

Rapport de stage de première année

Foulques Geraud

11 octobre 2017

Table des matières

1	Introduction	3
2	Sujet et recherche du stage	4
2.1	recherches, rencontre de l'entreprise	4
2.2	Délimitation du sujet	4
3	Entreprise et environnement	5
3.1	Ethereal World	5
3.2	La startup, le cluster	5
3.3	Matériel	6
3.4	L'équipe	6
4	Déroulé du stage	7
4.1	Installation, observation, planification	7
4.2	Recherche d'un framework, démarrage du code	8
4.3	premières vues	8
4.4	Contrôleurs, modèle	9
4.5	Le virage Laravel	10
4.6	Fin du stage et possibilités futures	11
5	Contraintes	12
5.1	Le framework Laravel	12
5.2	Méthodes de codage, flexibilité	12
5.3	Ressources	13
5.4	Le dictionnaire	14
6	Problèmes et résolutions	15
6.1	Approche d'une sécurisation	15
6.2	Prise en main d'un Framework	16
6.3	Réseau, Bases de données, chaos	16
7	Conclusion	18
7.1	Réalisations et apports du stage	18
7.2	Remerciments	18

1 Introduction

Le présent rapport se trouve bien particulier, puisque le rapport d'origine a été perdu suite au décès subit de mon ordinateur. Intégralement réécrit de mémoire, on n'y trouvera ni code, ni image, toutes ces sources ayant été perdues. Le rapport se constituera donc de bonne foi sur des faits et des explications détaillées, relatées ici avec l'assentiment de mon maître de stage.

Il s'articule autour de la présentation d'abord du stage et de son sujet, puis de l'entreprise, avant de présenter un déroulé chronologique du stage proprement dit, suivi d'une digression sur les contraintes du stage et sur leur implémentation. Vient ensuite une présentation des principaux problèmes rencontrés et résolus, et enfin une conclusion.

La première partie à venir va donc expliquer comment s'est déterminé mon sujet de stage.

2 Sujet et recherche du stage

2.1 recherches, rencontre de l'entreprise

Le parcours réalisé à la recherche d'un stage a été difficile et longtemps infructueux. Néanmoins, sur les recommandations d'autres élèves, plus expérimentés, j'ai eu l'occasion de rechercher un stage dans ces hauts lieux de l'informatique que sont les incubateurs de TSP et de l'ENSIIE. C'est finalement dans ce dernier que j'ai pu faire la rencontre de M Y et de ses associés d'Ethereal World.

Le C-19 étant (pour être exact) le cluster de l'ENSIIE, on y trouve de nombreuses entreprises dans le domaine de l'informatique, comme Dronetix, From The Bard ou la Junk Food Factory. Au sein de cette pépinière, les membres d'Ethereal World ont eu la gentillesse de m'accueillir et de montrer beaucoup de bienveillance face à mon projet professionnel et à leurs besoins.

C'est finalement après une longue et riche conversation que nous avons grossièrement délimité le sujet et les conditions du stage que je devais chez eux, concernant la création d'un site web.

2.2 Délimitation du sujet

A la suite d'un second entretien avec l'équipe d'Ethereal World, j'ai pu découvrir plus en détail le projet de cette start-up, lequel est le lancement d'un jeu vidéo sur mobile, Last Heart.

Pour faciliter le lancement dudit jeu sur certaines plateformes, pour des raisons de commercialisation, pour refléter l'entreprise qui l'a créé et pour cristalliser une communauté autour de Last Heart, créant ainsi une base commerciale pour les prochaines sorties de la start-up, il était nécessaire de construire un site web orienté vers tout ces points, qui fût plus qu'un site vitrine, et qui portât le projet en avant.

Bien que les membres d'Ethereal World aient largement les compétences de le faire, ils avaient néanmoins besoin de la main-d'oeuvre que je représentais pour eux, puisqu'ils approchaient du *rush* de sortie de leur jeu.

Le sujet de mon stage était donc précisément **la conception, la création et le lancement d'un site web dynamique et simple, hautement flexible, accessible sur mobile comme sur pc.**

3 Entreprise et environnement

3.1 Ethereal World

Ethereal World est une association de loi 1901 dont le statut doit devenir celui d'une entreprise d'ici cet automne. Le projet, Last Heart, est un jeu mobile permettant à son utilisateur de déplacer un personnage personnalisable dont le but est de survivre à des attaques successives de monstres, dans des environnements variés.

Ce projet est bâti en parallèle d'un autre, bien plus vaste et complet, nommé Discovery, construit tout autour d'un univers particulier. Le projet Last Heart a été lancé aux alentours de janvier dernier, c'est donc dans un projet quasiment terminé que je me suis intégré, et qui sortira cet automne dans ses premières versions.

