

BÉRIOT Alexis

MIN1

Q1. En tout, 18 fichiers différents ont été téléchargés pour les trois vues en moins d'une centaine de ms à chaque fois. Les fichiers .css sont téléchargés systématiquement à chaque nouvelle page web. Le fichier favicon.ico est tout le temps en 404 (not found) sur chaque vue de l'app.

État	Méthode	Domaine	Fichier	Initiateur	Type	Transfert	Taille	0 ms	160 ms
200	GET	localhost:8080	welcome.html	/src/:11 (document)	html	3,98 Ko	3,65 Ko	1 ms	
404	GET	localhost:8080	favicon.ico	FaviconLoader.jsm:191 (img)		NS_BINDING_ABORTED			
200	GET	localhost:8080	style.css	stylesheet	css	2,57 Ko	2,25 Ko	5 ms	
200	GET	localhost:8080	bootstrap.css	stylesheet	css	188,13 Ko	187,80 Ko	5 ms	
200	GET	localhost:8080	welcome.js	script	js	1,89 Ko	1,55 Ko	5 ms	
200	GET	localhost:8080	logo_take_my_money.png	img	png	20,15 Ko	19,83 Ko	6 ms	
404	GET	localhost:8080	favicon.ico	FaviconLoader.jsm:191 (img)	x-icon	151 o	0 o		2 ms
7 requêtes 215,08 Ko / 216,87 Ko transférés Terminé en : 129 ms load: 73 ms									

Vue : welcome

État	Méthode	Domaine	Fichier	Initiateur	Type	Transfert	Taille	0 ms	160 ms
200	GET	localhost:8080	game.html?name=zebi7&size=2	welcome.js:38 (document)	html	2,58 Ko	2,25 Ko	4 ms	
200	GET	localhost:8080	style.css	stylesheet	css	2,57 Ko	2,25 Ko	7 ms	
200	GET	localhost:8080	bootstrap.css	stylesheet	css	188,13 Ko	187,80 Ko	8 ms	
200	GET	localhost:8080	back.png	img	png	2,78 Ko	2,47 Ko	10 ms	
200	GET	localhost:8080	game.js	script	js	6,27 Ko	5,93 Ko	9 ms	
200	GET	localhost:8080	card.js	script	js	2,41 Ko	2,07 Ko	9 ms	
200	GET	localhost:8080	logo_take_my_money.png	img	png	20,15 Ko	19,83 Ko	15 ms	
200	GET	localhost:8081	board?size=2	game.js:109 (xhr)	json	284 o	17 o	4 ms	
200	GET	localhost:8080	back.png	img	png	mis en cache	4,14 Ko	0 ms	
404	GET	localhost:8080	favicon.ico	FaviconLoader.jsm:191 (img)	x-icon	151 o	0 o	9 ms	
200	GET	localhost:8080	card-1.png	game.js:47 (img)	png	127,08 Ko	126,76 Ko	14 ms	
200	GET	localhost:8080	card-0.png	game.js:47 (img)	png	166,34 Ko	166,02 Ko	8 ms	
12 requêtes 519,54 Ko / 518,73 Ko transférés Terminé en : 124 ms DOMContentLoaded: 80 ms load: 86 ms									

Vue : game

État	Méthode	Domaine	Fichier	Initiateur	Type	Transfert	Taille	0 ms	160 ms
200	GET	localhost:8080	score.html?name=zebi7&size=2&time=25	game.js:131 (document)	html	2,34 Ko	2,01 Ko	4 ms	
200	GET	localhost:8080	style.css	stylesheet	css	2,57 Ko	2,25 Ko	1 ms	
200	GET	localhost:8080	bootstrap.css	stylesheet	css	188,13 Ko	187,80 Ko	4 ms	
200	GET	localhost:8080	happy_homer.jpg	img	jpeg	10,67 Ko	10,35 Ko	11 ms	
200	GET	localhost:8080	score.js	script	js	1,83 Ko	1,49 Ko	6 ms	
200	GET	localhost:8080	logo_take_my_money.png	img	png	20,15 Ko	19,83 Ko	7 ms	
404	GET	localhost:8080	favicon.ico	FaviconLoader.jsm:191 (img)	x-icon	151 o	0 o	3 ms	
7 requêtes 223,73 Ko / 225,83 Ko transférés Terminé en : 86 ms DOMContentLoaded: 79 ms load: 83 ms									

Vue : score

Q2. On peut réutiliser les différents composants déjà codés pour d'autres utilisations et on n'a besoin de ne modifier que le composant visé dans le cas d'une mise à jour/maintenance.

