Ergonomie & développement d’une SPA animée

**ZOMBIE SURVIVORS**

|  |  |
| --- | --- |
| **Auteur 1** | Ngoc NGUYEN |
| **Auteur 2** | Tanguy QUEGUINEUR |
| **Auteur 3** |  |
| **Auteur 4** |  |
| **Auteur 5** |  |
| **Date** | 18.12.2022 |
| **Référence** | WEB2-2022-PROJECT-GROUP-17 |
| **Version** | 1.0 |

Contents

[1 Consignes et évaluations 3](#_Toc122298162)

[1.1 Consignes générales 3](#_Toc122298163)

[1.1.1 Création des groupes sur le site du cours 3](#_Toc122298164)

[1.1.2 Création d’un groupe sur GitHub Classroom et du web repo associé 3](#_Toc122298165)

[1.1.3 Projet 5](#_Toc122298166)

[1.2 JavaScript & Node.js : consignes techniques, timing et évaluations 6](#_Toc122298167)

[1.3 Ergonomie : consignes techniques, timing et évaluations 11](#_Toc122298168)

[2 Objectif du projet 11](#_Toc122298169)

[3 Planning des tâches et cas d’utilisation 12](#_Toc122298170)

[4 Besoins techniques 12](#_Toc122298171)

[4.1 Système 12](#_Toc122298172)

[4.2 Frontend 12](#_Toc122298173)

[4.3 API 13](#_Toc122298174)

[5 Choix technologiques 14](#_Toc122298175)

[5.1 Frontend 14](#_Toc122298176)

[5.2 RESTful API 14](#_Toc122298177)

[6 Conception & Implémentation 14](#_Toc122298178)

[6.1 Code repositories 14](#_Toc122298179)

[6.2 Secrets éventuels pour vos API ou base de données 14](#_Toc122298180)

[6.3 Documentation de votre API 15](#_Toc122298181)

[6.4 Déploiement de vos applications 15](#_Toc122298182)

[6.5 Code réutilisé 15](#_Toc122298183)

[7 Analyse des résultats par le groupe 16](#_Toc122298184)

[7.1 Evaluation du résultat par rapport au planning des tâches et des cas d’utilisation 16](#_Toc122298185)

[7.2 Audit ergonomique de votre projet 16](#_Toc122298186)

[7.3 Difficultés techniques rencontrées 17](#_Toc122298187)

[7.4 Conseils pour appliquer cette technologie 17](#_Toc122298188)

[7.5 Quels sont les points positifs à la manière dont s’est déroulée la collaboration au sein du groupe ? 18](#_Toc122298189)

[7.6 Quels sont les points qui seraient à améliorer pour de futures collaborations ? 18](#_Toc122298190)

[8 Présentation vidéo 18](#_Toc122298191)

# 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

Notre projet concerne la création d’un jeu 2D avec Phaser, ainsi que d’un site web et d’une API pour gérer les utilisateurs et les scores. Notre inspiration vient d’un jeu mobile « Survivor.io » et Vampire Survivors (un jeu à grand succès crée avec Phaser). Le but du jeu est de pouvoir résister le plus longtemps à des vagues de zombies ennemis. Le joueur peut récolter des gems et acheter des améliorations. Le jeu convient à tout le monde mais vise un public qui aime les jeux vidéo.

Nous voulions que notre jeu soit d’abord amusant à jouer et aussi graphiquement joli, nous avons donc passé beaucoup de temps à créer la logique du jeu et à la rendre le plus intéressant possible et nous avons créé ou ajouté des assets pour les ennemis, la carte, l’UI, etc.

Pour la création du site nous n’avions pas beaucoup de pages à réaliser, nous avons donc essayé de les rendre le plus agréable possible.

# Planning des tâches et cas d’utilisation

* URL vers votre GitHub Project public : <https://github.com/orgs/e-vinci/projects/16>

# 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

Pour l’API, nous avons utilisé le package express-rate-limiter, qui permet de limiter le nombre de requêtes dans une période de temps pour une IP. Nous l’avons appliqué sur les requêtes d’inscription, pour empêcher un utilisateur malveillant de créer un grand nombre de comptes à l’aide par exemple d’un bot.

Nous avons utilisé la librairie validator aussi bien dans le frontend que dans l’API pour gérer la validation des pseudos et mot de passe lors de la création de comptes.

Pour le frontend, nous avons utilisé Phaser pour faire notre jeu principal. Cela nous a permis de réaliser toutes les fonctionnalités que nous voulions ajouter au jeu.  
Nous avons aussi utilisé Animate on scroll pour faire des animations sur la page d’accueil afin d’attirer le regard.

