Projet Eden

(Idée de Pendu)  
langage : JAVA

**Introduction**

Nous nous proposons de programmer un jeu basé sur le célèbre principe du pendu que tout le monde connaît : Un mot est à deviner. Seules la premières et la dernière lettre sont affichées et les chances sont limités pour proposer les bonne lettres qui composent le mot par le nombre de traits qu’l faut pour dessiner le « pendu » (9 par défaut). En cas de défaite, un mini-jeu de snake permet, s’il est réussi par le joueur, de regagner une vie.

En lançant l’application le joueur arrive sur la page d’accueil qui est en fait un menu où il choisit l’un des modes de jeu suivants.

**Mode de jeu classique (2 joueurs)**

Chaque joueur propose un mot de longueur de caractères identique pour plus d’équité. Le nombre de caractères apparait sous forme de tiret « \_ » (si le mot est un mot composé, le trait d’union est déjà présent). Tour à tour, un joueur propose une lettre pour le mot de son adversaire, si cette lettre est présente une ou plusieurs fois, elle apparaît sur le mot du joueur qui l’a proposé autant de fois qu'elle se trouve dans le mot. Si un joueur a proposé une lettre qui n’est pas présente dans le mot, le premier trait du pendu s’affiche. Le jeu se poursuit jusqu’à ce qu’un des joueurs trouve le mot qui lui a été donné, ou jusqu’à ce qu’un des joueurs soit éliminé en ayant complété le dessin du pendu.

Au niveau graphique, la lettre déjà proposée est grisée sur le clavier virtuel. *(Peut être modifié)*

Si un des joueurs n’a plus de « chance » pour proposer une lettre, un mini jeu de snake apparait, lui permettant ainsi de récupérer une chance supplémentaire.

**Mode de Jeu contre CPU**

Il n’y a qu’un seul joueur.

Un Base de Données est mise en place. Cette Base contient un ensemble de mots, qui seront choisis aléatoirement par l’ordinateur.

Lorsqu’ une partie débute, un nombre d’Under scores égal à la longueur du mot sélectionné seront affichés.

De plus, la première et la dernière lettre du mot sont systématiquement dévoilées.

L’utilisateur a accès à un clavier virtuel, sur lequel il peut cliquer sur la lettre de son choix. Cette dernière sera affichée plusieurs fois si elle apparait plusieurs fois dans le mot. Sinon, le bouton de la lettre cliquée sera barré et une partie en plus du pendu sera affichée.

Dans tous les cas, le bouton de la lettre qui vient d’être appuyée sera grisé.

Si le joueur gagne, un petit message de victoire sera affiché.

Un pendu intégralement dessiné sera synonyme de défaite.

**Mode contre la montre**

Dans ce mode les deux joueurs doivent deviner un même mot proposé aléatoirement par le jeu. Les deux joueurs jouent chacun sur un écran différant et doivent choisir les lettres qu’ils pensent que le mot contient. Si celle-ci si trouve, elle s’affiche dans le mot sinon le dessin du pendu évolue

Deux fins de jeu sont alors possible :

L’un des joueur découvre le mot et gagne la partie, l’autre perd.

L’un des joueur perd toutes ses chances et perd la partie, l’autre gagne.

**Libellés et messages**

**En cas de victoire :** Le joueur reçoit un message « Félicitations ! Vous avez gagné ☺ »

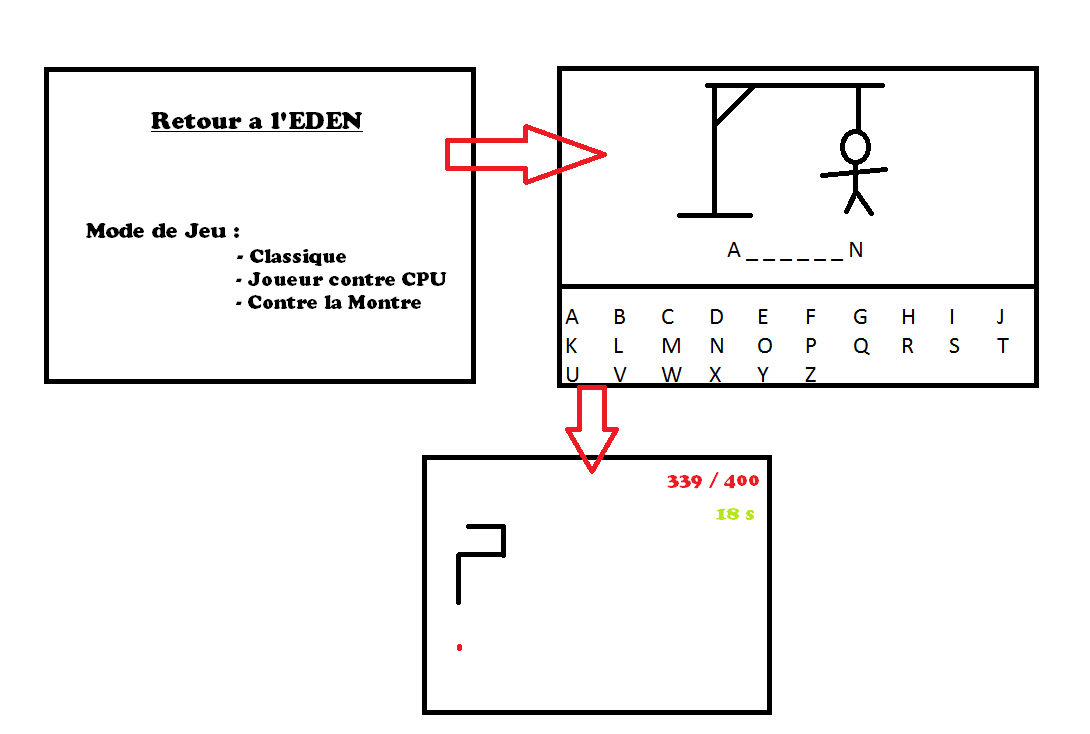
**En cas de défaite :** « Désolé ! Vous avez perdu ☹ ».

