Ergonomie & développement d’une SPA animée

**SEAL RESCUE**



|  |  |
| --- | --- |
| **Auteur 1** | Kimberley Blomme |
| **Auteur 2** | Alicia Boltryk |
| **Auteur 3** | Omar Jetti |
| **Auteur 4** | Antoine Putmans |
| **Auteur 5** | Guilherme Yamamoto |
| **Date** | 18.12.2022 |
| **Référence** | WEB2-2022-PROJECT-GROUP-05 |
| **Version** | 1.0 |

Contents

[1 Consignes et évaluations 3](#_Toc115852290)

[1.1 Consignes générales 3](#_Toc115852291)

[1.1.1 Création des groupes sur le site du cours 3](#_Toc115852292)

[1.1.2 Création d’un groupe sur GitHub Classroom et du web repo associé 4](#_Toc115852293)

[1.1.3 Projet 5](#_Toc115852294)

[1.2 JavaScript & Node.js : consignes techniques, timing et évaluations 6](#_Toc115852295)

[1.3 Ergonomie : consignes techniques, timing et évaluations 11](#_Toc115852296)

[2 Objectif du projet 11](#_Toc115852297)

[3 Mind map du projet 12](#_Toc115852298)

[4 Persona 12](#_Toc115852299)

[5 Axiomes de Morville 13](#_Toc115852300)

[6 Planning des tâches et cas d’utilisation 13](#_Toc115852301)

[7 Besoins techniques 15](#_Toc115852302)

[7.1 Système 15](#_Toc115852303)

[7.2 Frontend 15](#_Toc115852304)

[7.3 API 16](#_Toc115852305)

[8 Choix technologiques 17](#_Toc115852306)

[8.1 Frontend 17](#_Toc115852307)

[8.2 RESTful API 17](#_Toc115852308)

[8.3 Wireframe 17](#_Toc115852309)

[9 Conception & Implémentation 17](#_Toc115852310)

[9.1 Code repositories 17](#_Toc115852311)

[9.2 Secrets éventuels pour vos API ou base de données 18](#_Toc115852312)

[9.3 Documentation de votre API 18](#_Toc115852313)

[9.4 Déploiement de vos applications 19](#_Toc115852314)

[9.5 Code réutilisé 19](#_Toc115852315)

[10 Analyse des résultats par le groupe 20](#_Toc115852316)

[10.1 Evaluation du résultat par rapport au planning des tâches et des cas d’utilisation 20](#_Toc115852317)

[10.2 Audit ergonomique de votre projet 20](#_Toc115852318)

[10.3 Difficultés techniques rencontrées 20](#_Toc115852319)

[10.4 Conseils pour appliquer cette technologie 20](#_Toc115852320)

[10.5 Quels sont les points positifs à la manière dont s’est déroulée la collaboration au sein du groupe ? 21](#_Toc115852321)

[10.6 Quels sont les points qui seraient à améliorer pour de futures collaborations ? 21](#_Toc115852322)

[11 Analyses individuelles des résultats 21](#_Toc115852323)

[12 Présentation vidéo 21](#_Toc115852324)

[13 Revues de projets par les pairs 22](#_Toc115852325)

# Consignes et évaluations

## Consignes générales

### Création des groupes sur le site du cours

Veuillez former un groupe de 4 ou 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 4 ou 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 (23/10), 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 4-5 étudiants remplis, il restera maximum 3 étudiants non liés à un projet. Si nécessaire un ou plusieurs groupes de 3 étudiants seront créés.

### 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.

**Veuillez passer à cette étape qu’une fois votre groupe déjà finalisé sur le site du cours.**

#### Création de l’équipe associée à un projet

Veuillez identifier le membre qui créera votre équipe sur GitHub.

Ce membre accédera à l’assignement via : <https://classroom.github.com/a/7av06CzK>

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**.

Graphical user interface, text, application

Description automatically generated

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

Après un refresh de la page qui suit, voilà ce qui apparait :

Graphical user interface, text, application

Description automatically generated

Un web repository a été créé pour votre équipe.

#### Joindre une équipe existante

Une fois l’équipe d’un projet créée, les autres membres accéderont aussi à l’assignement via : <https://classroom.github.com/a/7av06CzK>.

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**.

Graphical user interface, text, application, email

Description automatically generated

Si vous préférez, vous pouvez visualiser cette vidéo qui montre [comment Joindre un GitHub Classroom Group Assignment](https://youtu.be/3TIVVCkne_0).

