

Ergonomie & développement d'une SPA animée

Shadow Fortress

Auteur 1	Rayane Belbachir
Auteur 2	Mohamed Mcheik
Auteur 3	Gabriel Paquet
Auteur 4	Roemer-Trinité Quiambao
Auteur 5	Mohamed Touali
Date	17.12.2023
Référence	WEB2-2023-PROJECT-GROUP-20
Version	1.0

Contents



1	Consignes et évaluations.....	3
1.1	Consignes générales	3
1.1.1	Création des groupes sur le site du cours.....	3
1.1.2	Création d'un groupe sur GitHub Classroom et du web repo associé	4
1.1.3	Projet.....	4
1.2	JavaScript & Node.js : consignes techniques, timing et évaluations	6
1.3	Ergonomie : consignes techniques, timing et évaluations.....	11
2	Objectif du projet	11
3	Mind map du projet	12
4	Persona	12
5	Axiomes de Morville	12
6	Planning des tâches et cas d'utilisation.....	12
7	Besoins techniques	12
7.1	Système.....	12
7.2	Frontend	13
7.3	API	13
8	Choix technologiques.....	14
8.1	Frontend	14
8.2	RESTful API	14
8.3	Wireframe	15
9	Conception & Implémentation.....	15
9.1	Code repositories	15
9.2	Secrets éventuels pour vos API ou base de données.....	15
9.3	Documentation de votre API.....	15
9.4	Déploiement de vos applications.....	16
9.5	Code réutilisé	16
10	Analyse des résultats par le groupe	16
10.1	Évaluation du résultat par rapport au planning des tâches et des cas d'utilisation	16
10.2	Audit ergonomique de votre projet.....	17

10.3	Difficultés techniques rencontrées	17
10.4	Conseils pour appliquer cette technologie.....	17
10.5	Quels sont les points positifs à la manière dont s'est déroulée la collaboration au sein du groupe ?	18
10.6	Quels sont les points qui seraient à améliorer pour de futures collaborations ?	18
11	Analyses individuelles des résultats	22
12	Présentation vidéo.....	18
13	Revue de projets par les pairs	24

1 Consignes et évaluations


1.1 Consignes générales

1.1.1 Création des groupes sur le site du cours

Veillez former un groupe de 5 étudiants sur le site associé au cours : <https://e-vinci.github.io/web2>. Pour ce faire, veuillez-vous authentifier en cliquant sur l'icône . Rendez-vous sur l'onglet **Projets** (<https://e-vinci.github.io/web2/project-page>). Il est recommandé que l'attribution des **groupes** se fasse par **discussions** entre les **étudiants**. Lorsque 5 étudiants ont un **intérêt commun** pour un **projet**, ils s'inscrivent au sein d'un groupe en cliquant sur l'icône .

Pour aider à la création de groupes, il est aussi possible de vous inscrire :

- à un **groupe vide**. Cela permettra à tous d'identifier les partenaires potentiels.
- à un **groupe où il y a déjà un ou plusieurs étudiants**. Dans ce cas, veuillez-vous entretenir avec ces potentiels partenaires sur le **sujet de votre projet**.

Si nécessaire, vous pouvez vous désinscrire d'un groupe où vous n'avez pas trouvé de sujet commun dans le but de rejoindre un autre groupe. Il suffit de cliquer sur l'icône .

A la date ultime de création de groupe (**Séance 9 du cours de JS, 16/10 ou 17/10 selon la série**), pour les étudiants toujours en recherche de partenaires, nous faciliterons (ou imposerons si nécessaire) la création des groupes, mais pas des sujets de projet.

Une fois tous les groupes de 5 étudiants remplis, il restera maximum 4 étudiants non liés à un projet. Si nécessaire un ou plusieurs groupes de 4 étudiants seront créés.

1.1.2 Création d'un groupe sur GitHub Classroom et du web repo associé

Pour chaque groupe de projet, vous allez hériter d'un web repository contenant un boilerplate via GitHub classroom.

Veillez passer à cette étape qu'une fois votre groupe déjà finalisé sur le site du cours.

1.1.2.1 Création de l'équipe associée à un projet

Veillez identifier le membre qui créera votre équipe sur GitHub.

