



ÉCOLE CENTRALE LYON

S8 APPLICATION WEB
RAPPORT

Le jeu du 2048

Élèves :

Nicolas EYCHÈNE
Nicolas DUMAS

Enseignant :

Daniel MULLER
René CHALON

9 avril 2021

Table des matières

1	Introduction	2
2	Cahier des charges	2
3	Présentation du jeu	2
3.1	Rappel des règles du 2048	3
3.2	Adaptation pour un jeu multijoueur	4
4	Technique	4
4.1	Technologies utilisés	4
4.2	Gestion du plateau	4
4.3	Communication serveur/client	5
5	Perspective d'amélioration	5

1 Introduction

Dans le cadre de l'électif "Application web" nous avons réaliser un jeu de 2048 en multijoueur.

2 Cahier des charges

Jeu en multijoueur

Le jeu créé doit se jouer en multijoueur, dans notre cas à deux joueurs. Dans notre cas, le jeu se joue sur un serveur local.

Chaque joueur devra donc se connecter au serveur local avec un pseudo unique. deux joueurs ne peuvent pas avoir le même pseudo sur le serveur.

Utilisation d'un lobby et de salons

L'application doit contenir un lobby dans lequel tous les joueurs arrivent lorsqu'ils se connecte sur le serveur avec dans ce lobby l'affichage des différents salon disponible. L'utilisateur doit pouvoir rejoindre un salon en cliquant dessus.

Gestion des connexion et déconnexion

Le serveur doit pouvoir détecter la connexion et la déconnexion d'un utilisateur pour mettre à jour l'affichage. L'application doit aussi clore une partie en cours si un des joueurs se déconnecte.

Dans l'application, il faut aussi détecter lorsqu'un joueur quitte le lobby pour entré dans un salon.

La connexion à un salon doit permettre aux deux premiers utilisateurs d'être joueurs dans ce salon et aux suivants d'être spectateurs et d'assister à la partie en cours.

Gestion des scores

Le jeu étant en multijoueur il faut définir un vainqueur à la fin de la partie. Pour cela nous utiliserons une notion de score. Ce score devra donc être propre à chaque joueur et actualiser au fil de la partie pour pouvoir déterminer le gagnant.

Possibilité de discuter avec les autres utilisateurs

L'application doit contenir un chat pour permettre aux joueurs de parler entre eux lorsqu'il sont dans le même salon ou s'ils sont tout les deux ensemble dans le lobby.

3 Présentation du jeu

Notre jeu est une version adaptée pour le multijoueur du célèbre jeu 2048.

3.1 Rappel des règles du 2048

Dans cette section est présenté les règles du jeu original du 2048. Il s'agit d'un jeu individuel. Le jeu est basé sur une grille 4x4 avec des numéros (ou du vide) dans chaque case. La grille de départ ne contient que quatre cases aléatoires avec des 2 ou des 4.

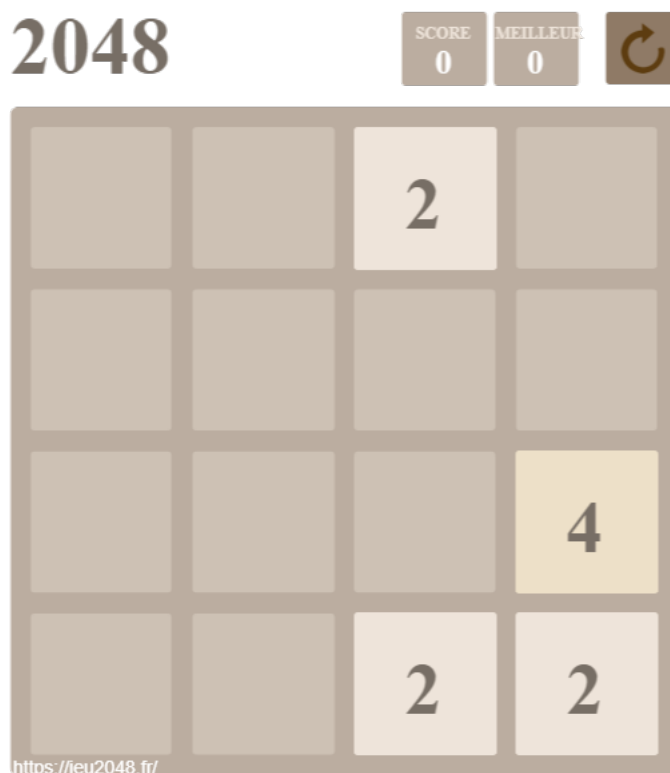


FIGURE 1 – Grille du 2048 au début d'une partie

A chaque tour quatre mouvements sont possibles : un déplacement vers la droite, la gauche le haut ou le bas. Lors d'un déplacement vers la droite par exemple, tous les nombres sont repoussés vers le bord droit du plateau. Si deux nombres identiques rentrent en collision ils fusionnent en leur somme (par exemple un 4 et un quatre qui fusionnent deviennent un 8). Après chaque mouvement un deux ou un quatre apparaissent sur une case vide aléatoire. Voici un exemple pour illustrer le propos

