Développement d’une SPA animée

**JavaScript SPA : Puissance 4**

|  |  |
| --- | --- |
| **Auteur 1** | Rayan BARMAN |
| **Auteur 2** | Alexis MICHIELS |
| **Auteur 3** |  |
| **Date** | 08.12.2019 |
| **Référence** | JS-PROJET-BARMAN-MICHIELS |
| **Version** | 0.1 |

Contents

[0 Consignes et évaluations 2](#_Toc23483430)

[0.1 Consignes générales 2](#_Toc23483431)

[0.2 Consignes techniques, timing et évaluations 3](#_Toc23483432)

[1 Objectif du projet 5](#_Toc23483433)

[2 Besoins fonctionnels et planning 5](#_Toc23483434)

[3 Besoins techniques 7](#_Toc23483435)

[4 Choix technologiques 7](#_Toc23483436)

[5 Code 8](#_Toc23483437)

[6 Reporting & Présentation 8](#_Toc23483438)

[6.1 Présentation 8](#_Toc23483439)

[6.2 Evaluation du résultat par rapport à la description fonctionnelle 8](#_Toc23483440)

[6.3 Ressources utilisées 8](#_Toc23483441)

[6.4 Difficultés rencontrées 8](#_Toc23483442)

[6.5 Conseils pour appliquer cette technologie 8](#_Toc23483443)

# Consignes et évaluations

## Consignes générales

Veuillez travailler par groupe de deux, sauf exception (1 groupe de trois par série si nombre d’étudiants impaire).

Veuillez créer une SPA mettant en œuvre :

* Des sujets et technologies qui vous tiennent à cœur ;
* Un backend Java tournant en mode « Jetty embedded » et offrant une API
* Une animation
* Des appels à vos API
* Au moins une librairie JS non vue en cours, sauf si vous choisissez anime.js

L’animation peut être 2D, 3D, sous forme de jeux ou de simples effets visuels…

## Consignes techniques, timing et évaluations

Les « deadlines » données ci-dessous sont les dates où au plus tard l’avancement des tâches doivent être présentées à l’enseignant pendant le cours.

| **Compétence** | **Tâches** | **Deadline** | **Points** | **Consigne** |
| --- | --- | --- | --- | --- |
| Reporting & présentation | Objectif du projet | **Week\_9** | 0.5 | Décrire l’objectif de votre projet au §2 en Week\_8.  Présenter ce document et faire valider votre objectif par un enseignant en Week\_9 lors d’un cours. |
|  | Besoins fonctionnels et planning | **Week\_10** | 0.5 | Décrire les besoins fonctionnels et le planning associé au §3 en Week\_9.  Présenter ceux-ci à un enseignant en Week\_9 et Week\_10 au plus tard. |
|  | Choix technologiques | **Week\_10** | 0.5 | Compléter le §5 en Week\_10.  Présenter celui-ci à un enseignant en Week\_9 et Week\_10 au plus tard. |
|  | Code | **Week\_11** | 0.5 | Présenter l’état d’avancement de votre code à un enseignant en Week\_11 au plus tard.  Soumettre votre code via Moodle (un devoir sera créé) en Week\_11. |
|  | Rapport | **Week\_11** | 0.5 | Compléter le §7 en Week\_11.  Présenter l’état d’avancement de votre rapport à un enseignant en Week\_11 au plus tard.  Soumettre ce document, dûment complété, via Moodle (un devoir sera créé). |
|  | Présentation | **Week\_12** | 1.5 | Présenter votre projet en Week\_12 selon les exigences du §7. |
| Animation (frontend) | Qualité |  | 2.5 | Qualité : Code bien structuré, UI et UX de qualité |
|  | Originalité |  | 2.5 | Soyez ambitieux, originaux, et vous serez récompensé. |
|  | Adaptation personnelle de la technologie |  | 1 | Appropriation personnelle du code (via commentaires dans le code, discussion lors des cours...). |
|  | Respect des spécifications techniques |  | 1 |  |
| SPA asynchrone (frontend & API) | Qualité |  | 3.5 | Qualité : Code bien structuré, UI et UX de qualité |
|  | Originalité |  | 3.5 | Soyez ambitieux, originaux, et vous serez récompensé. |
|  | Adaptation personnelle de la technologie |  | 1 | Appropriation personnelle du code (via commentaires dans le code, discussion lors des cours...). |
|  | Respect des spécifications techniques |  | 1 |  |
|  | **TOTAL** |  | **20** | Il est à noter que différents membres d’un groupe pourront être côtés différemment en fonction de leur engagement sur le projet visible lors des sessions de cours. |

# Objectif du projet

Création d’une SPA où il y aura la possibilité de jouer à 2 sur 2 machines différentes localement ( les deux utilisateurs doivent être connecter au même réseau LAN) au jeu connu sous le nom de puissance 4. La SPA aura aussi la possibilité de créer un moyen d’enregistrer des utilisateurs sur le site et de proposer un moyen de connexion à celui-ci.