Ce membre accédera à l'assignement via : <https://classroom.github.com/a/zJz7A4kY>

Ce membre devra créer une équipe reprenant le numéro de projet donné sur <https://e-vinci.github.io/web2/project-page> : si le nom de projet indiqué est **Projet N°4** : ... , il créera une équipe portant le nom **group-04** puis cliquera sur **Create team**.

Ce membre devra encore cliquer par la suite sur **Accept this assignment**.

Un web repository aura été créé pour votre équipe.

1.1.2.2 Joindre une équipe existante

Une fois l'équipe d'un projet créée, les autres membres accèderont aussi à l'assignement via : <https://classroom.github.com/a/zJz7A4kY>.

Ces membres joindront l'équipe existante en cliquant sur **Join** au sein de la bonne équipe. Par exemple, pour les membres du **Projet N°4**, ils cliqueront sur **Join** dans l'équipe **group-04**.

Si vous le souhaitez, vous pouvez visualiser cette vidéo qui montre [comment Joindre un GitHub Classroom Group Assignment](#).

1.1.3 Projet

Vous allez créer une SPA mettant en œuvre :

- Des sujets et technologies qui vous tiennent à cœur ;
- Une RESTful API tournant sous Node.js & Express ;
- Un frontend animé ;
- Un frontend consommant votre RESTful API et éventuellement des APIs tierces ;

- Au moins une librairie JS non vue en cours pour le frontend (anime.js ou phaser.io sont autorisées) ainsi qu'une librairie non vue pour l'API.

Pour votre frontend animé, l'animation peut être 2D, 3D, sous forme de jeux ou de simples effets visuels...

Au niveau de la présentation de votre projet, veillez à :

- Prendre en compte l'expérience utilisateur dès le début
- Optimiser le choix de vos technologies en fonction de l'expérience utilisateur
- Appliquez un maximum de théorème psycho-marketing
- Respectez les règles de Usability et auditez votre projet
- Respectez le GDPR

1.2 JavaScript & Node.js : consignes techniques, timing et évaluations

Tâche	Compétences	Critères	Deadline	Pt	Consignes
Objectif du projet	C6) Documenter et présenter en vidéo le développement d'une SPA		Séance 9 : 16/10 ou 17/10		<p>Donnez un nom à votre projet et décrire l'objectif de votre projet au §2 de ce document ainsi que sur https://e-vinci.github.io/web2/project-page, complétez :</p> <ul style="list-style-type: none">- Le nom du projet : Projet N°X : Nom de votre projet- le champs « Description ». <p>Discuter de votre objectif avec un enseignant et assurer vous que cet objectif soit validé avant d'aller plus loin dans votre projet.</p>
Planning des tâches et cas d'utilisation	C6)		Séance 12 : 24/10 ou 27/10		<p>Décrire le planning des tâches et cas d'utilisation selon les instructions données au §6.</p> <p>Présenter votre planning à un enseignant, afin qu'il puisse vous aider à bien prioriser les tâches.</p>
Indiquer l'URL de votre code repository	C6)		Séance 12 : 24/10 ou 27/10		<p>Votre code doit être accessible par tout le monde via un web repository public qui vous sera assigné par GitHub Classroom. Cela permettra notamment aux enseignants de suivre vos avancées tout au long de votre projet. Veuillez indiquer votre URL sur https://e-vinci.github.io/web2/project-page.</p> <p>Plus d'information aux §1.1.2 et §9.1.</p>

Tâche	Compétences	Critères	Deadline	Pt	Consignes
Choix technologiques	C6)		Séance 15 : 13/11 ou 14/11		Compléter le §8. Discuter de vos choix technologiques avec un enseignant.
Rapports individuels d'activités	C6)	Rapports de qualité <i>Indicateurs : formulation de qualité, analyse de qualité, respect des consignes</i>	12/11 19/11 26/11 3/12 10/12 17/12	1 solo	Des sessions individuelles de feedback sont organisées via TEAMMATES permettant à chacun de répondre à des questions dont les réponses sont confidentielles ou anonymisées au sein d'un groupe. Des e-mails seront envoyés vous invitant à compléter un formulaire hebdomadaire, à compléter pendant le WE. À partir de la 2 ^{ème} soumission, tout formulaire hebdomadaire non complété amènera à une pénalité individuelle de 0.5 point. Si vous manquez deux soumissions, vous aurez l'obligation de montrer que vous êtes actif sur le projet sous risque d'être écarté du projet.
Soumission du rapport de groupe	C6)	Idem	17/12	1	Compléter le §10 ainsi que tous les paragraphes qui n'auraient pas été finalisés de ce document. Soumettre ce document, via Moodle (un devoir sera créé) ainsi que dans le répertoire /report de votre repo.