Au début du stage, le game design est complété, et le jeu en est à ses premières versions avant lancement. Le principal travail est un travail graphique sur les textures, la luminosité, l'ajout de paramètres dans le jeu et la création de contenu (personnages, lieux...) .

3.2 La startup, le cluster

Située 19 Cours Blaise Pascal, à Evry, Ethereal World dispose de locaux ouverts et agréables qui laissent toute l'équipe en relation, à l'exception des graphistes qui ont la chance d'utiliser la plateforme technologique du Cluster, richement équipée. Les locaux disposent aussi d'une cafétéria, et leur proximité avec l'ENSIIE favorise de nombreux échanges entre l'équipe et les élèves.

La place de l'entreprise au sein de l'incubateur m'a permis de découvrir une quantité remarquable de choses sur son fonctionnement, ses financements, l'installation de ses équipements, les projets qui y sont menés, etc. Cette immersion dans le monde des start-ups m'a permis de me forger un avis construit sur une connaissance beaucoup plus étendue de la vie en incubateur.

Par ailleurs les relations de l'entreprise avec un *FabLab* d'Evry, La Fabrik', ont été elles aussi enrichissantes, et constituent un savoir qui me sera utile dans les domaines de la création et de la gestion d'entreprises, de projets, les recherches d'opportunités dans le monde du travail et de l'informatique.

3.3 Matériel

Pour ce qui concerne le matériel informatique utilisé pendant le stage, j'ai pu employer mon ordinateur personnel pour des raisons pratiques. En termes de logiciels/technologies, la liste s'allonge puisqu'il a fallu employer en premier lieu cloud9, puis changer d'environnement au profit d'un autre IDE, temporairement Codelite (lourd et peu fonctionnel par rapport au travail mené) puis SublimeText (léger, agréable mais proposant trop peu de contenu).

La base de données et l'hébergement ont été gérés grâce à WampServer, le framework utilisé fut Laravel (géré en terminal, il a fini par me faire beaucoup utiliser nano), impliquant d'apprendre à utiliser Blade pour les fichiers `.blade.php`.

Du point de vue graphique, les balises Bootstrap ont été employées dans le code HTML, avant que l'on ne s'attaque au Bootstrap lui-même en le compilant (j'ai donc dû faire une rapide promenade en ruby, afin d'installer sass, un compilateur ruby permettant de compiler du sass `[.scss]` en fichier css qui soit un Bootstrap utilisable).

Naturellement, les langages php, php orienté objet notamment ; html, javascript, css, sass, sql et blade (susmentionné) auront servi pour le code ; enfin l'éditeur graphique GIMP aura servi pour certaines ressources graphiques (logos, boutons).

Pour la documentation, les sites de Bootstrap et W3school, hélas parfois contradictoires, ont été utilisés, ainsi que l'excellente documentation en ligne de php. Les sites comme OpenClassrooms (peu utile et chaotique) ou Stackoverflow (très pratique quoique pas toujours de référence) ont également fait leur apparition. Les sites de Blade, de Laravel, de Sassy et bien d'autres ont été eux aussi consultés.

3.4 L'équipe

Les membres d'Ethereal World sont organisés autour de trois piliers : Tout d'abord Vitéra Y, patron de la boîte, ancien de l'ENSIIE. Puis viennent Thomas Laurent, mon maître de stage, et Martin Herbeth, *co-founder*. L'équipe comptait aussi trois graphistes et d'autres stagiaires, camarades de promotion. J'ai pu noter avec ravissement que l'esprit d'équipe est chaleureusement entretenu au sein de l'entreprise, par le travail constant de ses membres centraux, qui apportent souvent un réconfort substantiel à leurs employés lorsqu'ils travaillent, ou les emmènent profiter de sorties hors des heures ouvrées.

4 Déroulé du stage

4.1 Installation, observation, planification

Les premiers jours d'installation ont été consacrés à la planification du site lui-même, de la *to do list* y correspondant, au listing des fonctionnalités attendues par Ethereal World et à l'observation approfondie des sites du marché des jeux mobiles. C'est donc assez rapidement qu'ont émergé les idées des développeurs de Last Heart pour le site :

