

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 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.

Bug

vous ✓

Bonjour monsieur Niel,

nous venons de recevoir votre message concernant le bug sur l'application "Openfactfood" 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 réparé dans un délais maximum de 2 jours.

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.

Nous avons donc mis en ligne la version précédente datant du 20 Juin 2020 afin de pouvoir travailler sur la version actuelle.

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 (2 semaines maximum).

En espérant répondre à vos attentes,

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>

Correction du bug

Bonjour monsieur Niel,

Nous venons de réparer le bug du 20 Mai 2019 sur l'application "Openfactfood". En effet l'utilisateur, depuis la fonctionnalité "recherche" ne trouvait pas certains aliment pourtant bien présent 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 mis à jour des produits le programme fait appel à ce fichier qui remplacera directement ces caractères spéciaux par un espace vide.

Nous avons remis la version actuelle en ligne à la place de la version du 20 Mai 2019. Bien Sûr cela n'a aucune influence sur l'expérience de l'utilisateur.

Nous travaillons actuellement sur l'application "jeux".

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









Jean Baptiste Servais

my_profil@hotmail.fr



0606060606

Branch: master ▾

Commits on May 21, 2019

Update tests.py ... pastrouveedespeudo committed 2 days ago	Verified		a412345	
Update tests.py ... pastrouveedespeudo committed 2 days ago	Verified		f9f329b	
Update views.py ... pastrouveedespeudo committed 2 days ago	Verified		7816b02	
Update caractere.py ... pastrouveedespeudo committed 2 days ago	Verified		7fc2d6a	

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

Update tests.py ... pastrouveedespeudo committed 8 days ago	Verified		4b7479e	
Add files via upload pastrouveedespeudo committed 8 days ago	Verified		1985412	
Update init.py ... pastrouveedespeudo committed 8 days ago	Verified			

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

Application jeux



Jean-Baptiste SERVAIS

Jeu 23/05/2019 11:16

Vous ▾

Bonjour monsieur Niel,

Nous venons de terminer la fonctionnalité "jeux" de l'application "Openfactfood" ainsi que les tests associés.

En effet vous nous avez demandé: "un jeux de 2 niveaux qui affiche des images aléatoires dont un des produits est de nutriscore A. Si l'utilisateur a juste alors un de ces messages [...] apparaît sinon, en cas d'erreur un de ces messages [...] apparaît."

Vous pouvez vous connecter sur le site. Cliquer sur l'icône de la barre de menu et jouer.

En espérant avoir répondu à votre demande,

Jean Baptiste Servais

my_profil@hotmail.fr

06066060606






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

juste 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 vus 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() et la deuxième fonctions 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 depuis la views 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. Nous avons sans faire exprès effacer macguyvert de notre github donc 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 apprend tout seul à aller au point d'arrivée tout seul.

Dans une 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 dans une database nous l'enregistrons. A chaque essais si le nombre de mouvement est plus petit à la longueur du nombre de mouvement de la database alors nous modifier 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 incrémentons dans un fichier une variable de + 1. Par exemple au début dk 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 modifié et le fichier contenant une variable s'incrémentera de + 1. A 100 dk se déplacera tout seul.

Pour cela nous avons gardé le menu ainsi que les images du cours et la génération du niveau dans les fonctions file(), menu() et generate_level().

A chaque fois que dk fait un mouvement nous l'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.