Q3. Si on enlève les *closures*, aucune carte n'est téléchargée sur la vue de jeu et on ne peut donc pas jouer.

État	Méthode	Domaine	Fichier	Initiateur	Type	Transfert	Taille	0 ms	160 ms
200	GET	localhost:8080	game.component.html?name=zebi7&size=2	welcome.component.js:37 (d...	html	2,11 Ko	1,78 Ko	4 ms	
200	GET	localhost:8080	style.css	stylesheet	css	654 o	322 o	8 ms	
200	GET	localhost:8080	card.component.css	stylesheet	css	1,09 Ko	789 o	4 ms	
200	GET	localhost:8080	bootstrap.css	stylesheet	css	188,13 Ko	187,80 Ko	3 ms	
200	GET	localhost:8080	back.png	img	png	2,78 Ko	2,47 Ko	7 ms	
200	GET	localhost:8080	game.component.js	script	js	6,18 Ko	5,84 Ko	4 ms	
200	GET	localhost:8080	card.component.js	script	js	2,31 Ko	1,97 Ko	5 ms	
200	GET	localhost:8080	logo_take_my_money.png	img	png	20,15 Ko	19,83 Ko	6 ms	
404	GET	localhost:8080	See that ? Without closures, I can override variables from other files that belongs t	game.component.js:108 (xhr)	xml	151 o	0 o	4 ms	
404	GET	localhost:8080	favicon.ico	FaviconLoader.jsm:191 (img)	x-icon	151 o	0 o	8 ms	
10 requêtes 220,78 Ko / 223,69 Ko transférés Terminé en : 127 ms DOMContentLoaded: 91 ms load: 95 ms									

Uncaught Error: 404

```
onreadystatechange http://localhost:8080/src/app/components/game/game.component.js:104
fetchConfig http://localhost:8080/src/app/components/game/game.component.js:93
init http://localhost:8080/src/app/components/game/game.component.js:28
<anonymous> http://localhost:8080/src/app/components/game/game.component.html?name=zebi7&size=2:57
<anonymous> http://localhost:8080/src/app/components/game/game.component.html?name=zebi7&size=2:58

onreadystatechange http://localhost:8080/src/app/components/game/game.component.js:104
(Asynchronous : EventHandlerNonNull)
fetchConfig http://localhost:8080/src/app/components/game/game.component.js:93
init http://localhost:8080/src/app/components/game/game.component.js:28
<anonyme> http://localhost:8080/src/app/components/game/game.component.html?name=zebi7&size=2:57
<anonyme> http://localhost:8080/src/app/components/game/game.component.html?name=zebi7&size=2:58
```

Les *closures* permettent de définir un environnement local pour la déclaration de variables, fonctions, etc regroupées sous forme de blocs de code. En les retirant, on enlève toutes les délimitations sur les blocs et on a ainsi des conflits sur des déclarations de variables, fonctions, ...

Q4. Le *package-lock.json* sert à stocker une représentation exacte des dépendances installées dans le projet à un instant donné. Il permet également de pouvoir mettre en cache le dossier *node_modules*, ce qui fait gagner du temps d'intégration.

Q5. On appelle cela le *Semantic Versionning* (semver), le symbole **^** permet de cibler les releases mineures et correctifs, cad : pour « ^5.1.3 » on peut avoir en version compatible 5.1.4 ou 5.2.1 et supérieure.

Q6. Les *devDependencies* sont des modules uniquement nécessaires pour le développement d'un projet. À l'inverse, les *Dependencies* sont également et surtout requises au moment de l'exécution du programme, et donc lors de sa mise en production.

