



Projet Applications Web

Bataille Navale

Cédric Ernewein
Isadora Nguyen Van Khan

Il y a deux phases principales à gérer :

La phase de placement des bateaux de chaque joueur (avant que cette phase soit terminée on ne peut pas encore tirer).

La phase de tir chaque joueur tir à son tour, le but étant de faire couler tous les bateaux ennemis. Une fois que tous les bateaux ennemis sont coulés, le joueur a gagné.

Le client

L'interface est constituée de deux grilles. L'une représente la grille du joueur, celle sur laquelle le joueur va placer ses bateaux et l'autre représente celle de l'adversaire, c'est-à-dire celle sur laquelle le joueur va tirer.

Lorsqu'un joueur se connecte sur la page on lui demande son pseudo de joueur, que l'on va ensuite afficher sur l'interface.

Sur l'interface est aussi présente une liste déroulante permettant de choisir quel bateau on veut placer.

Lors de la phase de tir, les cases sur lesquelles on a tiré se colorent soit en bleu : on n'a pas touché de bateau adverse, soit en vert : on a touché un bateau. Si l'adversaire touche un de nos bateaux, la case correspondante sur notre grille se colore en vert.

Les cases sur lesquelles se trouvent les bateaux du joueur sont colorées en rouge.

La coloration des cases est gérée à l'aide de jquery.

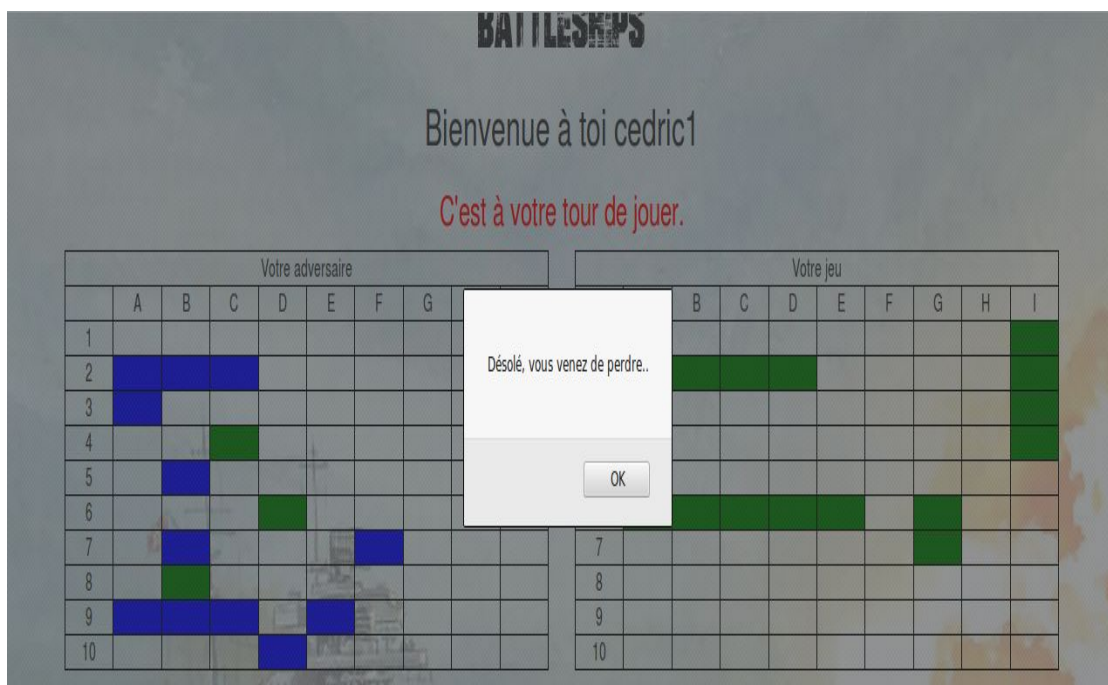


Toutes les actions effectuées sont envoyées au serveur et traitées par celui-ci. Par exemple la phase des bateaux :

Lorsque le joueur clique on teste s'il est bien en train de placer les bateaux et on teste aussi s'il les place bien (pas de travers, si la taille est bonne..). Si tout est bon, on envoie une requête au serveur « `socket.emit('placement_bateaux', {data})` ».