Tâche	Compétences	Critères	Deadline	Pt	Consignes
					Effacer toutes les consignes mises <i>en grisé</i> dans ce document avant de soumettre ce rapport sur Moodle.
Soumission de la vidéo	C6)	Vidéo de qualité <i>Indicateurs : présentation du projet de qualité, analyse de qualité, respect des consignes</i>	17/12	2	Présenter votre projet selon les exigences du §11.
Soumission du code du frontend	C2) Création d'IHM pour SPA inclus : C5 : Intégrer au développement d'une SPA des technologie non vues en cours inclus si nécessaire : C3) Sécurisation de SPA	Qualité de l'IHM produite <i>Indicateurs : esthétique, fonctionnel, codage de qualité, respect des consignes, ambitieux & original, utilisation d'une librairie pour l'IHM non vue en cours</i>	17/12	8	Réaliser un frontend et un backend de Qualité : Code bien structuré, UI et UX de qualité, API bien documentée (documentation des opérations de votre API, requêtes permettant de tester votre API...) Être ambitieux et original. Démontrer une appropriation personnelle du code (via commentaires dans le code, discussion lors des cours...).
Soumission du code du backend	C1 : Création de services web inclus : C5 : Intégrer au développement d'une SPA des technologie non vues en cours inclus si nécessaire : C3 : Sécurisation de SPA	Qualité du web service produit <i>Indicateurs : fonctionnel, codage de qualité, respect des consignes, ambitieux & original, utilisation d'une librairie pour le service web non vue en cours</i>	17/12	5	Respecter les spécifications techniques décrites dans ce document. Déployer votre frontend et votre backend chez un provider gratuit. NB : votre RESTful API doit être un minimum différente des APIs fournies dans les démos du cours de JS.

Tâche	Compétences	Critères	Deadline	Pt	Consignes
Déploiement tant de votre frontend que backend	C4) Déploiement d'applications web	Déploiement de la SPA sur le cloud <i>Indicateurs : fonctionnel, performances de chargement acceptables</i>	17/12	2	
Réaliser un minimum de 5 revues sur le site web	C7) Analyser le développement de SPA faites par des pairs	Revue de projets compréhensibles & constructives <i>Indicateur : présence d'un minimum de 5 revues</i>	Avant examen de 1 ^{ère} session	1 solo	Via https://e-vinci.github.io/web2/my-reviews-page , vous devez revoir les vidéos de présentation de 5 groupes (sauf le vôtre), exécuter leurs applications, et fournir votre critique de chacun de ces projets. Vous pourrez fournir la critique d'autant de projets que vous le souhaitez. Plus d'info sur la revue de projet au § Erreur ! Source du renvoi introuvable.
	TOTAL POINTS			20	<p>Il est à noter que des membres d'un même groupe pourront être cotés différemment en fonction de leur engagement sur le projet. L'engagement d'un étudiant est visible via les rapports individuels d'activités (outil TEAMMATES) via GitHub (GitHub Project, Issues, Milestones, commits...) et lors des sessions de cours.</p> <p>Les étudiants non actifs risquent d'être écarté du projet, spécialement s'ils ne soumettent pas leurs rapports individuels.</p> <p>Les étudiants n'ayant pas réalisé au moins un use case significatif seront considérés inactifs.</p>

Tâche	Compétences	Critères	Deadline	Pt	Consignes
					Les étudiants n'ayant pas participé significativement au projet recevront d'office une lourde pénalité au niveau de leurs points, voire un 0/20.

