






# Rapport de TP : APP DESIGN






Aurélien Desormiere – Olivier Marcenac

1. While going through the 3 views of the application, how many files did your browser download in total? What was the total size of transferred data?






Pour la 1ere page on a 5 fichiers :

Name	Status	Type	Initiator	Size	Time
 welcome.html	200	document	Other	4.2 kB	318 ms
 style.css	200	stylesheet	welcome.html	2.6 kB	13 ms
 bootstrap.css	200	stylesheet	welcome.html	193 kB	19 ms
 welcome.js	200	script	welcome.html	1.8 kB	46 ms
 logo_take_my_money.png	200	png	welcome.html	20.7 kB	52 ms
5 requests   222 kB transferred   220 kB resources   Finish: 401 ms   DOMContentLoaded: 400 ms   Load: 402 ms					

Pour la 2eme page on a :

Name	Status	Type	Initiator	Size	Time
 game.html?name=orel&size=2	200	document		2.5 kB	321 ms
 style.css	200	stylesheet	game.html?name=orel...	2.6 kB	12 ms
 bootstrap.css	200	stylesheet	game.html?name=orel...	193 kB	18 ms
 game.js	200	script	game.html?name=orel...	6.6 kB	45 ms
 card.js	200	script	game.html?name=orel...	2.8 kB	50 ms
10 requests   532 kB transferred   528 kB resources   Finish: 514 ms   DOMContentLoaded: 412 ms   Load: 415 ms					

Et pour la dernière page on a :

Name	Status	Type	Initiator	Size	Time
 logo_take_my_money.png	200	png	score.html?name=orel...	20.7 kB	36 ms
 happy_homer.jpg	200	jpeg	score.html?name=orel...	11.0 kB	70 ms
 score.js	200	script	score.html?name=orel...	1.7 kB	29 ms
 style.css	200	stylesheet	score.html?name=orel...	2.6 kB	12 ms
 bootstrap.css	200	stylesheet	score.html?name=orel...	193 kB	19 ms
6 requests   231 kB transferred   229 kB resources   Finish: 110 ms   DOMContentLoaded: 74 ms   Load: 112 ms					

Ainsi on a au total un peu moins d'1 MB de données transférées pour une quinzaine de fichiers téléchargés.

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

La programmation orientée composant, ou POC est basée sur le fait d'empiler ou d'emboîter des composants entre eux pour former le programme. La maintenance de la POC est plus facile puisqu'on peut mettre à jour les composants séparément sans mettre à jour tous le programme.

**3. If you look at the source code, every JS file encapsulates its code in a closure Try to remove the 2 closures from both card.component.js & game.component.js. What happens? Why?**

**4. By convention, all NPM dependencies use the same 3-digit style version numbers? How is it called?**

Cette convention d'écriture s'appelle le semantic versioning, ou versionnage sémantique en français. Son principe est le suivant :

1. Le premier chiffre est attribué aux modifications majeures
2. Le deuxième chiffre aux modifications mineures
3. Le troisième chiffre pour les corrections ou patches

**5. What means the "^" symbol next to the bootstrap version?**

Le symbole « ^ » nommé « caret » à côté de la version de bootstrap signifie que la version de bootstrap sera celle dont le numéro de version est affichée ou toute version plus récente donc le premier chiffre d'a pas été changé, c'est-à-dire sans modification majeure.

**6. 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 ?**

Le fichier package-lock.json est généré à chaque opération où npm modifie l'arborescence node\_modules ou package.json.

Ce fichier décrit l'arborescence générée afin que des installations ultérieures puissent générer une arborescence identique.

**7. What is a `devDependency` ? What are the differences with a "Dependency"?**

devDependency est un module nécessaire au développement du code. La différence avec un Dependency est que la devDependency n'est pas nécessaire à l'exécution du code contrairement à la Dependency.

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

Les classe ES6 ne peuvent pas réaliser les actions suivantes contrairement aux classes Java :

- L'encapsulation
- La présence d'interfaces
- L'utilisation de types

**9. What are the differences between "var" and "let"?**

La différence entre « var » et « let » se fait au niveau de la portée des variables définies : pour « var » la portée est celle de la fonction alors que pour « let » la portée est le bloc courant.