### 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

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

| **Tâche** | **Compétences** | **Critères** | **Dead-line** | **Pt** | **Consignes** |
| --- | --- | --- | --- | --- | --- |
| Objectif du projet | C7) Documenter et présenter en vidéo le développement d'une SPA |  | **23/10** |  | 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 :   * Le nom du projet : Projet N°X : Nom de votre projet * le champs « Description ».   Discuter de votre objectif avec un enseignant et assurer vous que cet objectif soit validé avant d’aller plus loin dans votre projet. |
| Planning des tâches et cas d’utilisation | C7) |  | **27/11** |  | Décrire le planning des tâches et cas d’utilisation selon les instructions données au §6.  Présenter votre planning à un enseignant, afin qu’il puisse vous aider à bien prioriser les tâches. |
| Indiquer l’URL de votre code repository | C7) |  | **27/11** |  | 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>.  Plus d’information aux §1.1.2 et §9.1. |
| Choix technologiques | C7) |  | **04/12** |  | Compléter le §8.  Discuter de vos choix technologiques avec un enseignant. |
| Rapports individuels d’activités | C7) | Rapports de qualité *Indicateurs : formulation de qualité, analyse de qualité, respect des consignes* | **18/12** | **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.  Tout formulaire hebdomadaire non complété amenera à une pénalité individuelle de 0.5 point. |
| Soumission du rapport de groupe | C7) | Idem | **18/12** | **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.  Effacer toutes les consignes mises *en grisé* dans ce document avant de soumettre ce rapport sur Moodle. |
| Soumission de la vidéo | C7) | Vidéo de qualité  *Indicateurs : présentation du projet de qualité, analyse de qualité, respect des consignes* | **18/12** | **2** | Présenter votre projet selon les exigences du §12. |
| Soumission du code du frontend | C1) Créer une IHM interactive, moderne & esthétique  Optionnel : C4) Intégrer l'authentification, l'autorisation et les sessions d'utilisateurs au sein d'une SPA | Qualité de l’IHM produite  *Indicateurs : esthétique, fonctionnel, codage de qualité, respect des consignes, ambitieux & original* | **18/12** | **5** | Réaliser un frontend et un backend de Qualité : Code bien structuré, UI et UX de qualité  Être ambitieux et original.  Démontrer une appropriation personnelle du code (via commentaires dans le code, discussion lors des cours...).  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. |
|  | C3) Créer une SPA intégrant une IHM & un web service | Qualité de l’intégration du service web à l’IHM  *Indicateurs : fonctionnel, codage de qualité, respect des consignes* | **18/12** | **2** |
| Soumission du code du backend | C2) Créer un service web de base  Optionnel : C4) Intégrer l'authentification, l'autorisation et les sessions d'utilisateurs au sein d'une SPA | Qualité du web service produit  *Indicateurs : fonctionnel, codage de qualité, respect des consignes, ambitieux & original* | **18/12** | **4** |
| Utilisation d’une librairie pour des animations ou un jeu  Utilisation d’une librairie pour le service web | C6) Intégrer au développement d'une SPA des technologies non vues en cours | Intégration de librairies non vues en cours  *Indicateurs : utilisation d’une librairie pour l’IHM, utilisation d’une librairie pour le service web* | **18/12** | **2** |  |
| Déploiement tant de votre frontend que backend | C5) Déployer une SPA sur le cloud | Déploiement de la SPA sur le cloud  *Indicateurs : fonctionnel, performances de chargement acceptables* | **18/12** | **2** |  |
| Réaliser un minimum de 5 revues sur le site web | C8) Analyser le développement de SPA faites par des pairs | Revues de projets compréhensibles & constructives  *Indicateur : présence d’un minimum de 5 revues* | **Avant examen de 1ère 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 §13. |
|  | **TOTAL POINTS** |  |  | **20** | Il est à noter que des membres d’un même groupe pourront être côtés 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.  Les étudiants n’ayant pas participés activement au projet recevront d’office une lourde pénalité au niveau de leurs points, voire un 0/20.  Les étudiants n’ayant pas réalisé au moins un use case significatif seront considérés inactifs. |

## 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 | **23/10** |  | 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 | **13/11** | 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 | **20/11** | 4 | Construire les wireframes détaillés de votre application. |
|  | Analyse des résultats et rapport associé | **18/12** | 2 | Auditez votre projet et vérifiez le respect des règles GDPR. |
|  | Présentation vidéo | **18/12** | 8 | Présenter votre projet en intégrant l’expérience utilisateur. |
|  | **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

Seal Rescue est un jeu de style Runner en 3D sur navigateur web. Nous avons pour but de sensibiliser le public à la cause animale, notamment les espèces en voie de disparition comme les phoques.

