



06.01.2019

Projet Web

« Bato »

Melvin Gay et Guetarni Anas

Table des matières

.....	0
Vue d'ensemble	2
Cahier des charges prévu.....	2
Utilisation de différentes librairies	2
Une architecture fondée sur une API.....	4
API Base de données MySQL	4
Différentes pages, différentes fonctionnalités	5
I. Connexion/Inscription	5
II. Jeu solo	5
III. Jeu multi-joueurs	6
IV. Classement solo/multi	6
V. Gamification	6
VI. Affichage des personnes connectées.....	7
Répartition des tâches.....	7
Une difficulté.....	8
Installation	8

Vue d'ensemble

« **Bato** » est un jeu de bataille navale développée dans le cadre d'un projet de 3^{ème} année à Hepia. Ce jeu permet de jouer seul ou à deux au mythique jeu de la bataille navale. Nous avons voulu réaliser une interface épurée afin de simplifier l'utilisation de la plateforme. Nous avons simplifié au maximum le code afin qu'il soit le plus lisible possible.

Cahier des charges prévu

Développement d'un jeu "bataille naval" sur browser à l'aide de Bootstrap et javascript (communication à l'aide de sockets) où deux joueurs pourraient s'affronter en direct sur deux ordinateurs différents.

En fonction de la vitesse de progression de notre projet, nous avons également pensé à rajouter une "gamification" à l'aide de niveau ou de classement total (une victoire rapportant un nombre de point différent d'une défaite).

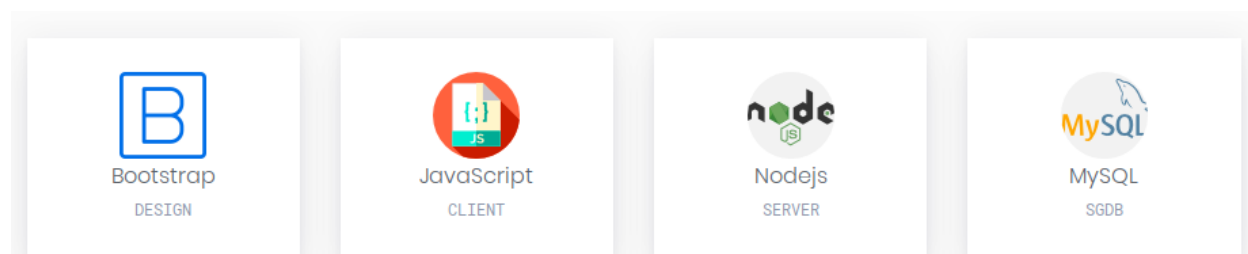
Interface de connexion et d'inscription (Bootstrap/HTML/Node Js/Socket.io)

Gamification (Base de données Mysql)

Interface de jeu (NodeJs / express / express-session)

Système de classement pour les utilisateurs, et système d'xp en fonction des parties (gagnées ou perdues)

Utilisation de différentes librairies



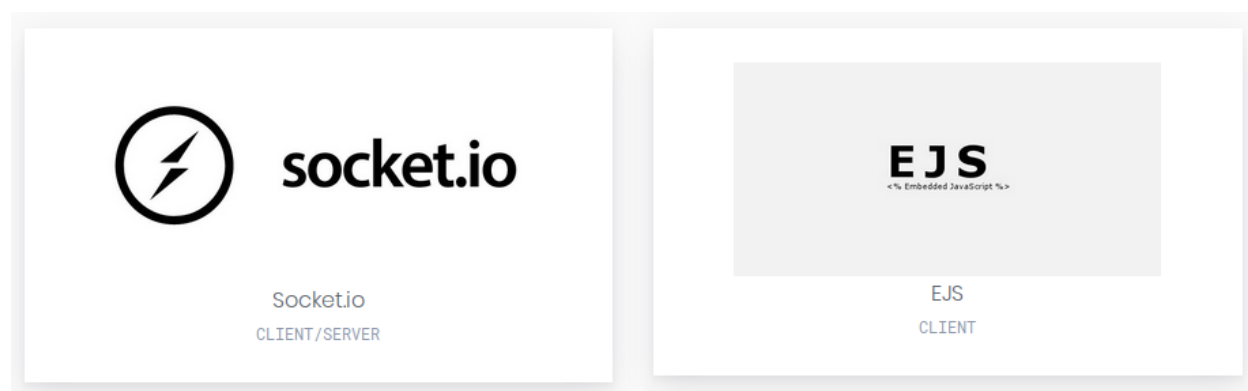
Ce projet a été conçu grâce à différents frameworks du côté clients mais aussi côté serveur.

