

Adopte un zikos !

Massinissan Ait-Ali

Grégoire Harba

Thomas Lesciellour

Ce qui nous a amené à réaliser ce projet (interview, brainstorming ...)

Depuis de nombreuses années, la musique a pris de plus en plus de place dans la vie de tous les jours. Il existe de nombreux services pour partager la musique, comme youtube qui a l'avantage d'offrir aussi un service de vidéo et de direct.

Il y a aussi des services plus répandus et moins intrusifs que YouTube tel que Spotify.

Avec la démocratisation de la musique et des applications pour y avoir accès, l'apprentissage des instruments a grandement augmenté. Il y a aussi beaucoup de gens qui apprennent à chanter très tôt et poursuivent leur passions longtemps.

Il existe aussi quelques jeux tel que guitar Hero ou Rocksmith pour apprendre à jouer d'un instrument ou de le simuler d'une façon plus arcade.

Lorsque le sujet libre nous a été proposé, nous ne savions pas du tout sur quel terrain partir. Dans le groupe, nous apprécions tous la musique. Nous voulions donc faire quelque chose qui avait rapport avec cette dernière. Cependant, il est dur de trouver une interaction autre que de brancher une guitar ou un micro à un PC et de recréer l'existant d'un point de vue applicatif, et même d'un point de vue interaction.

Puis nous avons pensé à une application qui viendrait s'interfacer avec plusieurs entrées. Dans un premier temps, nous avons souhaité une application qui utilise une guitar et qui permet à un guitariste de donner un concert en ligne. Le but n'étant pas de voir la performance du guitariste, mais de pouvoir avoir une interaction entre le public et le musicien.

Puis nous est venue l'idée de ne pas se limiter à simplement la guitar, mais d'aller plus loin en permettant à la star de pouvoir utiliser un micro et une webcam. Après avoir mis en place l'ensemble de ces idées, nous nous sommes rendu compte que d'avoir une entrée dans l'application qui était une guitar n'était pas le plus pratique. Nous avons donc décidé d'utiliser le clavier en tant que guitar pour simplifier la mise en place du projet et le rendre plus accessible aux utilisateurs.

Le projet final sur lequel nous nous sommes donc arrêté est une application web qui permet de mettre en relation un artiste avec des spectateurs et avoir des interactions entre les deux parties.

- Description de l'interaction 3D choisie et de son fonctionnement (storyboard résumé)

Lorsqu'un utilisateur lance le site, il fait face à un choix crucial. Il peut soit devenir une star soit un membre du public.

La suite des interactions est donc dépendante du rôle. Commençons par les interactions du public.

Avant toute chose, il est important de noter que le spectateur interagit avec la star à l'aide de son téléphone portable.

1. Il est possible de d'applaudir pour encourager la star. Cette action est prise en compte avec l'accéléromètre du téléphone.
2. Il est possible d'envoyer des projectiles sur la star, avec un swipe vers une cible.

La star de son côté a un panel d'interactions plus large :

1. Avec sa webcam, il dispose d'une reconnaissance faciale pour mimer certaines animations faciales, tel que tirer la langue à la façon du bassiste de KISS, Gene Simons. Ou encore headbanger
2. Il peut jouer de la guitar avec son clavier, ce qui permettra de jouer pour le public. De plus, s'il joue un solo endiablé, la scène prendra feu.
3. La voix du chanteur peut aussi être transmise par l'application.

- Moyens techniques mis en place et répartition du travail

Pour les moyens techniques mis en place nous avons choisi de partir sur une intégration web afin de faciliter l'accès à l'application. Pour la partie serveur nous avons choisi d'utiliser Node.JS avec Express.js et Socket.io afin de garantir une certaine instantanéité dans les échanges.

Pour l'intégration de la partie processing dans le rendu de la scène nous sommes parti sur la bibliothèque javascript P5.

Nous souhaitons bénéficier des dernières technologies dans nos développement nous avons donc mis en place un système de compilation de nos styles et de nos sources javascripts. Nous avons donc pu profiter d'un développement plus simple et structuré avec le SCSS et des dernières avancées d'ECMAScript pour la partie JS.

Enfin pour respecter les plus possible les règles du DevOps, nous avons mis en place un système très simple de gestion du projet en profitant de NPM.

Ainsi une fois le projet récupéré, il suffit de taper la commande `npm install` pour installer les dépendances du projet. Ensuite il faut compiler les sources du projet avec la commande `npm run build`. Et pour lancer le serveur il ne manque plus que la commande `npm run start`. Cette dernière commande va alors lancer le serveur web par défaut sur le port 8000.

Une fois les sources du projet compilées et le serveur lancé, on peut accéder à l'application via l'adresse : <https://localhost:8000/index.html>

- Retour utilisateurs et ouvertures sur les améliorations possibles