Accessible et intuitif, le jeu se veut tout-public et familial afin que son message puisse parvenir à la plus grande audience possible.

Le personnage joué est un phoque devant éviter les morceaux de glace fondue et des déchets tel que des tonneaux ou des bouteilles d’eau sur trois pistes.

En effet, au-delà de l’aspect amusant et compétitif du jeu, de nombreuses fonctionnalités sont proposées au joueur. Parmi elles, on y retrouve des live-camera où des animaux vivent (ou séjournent), des messages de sensibilisation ainsi que des appels aux dons pour des associations de conservation des espèces menacées.

Nous avons décidé de soutenir cette cause car nous estimons que nous pouvons apporter notre pierre à l’édifice par l’intermédiaire de cette application. À ce jour, les enjeux climatiques sont critiques, et nous avons la ferme volonté de contribuer à ce combat.

# Mind map du projet

*Une image contenant texte, équipement électronique

Description générée automatiquement*

# Persona

###### Premier persona

###### Insertion de l’image...Profil

Prénom : Célestin Gabin

Nom de famille : Dubois

Age : 27

Emploi : Jeune artiste au chômage qui ne trouve pas de travail car il est trop occupé entre grèves, manifestations et séjours sur Twitter.

###### Personnalité

Gauchiste aguerri, il hait les politiciens et les multinationales, mais il fait la file tous les matins chez Starbucks en scrollant sur son iPhone 14 Pro Max. Il signe des pétitions mais ne fait jamais de dons.

Dans quelques années, sa phrase fétiche sera : “C’était mieux avant !”

###### Objectifs

• Sauver la planète tout en restant sur son canapé à regarder des documentaires sur Arte

• Profiter de la vie

• Souvent contrarié et déçu par le service clientèle, veut parler au manager !

###### Interaction avec notre jeu

En buvant son café à 4€, il vapote son CBD bio tout en jouant à un petit jeu d’écolo bobo pour se donner bonne conscience.

###### Second persona

Une image contenant lit, intérieur, personne, bébé

Description générée automatiquement

###### Profil

Prénom : Timothée

Nom de famille : Nicolas

Age : 7 ans

###### Personnalité

Timothée, jeune gamer, adore les Rainbow Friends. Dès qu’il le peut, il gratte le téléphone de sa mère pour installer dix mille jeux et applications du Play Store et regarde des vidéos de gaming sur YouTube.

###### Objectifs

• Devenir Youtubeur

• Ne pas se faire punir

• Regarder discrètement des vidéos qu’il n'a pas le droit de regarder sur YouTube

• Dormir après 22h

###### Interaction avec notre jeu

Une fois qu’il a gagné ses premières pièces et acheté son premier skin, il ne pourra plus s’arrêter de jouer tant qu’il n’aura pas tous les magnifiques skins du jeu...

# Axiomes de Morville

UTILE : Seal Rescue est un jeu. Il apportera de l’amusement dans la vie de ses utilisateurs en répondant aux besoins de sensibilisation vis-à-vis du climat et de l’extinction des espèces animales.

UTILISABLE : Il sera facile à prendre en main car le type de jeu Runner est très courant et codifié. Nous pourrons donc réutiliser les mêmes codes afin de le rendre intuitif.

DESIRABLE : Les graphismes seront faits d’un design épuré et coloré de couleurs pastel et douce pour offrir une expérience visuelle agréable.

CREDIBLE : La simplicité du design suivra la tendance des sites les plus consultés (Facebook, YouTube, Google, …). On gardera des menus du même design épuré et des mêmes coloris que le jeu avec d’offrir une cohérence visuelle pour l’utilisateur.

ACCESSIBLE : Le jeu sera disponible sur le navigateur. Il sera donc facile d'accès à toute personne possédant un ordinateur. Il sera gratuit pour être accessible à tous les publics.

TROUVABLE : Le jeu sera potentiellement hébergé sur une plateforme en ligne.