Nous avons tout d'abord opté pour Bootstrap qui nous a permis d'aller chercher un front-end épuré et efficace grâce à un template. De nombreux choix se sont proposés à nous mais nous avons choisi de nous pencher sur un template v4 avec ses multiples fonctionnalités.


Ensuite l'incontournable Javascript nous a permis d'implémenter toutes nos fonctions côté client, mais nous a également permis l'implémentation des sockets afin de communiquer entre le serveur et les clients.

NodeJS nous a permis de mettre en place un serveur et un routage dynamique. Nous avons utilisé ainsi différents modules de NodeJS dont express (routes), express-session (session utilisateurs), body-parser (analyse les requêtes http), helmet (sécurité des headers http), crypto (chiffrement des mots de passe), et d'autres modules (fs, moment, formidable, path, mysql).

Au niveau du SGBD notre choix s'est porté sur MySQL, un outil facile d'accès que nous avons l'habitude de manipuler. Nous avons choisi une structure simple avec deux tables (USERS et LEVEL) afin de pouvoir développer une API de manière simple côté serveur nous permettant d'effectuer des requêtes SQL.



D'autres librairies nous ont aussi permis de réaliser ce projet :



Pour la gestion des sockets nous nous sommes tournés vers Socket.io qui nous a permis, grâce à ses envois de sockets client/serveur et serveur/client, de réaliser le jeu de la bataille navale avec des échanges rapides.

