Ergonomie & développement d’une SPA animée

|  |  |
| --- | --- |
| **Auteur 1** | ALAGOZ Efe |
| **Auteur 2** | FUENTES GONZALEZ Lucas |
| **Auteur 3** | JOHNEN Thomas |
| **Auteur 4** | MALKA Raphael |
| **Auteur 5** | OFFER Samy |
| **Date** | 18/12/2022 |
| **Référence** | WEB2-2022-PROJECT-GROUP-14 |
| **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 héritez 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

Le but de notre projet est de fournir une plateforme qui permet aux utilisateurs de se fixer des tâches à accomplir et de les réaliser de manière ludique. En choisissant la difficulté de chaque tâche et en accomplissant ces tâches, les utilisateurs gagnent de l'expérience pour leur personnage, qui monte en niveau et devient de plus en plus fort. Cette progression permet aux utilisateurs de défier et de vaincre des ennemis virtuels, ce qui peut être une source de motivation supplémentaire pour atteindre leurs objectifs. Le projet vise particulièrement les personnes qui ont du mal à se motiver et qui cherchent une façon amusante de s'engager dans des activités productives.

# Planning des tâches et cas d’utilisation

* URL vers votre GitHub Project public : <https://github.com/orgs/e-vinci/projects/28>

# 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

Chance : cette librairie permet de générer des noms et prénoms aléatoires pour la liste d’ennemies.

Uuidv4 : permet de générer des id aléatoire et unique pour chaque ennemies, taches, players, users.

Js-cinfetti : permet de générer des confetti lorsque l’on valide une tache.

Lien des sites des technologies utilisées :

<https://www.npmjs.com/package/chance>

<https://www.npmjs.com/package/uuidv4>

<https://www.npmjs.com/package/js-confetti>

## Wireframe

*Veuillez 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.*

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

# Conception & Implémentation

## Code repositories

* URL pour le web repository public associé à votre projet :

<https://github.com/e-vinci/web2-2022-project-group-14.git>

## Documentation de votre API

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

|  |  |  |
| --- | --- | --- |
| URI | Méthode HTTP | Opération |
| auths/register | POST | Register a user : enregistre un utilisateur |
| auths/login | POST | Login a user : permet a un utilisateur de se connecter a son compte |
| auths /addenemy | POST | Add a enemy : rajoute un enemi a la liste d’ennemi de l’utilisateur |
| auths /removeEnemy | POST | remove a enemy : retire un enemi de la liste d’ennemi de l’utilisateur |
| auths /readAllEnemies | GET | Read all enemies : permet de donner la liste des ennemies de l’utilisateur |
| auths /fight | POST | Fight : permet de faire combattre l’utilisateur contre le premier ennemi de sa liste |
| playerCharacter/ | POST | Get xp : permet de récupérer l’XP du user |
| playerCharacter/player | GET | Player : permet de récupérer le joueur |
| playerCharacter/attack | GET | Attack : permet de récupérer l’attaque |
| playerCharacter/currentHP | GET | CurrentHP : permet de récupérer les points de vie actuels |
| playerCharacter/level | GET  GET | Level : permet de récupérer le niveau |
| playerCharacter/maxHP | GET | MaxHP : permet de récupérer les points de vie max |
| playerCharacter/currentXP | GET | CurrentXP : permet de récupérer les points d’expérience actuels |
| playerCharacter/XPToLvlUp | GET | XPToLvlUp : permet de récupérer les points d’expérience nécessaires pour passer au niveau suivant |
| taches/ | POST | Create a task: permet de créer une nouvelle tâche. |
| taches/:id | GET | Display a specific task: permet de récupérer une tâche spécifique. |
| taches/:id | DELETE | Delete a task: permet de supprimer une tâche. |
| taches/:id | POST | Validate a task: permet de valider une tâche. |

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

## Déploiement de vos applications

* URL de votre frontend déployé : <https://steinzu.github.io/>
* URL de votre RESTful API déployée : <https://todogroupe14.azurewebsites.net/>

## 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** |
| --- | --- | --- | --- |
| Rien a part ce qu’il y a dans le cours et dans le boilerplate | Rien a part ce qu’il y a dans le cours et dans le boilerplate | Rien a part ce qu’il y a dans le cours et dans le boilerplate | Rien a part ce qu’il y a dans le cours et dans le boilerplate |

