Nguyen Khac Huy n°16293
Khazoum Noëlle n°16290
Musungayi Victor n°16282

Rapport de projet info

Introduction

Pour ce projet, nous avons décidé de faire une petite application sous forme de jeu interactif. Après avoir décidé d'un *Pseudo*, l'utilisateur est invité à choisir entre plusieurs décisions qui vont l'amener à des fins au contenu varié. Chaque choix de la partie est ensuite enregistré et un résumé du cheminement est affiché via la page *leaderboard* ainsi que toutes les autres parties qui ont été jouées depuis le début.
Etant un trinômes, une application kivy nous a été demandé. Celle-ci est constitué d'une simple interface qui affiche le nombre de parties jouées ainsi que le nombre de partie perdante et gagnante. Cette interface nous permet de voir facilement qui a gagné ou qui a perdu ainsi qu'une possibilité de supprimer ces données du site.

Python

Notre site comprend une multitude de routes qui permettent de rajouter un choix dans la base de données, mais en terme de routes ayant des fonctions différentes, nous en avons un petite dizaine (/index, /leaderboard, /choix*, /addchoix**, /admin, /login, /default, /getscores, /deletescores).

- *index* : page d'accueil qui permet d'accéder au leaderboard, page admin et de commencer le jeu
- *leaderboard*: page qui indique les choix pris par le joueur en forme de liste avec des phrases différentes en fonction du résultat. Prends en compte l'erreur KeyError qui s'affiche lorsque un joueur ne finit pas sa partie et veut accéder au leaderboard.
- choix* : page avec différents choix à prendre, redirige vers une nouvelle page choix.
- *addchoix** : Fonctions qui rajoute le choix dans la base de données.
- admin : page administrateur qui permet d'accéder à l'interface Kivy et de shutdown le serveur en cas de problème
- *login*: permet de vérifier que le mot de passe et le nom d'utilisateur est le bon
- *default*: indique un message lorsque la route est mal écrite
- *getscores, deletescore* : permet de récupérer et de supprimer des données grâce à kivy

Kivy

Notre interface nous permet de voir et de supprimer les données. Les parties gagnantes et perdantes sont séparées

- *loaddata* : permet de retourner les données et 2 listes, une gagnante et une perdante
- *result* : permet d'afficher les détails
- *delete* : permet de supprimer
- *redirecturl* : redirige vers le site

Html/CSS

Nous avons codé un vingtaine de page, dont plus de la moitié pour les choix. On a utilisé un fichier CSS (situé dans public/css) pour pouvoir agencer nos différents éléments de manières homogènes. Toutes le pages ont le même thème avec le texte en encadré et une image de fond. Seul le leaderboard possède un CSS différents car nous avons pris une image différents et donc les couleurs utilisées précédemment ne donnaient pas bien. Nous avons incorporés des boutons, des objets fixes, un fond, un agencement des textes, des images.

Utilisation

Pour exécuter ce programme, il suffit de lancer le fichier *server.py* situer dans le dossier. Après avoir accéder au localhost:8080, il sera demandé un pseudo que vous pouvez choisir et par la suite une série de choix vous seront proposée. A la fin de chacun des choix, des liens vous sont proposés qui redirigent soit vers l'écran d'accueil, soit vers la page du leaderboard.
Pour accéder à l'interface Kivy, il suffit d'accéder à la page *admin* supposé être accessible que par l'administrateur, il faut donc taper un mot de passe et un nom d'utilisateur. Le *username* et *password* à utiliser sont *admin* et *admin*. Sur l'interface, on sait voir et manipuler les données recueillies.
Si par hasard l'url de la route est mal écrite, il vous sera proposé un retour vers la page d'accueil.

Conclusion

Le site que nous avons fait utilise une multitude de procédés que nous avons appris en cours et mis en pratique lors de ce projet, que ce soit manipuler un dictionnaire, une liste ou bien gérer une erreur. Cet exercice nous a permis de tout mettre en commun et de devoir faire preuve de réflexion pour résoudre tous les bugs que nous avons rencontré.