

CHAOS DASHBOARD – RAPPORT DE PROJET

Eva BENHARIRA
Mathilde DUMOULIN
Guillaume HAERINCK
Laurine LAFONTAINE

IMAC 2 - Promotion 2021 - Projet de développement Web
Victor Diale



SOMMAIRE

[Spécifications](#)

[Introduction](#)

[I - Planification du chaos](#)

[A - L'origine du tout](#)

[B - Revêtement de la chapelle](#)

[C - Les armes de l'inquisition](#)

[D - Diviser pour mieux oeuvrer](#)

[II - Déroulement de la croisade](#)

[A - Les fondations du malheur](#)

[B - Pactiser avec le diable](#)

[C - L'enfer disponible pour tous](#)

[Conclusion](#)

[Annexe](#)

[Journal de progression](#)

SPÉCIFICATIONS

L'objectif de ce projet est de créer un site web de type dashboard. Un dashboard a pour but de structurer et de présenter un ensemble d'informations de façon claire et lisible. L'utilisateur peut éventuellement agir sur cette présentation.

Notre dashboard doit récupérer des données brutes grâce à une API externe et utiliser une librairie de data-visualisation pour les afficher. En outre, le site doit être construit par le biais d'un framework frontend moderne tel que HyperApp.

Le site est déployé à cette adresse: chaosdashboard.guillaumehaerinck.com

INTRODUCTION

Le paysage web est en proie à un cycle infernal de nouveautés et d'innovations. On pourrait couper une tête de la bête qu'une autre apportant son lot de nouveaux frameworks et d'acronymes irait pousser juste à côté. C'est au sein de ce cyclone que demeure pourtant une oasis de calme relatif.

L'oeil de ce cyclone, nous en sommes convaincu, se prénomme "disegno", ou de son nom non-latin, le "design". Bien qu'en apparence inoffensive, la constance de ce mirage a le pouvoir de troubler la motivation du plus ardu des développeurs et même de finir par aspirer son âme.

Armé de nos clavier en forme de crucifix inversés, nous nous sommes jurés de nous aventurer sur les traces de nos Anciens afin d'y trouver le secret de leur longévité. Vous pouvez nous accompagner dans cette aventure, mais il se peut que vous n'en ressortiez pas indemne.

I - PLANIFICATION DU CHAOS



"Il n'est pas de plus grande douleur que de se souvenir des temps heureux dans la misère." - **Dante, La divine comédie**

A - L'ORIGINE DU TOUT

Notre parti pris n'a rien d'innocent, nous sommes coupables d'avoir été séduits par une idée qui dorénavant nous habite et nous hante. "[Remember when frontend was fun ?](#)". Nous souhaitons voir **plus de folie et de surprise** de la part des développeurs. Nous souhaitons surfer sur le web en eau trouble, nous décharger du pouvoir de

voyance que nous possédons alors. Nous allons donc devenir des païens face au web moderne.

L'objet de notre dévotion se devait d'être aussi inhabituel sur la forme que sur le fond. Il s'agissait ainsi de marquer au fer rouge notre volonté et de prévenir quiconque s'approchait de notre terre promise qu'il s'agissait là d'un projet inhabituel. Nous avons donc décidé d'exploiter la **thématique du chaos**.

Avec un sujet aussi diabolique, la boîte de Pandore des interactions et des données farfelues est ouverte. En plus de présenter les informations comme sur un dashboard standard, nous avons souhaité **pousser les possibilités d'interactions** que nous offre le web moderne. L'utilisateur doit parfois chercher l'information via quelques mini-jeux. Cela nous permet d'apprendre ces technologies sous un nouvel angle et de tester leurs possibilités dans un bac à sable.

B – REVÊTEMENT DE LA CHAPELLE

L'identité visuelle du dashboard est tout droit inspirée des **sites des années 2000**, à contre-courant de ce qui se fait aujourd'hui en matière de design. Ce choix se traduit notamment par la présence de typographies style wordart, d'une grande quantité d'informations, de GIFs et d'animations. Il s'agit là dans l'imaginaire collectif d'une période d'ébullition en matière de design web. Nous rattacher à ce mouvement simplifie la présentation de notre parti pris à un éventuel public.

Sur cette base, nous sommes partis aux quatre coins du monde des API pour trouver le Graal. Les sites de la [World Bank](#), ainsi que de [l'ONU](#) furent une bonne base pour trouver le sujet de chacun de nos **7 widgets de data-visualisation**. À savoir le taux de suicides, de corruptions, de pauvreté, de crimes, de kidnappings, les morts les plus courantes ainsi que celles dues à l'utilisation de drogues.

