

## INFO-H-100 : Introduction à la programmation

### Projet 2 : TopWords

---

Le but du projet est d'implémenter en *Python 3* le jeu *TopWords* (marque déposée)<sup>1</sup>.

Ce jeu est une variante du très connu Scrabble. Il se joue avec 2 à 4 personnes et le but est de marquer le plus de points possibles en plaçant les meilleurs mots sur le plateau de jeu.

Le plateau de jeu est une grille de 10 sur 10 (soit 100 cases au total) dont les quatre cases centrales sont colorées. Il y a 100 lettres en tout dans un sachet.

Chaque joueur dispose d'un chevalet contenant 7 lettres tirées aléatoirement du sachet. Au premier tour et après avoir joué, le joueur tire aléatoirement des lettres dans le sachet afin de compléter son chevalet. Les mots sont placés de gauche à droite ou de haut en bas. Le premier mot placé doit passer par l'une des quatre cases colorées au centre du plateau de jeu. Les mots suivants doivent s'appuyer sur des mots déjà placés. Cela peut se faire de l'une des deux manières suivantes :

- Le nouveau mot est lié à (et réutilise) une ou plusieurs lettres d'un mot déjà posé sur le plateau. La ou les lettres déjà présentes sur le plateau ne doivent donc pas être présentes dans le chevalet du joueur.
- Certaines lettres du nouveau mot se superposent à (c-à-d écrasent) une ou plusieurs lettres déjà placées sur le plateau, de manière à changer le(s) mot(s) préexistant(s).

Le joueur qui parvient à placer 7 lettres en posant un mot reçoit un bonus de 20 points.

Chaque lettre placée représente 1 point ; les lettres empilées se valorisent à hauteur de la pile. Soit 3 lettres superposées correspondent à 3 points.

## Programme à réaliser

On vous demande de réaliser un jeu *TopWords* en Python. Votre programme devra se dérouler comme suit :

1. Votre programme chargera en mémoire le dictionnaire fourni sous forme d'un fichier texte et initialisera le sachet de lettres (le nombre d'exemplaires par lettre étant indiqué dans un autre fichier texte). Ces deux fichiers sont disponibles sur la page wiki du cours<sup>2</sup> et devront obligatoirement être lus à partir du répertoire dans lequel se trouve votre programme. Leur format est détaillé ci-dessous.
2. Ensuite, un tirage aléatoire de 7 lettres dans le sachet sera effectué pour initialiser le chevalet du joueur.
3. Pour placer un mot, le joueur entre : i) le mot complet (même s'il utilise des lettres présentes sur le plateau), ii) sa direction (horizontale ou verticale) ainsi que, iii) la position de la première lettre du mot à placer sur le plateau. L'origine du plateau se trouve en haut à gauche et les indices des positions commencent à 0. Le premier mot doit obligatoirement passer par l'une des quatre cases centrales colorées.
4. Une fois le mot entré par l'utilisateur, le système doit vérifier les conditions suivantes (voir aussi les exemples concrets donnés à l'avant-dernière page) :
  - (a) Le mot existe dans le dictionnaire fourni. Les diacritiques sont ignorées.
  - (b) Le mot entré ne dépasse pas les bornes du plateau.
  - (c) Le mot et sa position sont compatibles avec les lettres actuellement sur le plateau de jeu. Donc, si une nouvelle lettre écrase une lettre préexistante, le mot préexistant ainsi modifié doit exister aussi dans le dictionnaire.
  - (d) Le joueur possède les lettres nécessaires pour jouer son mot.
5. Si les conditions sont vérifiées, le mot est placé sur la grille et les lettres nécessaires sont retirées du chevalet. Les points de ce nouveau mot sont calculés en sommant :
  - les points liés au mot-même, en calculant la somme des lettres, y compris les lettres (non changées) se trouvant déjà sur le plateau de jeu ;
  - les points de tous les mots perpendiculaires qui ont été *affectés* par le placement des lettres du nouveau mot. C-à-d que les mots perpendiculaires qui ne sont pas transformés en un autre mot par le placement d'une lettre ne sont pas comptés.Nous ne vous demandons pas de prendre en compte les autres règles officielles (cf. Annexe) qui ne sont pas explicitement mentionnées dans l'énoncé.
6. Si le sachet n'est pas vide, le système tire aléatoirement dans le sachet le nombre de lettres consommées par le placement du mot (pour arriver à un total de 7 lettres, ou moins si le nombre de lettres restantes ne le permet pas) et les ajoute au chevalet. Sinon, la partie est terminée.

---

1. Voir [http://www.toutpourjouer.ch/data/files/manual\\_5048\\_fr.pdf](http://www.toutpourjouer.ch/data/files/manual_5048_fr.pdf) pour les règles complètes.