1.3 Ergonomie : consignes techniques, timing et évaluations

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

Compétence	Tâches	Deadline	Points	Consigne
Reporting & présentation	Objectif du projet	17/10		Décrire l'objectif de votre projet au §2 de ce document. Discuter de votre objectif avec un enseignant et assurer vous que cet objectif soit validé avant d'aller plus loin dans votre projet.
Conception	Définir la vision marketing	5/11	6	Décrire le Mind map du projet. Créer le persona de (s) l'utilisateur (s) ciblé (s) par le projet. Répondre aux axiomes de Morville.
Analyse d'applications web	Architecture UX	5/11	4	Construire les wireframes détaillés de votre application.
	Analyse des résultats et rapport associé	17/12	2	Auditez votre projet et vérifiez le respect des règles GDPR.
	Présentation vidéo	17/12	8	Présenter votre projet en intégrant l'expérience utilisateur.
	TOTAL		20	Il est à noter que différents membres d'un groupe pourront être cotés différemment en fonction de leur engagement sur le projet visible lors des sessions de cours.

2 Objectif du projet

Notre équipe est passionnée par l'univers du jeu vidéo, ce qui nous motive à développer un jeu de type tower defense. Nous avons tous grandi en explorant ce monde fascinant, et cette passion perdure aujourd'hui. Notre jeu vise un public diversifié, allant des jeunes enfants découvrant les jeux

vidéo pour la première fois aux adultes qui ont connu l'apogée du genre lors de leur jeunesse, ainsi qu'aux personnes âgées curieuses de s'immerger dans la culture des jeunes d'aujourd'hui. Nous aspirons à créer une expérience ludique et accessible, offrant un divertissement de qualité à un large éventail de joueurs tout en rendant hommage à la richesse et à la nostalgie de l'univers du tower défense. Notre jeu combine à la fois le plaisir de l'amusement et un aspect compétitif. Vous pouvez y jouer sans avoir besoin de vous connecter, ce qui signifie que n'importe qui peut lancer une partie et en profiter instantanément. Cependant, pour ceux qui recherchent une expérience plus compétitive, il est nécessaire de créer un compte et de se connecter. À la fin de chaque partie, votre score est comparé au meilleur score de l'utilisateur et est sauvegardé s'il est supérieur. Cela vous permet de défier vos propres performances et de rivaliser avec d'autres joueurs pour obtenir le meilleur score possible.

3 Mind map du projet

Remis sur le repository /ergonomics/marketing-view .

4 Persona

Remis sur le repository /ergonomics/marketing-view .

5 Axiomes de Morville

Remis sur le repository /ergonomics/marketing-view .

6 Planning des tâches et cas d'utilisation

- URL vers votre GitHub Project public : <https://github.com/orgs/e-vinci/projects/91>

7 Besoins techniques

7.1 Système

TRS01 : Vous devez développer une Single Page Application (SPA) à l'aide de JS et Node.js.

TRS02 : Votre RESTful API doit être indépendant de votre frontend ; vous aurez donc deux applications distinctes, une pour le frontend et l'autre pour la RESTful API.

TRS03 : Vous devez utiliser GitHub sur votre projet afin de gérer le développement de chacun des membres d'une équipe.

Nous vous recommandons d'appliquer un workflow vu dans votre cours de DevOps : pour chaque cas d'utilisation / feature que vous développez, essayez de créer une branche correspondante. De plus, il serait intéressant que vous mettiez en œuvre des revues de code au sein de votre projet via des Pull Request sur Github.

7.2 Frontend

TRF01 : Votre frontend doit utiliser Webpack en tant que package bundler.

TRF02 : Le frontend, développé en HTML / CSS (bootstrap ou autre) / JavaScript, doit consommer au moins une de vos RESTful API.

Votre frontend peut consommer des API externes, des APIs que vous n'avez pas développées vous-même (e.g. API de youtube, de google maps...)

TRF03 : 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. Attention à ne pas juste offrir une minuscule animation à l'aide d'une librairie ne demandant aucun code JS, comme certaines librairies mettant tout en œuvre à l'aide de CSS.

TRF04 : Votre frontend doit mettre en œuvre au minimum une librairie JS non vue en cours.

