

Mathieu Coutant, Iris Marjollet, Margot Pellegrin, Sarah Zakon

PROJET LAELITH

Copilot pour un éditeur de quêtes

Write your prompt here

Send

Your story will appear here...

Validate

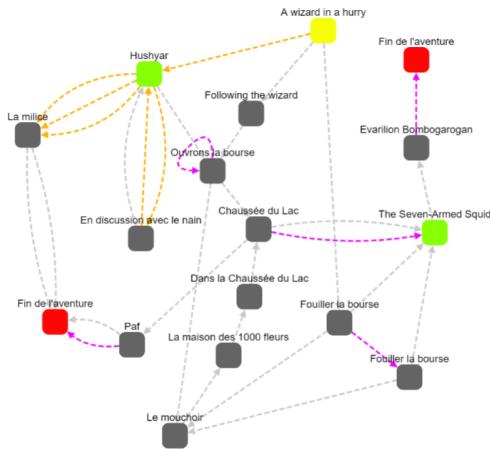


I. Présentation du projet	2
II. Contexte	3
Client	3
Public visé	3
Contexte technique	3
III. Cheminement	3
Situation initiale	3
Identification des problèmes	4
Nos inspirations	5
IV. Solutions	7
Solution CSS/design system	7
Barre de navigation	7
Conclusion	9

I. Présentation du projet

Notre projet consiste en l'amélioration d'un éditeur de quêtes dans le monde de jeu de rôle de Laelith. Nous avons apporté de nombreuses améliorations en termes d'UI et d'UX pour rendre l'utilisation de l'éditeur plus agréable dont la principale est l'ajout d'une IA génératrice pour faciliter l'écriture des quêtes.

Une quête est une "Histoire dont vous êtes le héros", elle consiste donc en une histoire de plusieurs petits chapitres à la fin desquels le joueur a plusieurs choix qui vont chacun le diriger vers un chapitre différent. La quête dans son ensemble peut alors être modélisée comme un graphe dans lequel chaque node représente un chapitre et les liens représentent les choix faits par le joueur.



Exemple de graphe représentant la quête "La bourse du magicien"

Pendant la résolution de la quête, le joueur doit donc parcourir ce graphe en partant du node de départ (ici en jaune) pour arriver à un node final (ici en rouge).

Le logiciel pour créer et modifier ces quêtes était déjà implémenté avant notre arrivée sur le projet, notre rôle a donc été de le faire évoluer pour le rendre plus accessible et plus facile d'utilisation.

Pour cela, nous avons effectué plusieurs modifications à l'éditeur :

- Intégration d'une IA génératrice qui permet de générer une quête cohérente à l'univers Laelith et adaptée aux attentes de l'utilisateur
- Centralisation de toutes les fenêtres de l'éditeur au sein d'un workspace
- Simplification de la navigation entre les différents nodes
- Implémentation d'un nouveau Design system

II. Contexte

Client

Notre client est la fondation Laelith. Laelith est une cité imaginaire qui a fait son apparition en 1985 dans le magazine Casus Belli, sous la plume de grands noms du jeu de rôle français. Aujourd'hui, la fondation a pour mission de protéger la propriété intellectuelle de l'univers tout en développant des outils qui permettent sa démocratisation, tel qu'un vaste ouvrage détaillant l'ensemble du lore de l'univers. Le site sur lequel nous avons travaillé fait partie de ces initiatives.

Public visé

Le public principalement visé par notre site est celui des joueurs de jeux de rôles. Le site de base permet aux utilisateurs de pouvoir découvrir l'univers de Laelith et de développer des quêtes. L'implémentation d'un copilot IA qui permet d'aider à l'écriture et au développement de quêtes permet ainsi au site initial d'atteindre un plus grand public, notamment pour les personnes qui débutent dans le monde des jeux de rôles et qui veulent commencer à développer leurs propres quêtes.

