



ÉCOLE CENTRALE LYON

L^AT_EX APPROFONDI
RAPPORT

Projet Application WEB

Élèves :

AL Mahdi LAHRACHE
Oussama HADDA

Enseignant :

MULLER DANIEL
CHALON RENÉ

5 avril 2023

Table des matières

1	Media Player	2
1.1	Introduction	2
1.2	Cahier des charges	2
1.3	Javascript coté serveur	2
1.4	Implémentation de l'application	3
1.5	Pistes de développement supplémentaires	4
1.6	Conclusion	4
2	Jeu de Pong	4
2.1	Cahier des charges	4
2.2	Comment commencer le jeu	5
2.3	Architecture générale de l'application	5
2.4	Améliorations envisageables	6

1 Media Player

1.1 Introduction

Nous avons choisi de concevoir une application web multimedia pour exploiter pleinement les fonctionnalités offertes par le langage JavaScript et la plateforme Node.js, en association avec les langages HTML et CSS. Notre projet vise à permettre à l'utilisateur de profiter d'une expérience musicale interactive en ligne. En explorant les fonctionnalités de ces langages et plateformes, nous avons travaillé à la conception de l'interface utilisateur, à la gestion des fichiers audio et à la mise en place de fonctionnalités telles que la recherche de musique et l'ajout de fichiers audio à la bibliothèque musicale. Nous avons ainsi pu approfondir notre compréhension des fonctionnalités de JavaScript et de Node.js, et des capacités de développement web qu'ils offrent.

1.2 Cahier des charges

- Permettre à l'utilisateur la possibilité de choisir la musique à jouer.
- Permettre à l'utilisateur de voir si un audio est disponible en tapant le titre de l'audio sur la barre de recherche.
- Permettre à l'utilisateur la possibilité de chercher si un audio existe en tapant le titre sur une barre de recherche.
- Permettre à l'utilisateur la possibilité de télécharger la musique.
- Permettre à l'utilisateur de sélectionner ou glisser les fichiers (mp3) pour les ajouter à la playlist.
- Permettre à l'utilisateur le contrôle de la lecture (pause, lecture, avance rapide, retour rapide)
- Affichage des métadonnées du fichier audio en cours de lecture (titre, artiste, album,...)
- Permettre à l'utilisateur la possibilité d'aimer une musique tout en enregistrant les musiques aimées dans un répertoire ou historique.
- Permettre à l'utilisateur de voir son historique.

1.3 Javascript coté serveur

Nous avons utilisé principalement le framework Express de Node.js pour créer notre serveur, ainsi que plusieurs modules pour faciliter la gestion des fichiers mp3. Nous avons commencé par créer la fonction `gettags`, qui utilise `jsmediatags.read()` pour extraire les tags des fichiers mp3 tels que le nom de l'artiste, le titre, la durée et l'image. Ensuite, nous avons utilisé la fonction `Remplirdb` pour remplir un objet db avec les tags des fichiers mp3 disponibles dans le dossier `public/songs/` et pour stocker ces informations dans un fichier JSON nommé `playlist.json`.

```
// fs sera utilisé pour lire / écrire .. et manipuler des fichiers
var fs = require('fs');

// Path sera utilisé pour manipuler les chemins des fichiers manipulés
var path = require('path');

// Récupérer la durée d'un fichier .mp3
var getMP3Duration = require('get-mp3-duration') // npm install --save get-mp3-duration

// jsmediatags.read() est la fonction utilisée dans la suite, c'est pour récupérer les données ( tags ) d'un mp3 file
var jsmediatags = require('jsmediatags')
```

FIGURE 1 – Modules utilisés et utilité dans la suite

Nous avons également créé la fonction `extraire()` pour extraire les fichiers audio présents sur le serveur et stocker leurs noms dans un fichier JSON. Enfin, nous avons créé une requête GET pour afficher le contenu de la page `index.html` et avons mis le serveur à l'écoute du port 8080.

1.4 Implémentation de l'application

Le lecteur de musique est organisé de manière ergonomique pour l'utilisateur. La liste numérotée des audios disponibles est facilement accessible en bas de la barre de recherche. L'audio qui est en lecture est mis en évidence en haut de la liste, accompagné du cover de l'audio. La barre de lecture en bas permet à l'utilisateur de naviguer facilement dans l'audio en cours de lecture.