Q7. Avec des classes Java, on peut créer des interfaces ou des classes abstraites, et aussi surcharger un constructeur, ce qui n'est pas possible avec ESNext.

Q8. Le mot clé *let* permet de déclarer des variables dont la portée est limitée à celle du bloc dans lequel elles sont déclarées. Le mot clé *var* permet lui de définir une variable globale ou locale à une fonction, sans distinction des blocs présents dans la fonction.

Q9. La syntaxe **function.bind(this)** retourne une copie de la fonction fournie **function** avec son paramètre *this*. La méthode *bind()* va créer une nouvelle fonction qui a pour contexte la valeur *this* passée en paramètre.

Si on supprime *.bind(this)* le nom, la taille et le temps ne sont pas récupérés par la vue du score.

Les *arrow functions* ne possèdent pas leurs propres valeurs pour *this* càd qu'une *arrow function* ne lie pas son propre *this* au sein de la fonction, on peut ainsi directement passer la valeur de retour de la fonction sans passer par une nouvelle fonction copie de celle exécutée avec le *this* en paramètre.

Q10. Une promesse est un objet qui est renvoyé et auquel on attache des *callbacks* au lieu de les passer en arguments à une fonction. Avec l'appel de fonction asynchrone, le premier avantage des promesses est le **chaînage** : appel d'une méthode sur une autre méthode du même objet. Un autre avantage est que les callbacks ne seront jamais appelés avant la fin du parcours de la boucle d'évènements courante. Enfin, des callbacks ajoutés avec un *.then()* seront appelés quelques soit le résultat de l'opération asynchrone.

Q11. Les mots-clés *async/await* ont été publiés dans la version d'ECMAScript 2016, ce qui correspond à la 7^e édition finalisée en juin 2016.

Q12. Le symbole « @ » dans l'expression « @babel/** » appelle une fonctionnalité appelée « *scoped packages* » qui va regrouper des packages *npm* apparentés et les relier entre eux avec une portée spécifiée.

BÉRIOT Alexis
MIN1

Q13. Babel transpile la classe WelcomeComponent en un fichier .js exécutable sous n'importe quelle version d'ECMAScript et donc n'importe quel navigateur. Le code produit est néanmoins beaucoup moins lisible, compréhensible et maintenable pour un développeur.

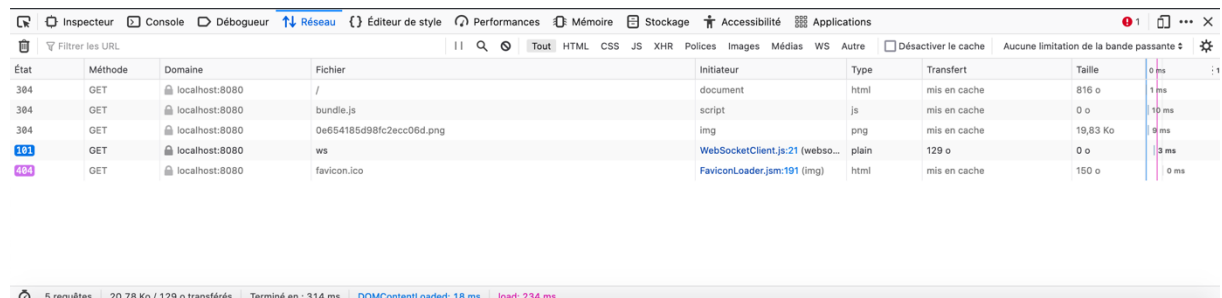
Q14. Le poids des fichiers transpilés est plus important que celui des fichiers originaux, ce qui semble logique au vu du travail de réécriture à effectuer sur le code original pour le transpiler.

Q15. Avec l'instruction « import * from module », on importe l'intégralité des exports contenu dans le module. À l'inverse, avec l'instruction « import { **monexport** } from module », on importe uniquement l'export **monexport** du module en question.

Q16. Le fichier utils.js est également transpilé par Babel car il contient lui aussi du code utilisant des fonctions ajoutées avec ESNext (let, .map, .reduce) qui doit donc être transpiler pour être compréhensible sous n'importe quel navigateur.

