

# Ergonomie & développement d'une SPA animée

AimClicker

<b>Auteur 1</b>	Avram Andrei
<b>Auteur 2</b>	Duraku Fisnik
<b>Auteur 3</b>	Esakwa Nkongo Nathan
<b>Auteur 4</b>	Rzine Hamza
<b>Auteur 5</b>	Michaël Van Aerschot
<b>Date</b>	12.12.2022
<b>Référence</b>	WEB2-2022-PROJECT-GROUP-01
<b>Version</b>	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.....	5
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.....	13
7.1	Système.....	13
7.2	Frontend .....	13
7.3	API.....	14
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 .....	15
9.4	Déploiement de vos applications .....	16
9.5	Code réutilisé.....	16
10	Analyse des résultats par le groupe .....	17
10.1	Evaluation du résultat par rapport au planning des tâches et des cas d'utilisation .....	17
10.2	Audit ergonomique de votre projet.....	17

10.3	Difficultés techniques rencontrées.....	17
10.4	Conseils pour appliquer cette technologie .....	18
10.5	Quels sont les points positifs à la manière dont s'est déroulée la collaboration au sein du groupe ? .....	19
10.6	Quels sont les points qui seraient à améliorer pour de futures collaborations ? .....	19
11	Analyses individuelles des résultats.....	19
12	Présentation vidéo .....	19
13	Revue de projets par les pairs.....	19

# 1 Consignes et évaluations


## 1.1 Consignes générales

### 1.1.1 Création des groupes sur le site du cours

Veillez 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.

### 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/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**.

e-vinci-web2

Accept the group assignment —  
**web2-2022-project**

Before you can accept this assignment, you must create or join a team. Be sure to select the correct team as you won't be able to change this later.

Create a new team

group-04

+ Create team

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

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

You're ready to go —  
**group-04**

You accepted the assignment, **web2-2022-project**.

Your team's assignment repository has been created:

<https://github.com/e-vinci/web2-2022-project-group-04>

We've configured the repository associated with this assignment ([update](#)).

Note: You may receive an email invitation to join [e-vinci](#) on your behalf. No further action is necessary.

Un web repository a été créé pour votre équipe.

### 1.1.2.2 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>.

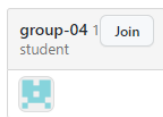
Ces membres joindront l'équipe existante en cliquant sur **Join** au sein de la bonne équipe. Par exemple, pour les membres du **Projet N°4**, ils cliqueront sur **Join** dans l'équipe **group-04**.

e-vinci-web2

Accept the group assignment —  
**web2-2022-project**

Before you can accept this assignment, you must create or join a team. Be sure to select the correct team as you won't be able to change this later.

Join an existing team



OR Create a new team

Create a new team

Si vous préférez, 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	Dead-line	Pt	Consignes
Objectif du projet	C7) Documenter et présenter en vidéo le développement d'une SPA		23/10		<p>Donnez un nom à votre projet et décrire l'objectif de votre projet au §2 de ce document ainsi que sur <a href="https://e-vinci.github.io/web2/project-page">https://e-vinci.github.io/web2/project-page</a>, complétez :</p> <ul style="list-style-type: none"><li>- Le nom du projet : Projet N°X : Nom de votre projet</li><li>- le champs « Description ».</li></ul> <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	C7)		27/11		<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	C7)		27/11		<p>Votre code doit être accessible par tout le monde via un web repository public qui vous sera</p>

Tâche	Compétences	Critères	Dead- line	Pt	Consignes
					<p>assigné par GitHub Classroom. Cela permettra notamment aux enseignants de suivre vos avancées tout au long de votre projet. Veuillez indiquer votre URL sur <a href="https://e-vinci.github.io/web2/project-page">https://e-vinci.github.io/web2/project-page</a>.</p> <p>Plus d'information aux §1.1.2 et §9.1.</p>
Choix technologiques	C7)		04/12		<p>Compléter le §8.</p> <p>Discuter de vos choix technologiques avec un enseignant.</p>
Rapports individuels d'activités	C7)	Rapports de qualité <i>Indicateurs : formulation de qualité, analyse de qualité, respect des consignes</i>	18/12	1 <b>solo</b>	<p>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.</p> <p>Tout formulaire hebdomadaire non complété amènera à une pénalité individuelle de 0.5 point.</p>
Soumission du rapport de groupe	C7)	Idem	18/12	1	<p>Compléter le §10 ainsi que tous les paragraphes qui n'auraient pas été finalisés de ce document.</p> <p>Soumettre ce document, via Moodle (un devoir sera créé) ainsi que dans le répertoire <b>/report</b> de votre repo.</p>

