Projet 1 NSI 1ere: Pendu

October 22, 2021

Le jeu du pendu, description rapide

Descriptif rapide du jeu:

- Choisir un mot aléatoirement
- tant qu'on a pas trouvé toutes les lettres ou perdu:
  - Proposer une lettre.
  - Comparer avec le mot choisit aléatoirement.
  - Si on a trouvé toutes les lettres, on a gagné.
  - Si on s'est trompé, dessiner un bout du pendu.
  - Si le dessin est terminé, on a perdu.
- la partie est finie.

Nous avons ensuite essayé de créer une partie du jeu sur le temps restant de la séance.

Une spécification plus précise

Dans cette partie, je vais vous présenter une spécification plus précise du jeu de pendu. Vous pourrez vous appuyer sur ce document pour réaliser le jeu.

Les attendus:

- Le mot à deviner doit être choisi au hasard.
- une fois le mot choisi, le jeu commence.
- a chaque tour l'utilisateur doit proposer une lettre.
- Tout le long de la partie, on veut afficher après chaque proposition de lettre:
  - Les lettres déjà essayées.
  - le mot qu'on cherche avec le caractère '_' pour les lettres que l'on a pas trouvé.
- un compteur de vie restante (optionnel: un dessin du pendu).
  - Si la lettre n'est pas dans le mot recherché, on perd une vie.
  - Si la lettre est dans le mot recherché, le compteur de vie n'est pas modifié.

Algorithme pour modifier l'affichage du mot (remplacer les "_" par des lettres)

Role: remplacer le symbole '_' par une lettre précise a la position qu'elle a dans un mot. Entrée: mot: Une chaîne de caractère. lettre: un caractère qui est dans la variable mot. tableau: un tableau contenant des lettres ou des '_'. La taille du tableau doit être égale a celle du mot. Sortie: la variable tableau modifiée. Pseudo-code:

```
i : Entier
i = 0
tant que  i < |mot| (longueur du mot).
    si mot[i] == lettre
        tableau[i] = lettre
renvoyer tableau
```

Un exemple d'exécution de l'algorithme

avec:

- lettre = 'e'
- mot = 'seul'
- tableau = ['_','_','_','_']

début:
i = 0
rappel :
$|mot| = 4$
boucle:

i	mot[i]	tableau	$i < \|mot\|$	mot[i] == lettre
0	s	['_','_','_','_']	Vrai	Faux
1	e	['_','e','_','_']	Vrai	Vrai (donc modification de tableau)
2	u	['_','_','_','_']	Vrai	Faux
3	l	['_','_','_','_']	Vrai	Faux
4	on a $i < \|mot\|$ Faux donc la boucle est finie			

on renvoie le tableau modifié qui contient ['_','e','_','_'].