

Conceptualisation

HangmanWeb a été développé en plusieurs étapes pour garder une approche claire et organisée. On a commencé par découper le projet en trois étapes : comprendre l'implémentation des routes, créer un prototype fonctionnel et améliorer l'interface graphique. Une idée importante, dès le début, a été de remplacer les menus texte traditionnels par des routes dynamiques. Ces routes ont défini la structure du jeu.

Sachant qu'on n'avait pas beaucoup d'expérience avec HTML et CSS, la priorité était d'avoir une version fonctionnelle du jeu avant de penser aux améliorations de design ou d'interface, si on avait le temps. Dès le départ, notre mot d'ordre était efficacité : avancer à chaque étape tout en laissant de la place pour ajuster si nécessaire.

Répartition des tâches

Pour répartir les tâches, on s'est appuyés sur nos points forts. On a commencé ensemble avec la page d'accueil où l'utilisateur entre son nom (pour le tableau des scores) et choisit un niveau de difficulté.

Ensuite, on a réparti les rôles :

- **Michel** a travaillé sur la recreation du jeu Hangman, tout en gardant les bases du projet précédent, et en ajoutant des nouveautés comme le choix des mots selon la difficulté.
- **Edwin** s'est occupé de la page des scores. Il a ajouté une fonctionnalité pour sauvegarder définitivement les noms et scores des joueurs dans un fichier .txt, une des exigences du projet.
- **Ensemble**, on a rapidement fini la page de fin, qui montre si le joueur a gagné ou perdu.

On a aussi créé trois listes de mots distinctes, une pour chaque niveau de difficulté, pour rendre le jeu plus intéressant. Cette répartition nous a permis de rester efficaces tout en jouant sur notre savoir-faire respectif.

Organisation et collaboration

Au début, on n'avait pas un plan très clair, ce qui a parfois causé des bugs ou des erreurs. Par exemple, ajouter des fonctionnalités sans en parler à l'autre pouvait créer des incohérences. Mais on a vite appris à mieux communiquer et à travailler ensemble. En utilisant des outils comme TeamViewer dans les dernières étapes, on a pu simplifier la

collaboration et résoudre des problèmes, comme des bugs qui faisaient apparaître la page de victoire trop tôt.

On a aussi utilisé des croquis et des diagrammes pour mieux comprendre les tâches complexes, comme la configuration des routes. Cela nous a permis à éviter des erreurs en écrivant le code.

Gestion du temps

La gestion du temps a été compliquée, surtout au début où on avançait lentement. On s'est d'abord concentrés sur la création d'une version fonctionnelle, en mettant de côté l'interface graphique pour plus tard. Vers la fin, quand le temps a commencé à manquer, on a accéléré le rythme. On a choisi un design simple et drôle, inspiré des mèmes, pour aller vite tout en rendant le jeu attrayant.

On a aussi ajusté notre manière de travailler : des sessions plus courtes et ciblées, et l'utilisation de Git pour garder des versions stables du projet. Si un bug survenait, on pouvait revenir facilement à une version précédente, ce qui nous a fait gagner beaucoup de temps.

Suivi des progrès

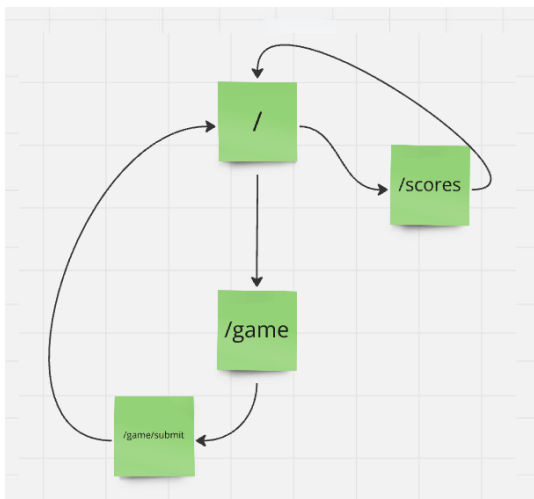


Figure 1 : Schéma de Routes¹

Pour suivre notre progrès, on a surtout mis sur une communication constante. À chaque modification, on se tenait informés, que ce soit par message ou en en parlant directement. Pour les tâches plus complexes, comme l'intégration des routes ou correction de bugs, les diagrammes étaient forts utiles¹. Ces outils visuels nous ont permis de mieux comprendre et résoudre les problèmes rapidement.

Même si notre suivi n'était pas très formel, il a bien fonctionné pour garder une vision claire et avancer efficacement. Nous avons également recours à des versions précédentes et stables pour réorganiser notre travail à partir de ce moment, tout en maintenant la transparence et en respectant les délais du projet.