- la flexibilité du langage : le site devait être codé pour être en anglais, et lisible en français grâce à un bouton.
- l'aspect communautaire : pour fédérer la communauté autour du jeu, des comptes devaient pouvoir être créés, accessibles, personnalisables, et devaient pouvoir commenter sur les informations, news etc.
- le caractère informatif : des rubriques de foire aux questions, et d'informations des développeurs ou d'évènementiel devaient être implémentées, éditables avec insertion d'hyperliens, de vidéos, changements de textes et autres, tout cela bien sûr commentable par les utilisateurs.
- la visibilité de l'entreprise : naturellement, faire un peu de publicité pour Ethereal World était le bienvenu, il fallait donc veiller à laisser une page de présentation de l'entreprise et de ses réalisations, de ses membres, et de ses politiques de confidentialité, encore une fois tout cela étant sujet à modification, devait être administrable sans toucher au code du site.
- les graphiques : tout jusqu'aux contours des boutons du site devaient avoir une animation avec variation de lumière, mouvements, en fonction de la durée du clic ou d'un simple passage de la souris pour accrocher le client potentiel

Finalement, le plan du site fut déterminé en une page d'accueil à quatre menus déroulants permettant d'accéder aux domaines News (nouvelles, galeries d'art), Communauté (Membres, Forum), Questions (FAQ, wiki) et Ethereal (site [De l'entreprise, pas encore réalisé - note], politique). La page d'accueil devait disposer de liens en pied de page vers les boutons du dernier menu (site et politique).

En tête de page, à droite du menu, le choix des langues (discret) et les options connection/inscription ou déconnection/mon compte selon que le visiteur du site soit connecté ou non. Enfin, un menu caché, contenant les panneaux d'administration de la FAQ, des news, des joueurs, *et caetera* réservé aux seuls administrateurs connectés.

Cela fait, il fut aisé de faire une liste complète des fonctionnalités nécessaires et désirées pour le site, et de voir lesquelles entraient en relation, et ainsi quelles étaient les tâches qu'il serait convenable de réaliser les premières.

4.2 Recherche d'un framework, démarrage du code

Les jours suivants ont été consacrés à la recherche d'un framework pour le site. J'ai pu en voir de nombreux, mais par exemple Django, bien que théoriquement en python, avait une syntaxe trop mal définie, trop vague et trop lourde, nécessitant une maîtrise qu'il m'était difficile d'acquérir en si peu de temps. Pour un autre exemple, Angularjs, bien que d'excellente réputation, était trop complet et encore une fois trop difficile à prendre en main pour être pris.

Mon choix s'est donc porté sur une absence de framework, puisque je n'en trouvais pas qui fût capable de répondre à mes attentes. Il était par ailleurs hors de question de réaliser un WordPress, ce qui en plus d'être sans intérêt pour moi ne procurait pas de fonctionnalité, bien que l'idée soit venue plusieurs fois à mon Maître de stage. Pour coder sans framework, j'ai réimporté une certaine quantité de fichiers du site que j'avais proposé en projet web, incluant les builder, mais excluant tout contenu html, js ou css, puisque la conception du site de Last Heart était radicalement différente.

La première implémentation que j'ai réalisée était celle de la base de données, qui a d'ailleurs beaucoup évolué au fil du temps. Elle contenait principalement une table des utilisateurs enregistrés, et des tables qui référençaient les fichiers situés dans les dossiers sources, et qui étaient soit

- une question de la FAQ en format txt
- sa réponse, sous le même format
- une image de profil sous le format [IDuser].jpg
- un texte d'information (news) au format txt
- une pièce jointe d'information à un format spécifique
- un commentaire au format txt

La sécurité a constitué une partie importante du code, qui sera détaillée dans la partie consacrée

4.3 premières vues

La base de données implémentée, j'ai pu m'attaquer aux éléments fondamentaux de mes pages web, à savoir l'aspect, les en-têtes, les menus et les pieds de pages. Pour l'aspect, mes employeurs ne pouvant me fournir une charte graphique spécifique au site, nous avons essayé à plusieurs reprises dans les trois mois de stage d'en établir une. Afin d'avoir, en attendant, des éléments sur lesquels tester l'intégration dans les pages, j'ai eu l'occasion d'employer GIMP sur les logos et images de l'entreprise aimablement fournis par mon maître de stage.

L'en-tête des pages web étant toujours la même, elle fut implémentée dans un fichier vue une bonne fois pour toutes, avec l'encodage qui allait, etc. Nous pouvons noter que la sagacité de mon maître de stage me fit importer les fichiers Bootstrap

de css et de js non pas de leur répertoire usuel, en ligne, mais bien de dossiers où je les avais téléchargés, puisque de rester toujours sur la dernière version en ligne cesserait d'être avantageux sitôt que celle-ci ne respecterait plus ma façon de coder, ce qui pourrait s'avérer dommageable au site et exiger des mises à jour majeures des vues.