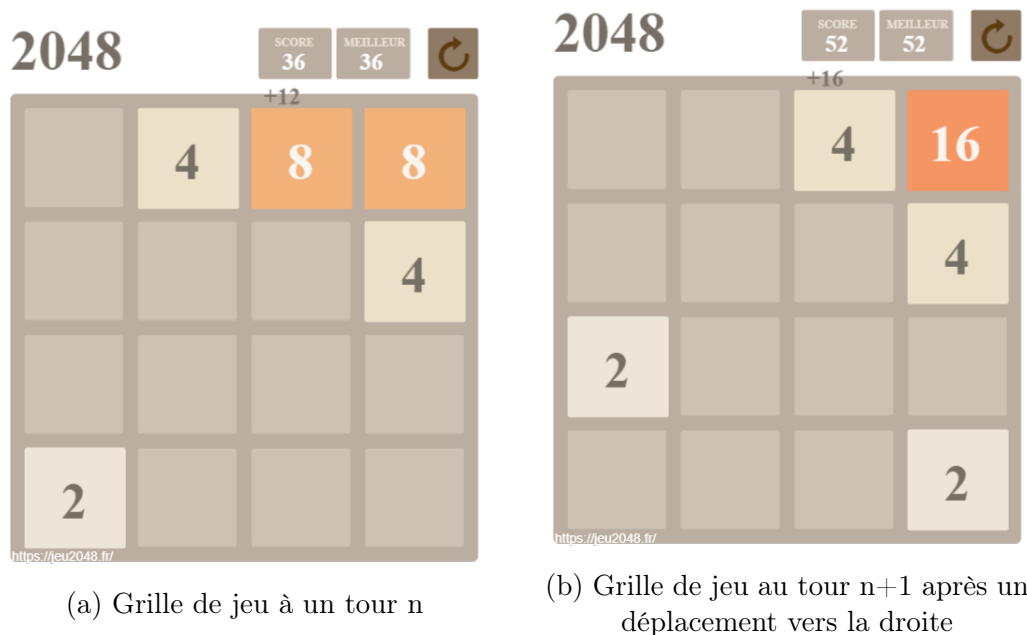


FIGURE 2 – Exemple d'un déplacement vers la droite

Le jeu s'arrête lorsque plus aucun déplacement n'est autorisé (grille remplie + aucun nombres identiques cotes à cotes)

3.2 Adaptation pour un jeu multijoueur

Nous avons adapté les règles du jeu pour qu'il devienne un jeu à 2 joueurs. Pour cela chaque joueur joue chacun son tour sur la même grille de jeu.

Dans une partie opposant un joueur A à un joueur B, si lors d'un tour le joueur A fusionne deux nombres de même valeur x alors le joueur A gagne $2x$ points puis c'est au joueur B de jouer.

Lorsque la grille ne peut plus être modifiée le jeu se termine et le joueur avec le score le plus élevé gagne

4 Technique

4.1 Technologies utilisés

Les codes ont été faits sous JavaScript avec la plateforme logicielle libre Node.js. La bibliothèque libre jQuery est également utilisée côté client pour simplifier l'écriture du script dans le code HTML. Enfin un document CSS a été créé pour gérer le style du projet.

Nous avons utilisé le module socket.io pour effectuer la connexion entre le serveur et le client et transmettre les requêtes entre les deux.

4.2 Gestion du plateau

Le plateau de jeu est représenté par une matrice 4×4 côté serveur que l'on transmet au client pour actualiser l'affichage chez les deux joueurs. Tous les changements au niveau

du plateau se font côté serveur, le client ne fait que transmettre une requête au serveur qui l'analyse et ensuite renvoie le plateau modifié.

Les modification du plateau s'effectue grâce à 4 fonctions qui gèrent chacune un mouvement (vers la gauche, le haut...) puis on effectue un test sur le plateau pour savoir si la partie est terminée ou non.

4.3 Communication serveur/client

La communication serveur/client s'effectue à l'aide de requêtes émises par le serveur ou le client et reçu par l'autre. Ces requêtes sont effectuées avec le module socket.io qui est très pratique et relativement facile d'utilisation.

Dans ce projet nous utilisons un grand nombre de requêtes pour actualiser l'affichage (plateau, scores...), pour gérer l'envoi de message via le chat mais aussi pour réinitialiser la partie après une victoire ou une déconnexion en affichant une fenêtre particulière pour chaque type de fin de partie.

5 Perspective d'amélioration

Pour améliorer ce projet, nous pourrions faire héberger ce dernier sur un serveur externe pour permettre à deux joueur deux jouer ensemble depuis deux ordinateurs différents, ce qui rendrait le jeu plus accessible pour faire une partie avec un ami.