Anime.js est autorisé pour votre animation.

TRF05 : Votre frontend doit respecter les droits d'auteurs, que ça soit pour les éventuels sons, images, vidéos, librairies et morceaux de codes utilisés. Cela est de votre responsabilité et non pas de celle de vos enseignants.

TRF06 : Vous devez déployer votre frontend sur GitHub Pages ou d'autres providers gratuits supportant votre application.

7.3 API

TRA01 : Vous devez créer une RESTful API afin d'offrir des opérations sur des ressources utiles à votre projet.

La RESTful API ne peut pas être uniquement un « copier/coller » de ressources offertes dans le cours (notamment les ressources users et auths). Vous pouvez utiliser les ressources offertes dans le cours, mais vous devez y apporter des ajouts significatifs.

TRA02 : Votre RESTful API doit mettre en œuvre au minimum un package non vu en cours.

TRA03 : Vous devez documenter les opérations de votre API conformément aux conventions REST. *Vous pouvez documenter votre API soit sous forme de tableau, comme vu dans le cours, soit à l'aide d'outils tel que Swagger.*

TRA04 : Les tests de votre API, les requêtes HTTP, doivent être données au sein de votre projet. Pour chaque opération de votre API, il doit exister au minimum une requête HTTP associée.

TRA05 : Votre API doit respecter les droits d'auteurs, que ça soit pour les éventuelles librairies utilisées, les morceaux de code, les sons, images, vidéos... Cela est de votre responsabilité et non pas de celle de vos enseignants.

TRA06 : Vous devez déployer votre backend sur Azure ou d'autres providers gratuits supportant votre application.

8 Choix technologiques

Nous avons choisi d'utiliser Phaser.io comme librairie frontend et MongoDB comme librairie backend. Ces deux librairies sont adaptées pour être utilisé ensemble. MongoDB et Phaser sont des choix qui correspondent au jeu que nous souhaitons développer.

8.1 Frontend

Nous avons utilisé Phaser 3 qui est une librairie pour des jeux en 2D. La gestion des collisions entre les ennemies et les projectiles tirés par nos tours est simplifier. Elle propose aussi plusieurs fonctions qui permet de modifier un sprite (des animations, redéfinition de la sprite, gerer le comportement). Nous avons aussi utilisé l'application Tiled pour créer la map principal de notre jeu. Pour les pages de notre site, nous avons utilisé Bootstrap et du simple CSS.

8.2 RESTful API

Concernant notre API, nous recherchions une librairie qui nous aiderait dans la gestion des données des utilisateurs. Il était nécessaire d'avoir une base de données puisque nous voulions avoir un tableau de score et qu'il fallait enregistrer des utilisateurs. Nous avons donc choisi MongoDB, cette librairie de gestion de base de données nous a permis de faciliter le traitement des informations des utilisateurs.

MongoDB nous a permis de gérer de manière efficace l'enregistrement des utilisateurs et des scores. Le traitement de ces données était simplifié et l'utilisation de cette librairie permettait d'obtenir un code plus lisible et optimisé.

8.3 Wireframe

Remis sur le repository /ergonomics/wireframes .

9 Conception & Implémentation

9.1 Code repositories

- URL pour le web repository public associé à votre projet : <https://github.com/e-vinci/web2-2023-project-group-20/tree/main>

9.2 Secrets éventuels pour vos API ou base de données

Fichier envoyé sur moodle.

9.3 Documentation de votre API

- Tableaux représentant les opérations de votre API ou lien vers la documentation de votre API :

URI	Méthode HTTP	Opération
scores	GET	READ ALL : Lis le score le plus élevé de chaque joueur.
scores	POST	CREATE ONE : Créer une ligne de score avec le nom d'utilisateur, la vague et le score dans le tableau des scores.
scores/allGames?username=value	GET	READ ALL : Lis toutes les parties d'un utilisateur
auths/register	POST	CREATE ONE : Enregistre un utilisateur.

auths/login	POST	CREATE ONE : Connecte un utilisateur
user	GET	READ ONE : Lis un utilisateur

