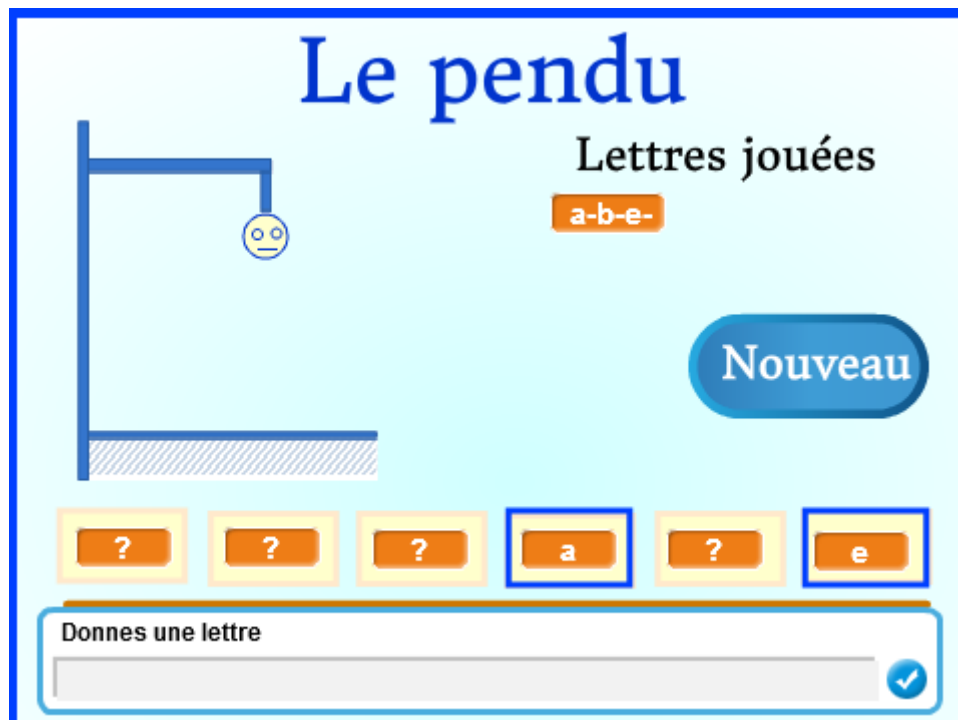


Le jeu du pendu



Le principe

L'ordinateur tire au hasard, un mot de 6 lettres dans une liste de mots.

C'est le mot secret à trouver.

Le joueur propose une lettre.

Si celle-ci existe dans le mot, elle est affichée dans tous les emplacements où elle apparaît dans le mot.

Si elle n'existe pas dans le mot, le pendu commence à apparaître.

Si le joueur a trouvé toutes les lettres du mot, avant que le petit personnage ne soit complet, il a gagné, sinon il a perdu.



Nous allons construire petit à petit ce projet.

Nous commencerons par mettre en place la mécanique du jeu, sans nous préoccuper de la partie visuelle.

Une fois que cette mécanique sera bien rôdée, que nous aurons vérifié que tout fonctionne, alors nous y ajouterons les éléments visuels.

Pendu version 1

Objectif : Placer dans un mot qui sera affiché, les lettres proposées par le joueur, aux emplacements qu'elles occupent dans le mot secret.

Matériel :

Un lutin « **Principal** » avec 2 costumes « gagné » et « perdu » : fournis.

Un arrière-plan **fond1** : fourni

Les variables du projet

A la création des variables, prenez soin de les déclarer locale ou globale, comme indiqué dans le texte. Cela permettra une fois le jeu complet, de ne voir dans chaque lutin, que les variables dont il a besoin.

« **motSecret** » (locale) contenant le mot à deviner.

« **lettresJouées** » (locale) en mode « grande lecture » contenant les lettres proposées par le joueur.

« **motAffiché** » (globale) le mot tel qu'il apparaît au fur et à mesure qu'on trouve les bonnes lettres du mot secret. Au départ ce mot affiche 6 points d'interrogation.

« **pos** » (locale) variable qui indique la position d'une lettre dans le mot secret ou dans le mot affiché.

« **temp** » (locale) variable de travail permettant de préparer le mot à afficher, avant de l'afficher.

Le scénario

Au démarrage, les variables sont initialisées aux valeurs adéquates.

Le lutin demande à l'utilisateur, le mot secret qui sera mémorisé dans la variable « **motSecret** ».

Créer un bloc « **Initialisation** » pour mettre en place toutes les initialisations

Créer un bloc « **TraiterLettre** » qui vérifie si la lettre fournie par le joueur est présente dans le mot secret.

Si oui, la lettre jouée est affichée dans « **motAffiché** » aux endroits où elle apparaît dans le mot secret. Cette lettre remplace un ou plusieurs points d'interrogation.

Dans tous les cas, la lettre jouée est affichée dans la variable « **lettresJouées** » suivi d'un «-».

Pendu version 2

Objectif : Détecter si le jour a gagné

Repartir de la version précédente.

Nous devons maintenant savoir si le joueur a trouvé le mot ou non.