La barre de menu fut vite implémentée, selon le plan énoncé plus haut. Il en alla de même pour le pied de page, qui disposait d'un bouton de retour au haut de page. Ces fonctionnalités prêtes et écrites dans les vues, il fallut les traduire ; car le site devait être bilingue, aussi devais-je m'afférer à la rédaction d'un dictionnaire, qui était un tableau associatif prenant en entrée une chaîne de caractère telle que dans les vues, identifiant unique de deux expressions linguistiques correspondantes, parmi lesquelles on choisissait la bonne en fonction de la variable de session de langues, placée par défaut sur l'anglais. Devant permettre la traduction de toutes les pages du site, il est vite devenu massif.

4.4 Contrôleurs, modèle

Une fois ce travail de vues implémenté, je me suis attaqué à celui des contrôleurs, en commençant à oeuvrer les pages en particulier. Pour chacune d'entre elles, j'ai établi un plan visuel façonné à partir d'objets Bootstrap qui correspondaient le mieux aux aspects que l'on voulait rendre pour mettre tel ou tel point en valeur. Ce faisant, il a fallu s'atteler au code des fonctionnalités en particulier, ce qui s'est divisé en trois parties distinctes quoique proches.

Tout d'abord le codage des formulaires. A partir du langage html implémenté dans le php par des fonctions **echo**, ce travail *a priori* simple s'est vite complexifié à cause de la lourde contrainte du dictionnaire. Par ailleurs, la diversité des objets concernés (champs de dates, textes, upload de fichiers pour les enregistrements de comptes ; champs d'upload de fichiers plus divers pour les news, etc) empêchait la réutilisation brute de code. L'organisation desdits formulaires de manière esthétique était un autre défi à relever, qui en l'occurrence n'avait pas la priorité. Tout ce travail n'était encore un travail que de vues, qui restait simple à côté des traitements mis en place derrière.

En effet, pour parvenir à présenter un système qui ne soit pas perclus de failles, il était nécessaire de préparer des injections SQL sécurisées dans la base de données. Ceci c'est fait selon les principes de bases, à savoir la recherche de caractères dangereux tels que les points-virgules, en renvoyant des messages d'erreur ; et l'échappement des caractères tels que les guillemets, simples ou doubles. En effet ceux-ci devaient être préservés dans la base puisqu'un utilisateur pouvait enregistrer dans son profil une signature, à savoir éventuellement une citation, qui se voudrait immanquablement accompagnée de tels caractères. Lors des injections et des extractions de telles chaînes de la base de données, il fallait donc aussi pen-

ser à l'échappement des caractères dangereux pour les autres langages (bien sur le html, avec les caractères comparateurs...), un gros travail a donc été effectué dessus.

Enfin, le rôle du contrôleur, qui consistait en une première vérification de cohérence des données entrées dans les champs du formulaire, puis en leur traitement, conduisant à un système de très nombreuses erreurs possibles (nous citerons un éminent professeur de l'ENSIIE, dont j'appliquais la maxime avant même de l'entendre : *Dites-vous que l'utilisateur est un con.*) afin de rendre un message d'erreur cohérent pour guider l'utilisateur qui remplirait mal les formulaires sur son écran. Les traitements concernant les téléversements d'images était nouveau pour moi, j'ai donc découvert comment grâce aux merveilleuses fonctions de php, je pouvais reconnaître leur format, le changer (écraser le nom de l'image par [ID user].jpg, en fait...), vérifier son poids en octets, l'enregistrer dans un dossier consacré... Bien entendu, pour surveiller la performance de mon code, je l'ai accompagné de nombreuses fonctions de vérification (Le fichier a-t-il bien été chargé ? Est-il bien renommé?..).

Continuant sur ce chemin, je progressais page par page, et le site se construisait en conséquence. Néanmoins, le système bâti n'était pas au goût de mon maître de stage qui demandait plus de flexibilité que le code php simplement segmentaire, et des avantages tels que le routage. Au bout d'un mois, celui-ci m'a fortement encouragé d'employer un framework qu'il maîtrisait et sur lequel il pourrait me conseiller, à savoir Laravel.

4.5 Le virage Laravel

Finalement, il a donc fallu employer le framework Laravel, construit avec un système de routage, et surtout employant du php orienté objet, ce qui a été pour moi une nouveauté. Nouveautés aussi les fichiers utilisant blade, et finalement à peu près toute la structure employée dans ce framework peu connu.

Il m'a fallu énormément de temps pour réussir à employer ce framework, au cours d'une prise en main qui sera détaillée ci-après. De longs jours ont été consacrés à lire la longue, et malheureusement peu claire documentation en ligne de Laravel, à essayer sans succès d'en trouver une pour blade, et à essayer de comprendre la structure dans son ensemble.