Tâche	Compétences	Critères	Dead-line	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	C7)	Vidéo de qualité <i>Indicateurs : présentation du projet de qualité, analyse de qualité, respect des consignes</i>	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 <i>Indicateurs : esthétique, fonctionnel, codage de qualité, respect des consignes, ambitieux &amp; original</i>	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.
	C3) Créer une SPA intégrant une IHM & un web service	Qualité de l'intégration du service web à l'IHM <i>Indicateurs : fonctionnel, codage de qualité, respect des consignes</i>	18/12	2	NB : votre RESTful API doit être un minimum différent des APIs fournies dans les démos du cours de JS.
Soumission du code du backend	C2) Créer un service web de base  Optionnel : C4) Intégrer	Qualité du web service produit <i>Indicateurs : fonctionnel, codage de qualité, respect des</i>	18/12	4	



Tâche	Compétences	Critères	Dead-line	Pt	Consignes
	l'authentification, l'autorisation et les sessions d'utilisateurs au sein d'une SPA	<i>consignes, ambitieux &amp; original</i>			
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 <i>Indicateurs : utilisation d'une librairie pour l'IHM, utilisation d'une librairie pour le service web</i>	<b>18/12</b>	<b>2</b>	
Déploiement tant de votre frontend que backend	C5) Déployer une SPA sur le cloud	Déploiement de la SPA sur le cloud <i>Indicateurs : fonctionnel, performances de chargement acceptables</i>	<b>18/12</b>	<b>2</b>	
Réaliser un minimum de 5 revues sur le site web	C8) Analyser le développement de SPA faites par des pairs	Revue de projets compréhensibles & constructives <i>Indicateur : présence d'un minimum de 5 revues</i>	<b>Avant examen de 1<sup>ère</sup> session</b>	<b>1 solo</b>	Via <a href="https://e-vinci.github.io/web2/my-reviews-page">https://e-vinci.github.io/web2/my-reviews-page</a> , 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.
	<b>TOTAL POINTS</b>			<b>20</b>	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

Tâche	Compétences	Critères	Dead-line	Pt	Consignes
					<p>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 n'ayant pas participés activement au projet recevront d'office une lourde pénalité au niveau de leurs points, voire un 0/20.</p> <p>Les étudiants n'ayant pas réalisé au moins un use case significatif seront considérés inactifs.</p>

### 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	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.
	<b>TOTAL</b>		<b>20</b>	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 application est un jeu de clic, le joueur peut choisir entre 3 modes de jeu, la partie rapide, la partie classé ou la partie troll. Il peut aussi se connecter afin d'accéder à la partie classée et se mesurer à tout les autres joueurs grâce à l'enregistrement du score et l'affichage des meilleurs

score. L'utilisateur peut aussi visionner les scores de ses ancienne partie classé ainsi que cherché des joueurs par leur pseudo et les demandé en amis.

La partie rapide (quick) peut être joué sans être connecté et offre un menu de personnalisation du temps, de la taille ainsi que de la couleur.

La partie troll est une partie augmentant le difficulté grâce a des formes au comportement inattendu.

Notre application peut tout principalement les joueurs sur ordinateur allant du joueur expérimenté au joueur casual.

Notre application est développée pour être utilisé uniquement sur pc.

Notre objectif est donc d'offrir un jeu ou tout type de joueurs peuvent s'amuser.

### 3 Mind map du projet

### 4 Persona

### 5 Axiomes de Morville

*Veillez décrire chaque axiome de Morville de manière assez complète vis-à-vis de votre application. Ne pas donner de réponse basique mais allez plus loin dans les besoins.*

*Exemple : App de la ville de Bruxelles*

- *Valable : "parce que les gens en ont besoin et ça les aidera"*

*--> Une application aide toujours mais en quoi cela rajoute de la plus-value ?*

*--> En quoi l'application va*

*réellement changer l'expérience utilisateur déjà existante pour ce type de projet ? Et si cela ne change pas, il faudra aussi le justifier. Ne pas recréer la roue n'est pas un défaut en soi, ça s'appelle de l'efficience.*

*Veillez soumettre vos axiomes au sein du répertoire /ergonomics/marketing-view de votre repo.*

## 6 Planning des tâches et cas d'utilisation

- URL vers votre GitHub Project public : <https://github.com/e-vinci/web2-2022-project-group-01.git>

## 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

Pour notre site nous avons utilisé 4 librairies et l'api canvas, une pour le back end et 3 pour le frontend.

Pour le backend nous avons utilisé le package postgres.js afin de faire le lien entre notre application et la base de données déployé sur le site elephantSQL(<https://customer.elephantsql.com/instance>). Ce package a donc été utilisé pour répondre à tous les use case backend ayant besoin des données de la base de données

Pour le frontend nous avons utilisé 3 librairies, gsap, jquery ainsi que fluid-meter.

