



ÉCOLE CENTRALE LYON

RAPPORT – APPLICATIONS WEB

Développement d'une Application Web Multijoueur : Jeu d'Échecs

Élèves :

Amani KRID

Youssef MILED

Enseignant :

Romain VUILLEMOT

René CHALON

Table des matières

1	Introduction	2
2	Objectifs du Projet	2
3	Architecture Générale	2
4	Fonctionnalités Développées	3
4.1	Authentification	3
4.2	Gestion des Parties	3
4.3	Base de Données	3
5	Interface Utilisateur	3
5.1	Accueil	3
5.2	Jeu	4
6	Scénarios d’Utilisation	4
6.1	Scénario Nominal	5
6.2	Autres Scénarios	5
7	Sécurité de l’Application	6
7.1	Protection des Données	6
7.2	Entrées Utilisateur	6
7.3	En-têtes HTTP	6
7.4	Limitations des Requêtes	6
8	Spécificités Techniques	6
9	Difficultés Rencontrées	7
10	Perspectives d’Amélioration	7
11	Conclusion	7

1 Introduction

Ce projet a été réalisé dans le cadre de l'unité d'enseignement "Appllications Web". Il s'agit d'une application web permettant à deux joueurs de s'affronter à un jeu d'échecs. Le système repose sur une architecture client-serveur, avec une communication bidirectionnelle via WebSockets.

Ce rapport présente les choix techniques, l'architecture de l'application, les fonctionnalités développées, ainsi que les difficultés rencontrées et les perspectives d'amélioration.

2 Objectifs du Projet

Le projet a pour ambition de concevoir une application de jeu interactive en ligne reposant sur les technologies web modernes, en particulier l'élément HTML5 `<canvas>` et SocketIO. Les objectifs principaux sont les suivants :

- **Exploiter l'élément HTML5 `<canvas>`** pour réaliser une interface graphique interactive, capable de représenter dynamiquement les éléments du jeu (plateau, pièces, déplacements).
- **Permettre le jeu en ligne** : une fois connecté, deux joueurs peuvent jouer sur la même interface.
- **Gérer les scores et les statistiques des joueurs** en mettant en place un système de persistance des données (hall of fame), permettant de suivre les performances, les victoires et les défaites.
- **Favoriser une architecture modulaire et réutilisable** pour faciliter l'évolution future du projet vers d'autres variantes ou d'autres types de jeux de plateau.

3 Architecture Générale

L'application repose sur une architecture MVC (Model, Vue, Controller) :

- **Front-end (Vue)** : HTML, CSS et JavaScript.
- **Back-end (Controller)** : Node.js avec Express, Socket.IO, et SQLite.
- **Communication** : WebSockets pour les échanges temps réel.
- **Base de données (Model)** : SQLite3 pour stocker les utilisateurs, parties et statistiques.

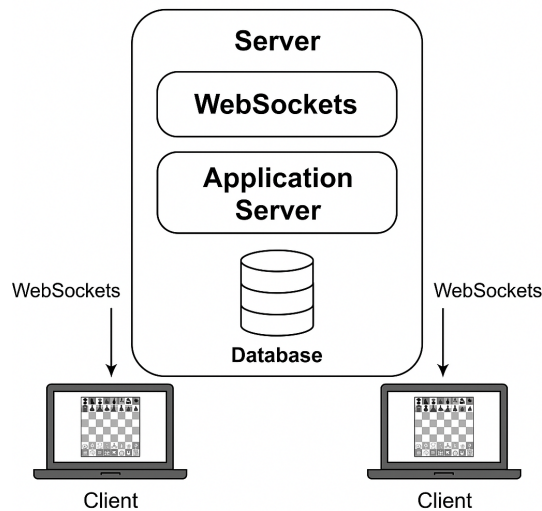


FIGURE 1 – Schéma de l’architecture client-serveur de l’application

4 Fonctionnalités Développées

4.1 Authentification

L’utilisateur peut s’inscrire ou se connecter via un formulaire. Les mots de passe sont sécurisés avec `bcrypt` et les sessions sont gérées via `express-session`.

