

Projet - Application frontale

Sur les 2 prochaines séances de TP, vous allez travailler sur le projet, binôme ou trinôme. Vous allez développer une application Vue.js d'un jeu pour deviner un mot de 5 lettres, ceci en utilisant l'ensemble des connaissances acquises durant les précédents TPs.

Note : pour rappel, ce projet comptera pour 30% de la note du module application frontale.

Démarrage du projet avec Git

- Utilisez le lien fourni sur Moodle pour générer un dépôt privé correspondant au TP, hébergé dans l'organisation <https://github.com/UE-FRONTEND> ;
- Clonez le dépôt GitHub ainsi obtenu ;
- Ajoutez les noms et prénoms des membres de votre projet ainsi que le groupe de TP dans le fichier *README.md* ;
- Créez un commit dans *master* pour ce changement et poussez le vers le dépôt distant.
- Prenez connaissance de la structure du projet avec IntelliJ ;
- Faites la commande `npm install` à la racine du dossier
- Lancez votre application à l'aide de la commande `npm run serve`

Note : il y a du `live reload`, ce qui signifie que vous n'avez pas à redémarrer votre application à chaque modification : vous les verrez directement apparaître dans votre navigateur.

Tâches utilisateur

Le but principal de l'utilisateur est de trouver un mot de 5 lettres.

Les composants graphiques de l'application ainsi que le comportement de l'interaction doivent permettre à l'utilisateur d'effectuer les tâches décrites dans le modèle de tâches "Trouver le mot". Pour faciliter une lecture en deux temps du modèle de tâches, vous trouverez, joints au sujet, deux versions du modèle de tâche "Trouver le mot":

- une version sans la représentation des données, informations, et périphériques
- une version avec les données, informations et périphériques

Documentation technique

Ce jeu utilisera un web service, exposé sous forme d'API Rest. Ce service est joignable à l'url suivante : <https://vue-project-backend-eta.vercel.app>

Vous pourrez trouver le code source du serveur [ici](#).

Voici une description de l'API exposée par l'API Rest au format JSON :

Vous avez 2 URI possibles :

- **/api/new-game (Méthode GET) :**

Permet de récupérer un **mot** à faire deviner.

Exemple : `{ "word": "jante" }`

- **/api/check-word (Méthode POST) :**

Permet de vérifier si le mot saisi par l'utilisateur est bien un mot du vocabulaire français.

- Body de la requête (format JSON) :

- **word** : le mot vérifier

Exemple : `{ "word" : "jante" }`

- Body de la réponse (format JSON) :

- **isWord**: **true** si le mot est bien un mot du vocabulaire français, **false** sinon.

Exemple: `{ "isWord": true }`

Pour faire jouer l'utilisateur, il faudra donc appeler **/api/new-game** en 1er pour récupérer un mot à faire deviner. Ensuite, pour chaque tentative de l'utilisateur, il faudra faire un appel à **/api/check-word** en renseignant le mot saisi par l'utilisateur. Si le mot saisi par l'utilisateur existe en français, alors il faut vérifier si les lettres saisies par l'utilisateur sont à la bonne place.

Exigences fonctionnelles (en plus du modèle de tâches)

Lisez le modèle de tâches avant de lire les exigences fonctionnelles ci-dessous.

- 1) Lorsqu'on arrive sur la page principale (path = /), vous devez présenter une page indiquant :
 - a) Vos noms et prénoms + groupe de TP
 - b) Un bouton démarrer une partie
 - c) Un bouton consulter les précédentes parties
- 2) Lorsque l'utilisateur clique sur le bouton valider son mot, alors faire une requête à l'API Rest pour savoir si le mot est valide.
 - a) Si le mot est valide, alors vérifier les lettres valides
 - b) Si le mot n'est pas valide, afficher un message d'erreur à l'utilisateur
- 3) Lorsque l'utilisateur veut consulter les statistiques, affichez une page récapitulant les différentes parties jouées. Cette page doit avoir une partie statistiques, avec :
 - a) Le temps moyen de jeu
 - b) Le nombre de tentatives moyennes
 - c) Le % de victoire
 - d) L'ensemble des parties jouées avec
 - i) La date de la partie
 - ii) Le nombre de tentatives
 - iii) Le temps de jeu
 - iv) Un indicateur pour savoir si l'utilisateur a réussi ou non
 - v) Le mot à deviner

Exigences techniques

L'application doit être développée en Vue.js.

Vous devez utiliser les dépendances suivantes :

- vue-router
- vuex
- axios

Votre application doit pouvoir se lancer à l'aide de la commande **npm run serve**. Votre application doit être du type SPA (single page app), c'est à dire que vous ne devez pas changer de page : vous devez donc utiliser vue-router. Vous devrez utiliser le web service mis à votre disposition pour faire jouer l'utilisateur.



Afin de pouvoir restituer les parties jouées (exigences 5 et 6), vous devez utiliser **vuex** : vous stockerez toutes les parties jouées ainsi que toutes les informations nécessaires à l’affichage.

Vous devez tester votre application avec Cypress. Voici la liste des tests attendus :

- 1) Vérifier que l’application se charge bien
- 2) Vérifier que lorsque vous cliquez sur le bouton “lancer une partie”, l’affichage propose bien à l’utilisateur de saisir un mot
- 3) Vérifiez que lorsque vous saisissez “zzzzz”, vous avez le message d’erreur “mot invalide” qui apparaît
- 4) Vérifiez que lorsque vous saisissez le mot ”jante”, celui-ci est bien accepté.
- 5) Vérifiez que lorsque vous appuyez sur le bouton “abandonner la partie”, vous arrivez bien sur une page de défaite
- 6) Lorsque vous accédez aux statistiques, vérifiez bien que vous avez la page des statistiques qui s’affiche (inutile que votre test valide que vos statistiques sont bonnes)

Vous pouvez vous servir des issues Github pour gérer votre projet et suivre un minimum les tâches que vous devez faire.

Aides

Pour la page de jeu: dès que l’écran est chargé (état **mounted** d’un composant), faites une requête pour récupérer un mot de jeu et lancez le chronomètre.

Pour le chronomètre : utilisez un ordonnanceur avec la méthode **setInterval** avec un pas de 1 seconde, pensez à bien arrêter cet ordonnanceur lorsque l’écran n’est plus affiché à l’utilisateur.

Travail collaboratif : plus vous ferez de composants, plus vous pourrez paralléliser vos développements et plus vous faciliterez la collaboration tout en minimisant les conflits Git.



Rendu du projet

Le projet est à rendre pour le 30/12/2022.

Pour rendre votre projet, vous devrez :

- 1) Compléter le README avec l'ensemble des informations nécessaires
 - a) Le nom et prénom de chaque participant
 - b) Le groupe de TP
- 2) Commiter et pusher votre code sur Github
Attention à bien embarquer les bons fichiers dans votre commit. Vérifiez bien que votre code est présent sur Github.
- 3) Créer un tag (= release sur Github) nommé 1.0.0

Ce tag devra être posé sur Github avant la date de rendu de projet. Tout tag posé après cette date limite entraînera un malus de 1 point / jour de retard.