- Requêtes HTTP se trouvent dans : **/api/REST Client**

9.4 Déploiement de vos applications

- URL de votre frontend déployé : <https://github.com/Niromin/ShadowFortress>
- URL de votre RESTful API déployée : <https://shadowfortress.azurewebsites.net>

9.5 Code réutilisé

Chemin du fichier où se trouve le code réutilisé	Auteur du code source réutilisé	URL où le code réutilisé est disponible	Raison de la réutilisation du code
<i>e.g. /webApp/src/index.js</i>	<i>Dogan Erisen</i>	<i>https://github.com/Azure-Samples/active-directory-b2c-javascript-msal-singlepageapp</i>	<i>Code pour recevoir un access token via Azure AD</i>

10 Analyse des résultats par le groupe

10.1 Évaluation du résultat par rapport au planning des tâches et des cas d'utilisation

Nous avons atteint la majorité de nos objectifs. Les objectifs restants n'étaient pas une haute priorité, nous avons préféré les mettre de cotés pour nous concentrer davantage sur les objectifs fonctionnels et importants.

Nous avons essayé de garder au maximum les objectifs fonctionnels initiaux. Nous n'avons quasiment pas dévié de ces objectifs. Cependant, il était parfois nécessaire d'ajouter certains

objectifs pour pouvoir en compléter d'autres. Par exemple, « s'occuper de l'authentification des opérations » était nécessaire à faire pour pouvoir compléter l'objectif « Pages principales (BACKEND) ».

Pour terminer, nous sommes satisfaits du résultat par rapport au planning des tâches et des cas d'utilisation. Nous avons respecté nos engagements et nous avons su nous concentrer sur les tâches importantes. Il est clair que nous aurions préféré compléter l'entièreté du planning mais l'objectif principal est atteint.

10.2 **Audit ergonomique de votre projet**

Les règles de gdpr sont respectées.

10.3 **Difficultés techniques rencontrées**

Parmi les difficultés techniques rencontrées durant le projet, nous avons dû faire face à la nécessité d'apprendre de nouvelles technologies. Certaines d'entre elles étaient totalement inconnues pour nous au début du projet. Cela a entraîné un apprentissage en profondeur, ce qui a pris du temps avant que nous puissions réellement créer notre application.

Nous avons rencontré des problèmes liés au backend pendant le développement de notre projet. Plus précisément, nous avons eu des difficultés avec les problèmes liés aux CORS de notre backend, ce qui a posé quelques soucis. Après des recherches approfondies, nous avons finalement réussi à résoudre ces problèmes, bien que cela ait été un défi.

10.4 **Conseils pour appliquer cette technologie**

Nous avons utilisé la librairie Phaser 3 pour la partie frontend du code car c'est une librairie assez simple à comprendre surtout pour créer des jeux web pour un petit projet. Le plus grand avantage de Phaser 3 est qu'il existe beaucoup de tutoriel qui sont disponible sur internet, sur youtube mais aussi sur le cours de web2.

Le site Phaser 3 vous procure un tutoriel pour faire un petit jeu de platform avec des animations et des physiques.

Sur Youtube il y a de nombreux créateur de contenu tel que Ourcade (Js,Ts)

(<https://www.youtube.com/@ourcadetv>), Ansimuz (Phaser 3 du style pixel art)

(<https://www.youtube.com/@ansimuz>), ou encore Zenva (<https://www.youtube.com/@Zenva>).

Ansimuz propose une playlist de 13 vidéos qui consiste de à faire un jeu de type "Space Invader". Chaque vidéo explique un point de base important de Phaser 3 (tel que les préparations, notions de scene, sprites, physique, etc).

Je me suis aussi inspiré sur le boilerplate que les professeurs nous ont fournis et les autres projets web des années précédents. Pour info, le jeu du boilerplate est le même jeu que sur le site Phaser 3, ce qui est une bonne base de code.

Le site itch.io (<https://itch.io/game-assets>) est un très bon site qui vous procure des assets gratuits. Une grosse partie de ces assets sont de licence **CC0**.

