

Ergonomie & développement d'une SPA animée

SantaFall

Auteur 1	Cattaruzza Alexis
Auteur 2	Debin Pierre-Alexandre
Auteur 3	Devos Thibaut
Auteur 4	Massart Xavier
Auteur 5	Ubah Chisom
Date	17.12.2023
Référence	WEB2-2023-PROJECT-GROUP-18
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	12
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	15
8.2	RESTful API	15
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.....	16
9.4	Déploiement de vos applications.....	16
9.5	Code réutilisé	18
10	Analyse des résultats par le groupe	18
10.1	Évaluation du résultat par rapport au planning des tâches et des cas d'utilisation	18
10.2	Audit ergonomique de votre projet	19

10.3	Difficultés techniques rencontrées	19
10.4	Conseils pour appliquer cette technologie.....	20
10.5	Quels sont les points positifs à la manière dont s'est déroulée la collaboration au sein du groupe ?	21
10.6	Quels sont les points qui seraient à améliorer pour de futures collaborations ?	21
11	Analyses individuelles des résultats	22
12	Présentation vidéo.....	22
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 §12.
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

Nous aspirons à développer un jeu vidéo sur navigateur mettant en scène le Père Noël descendant une montagne tout en évitant divers obstacles sur sa route. Ce jeu comportera une boutique et un tableau des scores gérés par une API RESTful.

Au cours de la descente, le joueur pourra accumuler des pièces, lui permettant d'acquérir des éléments dans la boutique pour personnaliser son expérience. Cette dernière proposera des 'skins' pour modifier l'apparence du personnage ainsi que des thèmes altérant l'environnement du jeu, des décors aux obstacles.

L'objectif principal consistera à parcourir la plus grande distance possible afin d'avoir le privilège de figurer parmi les meilleurs joueurs dans le tableau des scores.

Notre ambition est d'atteindre un large public et prendre tout type de profil en charge, de ce fait le jeu se voudra accessible à tous. Jouable sur ordinateur et téléphone, sa prise en main sera simple par la présence d'une seule action : sauter ! De plus, le jeu sera jouable hors ligne, bien que certaines fonctionnalités telles que le score ou la boutique seront limitées dans cette configuration.

Ce projet nous tient à cœur car il marque un premier saut dans l'univers du jeu vidéo. Notre objectif est de créer un jeu à la fois stimulant et accessible à tous, et nous espérons relever ce défi avec succès.

3 Mind map du projet

4 Persona

5 Axiomes de Morville

6 Planning des tâches et cas d'utilisation

- URL vers votre GitHub Project public :
<https://github.com/e-vinci/web2-2023-project-group-18/tree/main>

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

Afin de réaliser ce projet, nous avons choisi d'employer plusieurs packages JavaScript. Pour la partie frontale de l'application, nous avons opté pour Phaser.js et Simplify.js.

Concernant l'UC : Generate Ground (#19) :
Phaser.js a été d'une importance capitale pour la réalisation du jeu en gérant la physique, les déplacements du personnage, et divers autres aspects. Cette librairie a évité la nécessité de recréer entièrement ces fonctionnalités, économisant ainsi un temps considérable. Simplify.js a été très utile pour la génération aléatoire du terrain. Cette librairie a permis de créer des courbes à partir d'une série de points, offrant des options pour contrôler leur variation.

Pour les UC de type backend (#2, #4, #13, #14, #15) :
Pour l'API, nous avons choisi PostgreSQL dès le départ afin de mettre en pratique nos connaissances en bases de données. Nous avons également utilisé dotenv pour protéger les données sensibles de chaque utilisateur.

Nous avons employé node-postgres pour la communication avec notre base de données, hébergée sur elephantsql.com, un service gratuit. Cette approche nous a grandement aidés dans la manipulation des données, permettant de stocker des informations cruciales et d'améliorer la qualité du jeu.

DotEnv, recommandé dans le cours, a été utilisé pour stocker de manière sécurisée des données sensibles comme les identifiants de connexion à la base de données et les secrets pour la gestion

des tokens, ces derniers étant essentiels pour certaines opérations liées aux utilisateurs spécifiques.

Pour les UC #20 et #13 :
TexturePackerGui a été employé pour séparer les images en un fichier .json (Phaser), permettant la gestion des différentes animations du joueur. Cette application s'est révélée très utile dans la création des animations pour le jeu.

8.1 Frontend

Du côté du frontend, voici les librairies et autres technologies utilisées :

1. Pour le jeu, nous avons choisi d'utiliser le Framework Phaser : <https://phaser.io/>
2. Pour le sol du jeu, nous avons utilisé la librairie Simplify.js : <https://mourner.github.io/simplify-js/>
3. Pour le jeu et le shop, nous avons utilisé une application tierce qui s'appelle TexturePackerGui : <https://www.codeandweb.com/texturepacker>

8.2 RESTful API

Nous avons utilisé le package node-postgres (<https://node-postgres.com/>), qui nous a permis de communiquer avec une base de données de type PostgreSQL. Nous avons utilisé ElephantSQL (<https://www.elephantsql.com/>), qui nous donne accès à une petite base de données avec des limitations telles que le nombre de connexions simultanées à la base de données, limité à 5 pour l'offre gratuite. Nous avons également utilisé le package Dotenv pour stocker toutes les informations sensibles.