Q17. Cette commande permet d'éditer un fichier de configuration *webpack.config.js* pour Webpack avec le flag --config. Ce fichier *webpack.config.js* sera le fichier ayant la nouvelle configuration webpack. Néanmoins, cette commande ne crée pas le fichier qui doit ainsi être créé manuellement dans le répertoire de l'application.

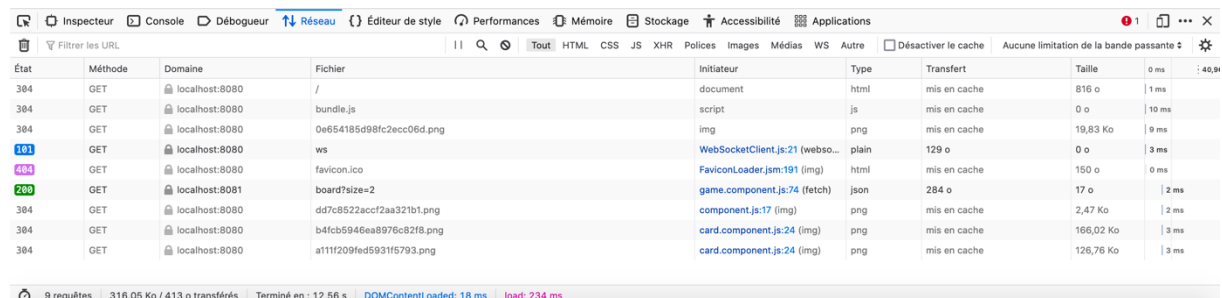
Q18. On a en tout téléchargé 9 fichiers différents sur les trois vues de l'application en 234ms pour chacune des vues. Le temps de chargement des 9 fichiers a ainsi été de 234ms.



État	Méthode	Domaine	Fichier	Initiateur	Type	Transfert	Taille	0 ms
384	GET	localhost:8080	/	document	html	mis en cache	816 o	1 ms
384	GET	localhost:8080	bundle.js	script	js	mis en cache	0 o	10 ms
384	GET	localhost:8080	0e654185d98fc2ecc06d.png	img	png	mis en cache	19,83 Ko	9 ms
191	GET	localhost:8080	ws	WebSocketClient.js:21 (webso...	plain	129 o	0 o	3 ms
413	GET	localhost:8080	favicon.ico	FaviconLoader.jsm:191 (img)	html	mis en cache	150 o	0 ms

5 requêtes 20,78 Ko / 129 o transférés Terminé en : 314 ms DOMContentLoaded: 18 ms load: 234 ms

Vue : welcome



État	Méthode	Domaine	Fichier	Initiateur	Type	Transfert	Taille	0 ms
384	GET	localhost:8080	/	document	html	mis en cache	816 o	1 ms
384	GET	localhost:8080	bundle.js	script	js	mis en cache	0 o	10 ms
384	GET	localhost:8080	0e654185d98fc2ecc06d.png	img	png	mis en cache	19,83 Ko	9 ms
191	GET	localhost:8080	ws	WebSocketClient.js:21 (webso...	plain	129 o	0 o	3 ms
413	GET	localhost:8080	favicon.ico	FaviconLoader.jsm:191 (img)	html	mis en cache	150 o	0 ms
280	GET	localhost:8081	board?size=2	game.component.js:74 (fetch)	json	284 o	17 o	2 ms
384	GET	localhost:8080	dd7c8522accf2aa321b1.png	component.js:17 (img)	png	mis en cache	2,47 Ko	2 ms
384	GET	localhost:8080	b4fcb5946ea8976c82f8.png	card.component.js:24 (img)	png	mis en cache	166,02 Ko	3 ms
384	GET	localhost:8080	a11f1209fed5931f5793.png	card.component.js:24 (img)	png	mis en cache	126,76 Ko	3 ms

9 requêtes 316,05 Ko / 413 o transférés Terminé en : 12,56 s DOMContentLoaded: 18 ms load: 234 ms