Après plus d'une semaine de lecture, d'exploration du framework et de ses possibilités, d'essais ; et me sentant prêt à me lancer, j'ai repris le travail par une importation / transposition des vues que j'avais rédigées. Malheureusement, le passage dans le framework changeait beaucoup la syntaxe (notamment dans les formulaires, ce qui sera détaillé dans la partie problèmes et résolutions).

Cela fait, il a fallu s'attaquer au routage des requêtes, qui devait être "flexible" (pour permettre à un utilisateur d'accéder, par exemple, sur une page détaillant une nouvelle lambda, de ne pas chercher `www.lastheart.com/news/newslambdacodecomplicé` mais `www.lastheart.com/news-xxx` -nuance tenant dans le fait qu'on ne cherche plus dans des répertoires de manière personnalisée mais de façon générique- ce qui permet surtout d'éviter de laisser apparente la structure des répertoires du site à d'éventuels utilisateurs mal intentionnés.

4.6 Fin du stage et possibilités futures

Le stage s'est donc terminé sur une adaptation des parties déjà réalisées à Lavel, ce qui a coûté énormément de temps et fait surgir de nombreuses difficultés. En effet tous les programmes de contrôleurs, écrits en php "classique", n'avaient rien à voir avec la façon de faire des langages orientés objet et demandaient donc des changements en profondeur.

Suite à ces adaptations, j'ai pu reprendre tranquillement le travail des pages jusqu'à la fin du stage. Le travail rendu est malheureusement bien loin des attentes de mon maître de stage en termes de graphismes, mais l'essentiel est là, et la structure du site, telle que demandée, est donnée.

Le contact enrichissant avec la société *Ethereal World*, la gentillesse de ses membres, l'étendue de leurs connaissances et celle de leurs expériences m'ont suffisamment motivé pour continuer à finir le site de Last Heart à titre personnel, et dans un cadre non professionnel. En effet il serait dommage de ne pas pousser jusqu'au bout la construction de ce site, dont la finition est en quelque sorte devenue pour moi une question d'honneur.

5 Contraintes

5.1 Le framework Laravel

Le framework Laravel, que j'ai eu de la difficulté à prendre en main, est constitué comme suit :

- un dossier **app** explicité plus loin ;
- dans un dossier config, la très longue et précise configuration du site ;
- le dossier **database** avec la BDD et le modèle ;
- le dossier **public**, avec du CSS, les sources (images), du JS et... index.php ;
- **resources**, qui contient du SASS non compilé, et l'autre partie du JS ;
- **bootstrap** pour le CSS bootstrap ;
- le dossier **routes** pour le routage ;
- **storage**, qui contient les fichiers générés par Laravel et les .blade compilés ;
- enfin les dossiers **test** et **vendor**, le dernier contenant les dépendances de **composer**.

Il est aussi à noter que le dossier *resources*, censé contenir les fonctionnalités du framework pour un site en plusieurs langues, n'en contenait pas, mais devait aussi recevoir les vues en plus d'une partie du CSS et d'une partie du JS. le dossier app contenait des dossiers pour les contrôleurs, pour les *events*, les exceptions, les *listeners*, etc. De manière générale, la requête d'une adresse du site est analysée dans routes, puis dirigée soit vers une vue (qui se veut composée avec blade), soit vers un contrôleur (en php objet donc) qui éventuellement procède à des opérations ou appelle une vue.

Laravel se targue de proposer un système d'authentification, mais celui-ci se restreint au strict minimum. Il est difficile de savoir ce que gère ou non le framework, du point de vue d'éventuels croisements de session, du point de vue de la sécurité, et d'autres ; et c'est ce qui m'a le plus déplu dans Laravel, avec l'absence de vrai système de langues : on ne prend pas un framework pour avoir une certaine prédisposition des dossiers, (qui est le point dont ils semblent le plus fiers) mais bien pour ne pas avoir à coder des systèmes que l'on retrouve dans tous les sites, comme un système d'authentification.

5.2 Méthodes de codage, flexibilité

Afin de réaliser un site web demandant un minimum d'entretien, il fallait veiller à ce que si une partie du code ou une seule disposition du site venait à changer, ce changement puisse être effectué en retouchant au minimum le site ; c'est-à-dire que l'on a insisté pour pousser le plus loin possible le DRY (*Don't Repeat Yourself*) dans la conception du site.

Bien que ce principe, poussé loin, devienne fastidieux à implémenter puisqu'il demande d'incessantes réinsertions de code, et tend à générer de l'arborescence de fichiers de plus en plus touffue, il garantit que chaque section du code utilise les mêmes fonctions et reste à jour par rapport aux autres.