8.3 Wireframe

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-18>

9.2 Secrets éventuels pour vos API ou base de données

Étant donné que notre API utilise le package Dotenv pour stocker toutes les informations sensibles, le fichier .env est fourni.

Le script pour la création de la base de données est aussi fourni.

Tout est spécifié dans le README de notre projet.

9.3 Documentation de votre API

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

URI	Méthode HTTP	Opérations
scores/	GET	Récupérer les scores de tous les utilisateurs dans l'ordre décroissant de score.
scores/:id	PUT	Crée ou mettre à jour le score d'un utilisateur (JWT obligatoire) doit être fourni le score
auths/register/	POST	Cree un compte utilisateur doit être fourni le nom d'utilisateur et le mot de passe (renvoie un token)
Auths/login/	POST	Se connecter doit être fourni un nom d'utilisateur indique un mot de passe (renvoie un token)
/collectibles/:id	GET	Récupérer le nombre de pièces qu'un utilisateur connecté possède. (JWT obligatoire)
/collectibles/add/:id	PUT	Créer ou mettre à jour le nombre de pièces d'un utilisateur connecté après chaque partie. (JWT obligatoire)
/shop/getskins	GET	Retourne la liste de tous les skins disponibles dans le shop dans l'ordre croissant de leur prix
/shop/getuserskins	GET	Retourne la liste de tous les skins d'un utilisateur
/shop/addskin	PUT	Ajoute un skin dans la liste des thèmes qu'un utilisateur possède
/shop/getthemes	GET	Retourne la liste de tous les thèmes disponibles dans le shop dans l'ordre croissant de leur prix
/shop/getuserthemes	GET	Retourne la liste de tous les thèmes d'un utilisateur
/shop/addtheme	PUT	Ajoute un thème dans la liste des thèmes qu'un utilisateur possède

- Les requêtes HTTP se trouvent dans :
/api/REST Client
- Les requêtes HTTP pour les pièces se trouvent dans :
/api/REST Client/collectibles.http
- Les requêtes HTTP pour le shop se trouve dans :
/api/REST Client/shops.http

9.4 Déploiement de vos applications

- Frontend : <https://e-vinci.github.io/web2-2023-project-group-18/>
- API: <https://santafall.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
api/routes/auths.js	Professeur JavaScript	https://github.com/e-vinci/jwt-api-boilerplate/blob/main/routes/auths.js	Code fourni dans le boilerplate répondant aux besoins
api/models/auths.js	Professeur JavaScript	https://github.com/e-vinci/jwt-api-boilerplate/blob/main/models/users.js	Code fourni dans le boilerplate répondant aux besoins
front-end/src/pages/HomePage.js	ChatGPT	Lorsque demandé « Donne-moi une fonction qui permet de me dire si je suis sous mobile ou non en JavaScript »	Code générique
front-end/src/Components/Game/GameScene.js	Emanuele Feronato	https://www.emanueleferonato.com/2019/08/23/the-basics-of-infinite-terrain-generation-for-a-horizontal-endless-runner-putting-all-together-and-adding-a-car/	Pas de réutilisation pure mais grande inspiration du code pour la génération du sol et des obstacles.

10 Analyse des résultats par le groupe

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

Au début du projet, nous nous sommes fixé des cas d'utilisation en fonction de nos idées et de nos ambitions pour réaliser notre projet. Chaque membre a eu un nombre de cas d'utilisation respectif. Nous avons tous progressé et résolu ceux-ci tout au long du projet.

Nous voici arrivés à la fin et nous avons réussi à compléter et résoudre tous les cas d'utilisation différents. Le jeu est au goût des attentes que nous avons au lancement du projet Web.

Chaque semaine, chacun des membres du groupe terminait un cas d'utilisation qui lui avait été assigné, et nous avons réussi à bien gérer le temps mis à disposition pour le projet.

10.2 **Audit ergonomique de votre projet**

Nous avons pensé notre jeu de manière à ce qu'il respecte le plus possible les règles d'ergonomie vues au cours. Ayant développé un jeu, de nombreuses règles ne s'appliquent pas à nous, mais voici deux exemples d'optimisation et de réflexion auxquels nous avons pensé lors de la création de notre projet :

- Où que vous soyez sur notre site, n'importe quelle autre page est accessible en maximum 3 clics.
- Lorsqu'un joueur connecté perd sa partie, les deux choix qui lui sont offerts sont de recommencer une partie ou d'aller vers la boutique. En revanche, lorsqu'un joueur non connecté perd sa partie, les choix sont de recommencer la partie ou de retourner à l'accueil.

En supprimant l'option de retourner à l'accueil du site, nous créons une boucle dans laquelle un utilisateur connecté restera un maximum de temps sur notre site. D'un autre côté, un utilisateur non connecté a le choix entre rejouer une partie ou retourner à l'accueil, endroit où il pourra créer un compte et profiter d'autres options de notre jeu, telles que la boutique.