Si le joueur a trouvé le mot, dans le mot affiché, il n'y a plus de point d'interrogation.

Au moment où nous traitons la lettre jouée, nous pouvons compter combien il reste de point d'interrogation dans le mot affiché.

Variable supplémentaire

« **nombre ?** » (locale) qui comptera le nombre de points d'interrogation restant dans le mot affiché.

Le scénario

S'il n'y a plus de point d'interrogation dans le mot affiché, alors le joueur a gagné.

Le lutin principal bascule sur son costume « gagné » et le programme

s'arrête :



Pendu version 3

Objectif : Détecter si le joueur a perdu

Repartir de la version précédente.

Nous devons savoir si le joueur a perdu.

Il a 8 essais : un essai n'est comptabilisé que si le joueur propose une lettre qui n'apparaît pas dans le mot secret.

Variables supplémentaires

« **essais** » (locale) qui comptabilise le nombre d'essais.

« **lettreCorrecte** » (locale) qui si elle est à 1, indique que la lettre jouée existe bien dans le mot secret. Sinon elle est à 0.

Le scénario

Le compteur d'essais sera initialement positionné à 8.

Il sera décrémenté à chaque fois que le joueur donne une lettre n'existant pas dans le mot secret.

Lorsque le nombre d'essais est à 0, le lutin principal bascule sur son

costume « perdu » et le programme s'arrête :



Pendu version 4

Objectif : Tirer le mot secret au hasard dans une liste de mots et structurer proprement le code dans le lutin principal.

Partir de la version précédente.

En ce point, la mécanique du jeu fonctionne. Nous allons rajouter une liste de mots de 6 lettres et tester le jeu, lorsque l'ordinateur tire un mot au hasard dans cette liste.

Les variables supplémentaires

« **ListeMots** » (locale) liste de mots de 6 lettres : pour cette version, mettre : acajou, balcon, centre, cerise, crépon, action, foudre, grelot

Le scénario

L'ordinateur tire un mot au hasard dans la liste et le joueur doit trouver le mot secret.

Lorsque le programme fonctionne, créer deux blocs :

« **BonneRéponse** » dans lequel sont placées les instructions qui traitent le cas où la lettre jouée est correcte.

« **MauvaiseRéponse** » dans lequel sont placées les instructions qui traitent le cas où la lettre jouée n'est pas correcte.

Si le joueur a perdu, alors il faut afficher le mot secret.

Pendu version 5

Objectif : Ajouter une zone d'affichage des lettres de la variable « **motAffiché** ». Cette zone sera gérée par un lutin « **Secondaire** »

Partir de la version précédente.

Notre jeu fonctionne, nous allons lui rajouter des éléments graphiques.

Changer l'arrière-plan du jeu : mettre fond5.svg.

Ajouter un lutin « **Secondaire** » et importer son costume : secondaire.svg

Cacher le lutin : clic sur le (i) en haut à gauche du lutin et décocher la case montrer.

Le lutin « **Secondaire** » est chargé d'afficher chaque lettre de « **motAffiché** » dans des variables **locales** c1, c2, c3, ...c6 toutes en mode « grande lecture ». Chacune de ces variables est placée dans un des rectangles jaunes situés au-dessus de la barre marron.

Il réalise cet affichage, lorsqu'il reçoit le message « **MiseAjour** ».

Mettre au point cette version : les bonnes lettres doivent s'afficher dans les variables c1, c2, ...c6 au cours du jeu.

Si le joueur perd, le mot secret s'affiche aussi dans ces variables.

Pendu version 6

Objectif : Améliorer l'affichage des lettres du mot affiché, afin de mettre en évidence les cases du mot occupées par une lettre.

Partir de la version précédente.

Si la lettre affichée dans une des variables c1, c2, ... c6 n'est pas un point d'interrogation, alors le lutin « Secondaire » « estampille » son costume, de façon à mettre cette lettre en évidence.

Créer un bloc « **Encadrer** » dans le lutin « Secondaire » qui a un paramètre « **lettre** ».

Si lors de l'appel de la procédure, la lettre fournie n'est pas un point d'interrogation, alors le lutin se place sur la case correspondant à la variable c1, c2, ...ou c6 à encadrer et estampille son costume.

La position de départ (centre du rectangle le plus à gauche) en $X = -184$,
en $Y = -90$.

L'écart entre le centre de deux cases consécutives est de 74.

Pendu version 7

Objectif : Finaliser notre jeu, en y ajoutant enfin notre petit pendu.

Partir de la version précédente.

Importer les deux lutins : Pendu et Nouveau.

Le lutin « Nouveau » lance un nouveau jeu. Le script de ce lutin est déjà prêt.

Le lutin « Pendu » affiche son costume « étape0 » lorsqu'un nouveau jeu démarre.

Il passe au costume suivant, lorsque le joueur a proposé une mauvaise lettre.

Le jeu est terminé, il ne reste plus qu'à ajouter des mots dans la liste de mots.

Pendu version 8

Dans cette version, la liste de mots contient 192 mots de 6 lettres.

Il n'y a pas de verbe, ni d'adjectif, ni de mot au pluriel.

Il peut y avoir des accents et des ç.