



Application Web

Rapport final

11 avril 2020

CLERC Billy - de JAHAM Charles

1 Sujet

Le sujet sur lequel nous avons choisit de mener notre projet est celui du *Pictionary en ligne*. Voici la description du sujet :

“Les joueurs peuvent créer un compte et se connecter. Une fois une partie créée (identifiée par un lien partageable avec d’autres joueurs), un joueur est sélectionné pour dessiner un mot choisi aléatoirement, les autres peuvent proposer des mots dans un ”chat” jusqu’à ce qu’un joueur devine le mot ou que le temps imparti s’écoule.”

2 L’application

L’application est hébergée sur *Heroku*, l’interface utilisateur est faite par l’intermédiaire de *React* et la partie serveur est réalisée par le biais de *Node.js/express*.

L’application elle-même ne comporte que trois pages. La page d’accueil, la page de jeu et la page d’erreur. Ces pages sont contenues dans le dossier *views*, lui-même contenu dans le dossier *src*.

La redirection vers ces pages est réalisée grâce au *Router* de *React* à l’intérieur du fichier *App.js*. N’importe quelle page non existante entraîne également une redirection vers la page d’erreur.

Les pages suivent toutes le même modèle avec une barre de navigation, un en-tête, un corps et un bas de page. Dans le bas de page, on retrouve notamment le remerciement à *W3Schools*, le site web d’où provient le code HTML et CSS du modèle original, qui a été modifié par la suite.

La partie la plus intéressante, le cœur même du projet, est contenu à l’intérieur de la page *Play*.

3 L'implémentation du pictionary

3.1 L'interface

Nous avons essayé de créer une interface simpliste mais agréable. En ce qui concerne la partie programmation, le code javascript utilisé à travers cette page est contenu dans le fichier *drawAndChat.js* (*public/scripts*). Cette page communique avec le serveur (*server.js*) grâce à l'utilisation de Socket.io. C'est à travers un échange de message entre ces deux éléments que la communication est établie et que les fonctionnalités sont mises en place.

3.2 Le nom d'utilisateur

Avant de pouvoir accéder au jeu, l'application demande un nom d'utilisateur. Ce dernier ne peut ni être null, ni composé uniquement d'espaces. Les espaces inutiles avant et après le nom d'utilisateur sont également automatiquement supprimés.

De plus, nous avons utilisé une *async function* ainsi qu'une *promise* – de la manière dont nous l'avons vu en cours – afin de pouvoir demander un nom d'utilisateur de manière asynchrone et tout de même l'utiliser à travers le code javascript.

3.3 La section des joueurs connectés

Tout d'abord, on aperçoit sur la gauche la liste des joueurs connectés. Sur cette dernière figure la liste des joueurs dans leur ordre de passage en tant que dessinateur. Le premier joueur dans cette liste sera le prochain joueur à dessiner un mot que les autres devront trouver.

De plus, sur cette liste figure le score qui est incrémenté d'un point par mot trouvé.

3.4 L'hôte

L'hôte, ou en tout cas le joueur qui décidera du lancement de la partie, est le premier joueur de la liste des joueurs connectés. Ce dernier verra un bouton au-dessus de la toile du dessinateur permettant de lancer la partie, alors que les autres joueurs verront le nom de l'hôte.

Une partie ne peut être lancée qu'à partir du moment où au moins deux joueurs sont connectés. S'il ne reste qu'un joueur ou moins à un moment donné de la partie, cette dernière s'arrêtera et le bouton pour lancer une partie sera de nouveau affiché (le score des joueurs encore connectés sera conservé).

3.5 Le salon de discussion

Par la suite, on observe du côté droit le salon de discussion. Sur ce dernier, les joueurs peuvent communiquer entre eux. Lorsqu'un joueur envoie un message, le serveur retransmet le message à tous les clients connectés.

Le serveur vérifie également si le message ne correspond pas au mot qui doit être trouvé. Si c'est le cas et que l'expéditeur n'est pas le dessinateur, un point est attribué au joueur ayant trouvé le mot et une nouvelle manche débute.

Le mot à deviner ainsi que le message envoyé sont tous deux comparés après avoir enlevé les majuscules et accents, ce qui permet au joueur ayant envoyé "pretre" d'avoir une réponse correct si le mot à deviner est "Prêtre".

Il est impossible d'envoyer des messages contenant uniquement des espaces dans le salon de discussion.

3.6 La toile du dessinateur

Enfin, au milieu de l'interface se trouve la zone de dessin à travers laquelle le dessinateur pourra exercer son art et les autres joueurs pourront observer la création d'une merveille artistique.

Dans un premier temps, le dessinateur doit choisir un mot parmi trois proposés. Actuellement, ces mots sont récupérés aléatoirement depuis le fichier *dictionary.txt*, lui-même contenu dans le dossier *public/listes*. Cette liste a été récupérée depuis le site http://www.normalesup.org/~fradurand/liste_mots/. La liste comprend en première ligne le nombre de mots dans la liste puis un mot par ligne.

Lorsque le dessinateur choisit son mot, un compte à rebours de 60 secondes débute et la boîte à outil du peintre est affichée, permettant de changer de couleur et de taille de trait. Si le mot n'est pas trouvé avant la fin du compte à rebours, une nouvelle manche débutera et personne n'aura eu le point.

Comme pour le salon de discussion, le serveur se charge de transmettre les coups de crayons en temps réel.

3.7 Déconnexion

Si un joueur se déconnecte, il perd son score et les autres joueurs le verront disparaître de la partie.

Si ce joueur était en train de dessiner, un nouveau dessinateur sera désigné.

4 Améliorations possibles

L'application que nous avons créée pourrait bénéficier de multiples améliorations. Voici une liste non exhaustive des implémentations ou changements supplémentaires que nous aurions aimé pouvoir faire :

- Améliorer l'interface.
- Faire en sorte que le nom d'utilisateur soit demandé d'une autre manière. Les prompt à l'écran ne représente pas la méthode optimale pour récupérer une information. De plus, il est possible de désactiver les pop-ups venant de l'application et ainsi ne plus pouvoir entrer de nom d'utilisateur.
- Instaurer une base de donnée.
Puisque les mouvements du dessinateur sont transmis en temps réel, il est impossible pour un nouveau joueur de voir ce qui a été dessiné avant son arrivée. Avec une base de donnée (ou même une simple liste retraçant les derniers coups de crayons), nous aurions pu permettre ce genre de chose. Nous aurions également pu créer un système de compte, bien que cela aurait également demandé d'instaurer un minimum de sécurité.
- Système de lien partageable.
Comme décrit dans le sujet, nous aurions pu mettre en place un système de lien partageable pour éviter tous les problèmes liés au fait de n'avoir qu'une seule zone de jeu pour la totalité de l'application.

5 Conclusion

Bien qu'il nous reste une semaine avant le rendu final, nous avons décidé de laisser le projet à l'état actuel. Nous sommes très satisfaits de l'application que nous avons fournie, même si de multiples améliorations sont possibles. Malheureusement, nous devons dès à présent travailler sur notre projet de fin de licence, ce dernier étant de grande importance.

Ce fut notre première application web, nous n'avions jamais codé en javascript par le passé, et nous sommes très content de ce que nous avons réussi à produire. Le lancement était difficile puisqu'il fallait apprendre de nouvelles notions mais nous sommes maintenant familier avec un aspect supplémentaire du domaine de l'informatique et nous en sommes ravis.

6 URL de notre dépôt GitHub

<https://github.com/Bart-soldier/Pictionary>

7 URL de notre site web

<https://clerc-dejaham-pictionary.herokuapp.com/>