2. <http://cs.ulb.ac.be/public/teaching/infoh100>

## Description des fichiers textes joints

Le fichier `french.dic` contient tous les mots acceptés dans le jeu *TopWords*. Chaque ligne de ce fichier contient un mot en majuscule.

Les jetons de lettres et leur nombre disponible durant la partie sont contenues dans le fichier `french.let`. Chaque ligne se compose de la manière suivante : la lettre suivie du nombre de jetons de cette lettre. Extrait du fichier :

```
D 3
E 15
F 2
G 2
```

## Apports personnels

Les apports personnels seront valorisés à hauteur de 3 points sur 20. En d'autres mots, un projet parfait sans apport personnel aura une valeur de 17 points sur 20.

Voici quelques exemples d'apports personnels possibles :

- Le mode multijoueur (difficulté moyenne)
- Gestion de la lettre "Qu" (difficulté faible)
- Affichage amélioré grâce à l'Art ASCII<sup>3</sup>
- Séparer le code en au moins deux modules (deux fichiers `.py`). Le premier module contiendra les fonctions logiques du projet (**Le modèle** : vérifier qu'un mot est dans le dictionnaire, ajouter un mot au plateau de jeu, etc...). Le second module contiendra les interactions avec l'utilisateur (**La vue** : c'est à dire l'affichage et les entrées) et importera le premier module. C'est donc ce second module qui sera exécuté pour lancer votre jeu *TopWords*.
- Interface graphique (difficile)

## Délivrables

Vous devez déposer une version de votre code sur le serveur Git se trouvant à l'adresse `http://wit-projects.ulb.ac.be/rhocode/INFO-H-100/Projets_2013/Projet-2`.

**Chaque fonction devra être correctement documentée.** C'est-à-dire pour chaque fonction : son prototype, une description succincte de la tâche qu'elle remplit, ses paramètres et leur domaine (le type et les valeurs possibles), le type de ses valeurs de retour et leur domaine et un exemple d'utilisation. Par exemple<sup>4</sup> :

```
def days_in_month(month, year):
    """Calcule le nombre de jours d'un mois.

    Arguments:
        month (int) : le mois considere, entre 1 et 12.
        year (int) : l'annee consideree.

    Valeurs de retour:
        int. Retourne le nombre de jours associes au mois et a l'annee considerees.
        Retourne -1 si les parametres ne respectent pas les domaines.

    Exemples:
        >>> print days_in_month(2, 2012)
        29
        >>> print days_in_month(13,2012)
        -1
    """
    <<code de la fonction>>
```

Rappelez-vous que la *docstring* est le texte qui sera affiché lorsque vous tapez `help(days_in_month)` dans l'interpréteur ; il s'agit donc de la documentation nécessaire à un **programmeur** pour *utiliser* la fonction. Ce texte ne s'adresse pas à l'utilisateur du *programme* (le joueur), et n'explique pas le fonctionnement *interne* de la fonction ; il explique uniquement **comment utiliser** la fonction.

## Consignes

Le projet se fera obligatoirement par groupes de deux étudiants de la même série.

La date limite de remise du projet et le **dimanche 16 mars 2014**. Tout projet rendu (placé sur le dépôt) après cette date sera sanctionné.

---

3. [http://fr.wikipedia.org/wiki/Art\\_ASCII](http://fr.wikipedia.org/wiki/Art_ASCII)

4. Basé sur [http://packages.python.org/an\\_example\\_pypi\\_project/sphinx.html#full-code-example](http://packages.python.org/an_example_pypi_project/sphinx.html#full-code-example)

Le code source sera rendu via le serveur *GIT*, de la même manière que pour le premier projet. On vous demande cette fois de créer votre dépôt dans le groupe `INFO-H-100>>Projets_2013>>Projet-2`. Le nom du dépôt devra être `<netid1>-<netid2>`, où les deux *netids* doivent être dans l'ordre alphabétique du nom de famille. Par exemple, `asilovy-mwaumans`.

Notez que vous ne devez pas placer les deux fichiers fournis (`french.dic` et `french.let`) sur le dépôt ; uniquement votre code source.

## L'évaluation

Vous défendrez votre projet lors d'une défense orale que vous passerez devant les assistants et qui se déroulera comme suit :

- Vous devrez faire *individuellement* une petite modification du code ; celle-ci sera évaluée par les étudiants-assistants (~5 min) et a pour objectif de vérifier que chacun connaisse et maîtrise son propre code.
- Vous passerez ensuite *en groupe* chez les assistants qui vous intervieweront et vous poseront des questions sur votre projet. Un feedback vous sera également donné (~10 min).

Les défenses auront lieu avant les congés de printemps. L'organisation détaillée des horaires de défenses sera effectué ultérieurement en coordination avec vos délégués.