Le serveur traite ce placement et renvoie au client la liste des bateaux qu'il a déjà placé afin de colorer les cases correspondantes. Ensuite, lorsque le joueur a placé ses 5 bateaux, le client envoie au serveur une requête « `socket.emit('fin_placement_bateaux', {data})` ». Si les deux joueurs ont fini ils peuvent passer à la phase de tir, sinon l'un doit attendre l'autre.

Pour la phase des tirs, chaque joueur choisit tour à tour une case sur laquelle tirer à l'aide de la grille "adversaire". Le client envoie ensuite la requête tir au serveur. Le serveur retourne la grille et dit si l'un des deux joueurs a gagné. Le message est ensuite affiché.



Le client envoie au serveur les signaux socket.io suivant:

- connection
- disconnect
- new_user (permet de connaître le pseudo du joueur et d'initialiser le jeu pour lui)
- placement_bateaux (permet de placer les bateaux du joueur dans la grille de jeu)

- fin_placement (signal renvoyé lorsque le placement des bateaux est terminé)
- tir (signal envoyé lorsque le joueur effectue un tir)

Les variables globales sont :

- socket (permet la connection au serveur)
- pseudo (nom du joueur)
- placement (true si on est en phase de placement, false sinon)
- mon_tour (true ou false)
- valeurs_bateaux (liste avec toutes les valeurs des bateaux)
- nbrBateaux (nombre de bateaux placés)
- listeBat (tableau associatif : {valeur d'un bateau : [nombre de cases, nombre d'extrémités placées, coordonnées de la première case si on a déjà cliqué dessus]})
- room (numéro de la room dans laquelle est l'utilisateur)

Le serveur

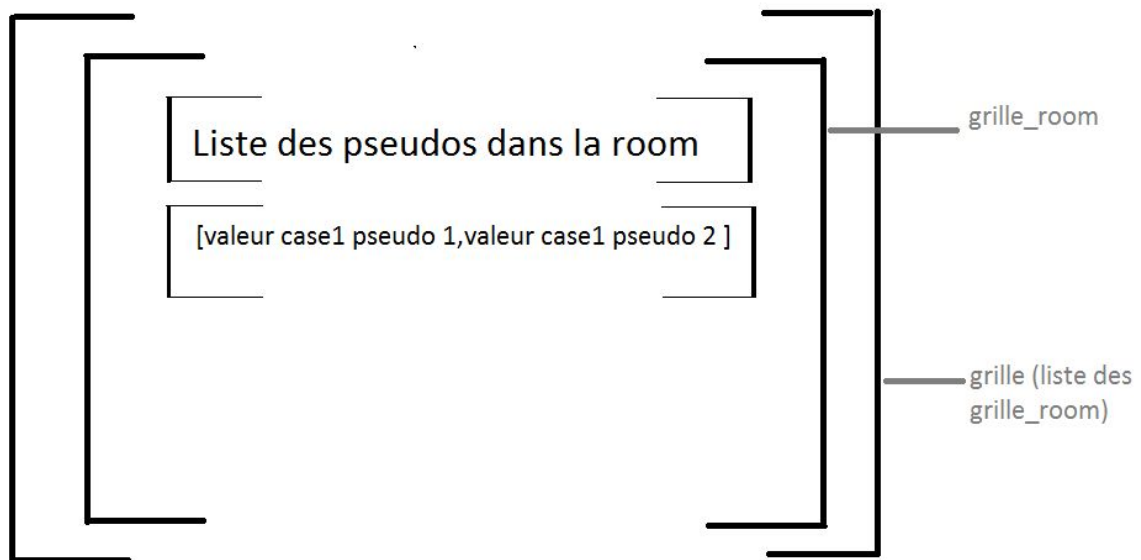
C'est le serveur qui traite toutes les actions des utilisateurs.

Le jeu est modélisé par une liste de matrice. Chaque matrice représente une salle de jeu. Le nombre de joueurs par salle de jeu est limité à 2, s'il y a plus de joueurs on crée de nouvelles rooms.

Les matrices de salle de jeu stockent les grilles de jeu des joueurs. Elle est de taille 10x10, comme la grille de jeu. Dans chaque case sont stockées deux valeurs, la 1ère correspond au 1er joueur (celui qui est arrivé en premier). 0 veut dire qu'il n'y a que de l'eau, 1 veut dire qu'il y a un bout de bateau dans la case, -1 veut dire que le bout de bateau a été touché.

Le serveur vérifie que l'emplacement des bateaux est bon (il n'y a pas d'autres bateaux) et dit si le tir a touché un bateau. Il renvoie toutes ses informations et la matrice mise à jour au client. A la fin du jeu, le serveur renvoie true ou false au client si l'un des joueurs a gagné.