## Frontend

* Phaser : https://phaser.io
* Validator : <https://www.npmjs.com/package/validator>
* Animate on scroll: <https://michalsnik.github.io/aos/>
* Font awesome : <https://fontawesome.com/>

## RESTful API

* Express-rate-limiter : <https://www.npmjs.com/package/express-rate-limit>
* Validator : <https://www.npmjs.com/package/validator>

# Conception & Implémentation

## Code repositories

* URL pour le web repository public associé à votre projet : <https://github.com/e-vinci/web2-2022-project-group-17>

## Secrets éventuels pour vos API ou base de données

## 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 | CREATE ONE |
| auths/login | POST | READ ONE |
| auths/logout | GET | READ ONE |
| scores | GET | READ ALL |
| scores?top=value | GET | READ ALL FILTERED |
| scores/{username} | GET | READ ALL FILTERED |
| scores | POST | CREATE ONE |
| scores/{id} | DELETE | DELETE ONE |
| users/{username} | GET | READ ONE |

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

## Déploiement de vos applications

* URL de votre frontend déployé : <https://tqueguin.github.io>
* URL de votre RESTful API déployée : <https://api-zombie-survivor.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** |
| --- | --- | --- | --- |
| /frontend/src/Componants /RegisterPage.js | Fredrik Söderström | https://writingjavascript.com/how-do-you-debounce-in-javascript | Code pour delayer une callback après lecture d’un input (pour éviter de spammer l’api) |

# Analyse des résultats par le groupe

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

Nous avons globalement atteint tous les objectifs que nous avions définis. Ne connaissant pas le framework Phaser lorsque nous avons commencé le projet, nous ne savions pas vraiment ce qui était réalisable ou pas et notre objectif était donc juste d’avoir un jeu fonctionnel avec des mécaniques de base : un joueur qui peut se déplacer, des ennemis qui apparaissent et une arme pour les tuer.

Au cours du projet et en comprenant mieux le fonctionnement de Phaser, nous avons pu rajouter des choses qui n’étaient en fait pas si compliquées à réaliser mais que nous n’avions pas prévues comme du son, un autre type d’arme ou une UI.

Le seul cas d’utilisation prévu que nous n’avons pas implémenté est un mode multijoueur. Cela aurait demandé beaucoup de temps et aurait été assez complexe.

## Audit ergonomique de votre projet

**GDPR :**

Notre site ne récolte pas de données à caractère personnel tel que définies par le RGPD, la seule donnée utilisateur que nous gardons étant le pseudo choisi par l’utilisateur. Les règles sont donc respectées.

**Audit ergonomique**

* Toutes les pages possèdent un raccourci vers la page d’accueil dans la barre de navigation.
* Le site est surtout prévu pour des joueurs sur PC avec un grand écran, mais il est quand même responsive. Cependant, l’affichage sur des très petits écrans comme un téléphone n’est pas toujours possible, par exemple sur la page du jeu.
* La page d’accueil contient en grand et au milieu deux boutons qui guident l’utilisateur (lancer une partie ou découvrir les règles du jeu).
* Les temps de chargement sont corrects.
* Les boutons importants (accès au jeu, connexion, inscription,…) sont placés de manière consistante dans la barre de navigation.

## Difficultés techniques rencontrées

Le développement du jeu en Phaser ne nous a pas posé de difficulté bloquante, parfois il nous a fallu passer beaucoup de temps pour trouver la manière de faire quelque chose ou résoudre un bug mais globalement le framework est bien conçu et n’a pas trop posé problème.

La plus grande difficulté à été la gestion des CORS, en particulier car nous utilisons des cookies et il a donc fallu chercher une solution pour qu’ils soient transmis car par défaut ce n’était pas le cas, donc quand l’utilisateur se connectait il en recevait pas de cookie. En cherchant nous avons trouvé la solution qui nécessitait juste d’ajouter quelques lignes de code mais nous avons perdu beaucoup de temps.

C’est le déploiement sur Github pages qui a posé des problèmes, à la base notre code marchait en local et nous n’avions aucun problème mais c’est quand nous avons commencé à le modifier pour le déployer et arrêté d’utiliser le proxy que toutes les erreurs liées aux CORS sont apparues.

## Conseils pour appliquer cette technologie

Nous avons choisi de réaliser un jeu avec le framework Phaser. C’est un outil très riche qui permet de faire énormément de choses, ce qui signifie également qu’il est assez complexe. C’est la première fois dans nos études que nous voyons un framework et il faut donc être prêt à passer du temps à lire de la documentation pour comprendre le fonctionnement de celui-ci. Heureusement celle de Phaser est très complète.

Le cours de Rudi Giot mentionné dans le module 3 du cours est une excellente ressource pour apprendre les bases.

Le site <https://labs.phaser.io/> contient de nombreux exemples, très utile pour comprendre le fonctionnement de phaser et voir ce qu’il est possible de faire avec.

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

La collaboration entre les deux membres du groupe était très bonne, le travail s’est réparti naturellement et nous avons chacun travaillé sur les différentes parties du projet (frontend, jeu avec Phaser, backend) ce qui nous a permis de ne pas nous limiter à un domaine précis et comprendre globalement tout notre code.

Nous avons bien travaillé ensemble pour résoudre nos problèmes et nous avons pu résoudre certains problèmes plus rapidement en communiquant par Discord pour partager nos problèmes.

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

Il aurait été souhaitable d’être plus de deux à travailler sur le projet car la charge de travail était assez grande. Malheureusement cela n’était pas en notre contrôle, par contre nous avons beaucoup repoussé la mise en place de fonctionnalités ce qui nous a emmené à effectuer une très grande partie du travail sur les dernières semaines. Il aurait fallu plus réfléchir aux objectifs concrets que nous voulions atteindre et essayer de les définir de manière précise afin de savoir exactement ce à quoi nous voulions arriver.

# Présentation vidéo

Lien vers la vidéo youtube : <https://www.youtube.com/watch?v=p6_vny0jCbc>