



ÉCOLE CENTRALE LYON

PUISSANCE 4  
RAPPORT

---

## Rapport de développement web

---

*Élèves :*

Simon LANGLAIS  
Quentin ALCAYDE

*Enseignant :*

René CHALON  
Daniel MULLER

5 avril 2022

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Cahier des Charges</b>	<b>2</b>
<b>3</b>	<b>Logique du Jeu</b>	<b>2</b>
<b>4</b>	<b>Utilisation du WebSocket : socket.io</b>	<b>2</b>
4.1	Gestion des salles . . . . .	2
4.2	Chat . . . . .	3
4.3	Plateau de jeu . . . . .	3
<b>5</b>	<b>Pages de notre application</b>	<b>3</b>
5.1	Page d'accueil . . . . .	3
5.2	Salle de jeu . . . . .	4
5.3	Plateau de jeu . . . . .	4
<b>6</b>	<b>Difficultés</b>	<b>5</b>
<b>7</b>	<b>Conclusion</b>	<b>5</b>

## 1 Introduction

Afin de rendre compte des possibilités qu'offre le langage javascript, naturellement accompagnée des langages html et css, nous nous sommes lancés dans la création d'un jeu multijoueur du jeu de société "PUISSANCE 4".

Initialement, nous comptions faire un morpion avant de nous rendre compte que la logique du jeu s'apparentait à celle du puissance 4. Cependant, afin de complexifier le projet, nous avons décidé d'accompagner le jeu d'un chat. En effet, le gestionnaire d'évènement asynchrone Nodejs couplé à l'API socket.io nous permettait d'implémenter cette fonctionnalité, pour toujours plus de parties endiablées sur Puissance4.

## 2 Cahier des Charges

Les consignes fournies dans l'énoncé ainsi que celle que nous nous sommes imposés peuvent être résumés de la manière suivante :

- Jeu du Puissance 4 jouable à deux personnes
- messagerie instantanée (chat) en parallèle du jeu.
- esthétisme travaillé et fluidité du jeu ainsi que du chat.
- Possibilité d'avoir un certain nombre de room disponible pour qu'il y ait plusieurs parties en simultané.

## 3 Logique du Jeu

Naturellement, nous allons créer une classe Puissance4 disposant de toutes les fonctionnalités dans le fichier puissance.js. En effet, l'orientation objet est ici particulièrement appropriée, compte tenu de la nature du jeu (plateau doté d'une grille, avec deux types de pions, au tour par tour).

Le jeu se déroule de la manière suivante : le joueur 1 appuie sur un des trous de la grille (tableau à deux dimensions). L'information est récupérée par le gestionnaire d'évènement qui va alors appeler la méthode win (en ayant au préalable vérifié que la colonne n'était pas pleine) qui va chercher si ce nouveau pion conduit à un gagnant. Si c'est le cas, un pop-up apparaît et le jeu se termine. Sinon, la méthode render est appelée qui va afficher le nouveau pion et laisser la tour au joueur 2).

## 4 Utilisation du WebSocket : socket.io

Nous avons utilisé le websocket : socket.io, le serveur a été développé en express.

### 4.1 Gestion des salles

On a utilisé les fonctionnalités de websocket afin de créer des salles de jeu dans lesquelles on retrouve le chat et le plateau de jeu. Chaque joueur sera redirigé vers la bonne salle de jeu via l'URL généré dans la page d'accueil du site.

## 4.2 Chat

On a réalisé un chat en ligne, tout d'abord on repère l'évènement de connexion d'un joueur dans la salle, et on envoie côté serveur un message de bienvenue. Côté serveur également, un message pour prévenir l'arrivée d'un nouveau joueur est envoyé automatiquement.

Enfin la gestion des messages est côté client et côté serveur, on récupère la requête d'envoi du message côté client, qui passe par le serveur et renvoie aux clients connectés dans la salle en actualisant le DOM.

## 4.3 Plateau de jeu

Le plateau est un tableau html généré par le code javascript. Toutes les interactions utilisateurs et serveur agissent donc sur le tableau, qui sera affiché par la méthode `render()`. Pour évoquer un peu de code, un des problèmes rencontrés fut que l'utilisation de l'objet `Puissance4` et de son interaction nous a contraint à chercher dans la documentation de `socket` afin de trouver une manière d'envoyer des évènements à l'autre utilisateur présent dans la room mais pas celui à l'origine de l'émission. Pour cela, côté serveur, nous avons utilisé la ligne de code :

```
socket.to(room).emit(event)
```

qui envoie l'évènement `event` à tous les utilisateurs de la room excepté l'envoyeur.

# 5 Pages de notre application

## 5.1 Page d'accueil

La page d'accueil de notre application web est composé d'une interface permettant de choisir la salle de jeu, ainsi que le surnom souhaité.