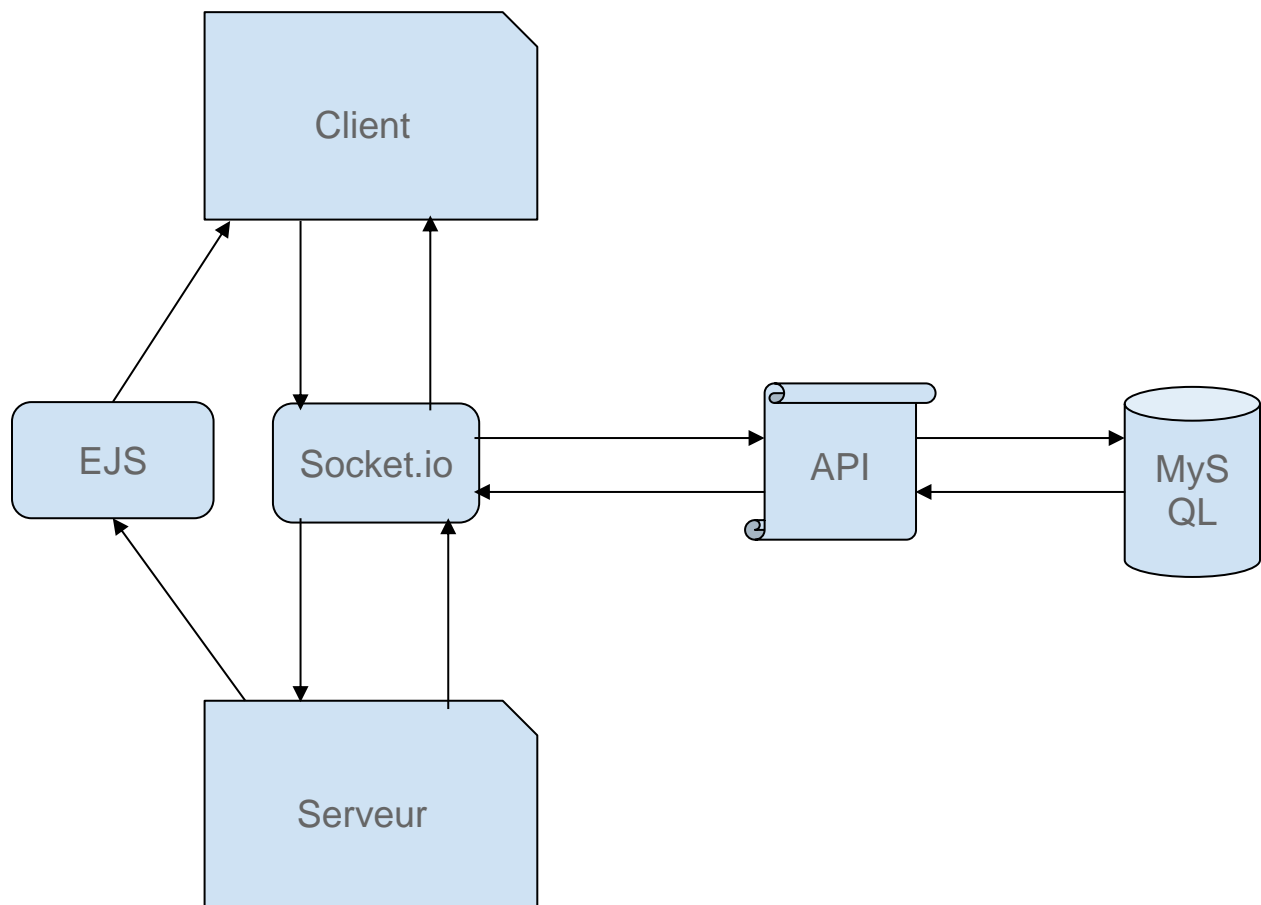
EJS est le 'view engine' que nous avons préféré à un simple HTML, ce dernier nous a permis d'envoyer des informations sur les pages en JSON mais uniquement du serveur vers le client lors des changements de routes. Nous l'utilisons pour sa facilité d'envoi d'informations.

Une architecture fondée sur une API

API Base de données MySQL

Afin de réaliser au mieux ce travail, nous avons développé un module de fonctions de base de données mySQL comparable à une API couvrant les différents besoins de Bato. Ce module permet de simplement appeler la fonction correspondant à la requête SQL du côté du serveur en lui passant les bons paramètres et le travail est joué, aucune redondance de requêtes et la simplicité est de mise.

Le schéma suivant présente le fonctionnement de communication de notre application.




Différentes pages, différentes fonctionnalités

I. Connexion/Inscription

Une page d'inscription et une page de connexion qui permettent l'accès à la plateforme. Il est possible de créer un compte à l'aide d'un nom d'utilisateur, d'un mot de passe et d'une confirmation de ce mot de passe. Le nouvel utilisateur est inséré dans la base de données avec le mot de passe encrypté. Il lui suffit ensuite de se diriger sur la page de connexion et d'y rentrer son nom d'utilisateur et son mot de passe pour se connecter.

II. Jeu solo



Nous avons implémenté un mode de jeu afin de pouvoir jouer seul contre une IA légère, ce mode de jeu est d'un niveau de difficulté très faible du fait de la simplicité actuelle de l'IA. En effet cette dernière tire de manière aléatoire sur le plateau, si elle ne touche aucun bateau elle passe donc son tour, dans le cas où elle toucherait un de vos navires elle retirerait aléatoirement sur une case adjacente à la case qu'elle vient de toucher sans aucune autre forme d'intelligence.


III. Jeu multi-joueurs

Le jeu en multijoueur est également disponible. Lorsque vous cliquez sur l'option « jouer en multijoueur » vous rejoignez temporairement une salle d'attente, dès que deux joueurs sont présents dans la salle d'attente ils sont redirigés vers une « chambre » de jeu. Dans cette chambre de jeu les deux joueurs vont alors pouvoir s'affronter. Chacun des joueurs place ses bateaux là où il le désire, attention le premier à avoir fini de placer ses bateaux gagne le droit de commencer à jouer ! Une fois les bateaux placés les derniers peuvent tirer chacun leurs tours sur le plateau de leur adversaire, un coup qui fait mouche donne le droit de retirer.

