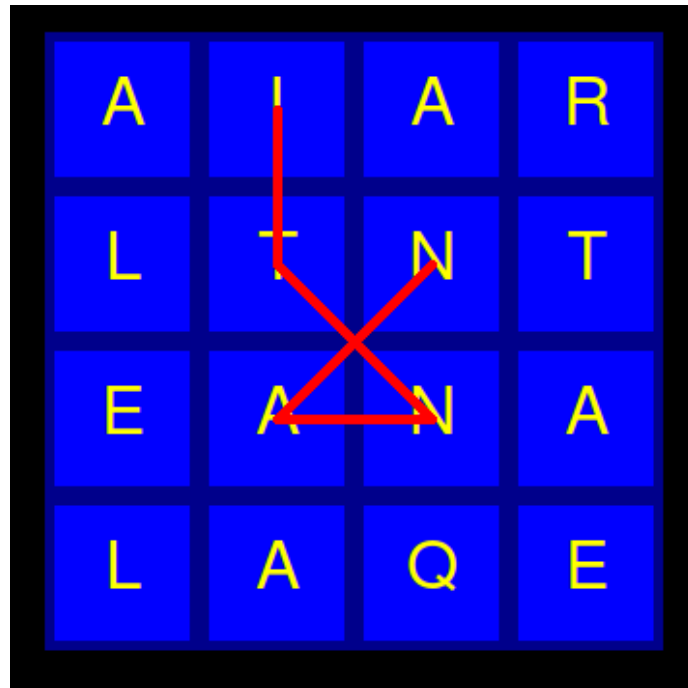


Projet "Jeu de mots"

Ce projet consiste à programmer un jeu de recherche de mots dans un damier.

Règles du jeu

L'interface est constituée d'un damier de 4x4 lettres tirées au sort aléatoirement¹.



Le jeu consiste à trouver des mots du dictionnaire dans ce damier en cliquant des cases contigües, orthogonalement ou diagonalement. On ne peut pas repasser sur une case déjà visitée durant le parcours du mot courant.

Dans ce damier, on peut par exemple trouver les mots SOFT, PESA, PARTIS, SOIF...

Etape 1 : projet de base

Vous devrez programmer le jeu de façon qu'un utilisateur puisse proposer des mots, qui seront validés ou pas en fonction de leur appartenance au dictionnaire français. Pour vérifier la validité des mots cliqués, vous chargerez en mémoire le dictionnaire fourni dans une structure python adéquate : un set.

Etape 2 : résolution

Au terme d'une partie, le jeu visitera automatiquement toutes les solutions possibles afin de présenter au joueur les mots qu'il aurait pu trouver. Cette partie pourra être réalisée de façon efficace...ou non. Tout travail sera considéré.

¹ Ce tirage au sort devra s'effectuer en tenant compte de la distribution des lettres en langue française. Vous pouvez par exemple utiliser la distribution des lettres au jeu de Scrabble.

Etape 3 : résolution optimale

Dans cette partie, vous mettrez en œuvre une structure de donnée arborescente afin de résoudre le jeu de façon optimale. Précisément, vous construirez un arbre de lettres (et nom de mots) permettant

- D'élaguer la recherche dès qu'un préfixe n'a aucune chance d'aboutir à un mot connu ;
- De déterminer la validité d'un mot en temps proportionnel à la taille du mot.

Avant d'entamer la programmation de cette partie, vous aurez intérêt à définir minutieusement une structure de donnée solide et pertinente.

Options

Tout ajout améliorant la jouabilité ou l'efficacité du jeu sera considéré. Exemples :

- Affichage d'un temps imparti
- Calcul du score du joueur en cours
- Mode compétition, où un même plateau est présenté deux fois à deux joueurs différents
- Sauvegarde automatique des scores avec le nom du joueur, rechargement au redémarrage

Votre travail

PENDANT LE DEROULEMENT DU PROJET

- Code python pas nécessairement objet (mais c'est conseillé). Utilisation de `tkiteasy` (ou de `tkinter`).
- Tous les paramètres des règles de jeu (taille du damier, temps imparti...) sont des variables globales en début de fichier, qu'on peut modifier facilement afin de changer le comportement général.
- Travail en **binôme obligatoire**.
- Vous suivrez la progression par étapes proposée dans ce sujet. La note maximale sera donnée aux projets qui atteignent la dernière étape.
- Toutes les fonctions doivent être documentées et commentées :
 - Un commentaire en début de fonction doit indiquer ce qu'elle effectue : quels paramètres reçoit-elle et que renvoie-t-elle ?
 - Les noms des fonctions et des variables doivent être judicieusement choisis.
 - Dans les endroits cruciaux du code, un commentaire doit expliquer les choix effectués.

UNE FOIS LE PROJET TERMINÉ

- Date de rendu des projets : **Samedi 25/3 21h**
- Vous déposerez votre projet **sur moodle**, sous la forme d'une **archive** au format **NOM1_NOM2.zip** où NOM1 et NOM2 sont les noms des deux membres du binôme. L'archive doit être débarrassée de tous les fichiers inutiles.
ATTENTION : le non-respect précis de cette consigne (mauvais nommage, mauvaise compression, dépôt en retard) entraîne une diminution de la note finale.
- Le projet doit être fonctionnel **SUR LES MACHINES DE L'UNIVERSITE**. À vous de tester votre projet sur ces machines avant de le rendre.
- Dans votre archive, vous joindrez un document **README** indiquant succinctement (une page) le travail accompli, les problèmes rencontrés, les bugs connus, les idées originales...etc.

Evaluation

Des soutenances en binôme auront lieu à la fin du module.

Les **compétences** qui seront évaluées sur ce projet seront :

- *Respect du cahier des charges "dépôt" (deadline, README, archive nettoyée et au bon format, programme fonctionne sans erreurs) : 1pt*
- *Qualité du code (commentaires, nommage, organisation générale) : 1pt*
- *Respect du cahier des charges minimal du programme (étape 1) : 5pts*
- *Respect du cahier des charges amélioré (étape 2) : 5pts*
- *Respect du cahier des charges avancé (étape 3) : 5pts*
- *Incorporer des options et solutions techniques innovantes : soyez imaginatifs ! : 3pts*

À l'oral, vous présenterez votre travail en étant capable de montrer que chaque membre du groupe a compris le fonctionnement général du programme. La note finale pourra être modulée en cas d'investissements inégaux des membres du groupe, ou bien de plagiat avéréⁱ.

Et maintenant ?

Vous pouvez (devez !) commencer votre projet dès maintenant, en respectant absolument les priorités suivantes :

- Choisir un.e binôme
- **Avant de coder quoi que ce soit, réfléchir ensemble aux structures de données à mettre en oeuvre**
- Se répartir le travail en vous mettant bien d'accord sur les spécifications de ce que chacun.e réalise

Un canal dédié au projet a été créé sur Discord.

Bon travail ! Amusez-vous bien...

ⁱ Les codes de l'ensemble de la promotion seront soumis à [JPlag](#), qui calcule un coefficient de similarité.