



Projet 4 NSI : Mini-Scrabble "Scrabble-Python"

1. Contexte du projet

Le Scrabble est l'un des jeux de lettres les plus populaires au monde. Votre mission est de créer un programme capable de calculer le score d'un mot saisi par l'utilisateur, de vérifier si ce mot existe dans un dictionnaire officiel et de sauvegarder les meilleurs scores (High Scores) dans un fichier.

2. Objectifs pédagogiques

- **Types Construits** : Utilisation de dictionnaires pour stocker les points de chaque lettre.
 - **Algorithmique** : Parcours de chaîne de caractères et calcul de cumul de points.
 - **Persistance** : Lecture d'un dictionnaire de mots (fichier `.txt`) et gestion d'un fichier de records.
 - **Validation** : Vérifier l'appartenance d'un élément à une liste de référence.
-

3. Travail demandé

A. Modélisation du barème

Vous devez utiliser un dictionnaire Python qui associe chaque lettre à sa valeur.

- *Exemple* : `{"A": 1, "E": 1, "B": 3, "C": 3, "K": 10...}`.

B. Gestion des fichiers

1. **Le dictionnaire (`mots_authorized.txt`)** : Un fichier contenant une liste de mots valides (un par ligne). Le programme doit charger ces mots dans une liste au démarrage.

Exemple : **Contenu du fichier `mots_authorized.txt` (contenant 50 mots)**

ARBRE
AVION
BATEAU

2. **Les records (`scores.txt`)** : Un fichier stockant les meilleurs scores réalisés au format `nom;score`.

4. Interface Utilisateur (Menu)

Le programme doit proposer les options suivantes :

1. **Calculer le score d'un mot** : L'utilisateur saisit un mot. Le programme vérifie s'il est dans le dictionnaire, puis calcule et affiche son score.
2. **Afficher le barème** : Afficher combien de points rapporte chaque lettre.
3. **Tableau des records** : Afficher les 5 meilleurs scores enregistrés.

4. **Ajouter un mot** : Permettre d'ajouter un nouveau mot au fichier `mots_authorized.txt`.
 5. **Quitter.**
-

5. Phases de travail

◆ Phase 1 : Le Moteur de Calcul (Socle)

- **Initialisation** : Créer le dictionnaire des points (A=1, B=3, etc.).
- **Logique** : Développer une fonction qui parcourt un mot et calcule la somme des points.
- **Interface** : Demander un mot à l'utilisateur et afficher immédiatement son score.

◆ Phase 2 : Validation et Dictionnaire (Approfondissement)

- **Fichiers** : Charger le fichier `mots_authorized.txt`.
- **Vérification** : Avant de calculer le score, vérifier si le mot existe dans la liste chargée.
- **Robustesse** : Gérer la casse pour que "Chat" ou "CHAT" soient acceptés de la même manière.

◆ Phase 3 : Gestion des Records (Extension)

- **Persistance** : Créer un système de sauvegarde des meilleurs scores dans `scores.txt`.
 - **Algorithmique** : Afficher le "Top 5" des joueurs après chaque partie en triant les données.
 - **Bonus de jeu** : Ajouter une option "Mot compte double" qui multiplie le score par 2.
-

6. Critères d'évaluation (/20 points)

Catégorie	Critère	Points
Calcul du score	Algorithme de parcours de la chaîne et cumul des points.	5 pts
Vérification	Recherche efficace du mot dans la liste des mots autorisés.	4 pts
Fichiers	Lecture du dictionnaire et sauvegarde des records.	6 pts
Interface	Qualité des menus et gestion des mots non valides.	3 pts
Code	Clarté, fonctions et docstrings et commentaires.	2 pts

7. Livrables attendus

1. Le fichier Python `scrabble.py`.
2. Le fichier `mots_authorized.txt` (contenant au moins 50 mots).
3. Le fichier `scores.txt`.
4. Diapo