Il y a possibilité de relancer une nouvelle partie.

**Améliorations à incorporer par la suite**

* Jeu de snake pour offrir une chance supplémentaire avant la fin de partie.
* Option de mode de difficulté pour changer le nombre de « chance » pour proposer une lettre.
* Tableau des scores avec la longueur de la chaîne et le timing.
* Définition du mot d’après Wikipédia à la fin de la partie.
* Animations graphiques.

**Interface Utilisateur**



*En cas d’échec, le jeu du snake est lancé*

*Ecran de Jeu (dans la partie inférieur les lettres que l’on peut proposer, dans la partie supérieur, le mot et le pendu).*

*Menu à l’ouverture de l’application*

**Scénario type :**

Lorsqu’un utilisateur démarre le jeu du pendu, une première fenêtre s’offre à lui. C’est le menu principal, composé du titre du jeu et de trois options de jeu simples.

Le joueur peut sélectionner le style de partie de son choix, parmi les 3 proposés. Admettons qu’il sélectionne le mode de jeu contre le CPU.

L’affichage se transforme. Le joueur a maintenant face à lui un clavier virtuel. Lorsqu’il est prêt, il appuie sur un bouton « Go ! » pour lancer la partie. Un nombre d’underscores égaux à la longueur du mot aléatoirement choisi par le CPU s’affiche and dessus du clavier virtuel. La première et la dernière lettre du mot apparaissent. Admettons que la première lettre du mot est un « m » et la dernière un « e ».

Le joueur décide alors d’appuyer sur « i ». La lettre étant dans le mot, elle apparait à la 9e position du mot. Le bouton du clavier virtuel appuyé se grise ensuite pour indiquer à l’utilisateur qu’il a déjà appuyé sur la lettre « i ». Chaque lettre qu’il sélectionnera sera ensuite grisée de la même façon.

Le joueur appuie ensuite sur « l ». La lettre étant deux fois dans le mot, elle apparait deux fois aux 3e et 8e positions.

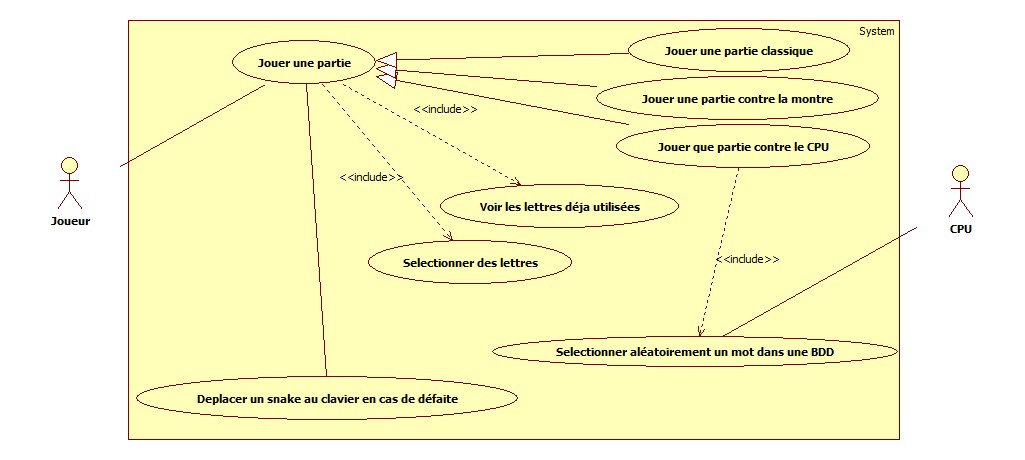
N’étant pas sur du mot, le joueur tente une nouvelle lettre au hasard. Il appuie sur « t ». La lettre n’est pas dans le mot. Ainsi, une partie du pendu est dessinée et s’affiche au-dessus du mot à deviner.

Enfin, le joueur appuie sur « o ». Un « o » apparait à la 7e position du mot.

Pensant avoir trouvé, le joueur appuie sur « a », « n », « c », « e » et le mot « Melancolie » apparait !

Le joueur est ainsi notifié de sa victoire par un petit label vert qui affiche à côté du pendu « Bravo ! Vous avez gagné ! »

**Diagramme de cas d’utilisation :**

****