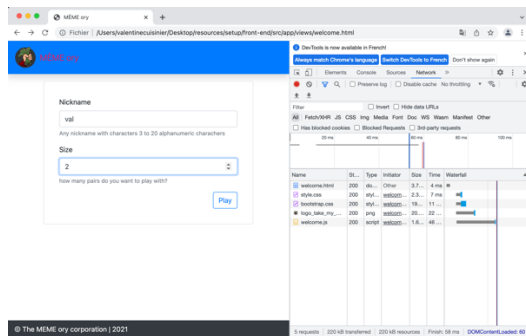


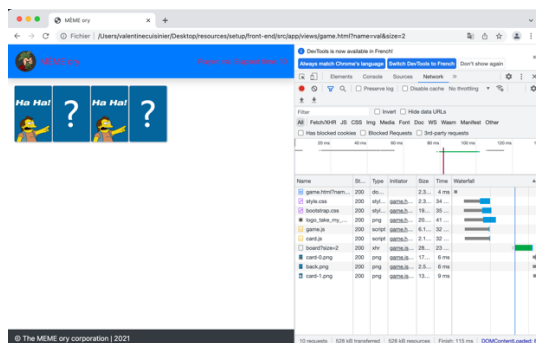
Compte Rendu TP Application Web Design

1. **Play the whole game with size=2. By browsing the 3 views of the application, how many files did your browser download overall? How many time did it took to load them all?**



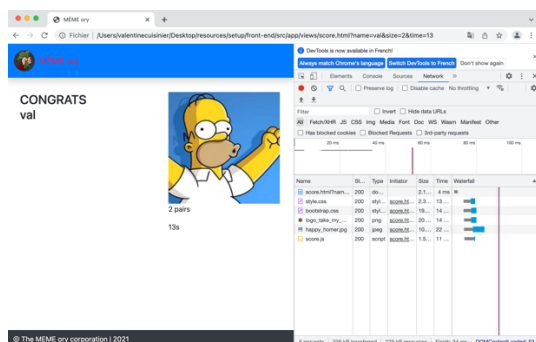
Welcome View

Chargement de 5 fichiers en 60ms



Game View

Chargement de 10 fichiers en 82 ms



Score View

Chargement de 6 fichiers en 52 ms

2. **Component-oriented programming for the web is considered more maintainable. Why?**

La programmation orientée composant (POC) est considérée comme plus facile à maintenir. En effet, si le développeur souhaite effectuer une modification sur un composant, il n'a besoin que d'une modification qui sera apportée à toutes les pages où ce composant est présent. Ce type de programmation permet une architecture plus claire et organisée du code, ce qui permet également un gain de temps.

3. **If you look at the source code, every JS file wraps its code into a closure:**

Lorsque l'on supprime les "closures", les deux variables environnement sont en conflit.

Nous avons donc deux possibilités pour que le jeu refonctionne :

- Enlever l'une des deux variables,
- Remettre les « closures ».

4. *As you can see, npm install command also generated a package-lock.json file along with package.json. What is the purpose of this file?*

Ce lock-file est mise à jour lorsque le dossier node_modules ou le fichier package.json est modifié par npm. Il permet de garantir que les mêmes dépendances sont installées mais aussi de bloquer toute mise à jour non voulu.

5. *By convention, all NPM dependencies use a 3-digit format for version numbers. How do you call this?*

- Le premier chiffre correspond au numéro de version majeure,
- Le deuxième chiffre correspond au numéro de la version mineure,
- Le troisième chiffre correspond au numéro des versions avec seulement des corrections de bugs.

Le symbole ^ permet d'indiquer que l'on souhaite effectuer des mises à jour qui ne modifie pas le nombre le plus à gauche (différent de 0).

6. *What is a devDependency exactly? What are the differences with a dependency?*

Une dépendance est nécessaire pour compiler et faire fonctionner le projet, que ce soit en phase de développement ou en phase de production. Tandis qu'une devDependency est une dépendance nécessaire seulement en phase de développement, elle n'est plus requise pour la production.

7. *Can you think of at least 2 things that are possible with Java classes, but cannot be done with ES6 classes?*

En JavaScript, il est possible d'effectuer des fonctionnalités telles que les génériques et les annotations de type, mais aussi les interfaces et les enums. Ces fonctionnalités précédentes ne sont pas prises en charge par ES6.

8. *What are the differences between var and let?*

En JavaScript, il y a 3 façons de déclarer des variables : var, let et const.

Les variables avec const ne peuvent pas être réaffectées contrairement à let et var.

La différence entre let et var va être la portée que la variable aura.

Avec let la variable sera locale, cela veut dire qu'elle sera accessible uniquement dans le bloc où elle a été déclarée.

Avec var, il y a deux possibilités :

- Si on la déclare dans une fonction, on ne pourra l'utiliser que dans cette fonction.
- Si on la déclare en dehors d'une fonction, on pourra l'utiliser dans tout le script.

Remarque : Il est possible de rédefinir une variable plusieurs fois avec var et let, (ce qui écrasera la valeur précédente) la seule différence réside dans le fait qu'une variable let ne peut pas être déclaré plusieurs fois.

9. What is the .bind(this) stuff? What does happen if you delete it? Is it needed when using an arrow function?

La méthode .bind(this) permet de créer une nouvelle fonction qui va lier un objet spécifique à une fonction originale. On pourra ensuite faire référence à cet objet attaché en utilisant cette nouvelle fonction. Cette nouvelle fonction est une copie de la fonction originale avec comme paramètre le this.