VALABLE : Le jeu aura plus à offrir que de l’amusement. Il sera également un outil de sensibilisation. Des diffusions de différents animaux en voie d'extinction seront directement disponibles dans notre jeu.

# Planning des tâches et cas d’utilisation

URL vers votre GitHub Project public :

<https://github.com/e-vinci/web2-2022-project-group-05>

# Besoins techniques

## 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.*

## 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.

## 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.

# Choix technologiques*.*

## Frontend

***Seal Rescue*** étant un jeu 3D, nous voulions initialement utiliser Three.js, comme proposer dans le cadre de ce cours. Lorsque nous nous sommes renseignés, nous sommes vite tombées face aux problèmes que nous allions rencontrer en utilisant ce moteur 3D. Three.js est une librairie 3D ne contenant aucun *game engine*. Il allait donc nous être très difficile de gérer tous les systèmes de collision ainsi que d’avoir un moteur physique pour simuler la réalité.

Nos recherches nous ont donc amenés sur BabylonJS. Il semblait répondre à tout ce que dont nous avions besoin en présentant assez de documentation et un environnement de travailler très évolué pour nous permettre de se l’approprier rapidement. Ceci-dit, malgré tous les supports offerts, nous avons vite réalisé les difficultés qui viennent avec la création d’un jeu en 3D.

Premièrement, la création du modèle en 3D pour notre personnage principal. Heureusement, l’un des membres de notre groupe avait déjà travaillé avec Blender et a pu nous faire une magnifique et sublime otarie (nommée Otto) en tant que personnage principal de ***Seal Rescue***. La difficulté fut ensuite de réussir à importer correctement ce modèle, réduire sa taille pour ne pas rendre le jeu trop lourd, et enfin de l’animer. De plus, il fallait créer des mouvements fluides pour rendre Otto plus réaliste. Ensuite, la création d’obstacles et l’interaction avec Otto nous a aussi posé un vrai défi. Nous avons décidé de faire avancer les obstacles vers l’écran au lieu de faire avancer toute la scène. Mais le jeu 3D n’est qu’une partie du site, il a aussi fallu faire toutes les interfaces en utilisant *Tailwind* car il semblait offrir plus de possibilités que Bootstrap, Vanilla JS.

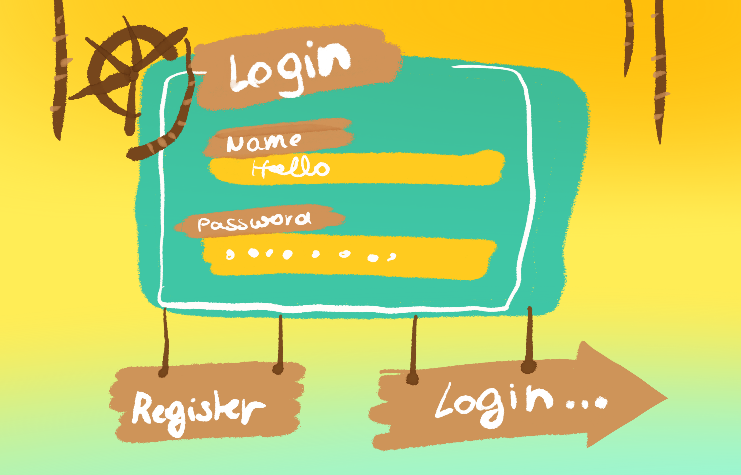
## RESTful API

Initialement, nous avions pensé à permettre à l’utilisateur d’avoir accès à des live streams d’animaux correspondant au skin de l’otarie sélectionné. Cela dit, nous n’avions trouvé qu’un seul site permettant d’utiliser ses API, mais il demandait un abonnement payant. Nous avons du coup pensé à une API extérieure qui nous fournirait des informations sur des espèces en danger. Nous voulions initialement utiliser Red List API venant de “International Union for Conservation of Nature’s Red List of Threatened Species” qui est une des plus grosses sources mondiales d’informations sur l’état d’extinction des animaux, plantes et champignons. Nous leur avons donc introduit une demande pour obtenir un token. Ceux-ci nous ont gentiment envoyer paître. Nous avons donc décidé d’utiliser API-Ninjas, plus précisément leur Animals API qui contient plein de faits intéressants sur des animaux. Nous avons utilisé une liste d’espèces qui seront tirées au hasard pour récupérer leurs plus grosses menaces pour l’afficher dans le footer.