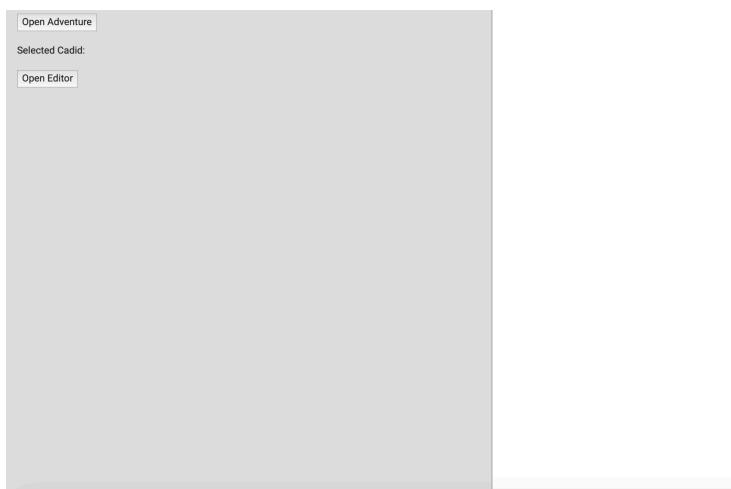
Contexte technique

La technologie utilisée pour notre site a été imposée par la structure du site déjà existant. Le backend du site a été codé en TypeScript et le front end en Svelte, un framework opensource. Pour l'IA, nous nous sommes servis d'une licence OpenAI fournie par la fondation. Nous avons communiqué avec l'API grâce à python.

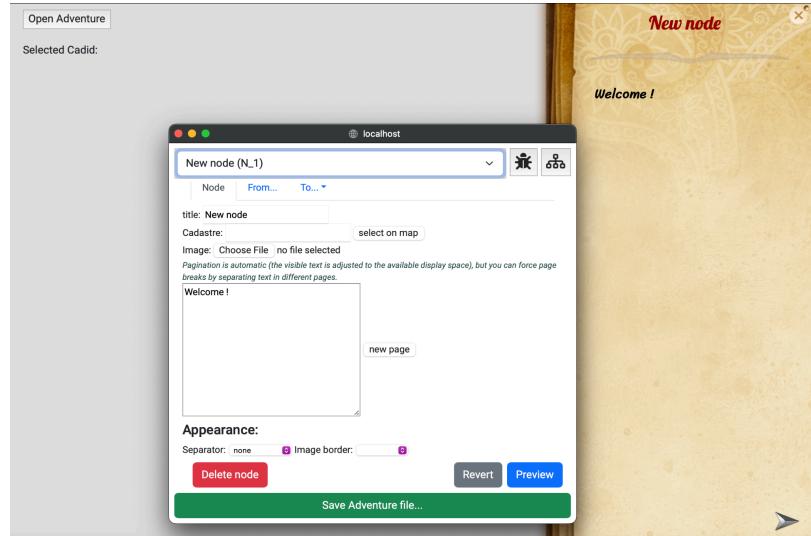
III. Cheminement

Situation initiale

Initialement, le site qui nous a été donné était à l'état presque brut avec un design non adapté à l'utilisation par le public. De plus, toutes les pages liées au site apparaissaient sous forme de popup destiné à une utilisation multi-écrans qui devenaient rapidement ingérables et contre-intuitives dès que trop de popup s'ouvraient en même temps.