Pour la partie serveur de l'application, l'équipe a utilisé le Framework Express, qui a permis de faciliter la tâche en permettant d'envoyer toutes les requêtes nécessaires, notamment pour la création de la page web.

La première fonction, appelée "extraire path", permet de renvoyer le chemin d'un dossier ou d'un fichier audio. Ensuite, la fonction "crawl" permet de récupérer tous les chemins des fichiers audio présents sur l'ordinateur, et d'écrire ces chemins dans un fichier JSON sous forme de paires clé-valeur, avec le numéro de chemin comme clé et le chemin comme valeur.

Dans la partie HTML de l'application, une fonction appelée "get data" est utilisée pour envoyer une requête AJAX afin de récupérer le fichier JSON contenant les chemins de tous les fichiers audio. Cette fonction inclut également d'autres fonctions, telles que des événements de clic pour ajouter des fichiers audio au tableau de lecture.

En ajoutant des fonctionnalités telles que la possibilité de créer des playlists ou de partager des musiques avec des amis, l'application pourrait être encore améliorée pour offrir une expérience de musique en ligne plus complète et conviviale pour l'utilisateur.

1.5 Pistes de développement supplémentaires

Le premier point de développement supplémentaire pour un lecteur de médias pourrait être l'ajout de fonctionnalités d'interaction sociale. Par exemple, une fonctionnalité de partage sur les réseaux sociaux qui permettrait aux utilisateurs de partager leur musique préférée avec leurs amis. Cette fonctionnalité pourrait être mise en œuvre en utilisant des API de réseaux sociaux telles que Twitter ou Facebook. De plus, une autre fonctionnalité

intéressante pourrait être la possibilité de créer des listes de lecture collaboratives. Cela permettrait à un groupe d'utilisateurs de collaborer sur une liste de lecture, ajoutant ou supprimant des chansons selon leurs préférences. Les utilisateurs pourraient également voter pour les chansons qu'ils préfèrent, créant ainsi une liste de lecture dynamique et interactive. De plus, une autre piste de développement pourrait être l'ajout d'une fonc-

tionnalité de reconnaissance vocale pour le contrôle du lecteur de musique. Cette fonctionnalité permettrait aux utilisateurs de contrôler la lecture de leur musique à l'aide de commandes vocales telles que "lecture", "pause" ou "suivant". Cette fonctionnalité pourrait être mise en œuvre en utilisant des API de reconnaissance vocale telles que la SpeechRecognition API de Google.

Enfin un autre point intéressant à développer est de pouvoir développer un modèle de suggestion de musique basé sur les anciennes musiques aimées et des fréquences des musiques entendues.

1.6 Conclusion

Ce projet Web nous a permis d'appliquer les compétences que nous avons acquises au cours du semestre en électronique d'application web, y compris en HTML, CSS et JavaScript, ainsi que dans la construction de serveurs en utilisant Node.js. Nous avons mis en pratique nos connaissances en utilisant ces technologies pour créer une application Web fonctionnelle qui répond aux besoins de l'utilisateur. Ce projet nous a également permis de découvrir de nouveaux concepts et de les intégrer dans notre travail, tout en nous offrant une expérience pratique précieuse pour notre avenir professionnel.

2 Jeu de Pong

2.1 Cahier des charges

- Créer un jeu de pong.
- Permettre à deux clients différents de jouer entre eux.
- Créer une window de chat qui permet aux joueurs de communiquer entre eux.
- Distinguer les messages envoyés par chaque joueur.
- Permettre à l'utilisateur de rentrer son nom qui sera ensuite utilisé dans le chat.
- Ajouter un son à chaque fois il y a une collision entre le et l'une des extrémités ou l'un des pagaies.
- Interdire aux pagaies de dépasser la fenêtre du jeu.
- Afficher le score qui doit changer à chaque fois l'un des joueurs marque un but.

- Afficher une fenêtre pop-up qui indique "Vous avez gagné" au gagnant et "Vous avez perdu" au perdant.
- Jouer un son à la fin du jeu qui dit : "Game over".