La SPA intégrera aussi un historique personnel et global qui permettra de voir le gagnant et le perdant de chaque parties

1.1

# Besoins fonctionnels et planning

* Enregistrer un utilisateur sur le site (mot de passe crypté)
* Connexion d’un utilisateur (Grâce au token)
* Création d’une instance de partie (Grâce au json)
* Envoie de données entre les utilisateurs et le serveur (json)
* Voir l’historique de tous les joueurs
* Voir l’historique personnel
* Voir les informations du joueur
* Modifier les informations du joueur
* Création du jeu puissance 4
* Intégration d’animation au jeu puissance 4
* Intégration d’animation sur le site en général
* Cryptage mot de passe
* Création de session (token)

| **Sequence** | **Cas d’utilisation** | **Nom du responsable** | **Ordre de priorité** | **Remarques** |
| --- | --- | --- | --- | --- |
| 1 | Base de données Json | Alexis | Haute |  |
| 2 | Création page d’accueil | Alexis | Haute |  |
| 3 | Création du jeu localement | Rayan | Haute |  |
| 4 | Création du login API | Alexis | Haute |  |
| 5 | Création du register API | Rayan | Haute |  |
| 6 | Cryptage du mot de passe | Alexis | Haute |  |
| 7 | Création des animations sur le site en général | Alexis | Haute |  |
| 8 | Échange de données entre client et serveur pour une partie | Rayan | Haute |  |
| 9 | Création historique personnel | Rayan | Moyenne |  |
| 10 | Beauté du site | Alexis | Moyenne |  |
| 11 | Création historique global | Rayan | Haute |  |
| 12 |  |  |  |  |
| 13 |  |  |  |  |
| 14 |  |  |  |  |
| 15 |  |  |  |  |

# Besoins techniques

*La spécification technique des besoins vous est donnée*

*N’hésitez pas à la compléter s’il manque qqch d’important.*

TR01 : La SPA devra être accessible localement via un browser.

TR02 : Un backend Java, tournant en mode « Jetty embedded », doit offrir au minimum une API.

TR03 : Le frontend, développé en HTML / CSS (bootstrap ou autre) / JavaScript, doit consommer votre API.

*Votre frontend peut consommer des API externes.*

TR04 : Votre frontend doit mettre en œuvre une librairie JS externe, ou l’API Canvas, afin de réaliser une animation.

*L’animation peut prendre la forme d’une animation 2D, 3D ou d’un jeu vidéo*

TR05 : Votre frontend doit mettre en œuvre au minimum une librairie JS non vue en cours, sauf si vous choisissez anime.js pour votre animation.

# Choix technologiques

*Les technologies utilisées sont les suivantes :*

*- JWT pour la création de token:* [*https://jwt.io/*](https://jwt.io/)

*- Ajax pour les appels de L‘API:* [*https://api.jquery.com/category/ajax/*](https://api.jquery.com/category/ajax/)

*- Bcrypt pour le cryptage des mots de passe*

*- BOOTSRAP pour la mise en page du site Web:* [*https://getbootstrap.com/*](https://getbootstrap.com/)

*- JETTY pour l’API: https://www.eclipse.org/jetty/documentation/*

*-JSON qui fait office de base de données:* [*https://www.json.org/json-en.html*](https://www.json.org/json-en.html)

# Code

*Veuillez créer un fichier .zip de votre code, ne comprenant pas les packages installés via npm, et le faire parvenir aux profs selon les consignes (devoir sous Moodle).*

# Reporting & Présentation

## Présentation

*Votre présentation se basera principalement sur la démonstration de votre SPA (exécution, en live, de votre API et du frontend) ainsi que des points repris dans ce §7. Elle doit viser une durée de 10 minutes, avec maximum 5 minutes de temps supplémentaire pour des questions & réponses.*

## Evaluation du résultat par rapport à la description fonctionnelle

Tous les besoins fonctionnels ont été atteint, on a misé sur une infrastructure stable pour que ça soit difficile pour un utilisateur connaissant le monde du WEB d’accéder à des ressources dont il n’aurait pas les droits. On a essayé de maximiser l’expérience utilisateur, pour pas qu’il n’ait qu’un simple jeu, mais une expérience globale.

## Ressources utilisées

Les ressources utilisées sont celles montrées en haut dans le document, ainsi que des sites spécialisé comme: <https://www.w3schools.com/> et <https://developer.mozilla.org/fr/docs/Web/JavaScript> pour tout ce qui est du javascript (comme créer des éléments et les ajouter au DOM, utiliser le JSON), <https://codepen.io/> pour tout ce qui est animation de base.

## Difficultés rencontrées

Pour la réalisation de ce projet, aucune grosse difficulté n’a été rencontrée. Si ce n’est qu’un minimum de réflexion pour mettre en œuvre les différents algorithmes. Nous avons dès le début bien découpé le travail pour ne faire qu’une seule chose à la fois et pas démarrer pleins de taches différentes et donc éviter de s’y perdre.

Nous nous sommes basés sur notre idée de base et nous avons continué à la suivre. Tout s’est donc très bien passé.

## Conseils pour appliquer cette technologie

Pour appliquer le même genre de technologie, il est important de se baser sur un exemple pour voir comment cela fonctionne dans un premier temps. Bien lire le code pour pouvoir se l’approprier. Lire la documentation est une chose essentielle lorsqu’on utilise une technologie nouvelle quelle qu’elle soit. Il ne faut surtout pas se décourager quand on ne comprend pas une notion directement, il faut persévérer.