Tout d'abord nous avons utilisé gsap afin de réaliser des animations dans notre application et de la rendre plus dynamique, elle a été utilisée dans toute les pages de partie ainsi que dans le menu. Elle permet de répondre aux use case Frontend : ajouter des animations, trouver des animations #33

Ensuite nous avons utilisé jquery afin de faciliter le changement de classe lors du clic pour passer de la page login à register. Nous avons utilisé cette librairie afin de changer la manière dont nous changeons de classe bien que nous aurions pu le faire en javascript. Elle permet de répondre à l'usage suivant Frontend : inscription page #6 et Frontend : connexion page #5

Ensuite nous avons utilisé fluid-meter afin d'afficher la progression dans le niveau dans l'utilisateur page de manière animée. Cette librairie permet de répondre aux cas d'usage Frontend : ajouter des animations, trouver des animations #33

Finalement, nous avons utilisé l'API canvas afin de faire apparaître les formes dans les zones de jeu et répondre à tous les cas d'usage faisant apparaître des pages game (troll, quick, ranked)

## 8.1 Frontend

Gsap → <https://greensock.com/gsap/>

Jquery → <https://jquery.com/>

Fluid-meter → <https://www.npmjs.com/package/fluid-meter>

API-canvas → [https://developer.mozilla.org/fr/docs/Web/API/Canvas\\_API](https://developer.mozilla.org/fr/docs/Web/API/Canvas_API)

## 8.2 RESTful API

Postgres.js → <https://www.npmjs.com/package/postgres>

## 8.3 Wireframe

*Veillez créer sur PowerPoint, des wireframes de votre application. Il en faut au minimum 3.*

*Exemple d'un jeu : il y aura au minimum, l'accueil inscription, le jeu, une page de résultat, page contact développeur, etc.*

*Veillez soumettre vos wireframes au sein du répertoire /ergonomics/wireframes de votre repo.*

# 9 Conception & Implémentation

## 9.1 Code repositories

- URL pour le web repository public associé à votre projet : <https://github.com/e-vinci/web2-2022-project-group-01>

## 9.2 Secrets éventuels pour vos API ou base de données

## 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
users/login	POST	READ ONE
users/register	POST	CREATE ONE
users/addScore	POST	CREATE ONE
users/getUser?pseudo=value	GET	READ ALL FILTERED
users/getUserFriends?id=value	GET	READ ALL FILTERED
users/addFriend	POST	CREATE ONE
users/getUsersScore	GET	READ ALL FILTERED
users/getBestUserScore?id=value	GET	READ ALL FILTERED
users/getAllUserScore?id=value	GET	READ ALL FILTERED

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

#### 9.4 Déploiement de vos applications

- URL de votre frontend déployé : <https://hamzarzine2.github.io/AimClick/>
- URL de votre RESTful API déployée : <https://apiaimclicker.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
Frontend/src/Components/Pages/LoginRegisterPage	Nothing4us	<a href="https://codepen.io/nothing4us/pen/JjZpBXL">https://codepen.io/nothing4us/pen/JjZpBXL</a>	Code pour afficher le formulaire login et register étant une bonne base pour ce que l'on voulait faire, cependant il a été modifié pour mieux colle a ce que l'on voulait



## 10 Analyse des résultats par le groupe

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

Concernant nos objectifs, nous avons rempli nos tâches principales qui étaient de faire 3 types de partie fonctionnelle et rendre la partie troll assez compliqué. Nous avons initialement prévu de ne faire que ça sans système de connexion cependant nous nous sommes vite rendu compte que cela n'était pas assez, nous avons donc pris le temps de faire un système de connexion ainsi qu'une user page collant nos envies. En outre, nous avons pu faire aussi un système d'« amis » permettant à l'utilisateur d'être amis avec un autre. Nous y avons intégré une fonction permettant la recherche d'un ami ou d'un utilisateur dont nous souhaitons être amis avec et aussi un bouton « Add as friend » afin que l'utilisateur puisse être ami(s) avec un ou d'autres, nous avons réaliser ce use case car nous manquions de code dans le backend.

### 10.2 **Audit ergonomique de votre projet**

Premièrement les règles GDPR ont bien été respecter car nous demandons à l'utilisateur s'il accepte de partager ses informations avec nous lors de son inscription.

Concernant l'ergonomie nous avons directement mis les boutons des différentes parties au milieu de l'écran pour attirer l'attention de l'utilisateur. Nous avons aussi fait disparaître les boutons next et previous dans toutes les parties de l'user dans la page user.

Nous avons aussi donné un choix de partie a l'utilisateur sans pour autant lui offrir trop de choix, chaque différente partie permet de répondre à un type d'utilisateur allant des compétitif au casual. Nous avons aussi donné la possibilité de personnalisé sa partie dans le quick game si l'user préfère avoir plus de temps ou une autre couleur.