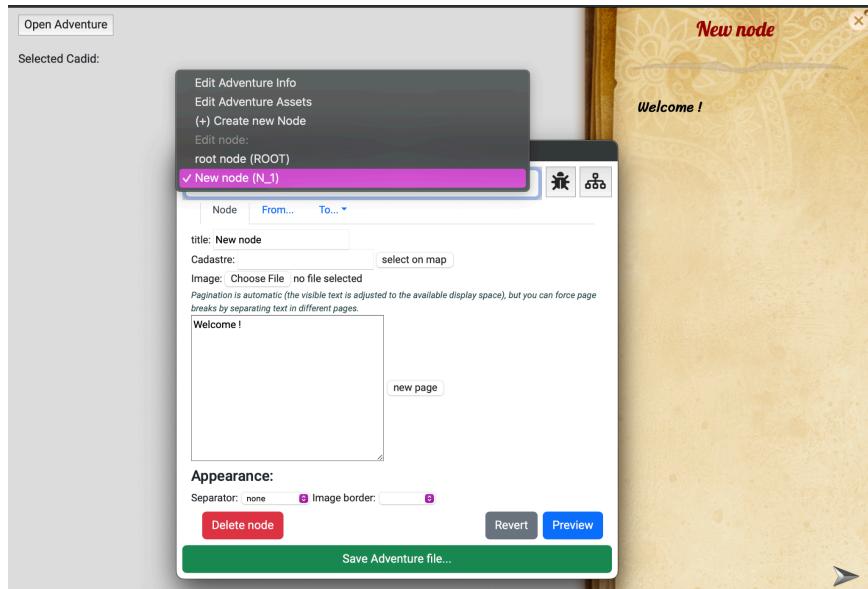


Page d'accueil original



Éditeur original

De plus, la barre de navigation entre les différents nodes ainsi que les différentes options d'édition était une barre déroulante dont les options n'apparaissaient qu'en cliquant dessus, ce qui rendait l'outil très peu intuitif et ralentissait l'usage.



Éditeur original

Identification des problèmes

Lorsque nous avons fait tester notre application la première fois sur cette version initiale de l'éditeur nous avons pu identifier plus précisément les problèmes de design que nous devions corriger.

Premièrement les utilisateurs ne réussissaient pas à créer de nouveau node car ils ne trouvaient pas le bouton. Ceci étant lié au fait que le bouton est caché dans un menu déroulant avec de nombreux autres boutons qui permettent notamment la navigation entre les nodes.

Les utilisateurs étaient rapidement submergés par le nombre de fenêtres qui s'ouvraient. Ils ne comprenaient pas quelle fenêtre était la principale ni les liaisons entre les différentes fenêtres. Comme certains boutons sur certaines fenêtres effectuent leurs actions sur une autre fenêtre, soit l'utilisateur ne comprenait pas ce que faisait le bouton, soit il changeait de fenêtre ce qui recouvrerait la précédente.

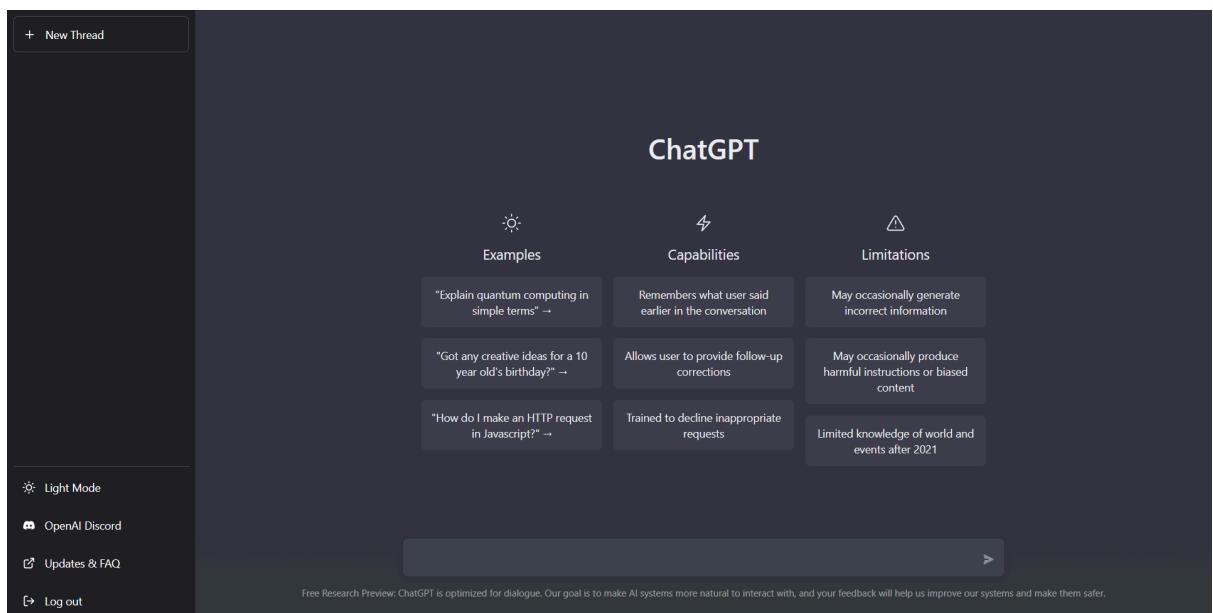
De plus, l'aspect front-end de l'application n'avait pas encore été implémenté. Il y avait donc de nombreux problèmes d'alignement, de position ainsi que de cohérence entre le design des différentes fenêtres.