Mon maître de stage m'a aussi appris à coder non pas pour moi-même, mais bien pour un éventuel successeur : insister sur l'abondance des commentaires, la clarté du code, l'aération des fichiers, leur répartition de manière claire et structurée... Entre autres, la rédaction d'un readme m'a plu, puisqu'elle me permettait d'exprimer de manière claire et concrète la manière dont j'avais pensé le site. Cet exercice m'a obligé de repenser la structure de certaines fonctionnalités, notamment dans la Foire Aux Questions, quand je me suis rendu compte qu'elle n'était pas claire...

La flexibilité de l'adressage a été un point difficile pour moi à mettre en oeuvre ; en effet en adaptant mon travail du php naturel vers Laravel, je me suis rendu compte que je faisais énormément d'appels à d'autres pages (liens avec des chemins, sujets à changer) dans mon code, sans respecter le fameux principe DRY. Afin de résoudre ce problème, j'ai pu utiliser le système de routage de Laravel, qui s'est avéré être ma disposition préférée de la part du framework.

Savoir dans quelle mesure le site devait être "flexible" représentait une autre difficulté. La volonté de mon maître de stage restait qu'il ne faille toucher au code du site que pour le moins d'actions possibles. Mais dès lors, comment s'arrêter dans les fonctionnalités ? J'ai permis aux utilisateurs de changer, par exemple, leurs images de profil. Mais doit-on permettre aux administrateurs du site de changer les images d'arrière-plan, les en-têtes de pages, les titres des menus, sans recoder le site ?

Bien que tout cela soit possible, et désiré par Ethereal World, il a bien entendu fallu se limiter à ce qui était faisable dans le temps imparti, c'est pourquoi j'ai choisi de réduire les possibilités des administrateurs au profit d'un site plus complet sur ses bases.

5.3 Ressources

Le fait de travailler sans source a été une autre difficulté ; jusques ici il n'existait aucun site pour Ethereal World ni pour Last Heart, je suis donc parti de rien. De plus l'absence de ressources graphiques (screenshots, boutons, logos, animations...) dans l'entreprise m'a poussé à me lancer, avec GIMP, dans de l'édition d'images, afin de créer des objets graphiques dont j'ai testé l'intégration dans l'esthétique du site.

La rencontre avec les infographistes de l'équipe a été très enrichissante, puisqu'ils n'ont eu de cesse de travailler, outre les personnages, les textures et les environnements du jeu, les logos et la charte graphique (menus, logos...) qui devait accompagner Last Heart. La rencontre de ces personnes et la découverte de leur travail a été pour moi un très bon souvenir, qui m'a laissé de fortes impressions.

Comme le *Game Design Document* n'existait pas au début de mon stage, j'ai eu la chance de voir des professionnels le construire au fur et à mesure de leurs avancées, de leurs échanges, de leurs idées. Avec ce GDD s'est formée la charte graphique, et c'est petit à petit le jeu que j'ai vu naître. En parallèle de cette construction, nous avons suivi la méthode agile SCRUM, en faisant notre stand-up tous les matins, et des sprint de une à deux semaines. Quoique je fus peu concerné par ceux-ci, mon travail étant tout à fait extérieur au développement du jeu, j'ai apprécié de découvrir ce fonctionnement d'une start-up.

5.4 Le dictionnaire

Dans sa première version, le site devait disposer pour chaque vue d'un texte en français et d'un texte en anglais. Mais grâce à l'intervention de mon MDS, j'ai pu m'épargner un travail considérable en utilisant le DRY : la rédaction du dictionnaire était fort utile, puisque pour traduire l'intégralité du site il n'y avait plus qu'à traduire un seul fichier php. En revanche, cette installation alourdit considérablement le code, lequel voit, au lieu de :

```
echo "<h1>Ceci est un titre</h1>";
```

l'expression plus recherchée :

```
echo "<h1>".getlg(ttl)."</h1>";
```

référant au dictionnaire

```
$Dico = array(
    ...
    "ttl" => ["This is a title","Ceci est un titre"],
    ...
);
```

Bien entendu, il a fallu faire un choix pour le fonctionnement du dictionnaire, et j'ai fait celui d'une carte php, qui, fonctionnant comme une table de hachage, permet des temps de recherches optimaux. L'idée est de se dire que le dictionnaire peut vite devenir extrêmement massif, étant donné la taille du site; et que donc il faut absolument éviter qu'un temps de recherche trop long ralentisse le fonctionnement général.

6 Problèmes et résolutions

6.1 Approche d'une sécurisation

La conception du site telle que je l'ai planifiée au départ, c'est à dire sans framework, incluait un système de sessions d'utilisateurs connectés. Afin de le réaliser, il a bien entendu fallu prendre en compte un système relativement sécurisé.

