

# Projet Langage C

---

## *Implémentation du jeu du pendu en langage C avec l'interface graphique libsx*

---

### Préambule

Ce projet est réalisé par Khady DIOP et Alex MEURILLON, 2 étudiants en ELSE3 à Polytech Nice-Sophia. Durant 2 mois, nous avons réfléchi et réalisé un jeu du pendu écrit en langage C. Dans ce document, vous retrouverez la présentation et le fonctionnement général du jeu ainsi que ses originalités.

### Présentation générale

Le projet HangMan\_C est une implémentation du jeu du pendu en langage C avec une interface graphique réalisée avec la bibliothèque libsx. L'architecture du projet est organisée autour de plusieurs fichiers .c et .h, chacune dédiées à une partie précise :

- `hangman.c` : `main.c` du programme. Il initialise l'affichage graphique, la partie de jeu, puis lance la boucle principale.
- `data.c` / `data.h` : Gestion de la logique du jeu (structure de la partie, gestion des erreurs, validation des lettres, détection de fin de partie, etc...).
- `readFile.c` / `readFile.h` : Lecture des dictionnaires (`dico_fr.txt`, `dico_uk.txt`), création de la liste de mot entre 6 et 8 lettres et sélection aléatoire du mot à deviner.
- `liste.c` / `liste.h` : Implémentation d'une liste chaînée générique utilisée pour stocker les mots du dictionnaire.
- `vue.c` / `vue.h` : Gestion de l'interface graphique (création et placement des widgets, affichage du pendu, mise à jour de l'affichage selon l'état du jeu).
- `callbacks.c` / `callbacks.h` : Définition des callbacks pour les interactions utilisateur (saisie de lettres, choix de la langue, sélection de la difficulté, affichage de l'aide, etc.).
- `Makefile` : Automatisation de la compilation et du nettoyage du projet.
- `dico_fr.txt` / `dico_uk.txt` : Dictionnaires de mots en français et en anglais.
- `README.md` : Fichier texte contenant des instructions pour l'utilisateur.

## Fonctionnement général

- Initialisation : Au lancement, le programme ouvre l’affichage graphique, initialise une nouvelle partie (par défaut en français), puis construit l’interface utilisateur (cf. Interface graphique)
- Chargement du dictionnaire : Le dictionnaire choisi est lu ligne par ligne, chaque mot (de 6 à 8 lettres) est stocké dans une liste chaînée.
- Affichage du mot à trouver : Au lancement d’une partie, les lettres du mot à trouver sont remplacées par ‘\_’. Au fur et à mesure que l’utilisateur trouve des lettres, celles-ci sont affichées
- Sélection du mot : Un mot est tiré aléatoirement depuis la liste pour être le mot à deviner.
- Boucle de jeu : L’utilisateur saisit des lettres via l’interface. Chaque lettre écrite par l’utilisateur est vérifiée : si la lettre est correcte, elle est révélée, sinon, une erreur est comptabilisée et le pendu se dessine progressivement.
- Gestion de parties : Le jeu gère les victoires, défaites. Il permet aussi de relance une nouvelle partie, de changer de langue et de configurer du niveau de difficulté (nombre d’erreurs autorisées).
- Interface graphique : L’interface propose des boutons pour accéder au menu, à l’aide, pour rejouer ou changer la difficulté, et affiche dynamiquement l’état du mot et du pendu.

## Originalités de notre programme

- Difficulté réglable : L’utilisateur peut choisir le niveau de difficulté, ce qui ajuste le nombre d’erreurs autorisé.
- Erreurs intelligentes : Lorsque l’utilisateur tape une lettre déjà utilisée et fausse, le jeu ne le prend pas en compte et n’incrémente pas le nombre d’erreurs. Plus besoin de réfléchir aux lettres sélectionnées, il suffit de jouer !
- Interface colorée : Les boutons sont représentés en couleur ainsi que la zone d’écriture.
- Entrée standard sur le clavier : Pour se différencier des jeux du pendu qu’on retrouve sur le Web, la sélection des lettres se fait grâce aux claviers.
- Choix de difficulté variable : Si vous vous rendez compte qu’au bout de 2 erreurs, la difficulté du jeu est trop importante, vous pouvez augmenter votre nombre d’erreurs pour la partie en cours !

## Conclusion

Grâce à ce projet, nous avons appliqué les connaissances acquises au cours de l’année en langage C. Nous avons utilisé des outils de partage comme Github afin de centraliser nos codes. Ce projet permettra de nous valoriser dans nos futurs entretiens d’embauche.