Vue : game

État	Méthode	Domaine	Fichier	Initiateur	Type	Transfert	Taille	0 ms
304	GET	localhost:8080	/	document	html	mis en cache	816 o	1 ms
304	GET	localhost:8080	bundle.js	script	js	mis en cache	0 o	10 ms
304	GET	localhost:8080	0e65418d98fc2ecc06d.png	img	png	mis en cache	19,83 Ko	9 ms
101	GET	localhost:8080	ws	WebSocketClient.js:21 (webso...	plain	129 o	0 o	3 ms
104	GET	localhost:8080	favicon.ico	FaviconLoader.jsm:191 (img)	html	mis en cache	150 o	0 ms
200	GET	localhost:8081	board?size=2	game.component.js:74 (fetch)	json	284 o	17 o	2 ms
304	GET	localhost:8080	dd7c8522accf2aa321b1.png	component.js:17 (img)	png	mis en cache	2,47 Ko	2 ms
304	GET	localhost:8080	b4fcb5946aa8976c82f8.png	card.component.js:24 (img)	png	mis en cache	166,02 Ko	3 ms
304	GET	localhost:8080	a11f1209fed5931f5793.png	card.component.js:24 (img)	png	mis en cache	126,76 Ko	3 ms
304	GET	localhost:8080	bee5ee7f5dfdda324e3a.jpg	component.js:17 (img)	jpeg	mis en cache	10,35 Ko	2 ms

Vue : score

Q19. Le *style-loader* récupère une chaîne de caractère générée par le *css-loader* pour la placer dans les balises `<style>` du fichier *index.html*. Le *css-loader* est le module qui va aider webpack à collecter les lignes de code CSS au sein des fichiers css pour enfin les ré-écrire dans une chaîne de caractère.

```

</style>
/* https://flatuicolors.com/palette/defo */ /*! * Bootstrap v5.1.3 (https://getbootstrap.com/) * Copyright 2011-2021 The Bootstrap
base64,eyJ2ZXJzaW9uIjozLCJ2b3VyY2VzIjpbIndlYnBhY2s6Ly8uL3NyYy9hcAvc3R5bGVzL19jb2xvcnMuc2NzcyIsIndlYnBhY2s6Ly8uL25vZGVfbW9kdWxlcys
/aWQ9MTU4MzQyXG4gIC8vIFNlZSBhbHNvIGh0dHBzO18vZ2l0aHViLmNvbS90d2JzL2Jvb3RzdHJhC9pc3NlZXMvMTc2OTVcbn1cb1xuLy8gU2h1bGwgZG1IRHrvIHbv
</style>
<style>
/* https://flatuicolors.com/palette/defo */ /***** * Card Component */ /* card { ... } */ .card-cmp { position: relative; display
</style>
<style>
/* https://flatuicolors.com/palette/defo */ /***** * Score Component */ /* score { ... } */ .score-cmp img.happy:hover { /* Start
</style>
Authors * Copyright 2011-2021 Twitter, Inc. * Licensed under MIT (https://github.com/twbs/bootstrap/blob/main/LICENSE) */ :root { --
ib290c3RyYXVAc2Nzcy9ib290c3RyYXUAc2NzcyIsIndlYnBhY2s6Ly8uL3NyYy9hcAvc3R5bGVzL3N0ewxLLnNjc3MlLCJ2ZWJhYmNrO18vL19ub2RlX21vZHVhZSXMvYm9vd
2l0aW9uIHROZSBtb2RhbcCB3aXR0IGdvJGRHrvBSbWYWRkaW5nXG4ubW9kYWwtZGh1bG9nIHhtc1AgcG9zaXRpb246IHJlbGZ0aXZlL01uICB3aWR0aDogYXV0b2tcbiAgbWYfZ2l
inline-block !important; width: 14%; } .card-cmp .card-wrapper { position: relative; transform-style: preserve-3d; transition: all .5
the shake animation and make the animation last for 0.5 seconds */ animation: shake 0.5s; /* When the animation is finished, start aga

```

Ci-dessus du code css ré-écrit dans les balises `<style>` de la page `index.html` grâce au *style-loader* et au *css-loader*.

Q20. Le préfixe « _ » dans le nom d'un fichier SCSS signifie que ce fichier est partiel càd que le compilateur du navigateur va simplement ignorer les fichiers SCSS commençant par un « _ ».