Ce sont donc sur ces 3 points que nous devions travailler. Pour cela, nous avons implémenté des solutions qui ont eu pour but d'améliorer l'expérience utilisateur en plus de l'ajout du copilot.

Nos inspirations

Afin de définir un visuel le plus intuitif possible pour l'utilisateur qui permettrait de naviguer entre les différentes sections de l'éditeur, nous nous sommes inspirés d'autres sites ou logiciels d'édition et de leur mise en page.

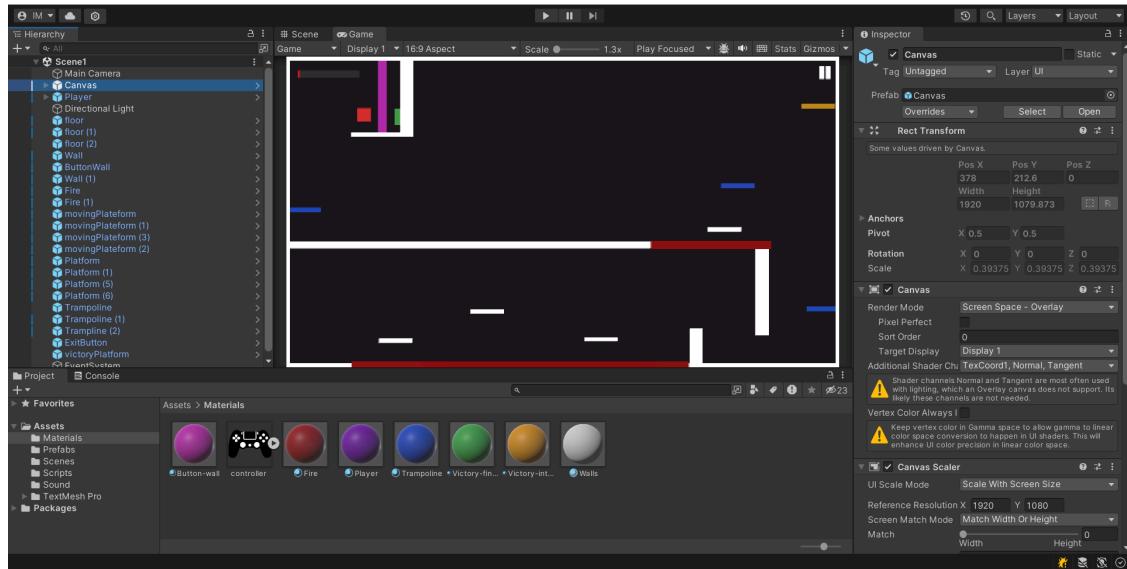
Une de nos principales inspirations est le site <https://chat.openai.com/>. Nous souhaitions utiliser la mise en page avec une section principale de l'éditeur et une section de navigation sous la forme d'une colonne située sur la gauche de l'écran. Cette barre pourrait ainsi permettre de naviguer dans l'éditeur, tout comme le site de ChatGPT permet de naviguer entre les différents canaux de communication.