4.2 Gestion des Parties

Une fois une partie commence :

- Chaque joueur a un temps de jeu total (10 minutes).
- Les coups sont transmis via Socket.IO.
- La partie est enregistrée avec les coups, le gagnant, et les temps restants.

4.3 Base de Données

Deux tables principales :

- **users** : id, username, password_hash, stats...
- **games** : id, player1_id, player2_id, moves, result, timers...

5 Interface Utilisateur

5.1 Accueil

Page d’accueil avec :

- Formulaire de connexion/inscription,
- Bouton d’instruction (affiche les règles du jeu).
- Animation d’attente en cas d’appariement.

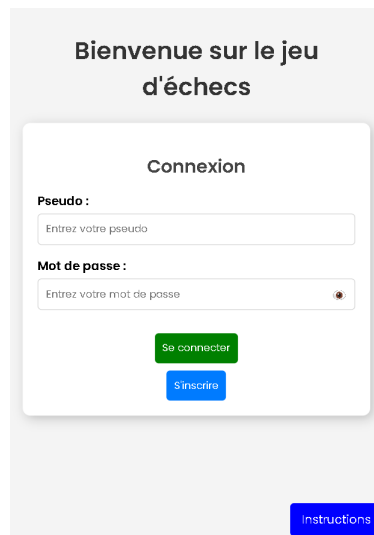


FIGURE 2 – Interface d’accueil du projet

5.2 Jeu

- Plateau d’échecs interactif.
- Indicateur de temps restant.
- Bouton pour une nouvelle partie.



FIGURE 3 – Interface du jeu du projet

6 Scénarios d’Utilisation

Notre application repose sur une logique d’interaction fluide entre les utilisateurs et le serveur. Plusieurs scénarios typiques ont été modélisés afin de mieux comprendre le déroulement du jeu et anticiper les cas particuliers.

6.1 Scénario Nominal

L'utilisateur s'identifie via l'interface. En cas d'identifiants valides, il est connecté, sinon il crée un nouveau compte. Il est ensuite redirigé vers le jeu. Les coups sont joués tour à tour, et le résultat final est enregistré.

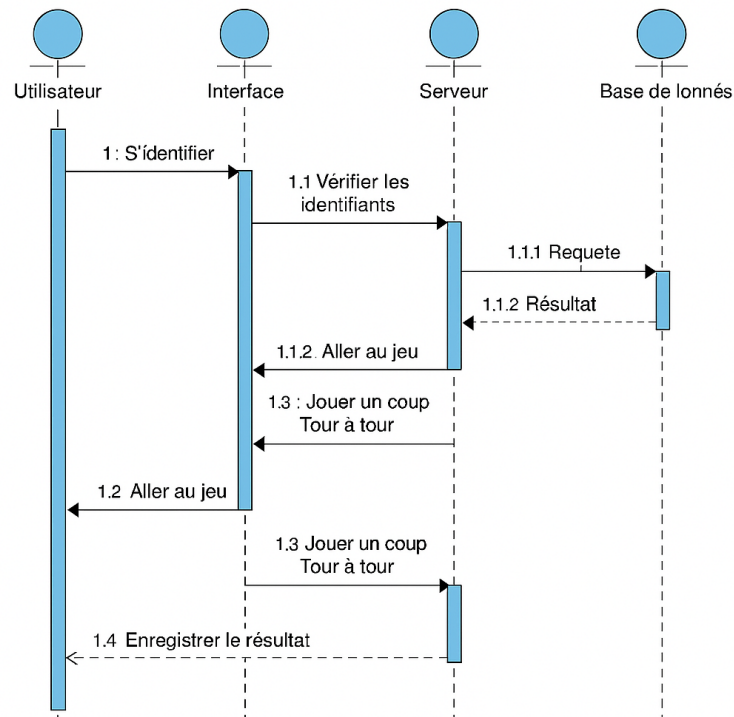


FIGURE 4 – Diagramme de séquence : scénario normal de connexion et jeu

6.2 Autres Scénarios

- **Abandon d'un joueur** : le joueur restant gagne automatiquement.
- **Timeout d'un joueur** : l'utilisateur reçoit un message puis est redirigé.
- **Echec et mat** : l'un des joueurs perd la partie lorsque son roi est menacé de capture au prochain coup et pour laquelle aucune parade n'est possible pour l'éviter.