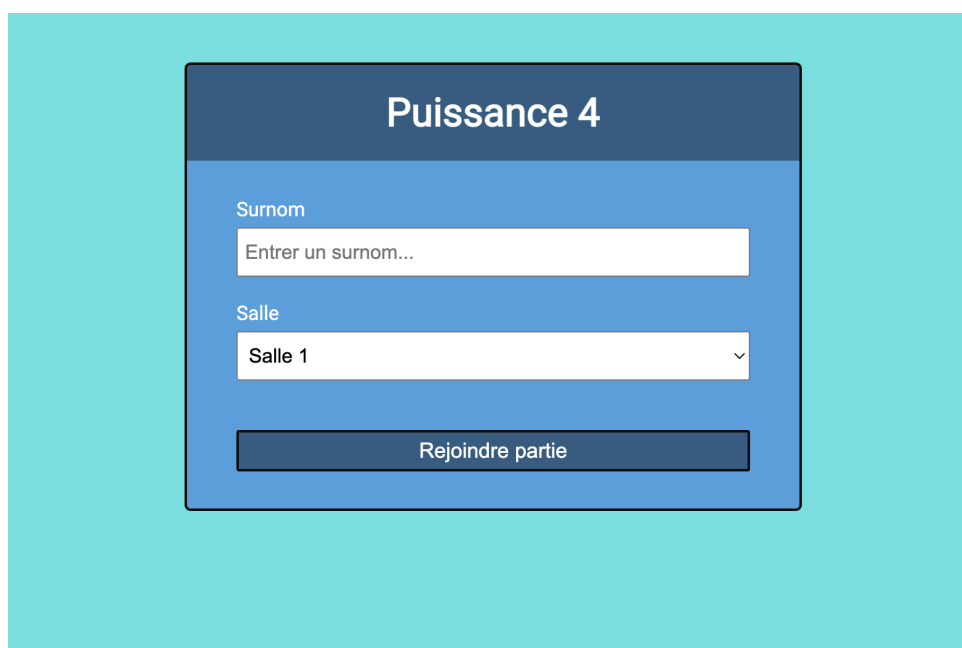


FIGURE 1 – Page d'accueil de notre site web

Après avoir renseigné les informations il suffit de cliquer sur le bouton rejoindre la partie pour entrer dans une salle.

## 5.2 Salle de jeu

Après être entré dans une salle de jeu, on peut voir le chat à gauche de la page web, tant que deux joueurs ne sont pas dans la salle, aucun jeu ne s'affiche. Dès que deux joueurs rentrent dans la salle le plateau de jeu s'affiche et les deux joueurs peuvent alors commencer la partie.



FIGURE 2 – Chat

## 5.3 Plateau de jeu

Il se met à jour automatiquement lorsqu'un joueur place un pion. Le premier joueur à jouer est automatiquement le premier à être entré dans la salle de jeu. Des messages d'alertes apparaissent lorsqu'un joueur tente de jouer en dehors de son tour et lorsque la partie est finie. Pour relancer une partie il suffit qu'un des deux joueurs clique sur le plateau de jeu une fois la partie finie.

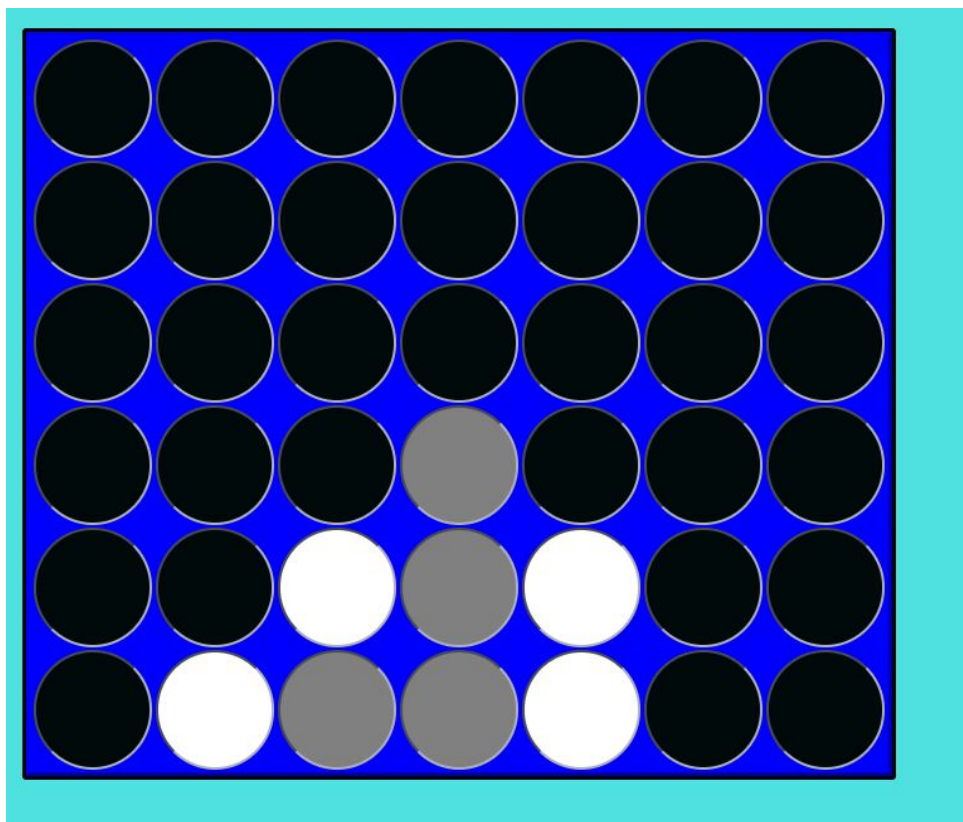


FIGURE 3 – Plateau du Jeu

## 6 Difficultés

Dans ce projet, nous avons commis l'énorme erreur de penser notre code de manière modulaire, c'est à dire de croire que nous pouvions coder chaque partie (chat, jeu, serveur) séparément et de ensuite les réunir. Nous avons en effet codé en premier notre jeu en local via l'utilisation du paradigme objet, en pensant qu'il serait aisé de l'adapter pour sa version en ligne. Ceci explique le retard accumulé ainsi que de nombreux bugs qu'il nous a fallu corriger. Cependant, à force de sueur et de persévérance, nous avons fini par y arriver et avons également acquis une connaissance poussée de certains mécanismes que prodiguent nodejs et socket.IO.

## 7 Conclusion

Ce projet nous aura permis de mettre en oeuvre les compétences développées dans ce module, coder en javascript et mettre en place une structure de serveur à travers ce jeu multijoueur. De plus il nous aura permis de remobiliser des compétences acquises l'an passé en HTML et CSS.