Chaque widget a été pensé et designé en amont avant d'être injecté dans la [maquette figma](#), réalisée pour nous donner une référence à suivre en matière de design et de disposition de l'information. Chacun des widgets possède des visuels, contenus, appels d'API et animations qui lui sont propres.

C – LES ARMES DE L'INQUISITION

En consultant les missives portant sur les offres de stages en intégration web, il nous est apparu qu'Angular et React étaient omniprésents. Notre épique équipe d'aventuriers ayant un niveau hétérogène en matière de sadisme et d'intégration front-end, nous avons jeté notre dévolu sur le dénommé **React** que deux d'entre nous avaient déjà croisé au détour de sombres ruelles.

Dorénavant armés de quelque chose de plus que notre foi, il nous fallait identifier à qui adresser nos demandes de données. Bien que prometteurs, les sites de l'ONU et de la World Bank disposent d'un accès API assez limité, seul le téléchargement de données brutes de base de données est réellement complet. C'est [DBNomics](#), un agrégateur de données auprès de ces sources que nous avons choisi comme API.

En effet, nos demandes sont pressantes et nombreuses en temps de développement. Afin de ne pas fâcher notre nouveau dieu en atteignant son quota de requêtes journalières, nous envoyons nos requêtes localement. Un **serveur de Mock** nommé [Mockoon](#) s'occupe de nous fournir nos données en temps et en heure, et sans limite. Il passe le relai à la réelle API lors d'un build "production".

Concernant les librairies Javascript, nous en avons utilisé trois :

- [Anime.js](#) permet de réaliser des animations de toutes sortes,
- [Matter.js](#) est une librairie de physique, qui nous a notamment permis de réaliser le pendu,
- [Charts.js](#) est une librairie fournissant des graphiques.

D - DIVISER POUR MIEUX OEUVRER

Afin de réaliser au mieux notre dashboard, nous nous sommes répartis les tâches en choisissant **chacun un ou deux composants** sur lequel travailler. Cela permettait à chacun d'exprimer ses plus viles envies, et nous empêchait de marcher sur le diocèse d'un autre membre de l'équipe.

Qui dit travail dit organisation. Nous avons utilisé un **tableau Kaban** sur [Trello](#) pour suivre l'évolution des tâches et Facebook messenger pour communiquer. Il y a eu beaucoup d'échanges d'informations lors de la mise en route, notamment par le biais d'appels de groupe pour expliquer le fonctionnement et l'architecture aux plus néophytes.

Par la suite, il s'agissait principalement de régler les problèmes et de se mettre d'accord sur les choix techniques (ces derniers sont abordés dans la partie II).

II - DÉROULEMENT DE LA CROISADE



"Pour tromper le monde, ressemblez au monde" - **Shakespeare, Macbeth**

A – LES FONDATIONS DU MALHEUR

Les premiers travaux ayant eu lieux dans notre bâtisse avaient pour but de tester l'agencement et la communication entre les composants React. À savoir **comment passer une information** d'un parent à un enfant, d'un enfant à un parent et à de multiples sources à la fois. Il s'agissait aussi de comprendre les outils donnés au développeurs pour faire du **rendu conditionnel** et de se rendre compte de la différence de philosophie par rapport à Vue et Angular.

Parce qu'il y a peu de partage dans la présentation et le traitement de l'information entre nos composants, nous n'avons pas adopté un design "atomique" qui sied pourtant si bien à React. En contrepartie nos composants sont dits "**self-contained**". Ils atteignent parfois les 200 lignes, mais nous estimons qu'ils sont assez simples et compréhensibles pour ne pas avoir à être divisés outre mesure.

La question s'est néanmoins posée à un moment concernant l'utilisation du pattern "flux" pour la communication entre de multiples sources à la fois. Typiquement, via l'utilisation de la librairie Redux ou du Contexte React. Il s'est avéré cependant qu'aucun appel d'API n'était commun et que nos composants n'auraient pas besoin d'une communication avancée.

Le dernier point architectural mis en place a été, plus tardivement, le **routeur** pour gérer de multiples pages. Sa mise en place est somme toute classique.

B – PACTISER AVEC LE DIABLE

La cérémonie de mariage passée, il était plus que temps de rejoindre notre ami commun pour la nuit de noces. Bien que le terrain ait pu sembler rugueux de prime abord, nous avons vite pris nos marques et adopté un style en accord avec nos prudes moeurs. On annonce la couleur avec une "**state machine**" simplifiée qui nous permet simplement d'afficher à l'utilisateur quand le component charge, est chargé, ou a décidé de nous rendre la vie difficile. En l'état, il ne s'agit que d'un enum qui change le rendu du component entier. Simple et efficace.