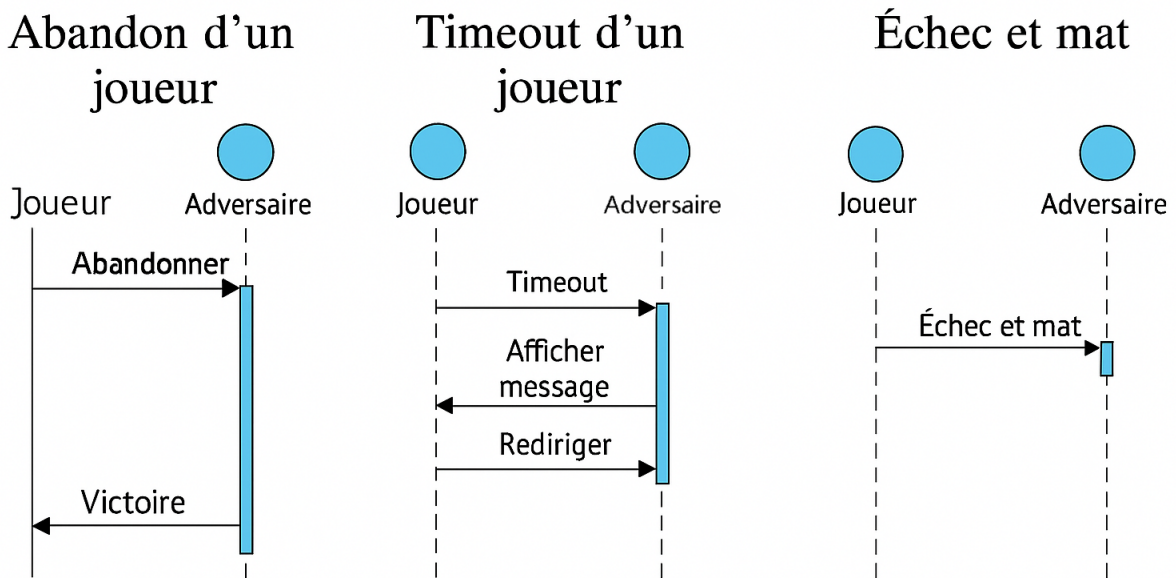


FIGURE 5 – Diagramme de séquence : différent scénarios de fin de partie

7 Sécurité de l'Application

Nous avons mis en place plusieurs mécanismes pour protéger l'application :

7.1 Protection des Données

Les mots de passe sont hashés avec `bcrypt` pour éviter toute fuite en clair.

7.2 Entrées Utilisateur

`sanitize-html` empêche toute injection de scripts malveillants dans les champs utilisateur.

7.3 En-têtes HTTP

`helmet` renforce la sécurité des communications HTTP via des en-têtes standardisés.

7.4 Limitations des Requêtes

`express-rate-limit` empêche les attaques par force brute en limitant les tentatives par IP.

8 Spécificités Techniques

- Communication bidirectionnelle avec `Socket.IO`.
- Gestion de sessions utilisateurs avec `express-session`.
- Stockage et persistance des données via `SQLite3`.
- Interface responsive avec `HTML/CSS/JavaScript`.

9 Difficultés Rencontrées

- Synchronisation et appariement des utilisateurs.
- Gestion des déconnexions imprévues.
- Problèmes de verrouillage dans SQLite en cas d'écriture simultanée.
- Implémentation d'un robot de jeu basique.

10 Perspectives d'Amélioration

- Intégrer une IA d'échecs comme Stockfish.
- Ajouter un système de classement ELO.
- Créer des profils utilisateurs avec historique.
- Permettre de jouer des parties en temps réel.
- Passer à PostgreSQL pour une meilleure scalabilité.

11 Conclusion

Ce projet a permis de mettre en pratique des compétences avancées en développement web, sécurité, et gestion de base de données. Il constitue une base solide pour le développement futur d'une plateforme de jeu plus aboutie.