**Enoncé**

**Titre du projet :** Tu le C

Vous allez devoir réaliser le jeu du pendu en langage C. Votre programme devra choisir un mot parmi la liste de mots du fichier « words.txt », en annexe, qui doit être deviné par le joueur.

Voici les règles à respecter :

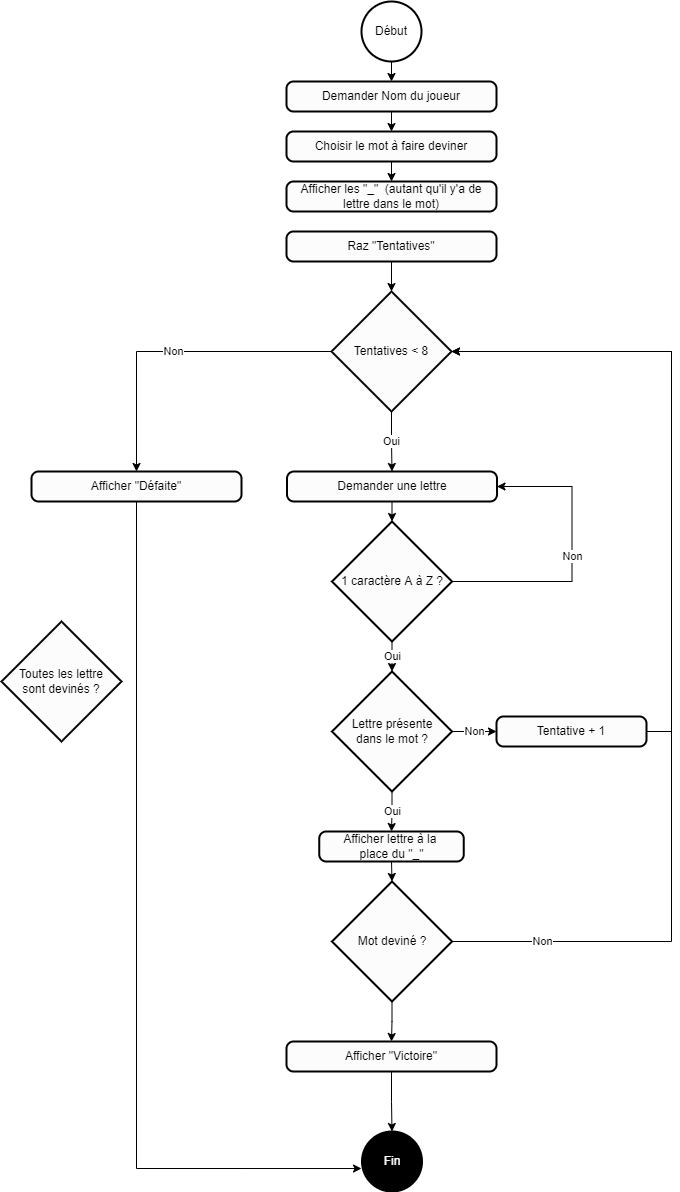
* Demander le nom du joueur
* Demander une lettre à la fois
* 8 tentatives maximum
* Afficher le « dessin » du pendu au fur à et mesure des tentatives
* Afficher un « \_ » pour chaque lettre du mot à deviner, et afficher la lettre à la place du « \_ » quand celle-ci est devinée
* Quand le joueur propose une même lettre déjà tentée précédemment, cela n’augmente pas le compteur de tentative (ne fait pas avancer le jeu)

Voici à titre d’exemple, l’échafaud final à la 8ème tentative :

\_\_\_\_  
 | |   
 | o   
 | /|\   
 | |  
 | / \  
 \_|\_  
 | |\_\_\_\_\_\_  
 | |  
 |\_\_\_\_\_\_\_\_\_\_|  
 `

A vous de « dessiner » les 7 autres étapes en retirant petit à petit des éléments de l’échafaud.

Voici un diagramme expliquant le déroulement du jeu :



**Déroulement**

Le projet se réalise seul.

|  |
| --- |
| Etapes |
| Lancement du projet, conception de la partie graphique, demander le nom du joueur, lancer la partie |
| Demander la lettre, système de tentative |
| Intégration du fichier words.txt, mot random |
| Finalisation, tests et présentation du projet |

**Conseils**

Si vous souhaitez améliorer le « rendu » graphique de votre jeu vous pouvez utiliser le « ASCII ART », voici un exemple : <http://patorjk.com/software/taag/#p=display&f=Big&t=PENDU>

Comme tous les jeux, celui-ci fonctionne sur une boucle infinie avec une gestion d’état :

* **Initialisation** : la boucle demande le nom du joueur
* **Lancement** : le nombre de tentative est à zéro, la boucle demande une lettre à deviner
* **Gain / Perte**: la boucle prend fin et affiche « victoire » ou « défaite »

Le jeu est gagné quand l’ensemble des lettres sont devinées. Nous vous conseillons d’utiliser 3 tableaux de caractères :

* 1 tableau qui contient le mot à deviner lettre par lettre
* 1 tableau de taille identique qui ne contient que des « \_ » à la place des lettres
* 1 tableau à taille dynamique qui stocke toutes les lettres tentées (bonnes et pas bonnes) afin de ne pas les compter à nouveau

Le jeu n’affichera alors que le tableau contenant des « \_ ». A chaque lettre correctement devinée, le « \_ » sera à remplacer par la lettre devinée.

Au début du projet, ne vous souciez pas du fichier « words.txt », renseignez un mot « en dur » dans votre programme avec lequel vous pourrez faire toutes sortes de tests. C’est seulement à la fin du projet que vous irez chercher un mot au hasard dans le fichier « words.txt »

**Bonus**

Stocker le nom des joueurs dans un fichier avec leurs scores respectifs, prévoir un écran qui affiche le TOP 10

**Pièces annexes**

Words.txt : Liste de mots