En effet les utilisateurs de sites web, face à la prolifération de leurs comptes, ont tendance à réutiliser partout les mêmes mots de passe. De sorte que, si jamais la base de données de notre site, si elle contenait en brut les mots de passe des utilisateurs, venait à être exploitée par un utilisateur malveillant, elle constituerait une source de problèmes très sérieux pour les autres utilisateurs et la protection de leur données.

Il m'est donc venu à l'esprit de protéger les données, et surtout les mots de passe des utilisateurs, par hachage de ces derniers. Ce fut une étude extrêmement intéressante que celle que j'ai réalisées des différentes manières de hacher, du fonctionnement de ce systèmes, et des outils php pour y parvenir.

Php propose en effet des fonctions relativement complexes pour protéger les données. Etant un peu perdu dans les nombreuses possibilités, j'ai poussé plus avant mes recherches pour savoir quel choix réaliser. Avais-je besoin d'un grain de sel ? Quel algorithme utiliser ? en quelle complexité ? Mon choix a fini par se fixer comme suit :

- l'algorithme classique de php : n'étant pas diplômé en sécurité, je n'avais pas l'aptitude de déterminer si telle fonction de hachage était plus efficace qu'une autre.
- Une complexité en $O(10)$, complexité suffisante qui évite, contrairement aux complexités 11 et 12, d'avoir du temps de chargement à la connexion.
- Une absence de grain de sel, puisque mon maître de stage m'a expliqué le peu d'utilité de ce paramètre difficile à implémenter et à mémoriser.

Tout cela a pu être entré dans les paramètres de la fonction `password_hash()` de php.

Mais la sécurité concernait aussi d'autres aspects : il a fallu coder les vérifications des tentatives d'injection de caractères problématiques en SQL par échappement des caractères ' et ", mais aussi par interdiction des caractères <, > et ;, les uns pour le html, les autres pour le SQL.

Ce désir de sécurité m'a aussi fait analyser les poids et formats des fichiers qui pouvaient être envoyés par l'utilisateur (images de profil) afin d'éviter l'inclusion de fichiers inappropriés ou malveillants. Pour le contenu moral des images, rien ne pouvait me permettre à mon niveau d'exercer quelque choix que ce fût, c'est pourquoi j'ai laissé cette charge à la modération humaine du site.

6.2 **Prise en main d'un Framework**

La prise en main du framework Laravel a pour moi été la principale difficulté du stage. On peut en effet se représenter le désarroi d'un pauvre stagiaire qui, habitué au php classique et amoureux du paradigme fonctionnel, est bombardé du jour au lendemain dans un framework de php orienté objet, sorti tout droit de ses cauchemars... En effet la philosophie des langages objets n'était absolument pas dans mes cordes, et la découverte de ce nouveau genre de php s'est faite dans un contexte peu favorable :

La prise en main du framework m'a directement confronté à une organisation des fichiers radicalement différente de celle que j'employais lorsque je concevais le site au début du stage. Cela a demandé du temps et de l'adaptation pour comprendre, par habitude puis par déduction, où j'étais censé trouver tel fichier devant m'aider.

Par ailleurs, l'utilisation de blade dans les fichiers php pour créer des formulaires, aussi obscure que nouvelle, a été un problème grave : ne trouvant pas de documentation pour m'éclairer, et ne connaissant pas les options ni même la vraie syntaxe du langage, j'ai choisi de garder mes formulaires tels que je les avais rédigés en html, langage qui gère suffisamment bien les options de chaque champ, bien qu'il puisse être un peu plus verbeux.

Le point véritablement douloureux du passage aux framework Laravel a été la traduction des fonctionnalités des contrôleurs en langage orienté objet. Par chance, les fonctions restaient les mêmes, mais leurs appels, et la définition des variables comme objets changeait beaucoup de choses. Ainsi l'outil de vérification des formulaires de Laravel, radicalement différent de mes fonctions php, a lui aussi dû être apprivoisé doucement...

Finalement, le problème de la prise en main du nouveau framework s'est résolu de la manière la plus évidente : par un travail constant, laborieux et difficile, jusqu'à ce que j'acquière une maîtrise relative de Laravel.

6.3 **Réseau, Bases de données, chaos**

De nombreux autres problèmes ont pu surgir de manière épisodique pendant le stage. Par exemple des problèmes de réseau. Pour réussir à parvenir à me connecter depuis ma propre machine à mon site, et sans ouvrir simplement un fichier php avec un navigateur, il me fallait héberger ledit site sur un hôte virtuel. J'ai donc installé un programme permettant de le faire, et réalisé une série d'actions dignes d'une danse de la pluie pour réussir : créer un hôte local virtuel nécessitait de toucher à quelques fichiers système cachés dans les tréfonds de mon OS, démarche délicate dans laquelle la présence de mon maître de stage a été celle d'un véritable phare.