Page du site de ChatGPT

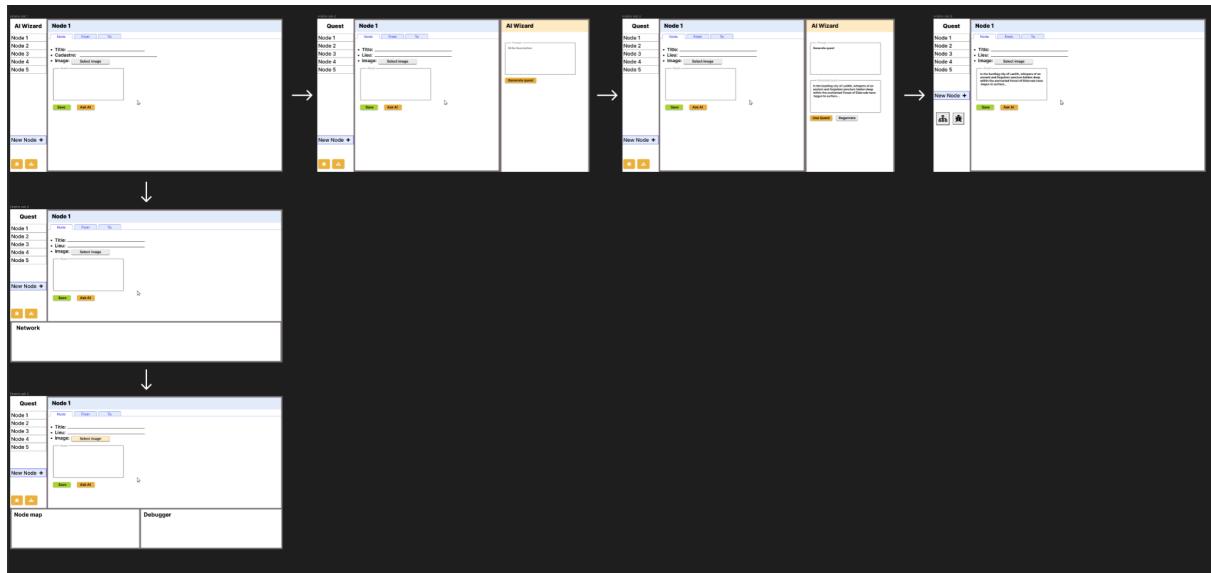
Afin de définir le nouveau visuel de notre éditeur de quête, nous nous sommes également inspirés des interfaces de logiciels comme Unity, avec un éditeur décomposé en

plusieurs sections. Nous sommes partis de ce modèle pour imaginer un éditeur qui aurait une mise en page avec une seule fenêtre, et chaque section pourrait s'ouvrir sur une partie de l'écran plutôt que sous la forme d'un popup.



Organisation en sections du logiciel Unity

À partir de ses inspirations nous avons créé une première maquette de notre site sur figma. Ceci nous a permis de visualiser les changements que nous voulions intégrer et c'est sur cela que nous avons basé nos modifications.

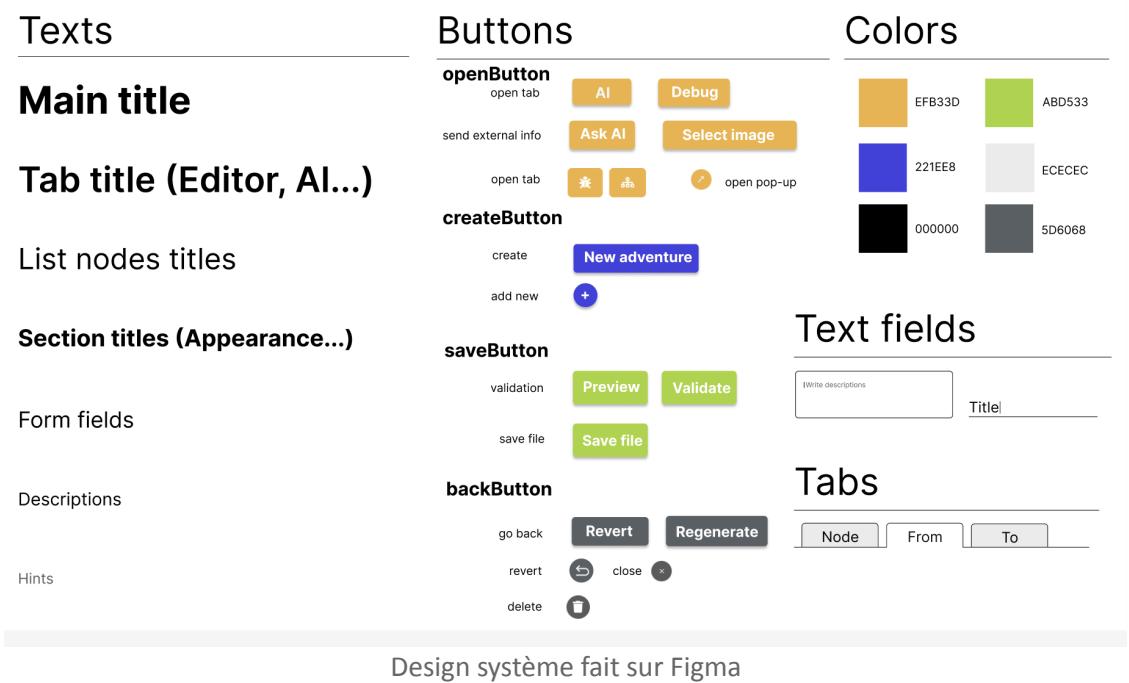


Maquette du site sur figma

IV. Solutions

Solution CSS/design system

Afin de moderniser l'UI nous avons tout d'abord développé notre Design system sur Figma :



Grâce à notre nouveau Design system, nous avons pu à la fois améliorer les visuels utilisés, mais également implémenter une cohérence au niveau des couleurs et des polices d'écriture.