En ce qui concerne les lois GDPR, au moment de l'enregistrement d'un nouvel utilisateur sur notre site, ce dernier doit accepter notre politique de confidentialité pour pouvoir créer un compte. Nous ne conservons aucune information d'un utilisateur non connecté pour éviter de violer les lois RGPD.

10.3 **Difficultés techniques rencontrées**

Au cours de l'avancement du projet, nous avons tous été confrontés à diverses difficultés, tant au niveau du code que de la compréhension de nouveaux outils.

Tout d'abord, la transition vers la création d'un jeu en 2D avec Phaser.js a été un défi majeur pour l'ensemble du groupe. La disponibilité limitée de ressources pour ce langage de programmation a rendu la tâche ardue. Appréhender la physique, assimiler la logique d'implémentation et comprendre les particularités propres à ce langage ont exigé un investissement important en heures de travail afin d'obtenir un résultat fonctionnel.

Ensuite, le déploiement de l'API sur Azure a été une étape difficile tout au long du projet, en particulier lors de l'utilisation de VS Code. L'assistance du professeur a été cruciale pour cette partie du projet, car la manipulation des envois de fichiers vers le serveur s'est avérée complexe à maîtriser.

Troisièmement, notre choix d'une authentification Stateless, bien que présentant plusieurs avantages, a été complexe à assimiler en raison de l'utilisation de JWT et des "tokens".

Enfin, la partie HMI et JWT nous a permis de comprendre l'importance des promesses et des "tokens" dans une application à une SPA. Ces défis ont constitué des apprentissages essentiels dans la maîtrise de ces concepts pour notre projet.

10.4 Conseils pour appliquer cette technologie

L'expérience acquise lors de notre projet a révélé que la plupart des défis étaient liés à l'adoption de nouvelles technologies, telles que l'utilisation de l'API node-postgres et du Framework Phaser pour le développement du jeu. Une leçon clé que nous aurions aimé connaître avant le début du projet concerne le choix au préalable des technologies. La décision de choisir les technologies en amont nous aurait permis d'explorer la documentation de base de chaque technologie et d'acquérir une compréhension préalable du fonctionnement de ces outils. Par exemple, avec Phaser, la découverte tardive des options de physique disponibles a grandement ralenti la conception de notre jeu.

Nos recommandations se focalisent sur une communication efficace des modifications apportées au code, assurant ainsi que tous les membres du projet soient informés des modifications apportées à celui-ci. De plus, une répartition équilibrée des tâches a été déterminante pour notre progression continue, garantissant que chacun ait un rôle clair et contribue activement au projet.

Pour finir, voici un site qui s'est avéré précieux lors de la réalisation du jeu avec Phaser :

- <https://www.emanueleferonato.com/>

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

Nous avons trouvé l'expérience collaborative très enrichissante, car elle nous a permis d'apprendre à travailler ensemble sur un projet concret. Cette collaboration a favorisé l'entraide et l'échange de conseils sur les méthodes de codage et l'organisation nécessaire pour mener à bien ce type de projet.

Chaque membre du groupe a joué un rôle significatif sur un cas d'utilisation important, offrant ainsi à chacun l'opportunité d'être un acteur majeur du projet à tour de rôle. La flexibilité temporelle a été une composante clé, permettant à chacun de travailler à son propre rythme et à des moments qui lui convenaient.

Nous avons tous mis tout en œuvre pour atteindre la meilleure version du projet possible, en investissant au maximum nos compétences. Cette expérience a clairement démontré l'importance cruciale de la collaboration au sein d'une équipe. L'atmosphère positive et collaborative au sein du groupe a permis de résoudre rapidement certains problèmes.

Dans l'ensemble, cette expérience collaborative a été valorisante, nous permettant de pleinement apprécier le développement conjoint de notre projet.

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

Les aspects à améliorer résident dans une communication plus approfondie sur nos avancées respectives, privilégiant des réunions régulières, qu'elles soient physiques ou virtuelles, afin de favoriser l'entraide lors des blocages.

De plus, une utilisation plus efficace des outils de collaboration tels que Git/GitHub est essentielle. Cela implique une structuration améliorée des "pull requests" accompagnées de commentaires plus précis, permettant aux "reviewers" de véritablement comprendre et valider le code avant d'être fusionné à la branche principale. En ayant plusieurs "reviewers" pour chaque "pull request", nous pourrions garantir une qualité et une compréhension accrues du code de tous, bénéficiant ainsi à l'ensemble de l'équipe.

L'intégration d'outils de planification serait également bénéfique pour assurer une progression continue de chaque membre, sans dépendre de la finition d'une tâche par un autre membre. Ces outils faciliteraient la répartition et la coordination des travaux, favorisant ainsi une avancée fluide et synchronisée du projet.

11 Analyses individuelles des résultats

12 Présentation vidéo

Lien vers la vidéo YouTube :

- <https://youtu.be/HUviPRoNQSU>