**10. What is the `.bind(this)` stuff? What does happens if you remove it? Use your web browser's debugger to guess what's going on.**

« `bind(this)` » est une méthode qui permet lier le « `this` » auquel la méthode fait référence à une fonction. Dans notre cas si on supprime « `bind(this)` » on ne pourrait pas afficher le nom du joueur car le `this` serait « `undefined` » et la fonction n'aurait alors pas de paramètre.

**11. The shorten syntax aside, what is the difference between**

```
setTimeout(function() { console.log(this._name); }, 750);
```

**And**

```
setTimeout(() => console.log(this._name), 750);
```

Dans le premier cas la fonction « `setTimeout` » traitera l'expression après 750 millisecondes alors que dans le deuxième cas la fonction va entrer en application après 750 millisecondes, et sera donc déjà traitée.

**12. What are the advantages of promises?**

Les promesses possèdent plusieurs avantages :

- Elles n'inversent pas le contrôle, c'est-à-dire qu'elles ne contrôlent pas l'exécution via des callbacks, c'est donc la fonction qui appelle qui garde le contrôle
- Elles sont standardisées avec ES6

**13. Inside which version of ECMAScript async / await has been released?**

Await/async est arrivé dans la version ECMAScript de 2017.

**14. What does the "@" symbol mean in "@babel / \*\*\*"?**

Le symbole @ permet à NPM de comprendre qu'il s'agit d'un scope. Si on enlève le symbole @, NPM va interpréter cela comme un repository Github, c'est pourquoi il faut toujours mettre @ avant un scope. Ici on veut télécharger des packages liés au scope « babel » donc on doit ajouter @ devant babel dans la commande « npm install. »

**15. Look at a files produced in the `dist /` folder. How did Babel transpile your "WelcomeComponent" class?**

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

La taille des fichiers transpilés est de 64 kO et de 1.8 MO pour les fichiers sources.

**17. What is the difference between `import \* from './Utils'` and `import {parseUrl} from './Utils'`?**

La commande « import \* » importe l'intégralité du contenu d'un module alors que la commande « import {export} » va importer un seul export d'un module.

Donc dans ce cas précis on ne veut pas importer tous le module utils entier mais seulement parseurl donc on va rentrer la commande « import {parseUrl } from './utils' » .

**18. Why will the `utils.js` be also transpiled?**

On transpile également utils.js parce qu'on l'importe dans le fichier main. Il faut donc aussi le transpiler si on veut transpiler le main.

**19. What does `webpack --config webpack.config.js` do?**

Cette commande permet de définir un fichier de configuration personnalisé. Ici nous avons créé nous-même le fichier webpack.config.js et on veut qu'il soit considéré comme le fichier de configuration, c'est pourquoi nous utilisons cette commande. La différence de ce fichier avec celui créé automatiquement et qu'il définit le point d'entrée comme étant seulement main.js au lieu d'index.js.

## 20. What does the `webpack-dev-server` do?

Webpack-dev-server permet de lancer un serveur web qui aura l'avantage de compiler automatiquement et donc de mettre à jour le serveur à chaque fois que le code change, donc sans passer par npm run build par exemple.

## 21. Play the whole game with `size = 2`. By browsing the 3 views of the application, how many files has your browser downloaded in total? What was the total size of the transferred data?

État	Méthode	Domaine	Fichier
304	GET	localhost:8080	bundle.js
304	GET	localhost:8080	0e654185d98fc2ecc06d.png
200	GET	localhost:8080	info?t=1620912238338
404	GET	localhost:8080	favicon.ico
101	GET	localhost:8080	websocket
200	GET	localhost:8081	board?size=2
⛔	GET	localhost:8080	dd7c8522accf2aa321b1.png
200	GET	localhost:8080	card-0.png
200	GET	localhost:8080	back.png
200	GET	localhost:8080	card-1.png
304	GET	localhost:8080	bec5ee7f5dfdda324e3a.jpg
🕒 12 requêtes   1,98 Mo / 296,92 Ko transférés   Terminé en : 12,84 s   DOMContentLoaded: 41 ms   load: 163 ms			

## 22. Can you guess how style-loader works exactly?

Style-loader permet de lire les fichiers css avec Webpack.

## 23. What does the prefix `\_` mean on a sass file?

Le symbole \_ indique à Sass que le fichier est un « Partial » c'est-à-dire un fichier qui est importé au moment de la compilation mais qui ne produit pas de fichier issu de celui-ci.