## Wireframe

Les wireframes ont été détaillés dans le rapport rendu pour le cours d’ergonomie de Patrick Kibale. Ceux-ci sont trouvables sur notre repository :

<https://github.com/e-vinci/web2-2022-project-group-05/tree/main/ergonomics/wireframes>





# Conception & Implémentation

## Code repositories

URL pour le web repository public associé à votre projet :

<https://github.com/e-vinci/web2-2022-project-group-05>

## Secrets éventuels pour vos API ou base de données

## Documentation de votre API

*Veuillez documenter les opérations de votre API, soit à l'aide de tableaux à donner ci-dessous, comme ceux vu dans le cours, soit en référençant la documentation qui aurait été générée à l'aide d'outils (Swagger par exemple). Votre fichier README de votre projet doit indiquer l'endroit où se situe la documentation de votre API.*

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

|  |  |
| --- | --- |
| ROUTE USERS | |
| GET /users | Retourne tous les utilisateurs |
| GET /users/user?username=user | récupère l'utilisateur nommé "user" |
| PATCH /users/balance?username=user | modifie la balance de l'utilisateur user |
| PATCH /users/highscore?username=user | modifie le highscore de l'utilisateur user |
| PATCH /users/skins?username=user | ajoute un skin dans la liste de skin de l'utilisateur user |
| PATCH /users/currentSkin?username=user | modifie le skin actuel dans la liste de skin de l'utilisateur user  les routes pour les users |
| ROUTE SKINS | |
| GET /skins | recupère tous les skins |
| GET /skins/skins?order=price | récupère les skin par ordre de prix |
| GET /skins/skinName?name=panda | récupère le skin panda |
| ROUTE AUTH | |
| POST /auths/register | enregistre le user |
| POST /auths/login | connecte le user |
| GET /auths/logout | deconnecte le user |
| ROUTE ANIMALS | |
| GET /animals | récupère tous les animaux |

* Requêtes HTTP se trouvent dans : <https://github.com/e-vinci/web2-2022-project-group-05/tree/main/api/REST%20Client>

## Déploiement de vos applications

* URL de votre frontend déployé : <https://e-vinci.github.io/SealRescue-Frontend/>
* URL de votre RESTful API déployée : <https://sealrescue-api2.azurewebsites.net>

# Analyse des résultats par le groupe

## Evaluation du résultat par rapport au planning des tâches et des cas d’utilisation

Voici les principaux objectifs que nous avons atteint :

Le premier et le plus important de nos objectifs était d’avoir un jeu runner 3D qui fonctionne sans crasher et sans memory leak. Cela dit, nous avons dû faire des sacrifices pour des questions d’efficacité et par manque d’optimisation, tel que le rendu et l’interaction entre les objets avec l’eau.

Le second est la présence d’un store avec la possibilité d’acheter et de sélectionner un skin.

Enfin, nous voulions un système de ranking pour pouvoir comparer les scores entre les utilisateurs.

Pour ce qui est des objectifs inachevés :

D’abord, nous avons eu du mal à obtenir les mêmes rendus que ceux prévus par nos wireframes.

De plus, nous voulions avoir une difficulté progressive dans le jeu.

Enfin, nous souhaitions un système de score basé sur le temps de survie plutôt que du nombre d’obstacles franchis.

Nous pensons que tout cela aurait été réglé si nous avions eu plus de temps pour le projet.

## Audit ergonomique de votre projet

Nous avons suivi de nombreuses guidelines présentées par : « *U.S. Department of Health & Human Services* ».

Nous ne fournissons que du contenu utile. Toutes les options proposées sont nécessaires à l’utilisation de notre application

Nous ne faisons apparaitre aucune fenêtre non sollicitée, aucun popup.

Nous avons suivi la loi de Fitts pour la création de la page principale. Nous demandons un minimum possible à l’utilisateur en diminuant au maximum sa charge de travail pour lancer le jeu. Il lui suffit de se connecter s’il le souhaite, et il peut directement jouer.

Nous offrons aussi une page de chargement car ***Seal Rescue*** peut mettre un certain temps selon l’ordinateur et la vitesse de connexion de l’utilisateur.

Nous n’avons aucun défilement horizontal.

***Seal Rescue*** respecte la loi de Hick-Hyman : il comporte un minimum d'informations sur la page d'accueil pour aider l’utilisateur à faire des choix dans sa navigation.