On ajoute ensuite un peu de piquant à notre soirée en lançant notre appel à l'API de DBNomics. Comme énoncé précédemment, les appels étant différents entre les composants, ils se chargent eux-même de la récupération et du parsing des données avec leurs besoins propres. Il est bon de noter que nous prenons en compte l'**interruption des requêtes en cours** en cas de destruction du component. Ne pas le faire entraînerait des fuites de mémoires dans l'application.

Avec de nombreux jouets à notre disposition pour présenter les données, nous avons dû expérimenter un certain temps pour intégrer nos idées de la meilleure façon. L'exemple le plus probant de cette démarche est sûrement l'intégration des **animations** pour les crimes et l'usage de drogue. Quand il s'agit de présenter une illustration en mouvement, il est possible de passer par une librairie comme [Anime.js](#), utiliser un gif ou une vidéo où l'on sélectionne la frame, une animation svg, css ou encore l'[API animation des navigateurs](#). Nous nous sommes finalement arrêtés sur l'utilisation d'animations svg pour la grande liberté de formes qu'elles offrent et la simplicité d'utilisation.

Un dernier point qui nous semble utile de mentionner est l'utilisation du **State** de React. Par son fonctionnement asynchrone, nous sommes déjà tombé dans le panneau de la concurrence de données lorsque que plusieurs `setState()` sont utilisés dans le même tas de fonctions. La solution a simplement été d'utiliser le callback de cette fonction pour hiérarchiser nos appels, mais à l'avenir l'utilisation des mots clés `async` et `await` nous semblent plus judicieux.

C - L'ENFER DISPONIBLE POUR TOUS

C'est bien connu, l'enfer est ouvert à tous et ne discrimine rien ni personne. Parce que notre équipe à bon coeur, nous avons souhaité élargir notre fanbase des plus vils d'entre vous jusqu'au plus innocent. Ainsi que votre téléphone ait la taille d'un ordinateur, possède des cornes où soit capable d'éteindre la lumière pour vous parler en latin, vous pourrez accéder à notre site pour votre plus grand malheur.

La mise en page générale du site utilise **CSS Grid**, mais si le contenu à besoin d'être cassé pour revenir à la ligne nous intégrons un container **Flexbox** dans la grille. Avec cette approche, nous n'utilisons qu'une seule ligne de **media queries** pour adapter le contenu sur mobile. Une critique pouvant émerger est le manque de support de ces versions de CSS sur internet explorer. Nous sommes cependant d'avis que, parce que notre site est destiné à amener l'enfer dans la vie de nos utilisateurs, il est inutile d'enfoncer le clou ceux provenant d'internet explorer. Leurs âmes les a déjà quitté il y a longtemps.

Lorsque notre site se sent trop à l'étroit, il sacrifie sa sidebar pour un modèle plus raffiné et compact. Nous avons troqué l'éternel menu burger sur mobile pour une **barre navigation sur le bas**. Le peu de pages que présente notre site nous permet de faire ce choix. Il s'agit là sans aucun doute d'une victoire savoureuse pour tous les pouces de la planète.

CONCLUSION

Nous sortons grandis de ce périple en terre ennemie. Notre but étant d'expérimenter et de construire des interactions uniques, nous laissons une ardoise honorable. Il faut dire qu'il y en a eu pour tous les goûts et que cette absence de contrainte forte nous a poussé à trouver toujours plus de folie dans notre travail.

Si cette quête en elle-même était une source intarissable de satisfaction, c'est aussi dans le travail d'équipe et dans la collaboration qui a pris forme que nous nous sommes revigorés. De niveaux différents, il a fallu s'arranger, expliquer et jouer des appétences de chacun pour construire notre oeuvre qui restera sans aucun doute inégalée pour les siècles à venir.

PS : La musique des enfers a été créé par nous-même.

ANNEXE

JOURNAL DE PROGRESSION

Date	Description
02/04	Choix du sujet
05/04	Listing des API et premier état des technique
06/04	Attribution à chacun de deux API
08/04	Recherche mode présentation des idées
12/04	Chacun avance sur un de ses éléments (visuels + test appel API)
15/04	Avancer la mise en forme du site
20/04	Séance de suivi avec Victor
22/04	Mise au point sur les avancées et ce qu'il reste à faire (notamment côté CSS)
25/04	Dernier point
26/04	Déploiement du site
26/04	Rendu