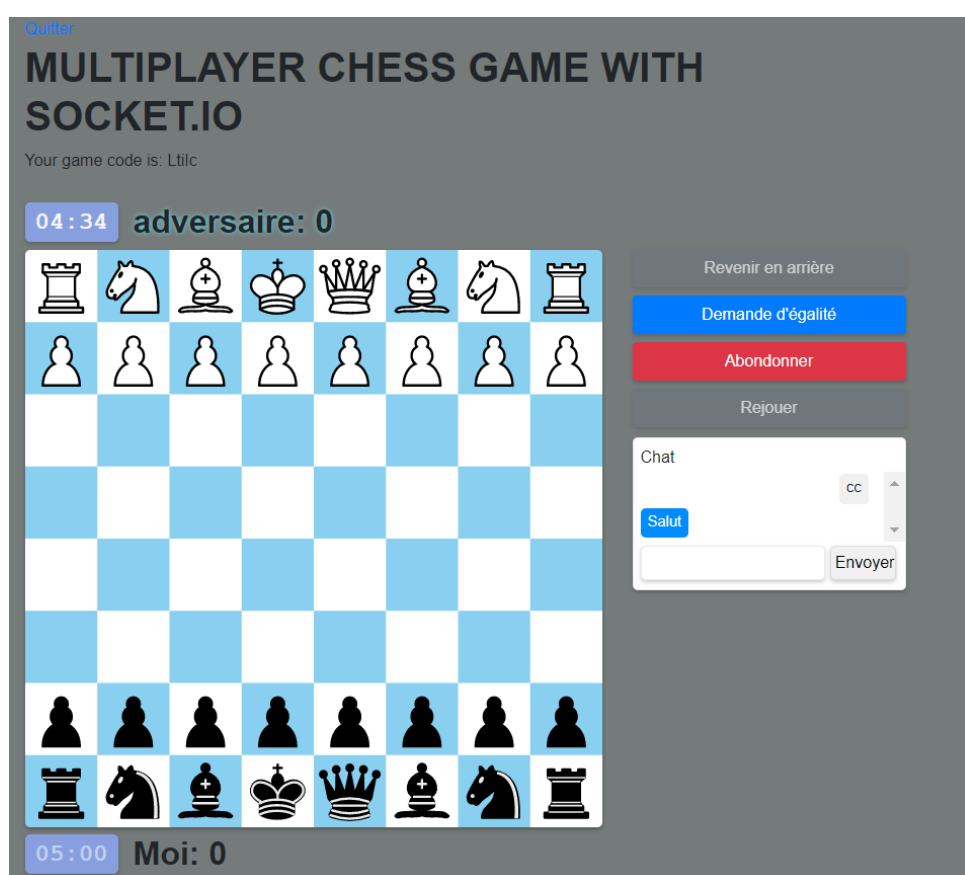


FIGURE 5 – Chess game