Les **critères d'évaluation** sont les suivants :

- La résolution du problème énoncé.
- Le respect scrupuleux de l'énoncé.
- Le respect des conventions et des règles de bonnes pratiques<sup>5</sup>.
- L'efficacité de vos algorithmes et structures de données. Si vous sortez du cadre de la matière vue au cours ou aux séances d'exercices, nous demandons que les nouveaux éléments du langage soient bien comprise et que leur utilisation dans votre programme soit maîtrisée. (Attention : vous pouvez être interrogé sur cette matière "volontairement étendue" également lors de votre défense.)

## Autres aspects pratiques

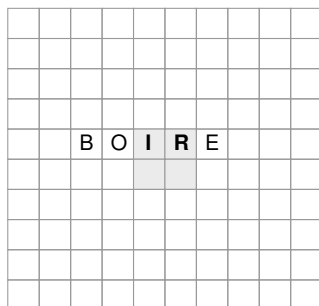
Les élèves assistants sont disponibles à 12h30 dans la salle Socrate afin de répondre à vos questions. Consultez leur horaire sur la page web des TPs.

---

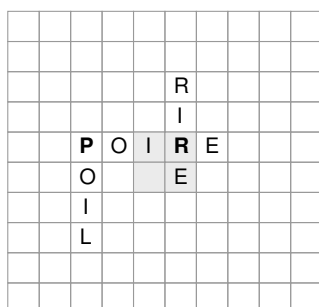
5. [http://cs.ulb.ac.be/public/\\_media/teaching/infoh100/info-h-100\\_-\\_regles\\_de\\_bonne\\_pratique.pdf](http://cs.ulb.ac.be/public/_media/teaching/infoh100/info-h-100_-_regles_de_bonne_pratique.pdf)

## Quelques précisions relatives aux règles du jeu

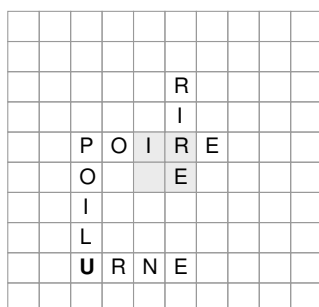
**Note :** Même si elle s'écartent éventuellement des règles officielles (cf. page suivante), ce sont *nos* règles qui prévalent !



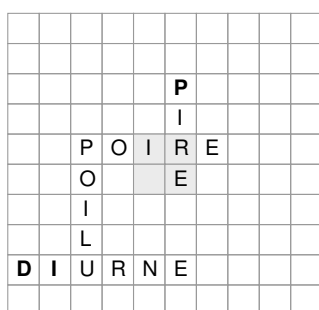
- Le premier mot placé doit passer par l'une au moins des 4 cases centrales grisées.
- Les mots ne peuvent être placés qu'horizontalement, de gauche à droite, ou verticalement, du haut vers le bas.



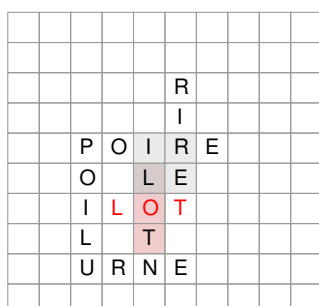
- Chaque nouveau mot doit s'"appuyer" sur un mot déjà placé sur la grille. Il y a deux possibilités :
  1. le nouveau mot (RIRE) s'appuie sur une lettre d'un mot déjà placé (BOIRE) et ne modifie pas celui-ci. Dans ce cas, le nouveau mot rapporte 4 points.
  2. le nouveau mot (POIL) s'appuie sur un mot existant (BOIRE) et le **transforme** (en POIRE). Ce nouveau mot rapporte 9 points (4 pour POIL et 5 pour POIRE).



- Pour le deuxième cas, on peut également transformer un mot existant *perpendiculaire* en l'**étendant** (POIL devient POILU). Le nouveau mot (URNE) ainsi placé rapporte 9 points (4 pour URNE et 5 pour POILU).



- On ne peut également étendre un mot existant "parallèlement". Exemples de ce que le programme doit aussi permettre :
  - placer un P à la place du premier R dans RIRE pour le transformer en PIRE
  - ajouter horizontalement DI avant URNE pour le transformer en DIURNE.



- Un nouveau mot placé ne peut "toucher" d'autres mots perpendiculaires que s'il les étend. Sinon, il faut au moins une case vide. Exemples :
  - on ne peut pas placer les lettres LOT horizontalement pour faire ILOT, car le mot vertical RIRET qui étendrait le mot vertical RIRE n'existe pas ;
  - on ne peut pas placer non plus ces mêmes lettres LOT verticalement, car le mot ILOT toucherait (en bas) le N de DIURNE ; le mot ILOTN n'existe pas.

