

HANGMAN-WEB

COMMENT AVEZ VOUS COMPOSER LE PROJET ?

Tout d'abord venant on a la naissance du projet qui a beaucoup varier lors du mois on est passé d'un projet avec le thème Saw qui est une célèbre saga de film d'horreur sur un tueur au puzzle à un jeux style futuriste le projet s'est composé en plusieurs dossier nous avons au début suivi les consignes que nous nous étions donner à l'aide d'un Trello ainsi qu'un miro.

<https://trello.com/invite/b/670d4f9d29b65a51a64c5a06/ATTIdbc6b4d2999d04072fbdc7d2a105a5d4BED0279/hangman-web>

QUELS SONT ETE LES PHASES CLES ?

PHASE 1:

- nous avons commencer par choisir le projet qui été d'abord sur le theme de saw puis vers un monde futuriste.
- puis la planification des taches sur le trello
- ainsi que la répartitions des dossier et des fichiers dans l'architecture.

PHASE 2:

- réflexions sur l'interface de chaque page (en tout 4) page d'accueil, page de jeu
- création du main.go avec le serveur dispo sur le port 8080

- repris de notre hangman, page de score et pour finir la page qui vous demande si vous voulez rejouer ou quitter pour revenir au menu.

PHASE 3 :

- correction des potentiel bug et implantation du jeu du pendu (hangman)
- finalisation du fichier README et création du document.

REPARTITIONS DES TACHES

les taches ont été un peu mal répartie nous faisons un peu ce qui nous plaisais (ou nous étions le plus à l'aise) puis on s'appelais sur discord pour voir un peu comment on avancer. sinon en général c'est à peu près une page chacune

Comme stratégie nous avons le Trello pour nous guider ou chaque tache était indiquer

CONSTRUCTION DU PROJET :

Le développement de Hangman Web a commencé par une phase de planification minutieuse. La première étape consistait à mettre en place l'infrastructure de base du serveur Go. J'ai choisi d'utiliser le package "net/http" standard de Go pour sa simplicité et sa robustesse. La création du serveur a nécessité une attention particulière à la gestion des routes et des sessions utilisateur. J'ai implémenté un système de routage qui dirige les utilisateurs vers différentes pages en fonction de leur état de jeu : un joueur avec une partie en cours est automatiquement redirigé vers la page de jeu, tandis qu'un nouveau joueur est accueilli sur la page d'accueil.

La gestion des sessions a été l'un des défis majeurs. Il fallait trouver un moyen de maintenir l'état du jeu pour chaque utilisateur sans surcharger le serveur. J'ai opté pour une solution utilisant des cookies sécurisés et une map en mémoire pour stocker les états de jeu. Chaque session est identifiée par un ID unique généré de manière cryptographique, assurant ainsi la sécurité et l'unicité des sessions.

L'Implémentation du Jeu

La logique du jeu elle-même a nécessité une réflexion approfondie. Le système de sélection des mots a été conçu pour offrir une expérience équilibrée à tous les niveaux de difficulté. Pour le niveau facile, j'ai sélectionné des mots courants de 4 à 6 lettres, tandis que le niveau difficile propose des mots plus complexes pouvant aller jusqu'à 12 lettres. Chaque mot est stocké en majuscules pour simplifier la comparaison avec les entrées utilisateur, et j'ai mis en place un système de normalisation pour gérer les caractères spéciaux.

La vérification des lettres proposées par le joueur utilise une approche optimisée : chaque lettre n'est vérifiée qu'une seule fois et le résultat est immédiatement reflété dans l'interface. Le système maintient un tableau des lettres déjà essayées pour éviter les doublons, et chaque tentative incorrecte déclenche une animation fluide de l'image du pendu.

Le Design et l'Interface Utilisateur

L'aspect visuel du jeu a été particulièrement stimulant à développer. Je voulais créer une interface qui soit à la fois moderne et fonctionnelle. Le choix d'un thème futuriste avec des effets néon n'était pas seulement esthétique ; il servait aussi à guider l'attention du joueur vers les éléments importants de l'interface.

Pour créer les effets de néon, j'ai utilisé une combinaison de box-shadows et de text-shadows en CSS. Chaque élément interactif possède plusieurs couches d'ombres avec différentes couleurs et intensités, créant un effet de profondeur et de luminosité. Les animations sont gérées avec des keyframes CSS soigneusement calibrées pour être fluides sans être distrayantes.

Le responsive design a nécessité une approche méticuleuse. J'ai utilisé des media queries pour adapter l'interface à différentes tailles d'écran. Sur mobile, les éléments se réorganisent verticalement, et la taille des polices s'ajuste automatiquement. Les animations sont également optimisées pour les appareils mobiles, avec des versions simplifiées qui consomment moins de ressources.

La Gestion des Scores et des Données

Le système de scores a été conçu pour être à la fois équitable et motivant. Les points sont calculés selon une formule qui prend en compte plusieurs facteurs :

Le niveau de difficulté (multiplicateur ×1 pour facile, ×1.5 pour moyen, ×2 pour difficile)

Le nombre de vies restantes (bonus pour chaque vie épargnée)

La rapidité de résolution (bonus de temps dégressif)

La longueur du mot (points de base proportionnels au nombre de lettres)

Les scores sont sauvegardés dans un fichier JSON qui est mis à jour de manière atomique pour éviter toute corruption de données. J'ai implémenté un système de verrouillage de fichier pour gérer les accès concurrents, garantissant ainsi l'intégrité des données même avec plusieurs joueurs simultanés.

Les Défis Techniques Rencontrés

Un des défis majeurs a été la gestion de l'état du jeu en temps réel. Lorsqu'un joueur propose une lettre, plusieurs actions doivent se produire simultanément :

Vérification de la validité de la lettre

Mise à jour du mot affiché

Actualisation du compteur de vies

Animation de l'image du pendu

Mise à jour de la liste des lettres essayées

Sauvegarde de l'état de la partie

Tout cela doit se produire de manière fluide et sans latence perceptible. J'ai résolu ce défi en optimisant chaque étape du processus et en utilisant des techniques d'animation CSS efficaces.

voici les étapes pour parvenir a la fin du projet même si nous avons eu beaucoup de mécontentement dans la visions du projet.

Quel stratégie avez vous adopter pour vous documenter ?

pour nous documenter nous avons beaucoup utiliser YouTube ainsi que certains site internet et les slides du cours Golang pour la page d'accueil j'ai suivi un tuto YouTube par exemple puis modifier a ma guise pour que cela corresponde a la vision que j'avais.