

PROJET 11

AJOUTER UNE FONCTIONNALITÉ

Jean baptiste Servais

Nous avons amélioré 2 projets car le premier projet ne nous semblait pas assez conséquent. La première amélioration de projet est un ajout d'un jeu dans le projet 8 "Openfactfood". La deuxième amélioration est sur le projet labyrinthe. En effet nous avons robotisé dk.

Dans un premier temps il fallait répondre à un bug. Pour cela nous avons décidé de recréer un bug que nous avons eu et qui nous avait pris du temps.

Bonjour monsieur Niel,

nous venons de recevoir votre message concernant le bug sur l'application "Openfoodfact" que nous avons réalisé avec vous le 20 Mai 2019. Afin de vous rassurer, nous pensons que le bug est minime et qu'il sera corrigé le plus rapidement possible.

Nous pensons que c'est un bug lié à la base de donnée et plus précisément lié à une erreur d'encodage de caractère. En effet l'utilisateur ne trouve pas certains produit depuis la recherche d'aliment comme "légume vert" ou "népalé argenté".

De plus, nous avons bien pris en compte votre demande d'ajout de fonctionnalité "jeux" à l'application "Openfactfood". Nous vous le livrerons dans les plus bref délais.

En espérant répondre à vos attentes et en nous excusant du bug,
cordialement,

Jean Baptiste Servais

my_profil@hotmail.fr
0606060606

Pour cela nous avons dû créer plusieurs fichiers remplaçant les caractères spéciaux.
<https://github.com/pastrouveedespseudo/projet-8>

Bonjour monsieur Niel,

Nous venons de réparer le bug du 20 Mai 2019 sur l'application "Openfoodfact". En effet l'utilisateur, depuis la fonctionnalité "recherche" ne trouvait pas certains aliments pourtant bien présents comme légume vert ou "népalé argenté".

Le bug était lié à un erreur d'encodage en base de donnée. Nous avons donc dû créer plusieurs fichiers remplaçant les caractères spéciaux par un espace vide. A chaque mise à jour des produits le programme fait appel à ce fichier qui remplacera directement ces caractères spéciaux comme '\x0e' par un espace vide.

Nous travaillons actuellement sur la fonctionnalité "jeux".

En espérant avoir répondu à vos attentes et en nous excusant encore, cordialement,

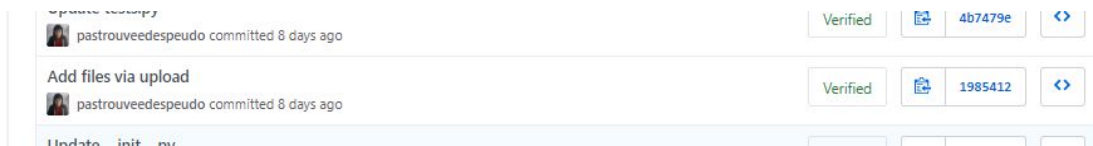
Jean Baptiste Servais

my_profil@hotmail.fr
0606060606

Commits on May 21, 2019

Update tests.py ...  pastrouveedespseudo committed il y a 12 jours Ajout des tests de la fonctionnalité "jeux" sur les pages et la sortie d'un produit de nutriscore A	Verified  a412345 
Update tests.py ...  pastrouveedespseudo committed il y a 12 jours Ajout de test concernant la base de donnée et les pages	Verified  f9f329b 
Update views.py ...  pastrouveedespseudo committed il y a 12 jours Ajout de la fonctionnalité "jeux" l'utilisateur à 2 niveaux. Il doit choisir le produit de nutriscore A parmi 3 à 7 autres produits, un feedback lui est alors renvoyé sous forme de message	Verified  7816b02 
Update caractere.py ...  pastrouveedespseudo committed il y a 12 jours Création d'un script de suppression du caractère '\xa0', cela produisait un bug dans la recherche d'aliment, par exemple légume vert	Verified  7fc2d6a 

Les commits suivants sont: "Update caractere.py" permettant de remplacer les caractères spéciaux par un espace vide.