## Annexe : Règles officielles



### BUT DU JEU

Etre le joueur totalisant le plus grand nombre de points en fin de partie.

### CONTENU

1 plateau de jeu Topword (en 2 parties), 1 plaque colorée à poser au centre du plateau, 1 base pivotante, 4 pieds en caoutchouc, 1 feuille d'autocollants, 4 supports pour les lettres, 100 lettres :

A x 9	H x 2	O x 6	V x 2
B x 2	I x 8	P x 2	W x 1
C x 2	J x 1	Qu x 1	X x 1
D x 3	K x 1	R x 6	Y x 1
E x 15	L x 5	S x 6	Z x 1
F x 2	M x 3	T x 6	
G x 2	N x 6	U x 6	

### ASSEMBLAGE

Assemblez le plateau de jeu comme indiqué *Figure 1*.

1. Insérez les 4 pieds en caoutchouc dans les trous de la base pivotante.
2. Emboîtez les deux parties du plateau.
3. Attachez le plateau sur la base pivotante.
4. Placez la plaque colorée de 4 cases au centre du plateau.
5. Collez les autocollants sur les côtés du plateau.
6. Assurez-vous que le plateau tourne bien.

### PREPARATION DU JEU

1. Posez le plateau et étalez les lettres, faces cachées, au centre de la surface de jeu. Assurez-vous que tout le monde peut atteindre le plateau et les lettres.
2. Chaque joueur prend un support pour poser ses lettres et le place devant lui, de manière à ce que personne d'autre que lui ne puisse voir ses lettres. Puis il prend 7 lettres au hasard et les pose sur son support.
3. Choisissez un joueur pour marquer les points. Ce joueur doit être équipé d'un papier et d'un stylo. Il servira aussi d'arbitre si besoin.

### DEROULEMENT DU JEU

Pour savoir qui commence à jouer, prenez chacun une lettre au hasard sur la surface de jeu. Le joueur qui tire la lettre la plus proche du « A » commence à jouer, puis la partie continuera dans le sens des aiguilles d'une montre.

### A Votre Tour

Créez un nouveau mot sur le plateau de jeu : c'est aussi simple que cela !

Pour commencer une partie, le premier joueur doit poser un mot horizontal ou vertical, dont au moins une lettre doit se trouver sur l'une des quatre cases colorées au centre du plateau.

A chaque tour, vous devez créer un nouveau mot de l'une des deux manières suivantes :

- Liez votre nouveau mot à une lettre d'un mot déjà posé sur le plateau (sur le principe des mots croisés).
- Empilez une ou plusieurs lettres sur d'autres lettres déjà placées sur le plateau, de manière à changer le mot préexistant.

**Remarque :** vous ne pouvez pas empiler une lettre sur une lettre identique, par exemple poser une lettre « A » sur une lettre « A ».

Quand vous avez posé votre mot, comptez vos points (voir section Comptage des Points), puis remplacez les lettres que vous avez utilisées par de nouvelles lettres tirées au sort (Vous devez toujours avoir 7 lettres sur votre support, sauf s'il n'y a plus de lettres à piocher.)

### Mots Autorisés

1. La règle de base est la suivante : si le mot est dans le dictionnaire, il est autorisé. Cependant, les mots suivants ne sont pas autorisés :

- Noms propres (noms de lieux ou de personnes)
- Abréviations et acronymes
- Mots contenant une apostrophe
- Mots contenant un trait d'union
- Préfixes et suffixes utilisés seuls.

Gardez un dictionnaire à portée de main pour vérifier les mots dont vous n'êtes pas sûr ou les mots contestés. Cependant, vous n'avez pas le droit de chercher dans le dictionnaire des idées pour composer des mots avec vos lettres !

2. Les mots doivent être posés verticalement ou horizontalement, mais pas en diagonale.
3. Dès que vous créez un mot ou changez un mot existant, les mots nouvellement formés doivent être des mots autorisés (voir section *Mots Autorisés*).
4. **Attention :** quand une pile est constituée de 5 lettres, vous ne pouvez pas empiler d'autres lettres dessus.

### COMPTAGE DES POINTS

Lorsque votre tour est terminé, comptez vos points comme expliqué ci-après (voir section *Créer des Mots*, pour les exemples) :

- Comptez 1 point pour chaque lettre du mot que vous avez créé lors de votre tour.

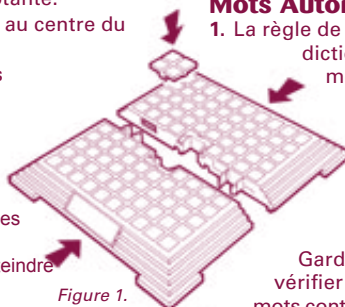


Figure 1.