Nous avons aussi pris en comptes les temps de chargement de la connexion et inscriptions en mettant un loading pour annoncer à l'utilisateur qu'il doit attendre.

### 10.3 Difficultés techniques rencontrées

La principale difficulté technique a été de trouvé des librairies externes répondant au besoin que nous avons, beaucoup de librairie manquait de documentation et était donc difficile à utiliser tandis que d'autres étaient documentés mais ne donnaient pas l'effet escompté.

La seconde difficulté a été de se répartir les tâches également et de s'adapter à la vitesse de travail de chacun, quelques personnes étaient bien avancées tandis que d'autre un peu en retrait. Ensuite, un problème rencontré était le debugge car parfois nous sommes restés bloqué des jours sur certain beug sans comprendre pourquoi cela ne fonctionnait pas.

Nous avons aussi rencontré une difficulté avec api Canvas car celle-ci pouvait faire apparaitre les formes que l'on voulait mais n'avait malheureusement pas de méthodes afin de récupérer si le clic était bien sur la forme apparue, nous avons donc nous documenter sur les méthodes offertes par notre browser afin de récupérer les endroits de clic et les comparé avec la forme.

Nous avons aussi voulu faire un système de niveau en recollant l'XP (expérience permettant au joueur de gagner un niveau). Mais malheureusement, nous n'avons pas pu avoir assez temps pour réaliser cela à cause de quelques soucis technique, surtout avec la librairie « Fluid-Meter », nous avons simplement affiché l'XP dans l'user page ainsi que dans les parties enregistrées, que ça soit dans la zone des meilleures parties ou de toutes les parties de l'utilisateur, toujours dans cette même page.

#### 10.4 Conseils pour appliquer cette technologie

En ce qui concerne les librairies, il est préférable de lire toute la documentation avant de pouvoir les utiliser et ne pas non plus se limiter à un seul site. Sinon, il y a de grandes chances de rester bloquer. Nous conseillons aussi d'utiliser des « console.log », cela est d'une grande aide permettant d'avoir plus de chances de dénicher des erreurs d'affichages si, par exemple dans une base de données, vos données ne s'affichent pas sur votre site.

En ce qui concerne au point de vue de la répartition des tâches et l'adaptation de la vitesse de chacun, il faudrait fixer des rendez-vous chaque semaine ou voir plusieurs, pour voir ou en est chacun pour pouvoir l'aider et donc pouvoir avancée plus vite dans le projet, peut être utiliser le logiciel de communication Teams ou tout simplement se rendre sur place pour favoriser l'échange en temps réel.

Ensuite, nous conseillons à tous de prendre du temps libre pour se documenter plus sur la situation du projet, que ce soit en regardant des vidéos youtube, qui eux redirigent vers des sites un peu plus pousser pour profiter du savoir des autres programmeurs qui peuvent vous aider dans vos prochaines collaborations.

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

Les points positifs de ce projet sont que nous avons pu collaborer avec des nouvelles personnes, qui dit nouvelles personnes dit forcément qu'il faut se comprendre mutuellement mais aussi de pouvoir au fur et à mesure que le projet avance de se tisser des liens qui permet ensuite de créer une meilleure atmosphère de travail. Lors de ce projet, chaque partie de travail nous rapprochait vers notre objectif commun qui était de finir l'idée sur laquelle tout le groupe était d'accord. Ensuite, notre communication s'améliorait de jour en jour grâce aux différentes difficultés que nous pouvions rencontrer au fil du temps, il était alors plus aisé d'aller parler à n'importe qui du groupe qui avait la réponse à ce problème. La manière dont nous avons travaillé était aussi un point efficace à pointer, nous avons tous eu chacun sa branche avec ses features, les merges request étaient faits sans destruction du code de quelqu'un qui avait déjà push son travail, c'était donc agréable.

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

Eh bien, quelques points à améliorer ? Je dirai que nous sommes plutôt bien débrouillés dans la globalité de la chose mais pour de futures collaborations il faudrait peut-être un vrai enjeu, ceci étant un projet d'école qui arrivait au moment où d'autre projet de grande taille arrivait, alors nous n'étions pas investis à 100% dans le projet mais ce que nous pouvons dire c'est que dans le point de vue de la collaboration, la communication est primordiale et comme dit plus haut, nous avons cette chance dans ce groupe, nous pensons donc que avec plus de ressource et de temps et évidemment un peu plus de sérieux de toute l'équipe, nous pouvons arriver à faire bien mieux que ce que nous avons montré avec la même façon de faire.

## **11 Analyses individuelles des résultats**

## **12 Présentation vidéo**

Lien vers la vidéo youtube : <https://youtu.be/86R-Fs1sAZA>

## **13 Revues de projets par les pairs**