2.2 Comment commencer le jeu

Pour lancer le jeu sur votre pc, il faut télécharger tous les fichiers et les mettre dans le même dossier. Ensuite, vous lancez le serveur en exécutant le fichier "server.js". Le serveur est sur le port 3000, donc vous allez sur : **localhost:3000**.

Il y a deux joueurs donc vous ouvrez deux onglets sur cette même adresse.

AJOUTER FIGURE : ECRAN DIVISE EN DEUX

Durant le jeu, on joue un son à chaque fois le ballon rentre en collision avec l'un des pagaies ou avec une extrémité de la fenêtre. Afin d'avoir une meilleure expérience, vous devez vous connecter avec des comptes différents sur chacun des onglets (ou être *invité* sur l'un des onglets et *connecté* sur l'autre), ensuite vous coupez le son de l'un des deux. Cela permet de ne pas entendre le même son deux fois.

Le jeu ne commence que quand **deux joueurs** sont connectés et que l'un des joueurs a appuyé sur **espace**. Le premier joueur qui se connecte prendra la pagaie à gauche et jouera avec les flèches *gauche* et *droite*. Le deuxième joueur prendra la pagaie à droite et jouera avec la souris. Ainsi lors de la division de votre écran en deux, il est conseillé que vous vous connectiez avec l'onglet à gauche premièrement. Le chat fonctionne indépendamment du jeu. Au début, on affiche sur le formulaire : "Entrez votre nom". Dès que les deux joueurs ont rentré leurs noms, on commence à afficher : "Tapez votre message". Les noms entrés seront ensuite affichés dans le chat quand on envoie un message.

Quand l'un des deux joueurs se déconnecte, le jeu s'arrête immédiatement, les scores sont remis à 0 et une pop-up s'affiche au joueur restant indiquant : "L'autre joueur a quitté".

Le jeu se termine quand le score de l'un des joueurs atteint 5. Dans ce cas on affiche une pop-up au gagnant qui dit : "Vous avez gagné" et une autre au perdant qui dit : "Vous avez perdu". En même temps, un son disant : "Game over" est joué.

2.3 Architecture générale de l'application

Nous avons un fichier javascript qui gère le serveur, un fichier html pour créer la page, un fichier css pour le style, un fichier javascript qui gère le jeu et un fichier javascript qui gère le chat.

Pour créer le serveur, nous utilisons *node.js* avec le framework *express*. Pour gérer les requêtes du jeu et du chat nous utilisons la librairie *socket.io*.

Le principe du jeu consiste à modifier la position verticale de chacun des pagaies après chaque action de l'un des joueurs (appui sur une touche ou mouvement de la souris). Cette position est ensuite envoyée au serveur qui la broadcast vers l'autre joueur. La position du ballon change constamment en augmentant son abscisse de dx et son ordonnée de dy . La fonction *move_ball()* qui change la position du ballon détecte aussi les collisions entre le ballon et les extrémités de la fenêtre du jeu ou l'un des pagaies.

La fonction *draw()* retrace tous les éléments du jeu après chaque milliseconde avec

leurs nouvelles positions.

Dès la connection d'un client, on prend son ID en utilisant *socket.io* et on le stocke dans une variable constante. Le serveur envoie aux deux clients une liste contenant les IDs des deux joueurs connectés. Cette liste sera ensuite utilisée dans le chat pour comparer l'ID du client à l'ID de celui qui a envoyé le message ce qui modifiera l'affichage dans le chat ("Vous"+message si on a envoyé le message, "Nom de l'adversaire"+message dans le cas contraire). Elle sera également utilisée pour savoir si ce client a gagné ou perdu pour ensuite lui afficher son statut avec un message dans une fenêtre pop-up.

2.4 Améliorations envisageables

Afin d'améliorer le jeu de pong, nous envisageons ajouter l'option choix de salle publique ou salle privée ce qui permettra aux joueurs de choisir leur adversaire. Une autre amélioration possible pourrait être de gérer plusieurs clients en même temps au lieu que seulement deux.

Pour améliorer le chat, on pourrait afficher les messages envoyés à droite et les messages reçus à gauche et colorer chacun des deux avec une couleur différente.