



Initiation à la programmation

Jeu du pendu

Objectifs du TP

- programmer le jeu du pendu

Matériel fourni

- un fichier `materiel.zip` contenant 4 fichiers
 1. `dico.txt` : un dictionnaire (fichier au format texte) contenant environ 68000 mots de la langue française ;
 2. `dico.pas` : le fichier source d'une unité offrant une fonction de choix au hasard d'un des mots du dictionnaire ;
 3. `pendu_et_affichage.pas` : le fichier source d'une unité offrant un certain nombre de procédures et fonctions pour gérer les sanctions et affichage du pendu ;
 4. `squelette_jouer_au_pendu.pas` : le squelette du fichier source du programme à réaliser.

1 Règles du jeu

Les règles du *jeu du pendu* sont les suivantes.

1.1 Le but du jeu

Il s'agit de retrouver un mot choisi aléatoirement par l'ordinateur. Le joueur peut faire un nombre fini de *propositions*.

1.2 Le déroulement du jeu

On choisit un mot au hasard dans le dictionnaire. C'est ce mot qui sera le *mot à trouver* par le joueur. Le joueur fait une *proposition*. Tant que les conditions de fin de partie ne sont pas atteintes, le joueur peut faire une nouvelle proposition.

1.3 Une proposition

Avant chaque proposition, on indique le nombre de lettres du mot à trouver, les lettres déjà découvertes sont affichées à leur place, et les lettres qu'on doit encore découvrir sont symbolisées par des astérisques (*).

Le joueur peut proposer

- soit une lettre,
- soit un mot complet,
- soit un abandon,
- soit un affichage des règles du jeu.

Lorsque la proposition est une lettre ou un mot, on dit qu'elle est *erronée*

- lorsque la lettre proposée n'appartient pas au mot,
- ou bien lorsque la lettre a déjà été proposée,
- ou enfin lorsque le mot proposé n'est pas égal au mot à trouver.

Dans ce cas, un morceau supplémentaire du corps du pendu est dessiné sous la potence. Le corps du pendu est constitué

- d'une tête
- d'un tronc
- de deux bras (droit et gauche)
- de deux jambes (droite et gauche)

Avant la première proposition, aucun morceau du corps du pendu n'est sous la potence.

Lorsque une lettre proposée est correcte, on doit découvrir la (ou toutes les) occurrence(s) de cette lettre dans le mot à trouver.

1.4 Fin de partie

la partie se termine lorsque

- le pendu est complètement dessiné. Le joueur a alors perdu.
- le joueur a découvert le mot à trouver¹. Dans ce cas, le joueur a gagné.
- le joueur a décidé d'abandonner, et il a alors perdu.

2 Le travail à faire

2.1 Cahier des charges

Vous devez réaliser un programme qui permet de jouer au pendu. Après chaque partie, l'utilisateur peut éventuellement vouloir rejouer une partie, il faut offrir cette possibilité. Enfin lors de chaque proposition, l'utilisateur doit pouvoir consulter la règle en tapant le caractère '?'.

2.2 Le matériel fourni

- un fichier nommé **dico.txt** contenant un dictionnaire de 68000 mots environ... Le dictionnaire est un fichier texte. Chaque ligne contient un mot et un seul. On peut remplacer ce fichier par tout autre fichier respectant le format décrit précédemment, il ne doit cependant pas contenir plus de 70 000 mots. De plus, ce fichier doit se trouver dans le même répertoire que l'exécutable.
- une unité nommée **dico.pas**. Cette unité contient une fonction sans paramètre qui choisit un mot au hasard dans le dictionnaire (fichier **dico.txt**).

```
function mot_au_hasard : STRING;
```

- une unité nommée **pendu_et_affichage.pas**. Cette unité contient
 - des procédures et des fonctions qui permettent de changer et de consulter l'état du gibet .

```
// la procédure Preparer_gibet initialise le gibet
```

```
// lors du début d'une partie pendu.
```

```
procedure Preparer_gibet;
```

```
// la procédure sanctionner permet d'ajouter un morceau de plus du corps du
```

```
// pendu sous la potence
```

```
// cette procédure peut déclencher l'exception (privée)
```

```
// deja_Pendu
```

```
procedure Sanctionner;
```

```
// la fonction retourne VRAI si et seulement si tout le corps du pendu est
```

```
// dessiné.
```

```
function Pendu : BOOLEAN;
```

- des procédures d'affichages sans paramètres.

```
procedure Afficher_gibet;
```

```
procedure Afficher_regle;
```

- un exécutable de démonstration **jouer_au_pendu_demo.exe**
- un fichier **squelette_jouer_au_pendu.pas** contenant le squelette du programme principal dans lequel toutes les déclarations ont été faites. Remarquez qu'une seule variable globale est déclarée, vous n'avez pas à en rajouter. De même, le programme principal est écrit, vous n'avez pas le droit de le modifier.

2.3 Découpage du travail

Il est conseillé de réaliser les procédures et fonctions suivantes :

```
// cette fonction sert à fabriquer une chaîne de n étoiles.
```

```
function repete_etoile(const n: CARDINAL) : STRING;
```

¹soit en découvrant les lettres une par une soit en proposant le mot complet

```

// cette procédure permet de modifier le mot_cache
// en faisant apparaître la lettre proposée
procedure decouvrir(lettre:CHAR);

// cette procédure demande à l'utilisateur une proposition.
// Si la proposition est erronée alors le joueur est sanctionné,
// sinon la lettre est découverte.
procedure faire_proposition;

procedure jouer;

```

3 Exemple

Ce qui suit est une trace partielle d'une exécution du programme `jouer_au_pendu_demo.exe`

```

-----
|/ |
| 0
| /|
|  |
|
---
```

```

e*iton*
votre proposition (*/?/lettre/mot)
v

```

```

-----
|/ |
| 0
| /|\
|  |
|
---
```

```

e*iton*
votre proposition (*/?/lettre/mot)
s

```

```

-----
|/ |
| 0
| /|\
|  |
|
---
```

```

e*itons
votre proposition (*/?/lettre/mot)
d
Bravo!
le mot etait "editons"
voulez vous rejouer (o/n)?
n
Merci d'avoir joué au pendu.

```