Pour l'instant, notre application ne possède qu'un seul salon de jeu donc seulement deux joueurs peuvent jouer en même temps, on pourrait donc rajouter plusieurs salons pour pouvoir lancer plusieurs parties en même temps et laissé au joueur le choix du salon dans lequel aller. Il aurait également possible de créer automatiquement un nouveau salon quand un nouveau joueur se connecte pour toujours permettre aux joueurs de jouer sans avoir à attendre qu'un salon se libère

En plus de cela, nous pourrions crée un CPU pour permettre au joueur de jouer seul contre une machine.

Enfin nous pourrions améliorer le design du projet pour le rendre plus attrayant (rendre les boutons de mouvements plus joli, mettre une couleur sur les cases et faire changer cette couleur en fonction de la valeur de la case pour rendre le jeu plus visuel...)

Annexe : Captures d'écran de l'application WEB

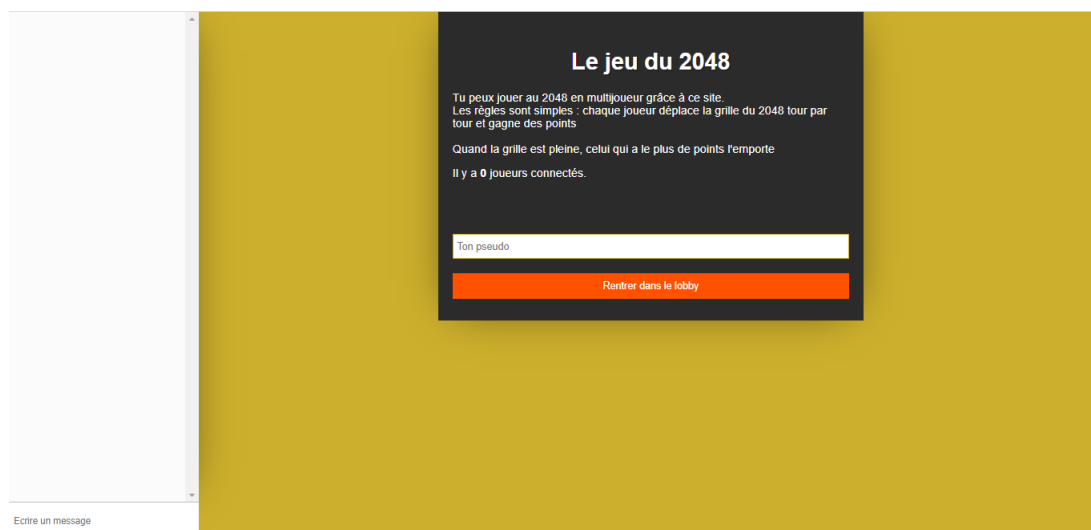


FIGURE 3 – Capture de l'accueil

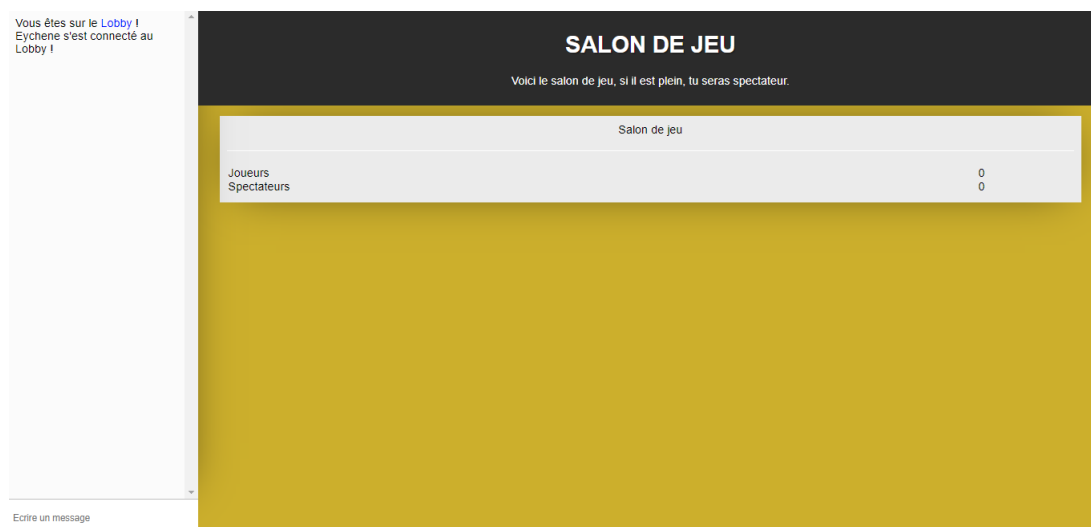


FIGURE 4 – Capture du lobby



FIGURE 5 – Capture du jeu : C'est votre tour



FIGURE 6 – Capture du jeu : C'est le tour adverse