Pour la phase de connexion et de déconnexion, le serveur initialise la matrice correspondant à la room (grille_room), ainsi que les tableaux associatifs (rooms, personne_room, pseudos).

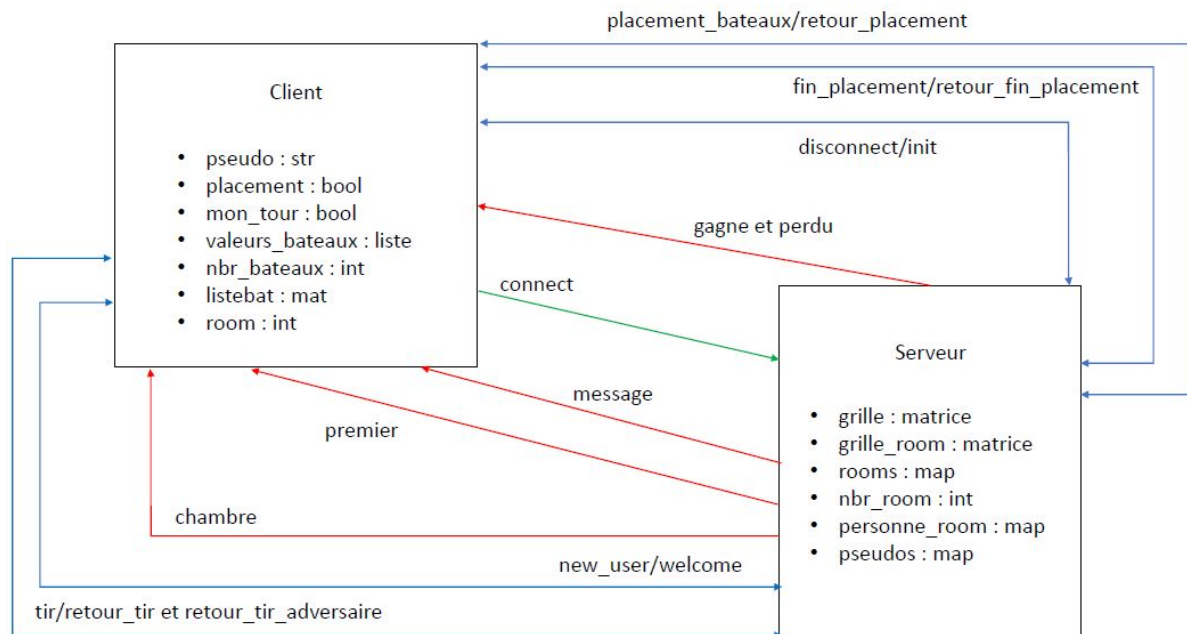


Le serveur envoie au client les signaux suivants:

- welcome (suite à new_user)
- chambre (renvoie la room de l'utilisateur)
- premier (renvoie l'ordre de passage pour la phase de tir)
- message (renvoie un message)
- init (renvoie un signal d'initialisation)
- retour_placement (suite au placement d'un bateau, permet au client de colorer les cases correspondantes)
- retour_fin_placement (suite à la fin du placement des bateaux)
- retour_tir (suite au tir du joueur, permet de colorer la case de la grille adverse suivant la réussite du tir)
- retour_tir_adversaire (suite au tir de l'adversaire, permet de colorer la case du joueur suivant la réussite du tir)
- gagne (signal envoyé si le joueur réussit à couler la flotte adverse)
- perdu (signal envoyé si le joueur s'est fait anéantir sa flotte, couplé avec gagné)

Les variables globales sont :

- io
- grille (liste des matrices représentant les salles de jeu : liste de grille_room)
- grille_room (matrices représentant une salle de jeu particulière)
- rooms (tableau associatif : {room : nombre de personnes étant dans la salle})
- nbr_room (nombre de salles existantes)
- personne_room (tableau associatif : {pseudo:room})
- pseudos (tableau associatif : {room : [pseudos des joueurs ayant terminé la phase de placement] })



Mode d'emploi

Il faut installer les nodes modules à l'aide de la commande "*npm install*".

Le serveur tourne sur le port 8080, pour le lancer il suffit de se placer dans le répertoire principal et lancer node app.js

Ouvrir un navigateur et se connecter sur adresse_ip_du_serveur:8080