IV. Classement solo/multi

Bato possède également un système de classement. Chaque partie remportée, que ce soit en solo ou en multijoueur vous rapporteront 40 points du classement correspondants (solo ou multi), ces points qui seront directement ajoutés dans la base de données. Attention une défaite en multijoueur vous fera perdre 20 points multijoueur. Ces classements sont accessibles sur la page classement que l'on peut rejoindre après la connexion depuis la page « home ».

V. Gamification



Les points décrit précédemment permettent plus qu'une simple place dans le classement de « Bato ». En effet ces points vous permettront d'atteindre différents niveaux qui sont caractérisés par un titre, ces derniers sont débloqués à différents paliers de points :

- « Touriste » de 0 à 499 points
- « Canapiste » de 500 à 999 points
- « Jetski » de 1000 à 1499 points
- « Titanic » de 1500 à 1999
- « Moine » à partir de 2000).

Une barre d'expérience avec un pourcentage parcouru du niveau actuel est disponible dans la page « home ».

VI. Affichage des personnes connectées

Au pied de la page « home » un petit bouton vous permet de voir quels sont les joueurs actuellement connectés sur le site.

Répartition des tâches

Pour ce projet nous nous sommes réparti le travail de la manière suivante :

Anas:

- Espace de connexion et d'inscription
- Système de classement
- Gamification
- Jeu multi côté serveur

Melvin:

- Base de données
- Jeu solo avec IA
- Jeu multi côté client
- Design Bootstrap

Difficultés

Notre difficulté a été dans le peaufinage du CSS. Bootstrap à ses secrets, que les flemmards ne pourront jamais avoir. Le jeu sur mobile n'est pas encore optimisé, on pense le faire par la suite mais nous n'avons malheureusement pas eu le temps.

Nous avons eu plusieurs difficultés à mener à terme ce projet, tout d'abord sans se mentir, Bootstrap à ses secrets, que nous avons mis un moment à percer. L'affichage des différents éléments de la manière dont nous le souhaitions était particulièrement gourmand en matière de temps, notamment pour le peaufinage. Le jeu n'est d'ailleurs pas totalement optimisé pour l'affichage sur mobile (les plateaux de jeux s'affichant mal sur un téléphone qui n'est pas tourné en mode paysage). Les sockets nous ont créé plusieurs soucis également, notamment pour la création de groupe spécifique afin de diffuser des sockets dans une chambre de jeu spécifique (eh oui, c'est embêtant de signaler à toutes les chambres de jeu qu'un joueur a gagné la partie !).

Pour ce qui est de l'algorithme du jeu en lui-même, il fut assez compliqué à mettre en place au départ car nous sommes vraiment partis de 0 sans vraiment trop d'idées d'où se diriger, mais de fil en aiguille et de fonctionnalités en fonctionnalités la compréhension de ce qu'il restait à faire à chaque étape était de plus en plus limpide.

Installation

Voici les différentes étapes nécessaires au déploiement de « Bato » sur votre ordinateur :

- Tout d'abord un fichier « bato.sql » est présent à la racine du le projet, il vous suffit de créer une base de données « bato » dans MySQL puis de copier le contenu du fichier SQL afin de construire les tables de cette base de données.
- Il faut ensuite modifier vos identifiants de connexion mysql dans le fichier lib/db.js selon votre configuration.
- Il faut mettre votre propre adresse IP dans le fichier view/partial/navbar.ejs a la ligne 46.
- Vous devez ensuite ouvrir la console NodeJS, se déplacer jusqu'à la racine du projet puis effectuer la commande « *npm install* » afin d'installer les modules nécessaires à l'exécution du site.
- Il suffit ensuite de lancer le projet avec la commande « *nodemon app.js* ».