La base de données elle-même m'a posé problème puisque, l'ayant mal définie dès le départ, j'ai dû la transformer très fréquemment. Il m'a fallu la réorganiser, changer sans cesse des attributs... sans compter les corrections du code php qui allaient avec ! C'est finalement en replanifiant la base de données au regard de l'ensemble du site que j'ai réussi à l'implémenter une bonne fois pour toutes. Le passage sur Laravel a aussi apporté d'innombrables problèmes de connexion à la BDD.

Toujours dans le but d'une forme de sécurisation, j'avais peur que si l'hébergeur du site vint à planter, la BDD soit détruite. Pour pallier à ce problème, j'ai implémenté mes contrôleurs de telle sorte que chaque ajout d'information dans la base soit un ajout de texte dans des fichiers csv, qui étaient réimportés à chaque lancement de la BDD.

Une autre source de problèmes, mais à un niveau moins informatique, était mon manque de connaissances par rapport au monde de la publicité pour savoir faire un site promotionnel. Fort heureusement, l'alignement de la concurrence sur des sites très semblables et mes compétences naturelles en communication m'ont permis de me faire une idée assez précise de ce qui pouvait être attendu.

La durée du stage elle-même était un autre problème au vu de la quantité de travail qui était à faire. Le site rendu à Ethereal World aujourd'hui est lacunaire et incomplet, dû à cette contrainte de travailler seul pendant deux mois sur un site au cahier des charges exigeant. Néanmoins, ce qui a été fait a été bien fait et constitue un apport considérable, qui me fait honneur, et m'a suffisamment plu pour me motiver à le porter à terme.

7 Conclusion

7.1 Réalisations et apports du stage

Le site de Last Heart remis à Ethereum est un site au visuel très simple, mais clair et *design* (grâce au Bootstrap). Les fonctionnalités sont présentes, et même si les animations et les visuels rêvés par Ethereum sont encore loin d'être réalisés, toute la structure fonctionne, la base de données est implémentée, les comptes d'utilisateurs sont utilisables. Bien que le système de news soit encore incomplet (il n'y a pas d'édition complexe possible), la FAQ, l'envoi de news, le carroussel d'accueil, les importations de fichiers, la personnalisation des comptes, l'interface administrateurs ont bien été réalisés.

Dans son ensemble le stage m'a énormément fait progresser, et sur de nombreux points. D'abord dans l'écriture de code orienté objet, dans laquelle j'avais d'énormes lacunes. Une grosse quantité de savoirs techniques m'ont été enseignées : bonnes attitudes de code par commentaires, DRY ; établissement d'hôtes virtuels, mise en ligne de serveurs, hébergement ; méthodes agiles, travail collaboratif. Ma maîtrise du développement web en général a bien été poussée.

Cet enrichissement en termes de connaissances se ressent particulièrement au niveau du web, puisque je maîtrise désormais jusques à la compilation du sass en scss, les bases de la sécurisation des données utilisateur, l'utilisation d'un framework en particulier, l'upload et le traitement de fichiers envoyés par les utilisateurs.

Si je devais tirer une leçon générale de l'enseignement de ce stage, je donnerais celle-ci : qu'en web les frameworks n'existent que pour coder chacun un type de site bien particulier, sans quoi ils sont plus problématiques que pratiques ; mais que pris dans leur domaine, ils sont d'excellents outils de développement.

7.2 Remerciements

Tout d'abord, il est naturel que je remercie M. Laurent, mon maître de stage, pour sa grande patience, sa gentillesse, sa conversation extrêmement intéressante, en particulier en développement, où il laisse voir les trésors de son immense érudition.

Je tiens particulièrement à remercier M. Y pour son accueil, sa gentillesse et sa bonne humeur constantes, ainsi que pour le soin qu'il prend de son équipe et de l'ambiance d'Ethereum. Merci aussi pour ses attentions quotidiennes et sa générosité.

Remerciements encore à M Herbeth, très certainement le meilleur collègue qu'on puisse avoir.

Mille mercis à M. Forest pour sa patience devant le décès de ma machine, et sa gentillesse de m'avoir accordé du temps pour recommencer le présent rapport.

Je rends grâce aussi à M. Santos, costagiaire indeffectible, compagnon irremplaçable pour la solitude des heures d'avance le matin.

Remerciements enfin à Mlle Stéphane et M. De Gani pour leur présence amicale et rassurante dans le monde du travail.