# Analyse des résultats par le groupe

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

L'application que nous avions imaginée au départ était une liste de tâches à accomplir. En d'autres termes, notre but était de pouvoir créer et effectuer des tâches que nous nous fixions dans le but de nous aider à les réaliser quotidiennement. Par la suite, nous avons voulu étendre notre idée à un jeu vidéo. Nous trouvions que faire une application web qui ne permettait que de gérer des tâches n'était pas suffisant et avons pensé à un système de combats contre une liste d'ennemis. Cela ajoutait aussi la dimension "joueur" pour l'utilisateur et nous a donc obligé à créer un personnage dans notre application pour l'utilisateur. Finalement, nous avons réussi à atteindre tous nos objectifs. Aussi bien notre idée initiale qui était de faire une liste de tâches à accomplir, mais aussi d'implémenter notre système de combat interactif entre l'utilisateur et nos ennemis. Nous avons travaillé dur pour atteindre nos objectifs et sommes ravis d'avoir réussi à créer une application aussi divertissante et utile pour les utilisateurs.

## Difficultés techniques rencontrées

Lors de la création des ennemis nous nous sommes rendu compte que créer un model a par entière pour faire la liste d’ennemis n’était pas une bonne idée car cela permettais a n’importe qui d’avoir accès a la liste d’un autre utilisateur. Il a donc fallu réécrire tout le code et le transférer dans le model user afin de lier les deux plus efficacement.

Concernant la partie des taches nous avons éprouvé des difficultés à associer les taches a des utilisateurs. Nous avons finalement réussi à associer les taches a chaque utilisateur connecté. Le but était de permettre a chaque utilisateur d’accéder a ses taches une fois connecté.

Concernant le déploiement du projet, nous avons rencontrés plusieurs problèmes. En effet, localement le projet fonctionne parfaitement. Nous n’avons pas su trouver la raison pour laquelle la version déployée ne fonctionne pas de la même manière…

## Audit ergonomique de votre projet

Selon les différentes analyses que nous avons vus en classe nous pouvons faire une analyse simple de notre site. Tout d’abord si nous analysons le site selon la loi de Hick-Hyman nous pouvons voir que l’utilisateur peut facilement trouver ce qu’il recherche et comprends facilement les différences entre chaque options. De plus selon l’effet d’isolation de Von Restorff nous voyons que certains éléments telles que le bouton « login » et « register » sont plus foncés que le reste afin de permettre à l’utilisateur de voir au premier coup d’œil la première action qu’il doit effectuer. Pour finir le site utilise différentes tailles et couleur de bouton afin de permettre à l’utilisateur de bien distingués les différentes options du site. La règles RGPD est Le Règlement Général pour la Protection des Données. Lors de l’inscription d’un utilisateur sur notre site internet nous demandons à celui-ci de cocher la case acceptant les conditions générales d’utilisation ce qui comprends les règles RGPD.

## Conseils pour appliquer cette technologie

Avant d’appliquer cette technologie SPA, il faut avant tout bien comprendre et établir la structure du projet sur lequel on va travailler afin d’éviter de devoir faire des gros changements car cela peut amener davantage de bug.

Nous nous sommes beaucoup aidés du syllabus web 2 du cours ainsi que certains sites qui nous ont aider à trouver des solutions à nos problèmes :

Lien 1 stackoverflow : <https://stackoverflow.com/>

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

Tout d’abord, chaque membre de l’équipe a jouer son rôle dans le projet. Des le début nous avons su nous répartir les taches de façon équitables et chaque membre a essayer d’aider les autres dans leur taches lorsque cela était possible. Ensuite, notre groupe ne manquais pas de communication. En effet, nos discution souvent du projet et nous tenions les autres au courant de nos avancement que ce soit par le tableau de bord instaurer au seins de l’équipe que par notre discussion discord.

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

Avant de commencer un projet, il faudrait que chaque membre du groupe établisse ses capacités ou les éléments qu’il connait le mieux afin d’établir des taches en fonction des capacités de chacun. Ensuite, pour améliorer la collaboration au seins du groupe, nous pourrions etablir des deadlines plus strictes.

# Présentation vidéo

Lien vers la vidéo YouTube : <https://youtu.be/yOC4drSP_FE>