Les règles de GDPR sont respectées par le fait que la demande de soit dissocies des autres options d’enregistrement. Nous demandons une acception active de la part de l’utilisateur pour accepter nos conditions. Cela dit, nous avons conscience qu’il n’est pas possible de retirer son consentement. Cela pourrait se régler en créant une page utilisateur en lui offrant la possibilité de supprimer son compte mais nous n’avons malheureusement pas eu le temps de faire cela.

## Difficultés techniques rencontrées

Ayant décidé de nous lancer dans un jeu 3D, il nous a fallu apprendre de nouveaux concepts propres à *Babylonjs* pour gérer les animations des objets à l’écran, notamment maîtriser les appels asynchrones et comprendre quand est-ce qu’ils étaient nécessaires. De plus, il nous a fallu apprendre à gérer les imports d’objets *glb* qui étaient nos modèles 3D. Cela a été très difficile car il n’y avait pas de support du cours pour nous aider, nous avons donc dû chercher énormément de notre côté. Cela nous a contraint à toujours avoir une partie de notre équipe dédiée à travailler sur *BabylonJS*.

De plus, le déploiement d’un jeu 3D semble pose beaucoup de difficultés pour *webpack.* Nous avons dû faire diverses modifications dans le *webpack.config*.

Ensuite, nous avions décidé d’utiliser *Tailwind* car nous pensions que cela serait plus flexible que VanillaJS et Bootstrap, mais cela n’était peut-être pas le cas finalement. Surtout pour la difficulté que nous cela nous a posé à apprendre.

Enfin, le déploiement est ce qui a posé le plus de problèmes. Nous avions initialement un problème d’origine lié aux cors. Mais le plus gros souci est venu avec les cookies. Nous ne comprenions pas pourquoi les cookies fonctionnaient lorsque nous travaillions en localhost mais pas pendant le déploiement. Lorsque nous avons enfin réglé le problème, nous avons constaté des énormes problèmes de sécurités. Malheureusement, cela a été constaté trop tard pour pouvoir être résolu avant le déploiement.

## Conseils pour appliquer cette technologie

Nous suggérons de ne pas déployer votre projet pendant la dernière semaine, cela nous a posé beaucoup de problèmes juste avant la deadline.

Nous avons entamé ce projet en 3D car nous avons quelqu’un dans l’équipe qui en avait déjà créé des jeux en 3D et qui est familière avec Blender. Nous aurions probablement fait un jeu en 2D sinon. Nous pensons cela dit que cela est réalisable même sans ces précédents acquis mais il vous faudra probablement une personne exclusivement dédiée à cette tâche.

Il vous faudrait également bien répartir les tâches. Au départ, nous avions tous commencé à apprendre Babylonjs mais c’était probablement une erreur, nous aurions peut-être mieux fait d’alloué une ou deux personnes exclusivement à cela et le reste s’occuper des autres aspects du projet.

Il vous faudrait tirer au mieux des gestionnaires de taches car c’est quelque chose que nous n’avons pas suivi à la lettre et cela nous a parfois amené à être plusieurs a travaillé sur les mêmes taches.

* <https://stackoverflow.com/>
* <https://doc.babylonjs.com/>
* <https://www.youtube.com/@BabylonJSEngine>
* <https://webpack.js.org/concepts/>
* <https://developer.mozilla.org/en-US/>

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

Nous étions toujours occupe à faire quelque chose. Il n’a eu personne du groupe qui a refusé de collaborer et tout s’est passé dans la bonne entente. La charge de travail était très grande mais cela ne nous pas découragé et nous avancions petit à petit. Nous avons énormément communiqué et tout le monde était très enthousiaste pour apprendre de nouvelles choses.

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

Nous aurions dû plus utiliser les outils GitHub pour traquer l’avancement de notre projet. Nous ne pensions à mettre constamment à jour les issues dès que nous faisions des changements. Heureusement nous avions une très bonne cohésion d’équipe, nous discutions donc souvent de ce qui était à faire et repartions les taches entre nous selon les préférences de chacun.

Plus jamais nous ne déploierons la dernière semaine de la remise du projet. Cela a vraiment été catastrophique.

# Analyses individuelles des résultats

Les feedbacks hebdomadaires ont été fait sur <https://teammatesv4.appspot.com>.

# Présentation vidéo

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