10.5 Quels sont les points positifs à la manière dont s'est déroulée la collaboration au sein du groupe ?

La collaboration au sein du groupe a engendré plusieurs points positifs, notamment l'entraide entre pairs lorsqu'un problème surgissait dans le développement de notre application. De plus, la répartition des tâches a permis de produire un code de meilleure qualité, car chacun avait plus de temps pour se concentrer sur une partie spécifique du projet. Cela a également permis d'appliquer les différentes compétences acquises lors de nos cours de Devops.

Il a également été nécessaire de bien gérer notre temps pour livrer le travail demandé dans les délais impartis.

10.6 Quels sont les points qui seraient à améliorer pour de futures collaborations ?

Pour les futures collaborations, voici les points à améliorer : une meilleure structuration du groupe en désignant un chef de projet chargé de superviser le travail. Cela permettrait au groupe de travailler plus sereinement et de se concentrer davantage sur le projet. Il est également important de diversifier les talents, car en tant qu'étudiants, nos compétences sont souvent similaires, ce qui peut entraîner des retards dus à l'apprentissage de nouvelles compétences pour certaines tâches. Effectuer des comptes rendus hebdomadaires entre les membres du groupe afin d'obtenir une vue d'ensemble de l'avancement du projet et, le cas échéant, de combler les retards.

11 Présentation vidéo

Voici les exigences associées à votre présentation vidéo :

- *Elle doit viser une durée de 5 minutes, et ne peut pas dépasser 10 minutes.*
- *Elle doit être visible sous youtube par n'importe qui possédant son URL. Sa visibilité doit donc être en "Unlisted" ou "Public", mais pas « Private » !*
- *Elle se basera principalement :*
 - o *sur la présentation de votre application web : exécution, en live, de votre API et du frontend ;*
 - o *la présentation de l'expérience utilisateur ;*
- *Elle pourra aussi se baser sur d'autre(s) point(s) éventuel(s) vous permettant de vendre au mieux votre travail.*
- *Votre présentation devra être bien visible et audible.*
- *Il serait bien que celle-ci soit bien structurée, notamment via l'affichage éventuel de titres.*

- Vous pouvez ajouter une bande son, et des images, mais seulement si celles-ci respectent les droits d'auteurs.
- Vous veillerez à ce que, dans la description de votre vidéo, vous fournissiez les liens vers le web repository associé à votre projet ainsi que l'URL vers le frontend déployé. L'idée est que si un projet intéresse des visiteurs de votre repo, ils aient accès à tout ce qui est nécessaire pour bien le comprendre, voire pour le réutiliser, sous réserve de bien citer vos ressources.

En plus de ces exigences, la présentation vidéo a pour but de vendre un projet qui vous tient à cœur. Il est possible que si le résultat soit accrocheur, les enseignants demandent votre autorisation afin de rendre votre projet public. Avec votre autorisation, nous pourrions notamment présenter votre projet lors de salons d'étudiants, soit via votre vidéo, ou directement en exécutant votre application déployée sur le cloud.

Le site <https://e-vinci.github.io/web2/> présentera les projets qui auront été sélectionnés pour être publics. De plus, vous pourriez utiliser vos projets comme portfolio pour vos futurs employeurs.

Pour créer votre vidéo, avant de la mettre sous youtube, veillez à ce que celle-ci soit bien visible et bien audible. Nous vous recommandons :

- de la réaliser au format 1920 X 1080
- d'utiliser un logiciel gratuit pour la réaliser. Voici ceux que nous pouvons vous conseiller :
 - o <https://obsproject.com/> : logiciel open source demandant un temps d'adaptation, mais permettant de faire énormément
 - o <https://www.loom.com/> : logiciel pouvant être utilisé gratuitement sous réserve d'accepter un logo. Très facile d'utilisation.
 - o <https://screencast-o-matic.com/> : logiciel pouvant être utilisé gratuitement sous réserve d'accepter un logo. Très facile d'utilisation.

Veuillez indiquer le lien vers la vidéo youtube que vous avez créée sur le site <https://e-vinci.github.io/web2/project-page>. Pour ce faire, veuillez modifier le champ « Vidéo de présentation ».

De plus, veuillez indiquer ci-dessous ce lien :

Lien vers la vidéo youtube : <https://www.youtube.com/watch?v=UYYNYC1fnk>