Le commit upload est la mise en github de la fonctionnalité jeux.

L'application se trouve sur ce site: <https://openfactfood.herokuapp.com/>

La première amélioration de projet est donc un jeu où l'on affiche des images aléatoires de produit. Le joueur doit trouver pour le niveau 1 l'image correspondant à un nutriscore A. Nous lui affichons des messages selon si la réponse est juste ou fausse. Pour le niveau 1, 4 images sont affichées et pour le niveau 2 8 images sont affichées. Pour cela nous avons fait une nouvelle application « jeux » et différentes fonctions affichant les images par l'orm, et d'autres pour vérifier la réponse.

De fait, depuis la vue nous avons une fonction « jeux » qui fait appel à une fonction niveau qui choisit la place de l'image aléatoirement sur le template. Afin d'obtenir les images nous avons dû faire plusieurs autres fonctions.

Pour cela nous devons faire appel à jeux.py qui va chercher dans l'orm un aliment de nutriscore A et 3 à 7 autres aliments de nutriscore différents à A. Vérification_réponses() est une fonction qui vérifie le nutriscore selon le paramètre image de l'orm. En effet depuis la page HTML nous envoyons en POST l'image sur lequel le joueur a cliqué. Puis nous avons fait deux fonctions qui recommencent les opérations afin de pouvoir faire une continuation du jeu. Puis nous les affichons sur le template.

Enfin nous avons fait les tests afin de vérifier la base de donnée mais aussi le template (cf les commits intitulés test.py).

Comme ce projet ne nous semblait pas assez conséquent nous avons fait une deuxième amélioration de projet. Nous avons amélioré le projet dklabyrinthe et nous nous sommes basés sur le code du cours. <https://github.com/pastrouveedespseudo/p11>

Pour cela nous avons factorisé le code rangé dans des fonctions. Le but est que dk apprenne tout seul à aller au point d'arrivée tout seul.

Dans un premier temps il parcourt le labyrinthe avec des aléatoires par exemple droite droite haut gauche droite ect... Une fois qu'il arrive sur l'arrivée nous relevons le nombre de mouvement qu'il a fait. Puis nous enregistrons cela dans une database. A chaque essai si le nombre de mouvement est plus petit que la longueur du nombre de mouvement renvoyé par la database alors nous modifions la database par les nouveaux mouvements ainsi de suite. Si dk fait le même nombre de mouvement 100 fois c'est que c'est le plus petit chemin. A chaque fois qu'il réussit à faire le nombre de mouvement nous ajoutons dans un fichier une variable de + 1.

En résumé, au début dk par exemple fait 250 mouvements. Puis il en fait 100 puis 50 puis 5. A chaque fois nous modifierons la base de donnée de 250 à 50 coups. Puis comme 5 est le plus petit coup à réaliser la database ne sera plus jamais modifiée et le fichier contenant une variable s'incrémentera de + 1 à chaque fois que le nombre de mouvement sera de 5. A 100 dk se déplacera tout seul par une fonction qui dit: si la variable = 100 alors dk se déplace tout seul.

A chaque fois que dk fait un mouvement nous ajoutons à la liste globale LISTE_CHOICE .

Dans les fonctions toright(), toleft(), totop() et todown() nous indiquons si c'est un mur est ne déplaçons pas dk mais le mouvement est pris en compte pour la database. Nous lui indiquons aussi que s'il arrive sur S soit l'arrivée alors on return checkpoint qui arrête la boucle du jeu et active les fonctions de database. L'insertion en database si le nombre de mouvement réalisé par dk depuis len(LISTE_CHOICE) est plus petit que le len() de la database. Par la fonction visualisation() du fichier database.py sinon on le renvoie sur le menu. Si la longueur de LISTE_CHOICE est égale à celle de la database nous écrivons dans un fichier par la fonction write() ou nous incrémentons une variable.

Nous avons donc modifier le projet 3 en ajoutant un bot qui se déplace tout seul via la méthode de brut force qui n'est pas la façon la plus optimisée.