Ensuite, dans le but de faciliter l'implémentation cohérente de notre design system, on a créé un fichier CSS à côté du projet qui contenait tous les types de boutons, textes fields, dans lequel on allait piocher lors de la modification de l'UI.

Barre de navigation

Afin de faciliter l'utilisation de l'outil et la navigation entre les différents nodes et différentes options d'éditions, nous avons implémenté un navbar sur le côté gauche de la fenêtre, ce qui permet à l'utilisateur d'avoir une vue d'ensemble de tous les nodes qu'il a créé ainsi que des différentes options auquel il dispose.

Nous avons aussi décidé d'ajouter à cette barre tous les boutons qui permettent d'ouvrir le *debugger*, d'ouvrir le réseau de nodes et le bouton sauvegarde que nous avons indiqué grâce à une communication iconographique à la place de textuelle comme avant.

Ainsi, nous avons regroupé tous les outils de gestions de fichiers inter-nodes au même endroit, facilitant ainsi l'utilisation.



d. Iframes

Éditeur de quêtes avec le copilote ouvert

Afin de répondre aux demandes de notre client qui préfèrent le format multi-écrans, nous avons adapté notre interface pour pouvoir popout nos iframes: en appuyant sur le bouton: ⚡ en haut à gauche de chaque iframe, l'iframe et le contenu s'ouvre dans une nouvelle fenêtre :

Éditeur de quêtes avec la fenêtre réseau ouvert

Enfin, afin d'afficher toutes les fenêtres dont l'utilisateur a besoin au même endroit, nous avons décidé d'ouvrir toutes les pages popup sous la forme d'iframes contenues dans la fenêtre initiale.

Ces iframes s'ouvrent facilement en appuyant sur le bouton "Ask wizard" (pour l'iframe du copilot) ou le bouton réseau (pour l'iframe du réseau).

The screenshot shows the 'Quest' application interface. On the left, there's a sidebar with sections like 'Nodes' and a list of nodes such as 'ROOT A wizard in a hurry', 'N11 Ouvrons la bourse', etc. The main area is titled 'Edit adventure info' and contains fields for 'Title' (La bourse du magicien) and 'Start Text' (Il m'est arrivé un truc...). Below these is a large network diagram window titled 'localhost'. The diagram shows nodes connected by various lines: a yellow node labeled 'A wizard in a hurry' connects to a grey node 'Following the wizard', which in turn connects to a grey node 'Ouvrons la bourse'. Other nodes include 'Fouiller la bourse', 'Hushyar - En discussion avec le nain', 'Le mouchoir', 'La maison des 1000 fleurs', 'Dans la Chaussée du Lac', 'La mille', 'La fin de l'aventure', and 'Paf'. A legend on the right side of the diagram window, titled 'Color Labels', lists categories with corresponding colors: end (red), +/- Beer (orange), start (yellow), main node (green), and +/- Lethal (purple). There are also buttons for 'Revert' and 'Preview' at the bottom of the diagram window.

Éditeur de quêtes avec la fenêtre réseau ouvert dans une page à part

Conclusion

Suite à l'implémentation de notre nouveau design nous avons effectué de nouveaux tests avec de nouveaux utilisateurs. Nous avons pu constater de nettes améliorations par rapport aux premiers tests. Les utilisateurs réussissent à créer de nouveaux nodes, à naviguer sur l'application, etc. Ces changements ont réellement permis de souligner l'importance de l'interface et d'un design system cohérent afin de guider au mieux l'utilisateur. Donner un sens aux choix esthétiques comme les couleurs de boutons ou séparations entre les sections permet d'avoir une interface plus claire et compréhensible d'un simple coup d'œil.

Grâce au design system détaillé sur Figma ainsi qu'à la page html listant tous les styles de boutons, zones de textes ou taille de police, le projet peut être continué en assurant une intégration des outils que nous avons mis en place.