Si l'on supprime la méthode, le this ne retournera pas l'objet souhaité : il y aura donc une erreur.

Cette méthode n'est pas nécessaire dans une arrow function puisqu'elle lie automatiquement le contexte du code environnant.

10. What are the advantages of Promises?

Les avantages des promesses :

- Possibilité de chaîner les opérations asynchrones,
- Simplicité et cohérence de lecture,
- Garantie que les opérations se déroulent dans l'ordre voulu,
- Meilleure gestion des erreurs.

11. What version of ECMAScript async / await was released in?

Les mots-clés async et await ont été publiés dans la version ECMAScript 2017.

12. What does the @ symbol mean in @babel/?***

Dans @babel/**, le symbole @ permet d'identifier l'auteur du fichier.

Ce qui permet d'avoir plusieurs bibliothèques avec le même nom, mais identifiées par un nom d'auteur différent.

13. Look at the files produced within dist/ folder. How babel transpile your class WelcomeComponent?

Dans le dossier dist/ nous ne trouvons que des fichiers JavaScript, ce qui paraît cohérent puisque ce sont eux qui vont être transpilés pour avoir du ES5.

Afin de transpiler la classe WelcomeComponent, Babel importe des fonctions ES5 équivalentes à celles utilisées en ES6 grâce au mot clé require().

14. What is the weight of the transpiled sources compared to your original sources?



Les sources transpilées pèsent 16 ko. Tandis que les sources originales pèsent 2,1 Mo. Cette large différence est due au fait que pour réaliser une même tâche, le code sous version ES5 nécessite plus de lignes que sous version ES6.

15. What is the difference between `import * from './utils'` and `import { parseUrl } from './utils'`?

La différence est que l'astérisque `*` désigne tous les éléments du fichier utils (composants, fonctions, classes), alors que `{}` permet de sélectionner uniquement un élément (ici la classe `parseUrl()`).

16. Why the `utils.js` will also be transpiled?

Tous les fichiers JavaScript nécessaires au fonctionnement de l'application vont être transpilés, c'est-à-dire ceux présents dans le dossier `src`.

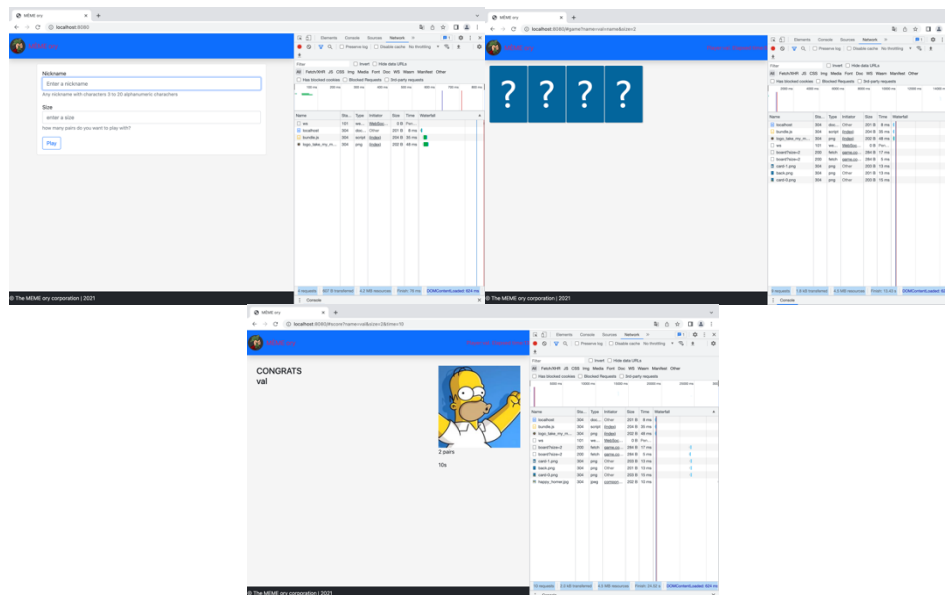
De plus, nous pouvons remarquer que ce fichier utilise des fonctions qui n'existent que sous ES6.

17. What does the `webpack --config webpack.config.js` do?

Cette commande permet de configurer Webpack.

Webpack c'est un bundler : il permet d'utiliser un serveur local, mais aussi permet de compiler tous nos fichiers pour les regrouper en un seul, ce qui apportera de la performance à notre application.

18. Play the whole game with `size=2`. By browsing the 3 views of the application, how many files did your browser download in total? How many time did it took to load them all?



Notre application est devenue une SPA, il n'y a donc plus de rechargement de page lorsque l'on navigue sur toute l'application.
Ce seul chargement a duré 754 ms et a exécuté 9 fichiers (+2 tableaux).

19. Can you guess how exactly style-loader works exactly?

Style-loader s'ajoute à la configuration webpack. Il va permettre d'injecter du CSS dans le DOM.

20. What does the `_` prefix means on a sass file?

Le préfixe underscore `_` est suivi du nom d'un fichier scss.
Il permet d'indiquer que ce fichier correspond à une partie du css.
Ce procédé permet d'organiser son code css en plusieurs fichiers logiques.
Ces différents fichiers logiques seront ensuite compilés dans un seul fichier, la feuille de style principale style.scss.