

*Objectif de la séance :*  
Apprendre à manipuler des IHM en mode texte

## 1 Déplacement d'un symbole avec le clavier

Le premier programme à écrire nécessite l'utilisation des unités Crt et Keyboard, respectivement pour afficher un symbole à une position donnée dans le terminal de sortie et pour reconnaître des événements issus du clavier.

Pour utiliser ces unités, vous devez vous appuyer sur leur documentation en ligne (<http://www.freepascal.org/docs-html/rtl/>) et vous pouvez vous inspirer des exemples vus en cours.

### Exercices

1. Créez un nouveau fichier `deplace.pas` qui contient le programme principal ci-dessous

```
begin
  InitKeyBoard()
  deplacement()
  DoneKeyBoard()
end.
```

Dans un premier temps, écrivez seulement, dans le corps de la procédure `deplacement` les instructions permettant de récupérer dans une variable un événement issu du clavier et d'écrire à l'écran le résultat de sa conversion en chaîne de caractères (fonction `KeyEventToString`). Les événements clavier sont affichés en boucle jusqu'à ce que la touche 'q' soit pressée. Repérez le texte correspond aux touches des flèches du clavier.

2. Modifiez la procédure `deplacement` pour qu'elle utilise 2 variables `posX` et `posY` indiquant les coordonnées d'un symbole (utilisons le caractère 'o' pour cela) à afficher. A l'aide de la procédure `GotoXY` de l'unité Crt affichez ce symbole à une position initiale de coordonnées (2,2) au début de la procédure `deplacement`.
3. Modifiez la procédure `deplacement` pour ne plus afficher la chaîne qui correspond à un événement clavier mais pour réagir aux touches de flèche du clavier. La réaction attendue est de déplacer le symbole dans le sens correspondant à la flèche pressée. Pour cela, vous devez d'abord effacer le symbole, calculer sa nouvelle position puis le ré-afficher.
4. Faites une dernière modification de la procédure `deplacement` pour imposer des bornes minimum et maximum au déplacement. Le symbole ne pourra pas aller en dessous des coordonnées 2 et au dessus des coordonnées 5 en abscisse et en ordonnée.

## 2 Déplacement dans un labyrinthe

La seconde étape du TP est d'afficher un labyrinthe dans lequel se déplacera le symbole. Téléchargez depuis moodle les fichiers `labyrinthe.pas` et `laby1.txt`. Le programme pascal est à compléter alors que le fichier texte contient la description d'un labyrinthe donné.

---

### *Exercices*

---

1. Lisez attentivement le contenu de ces fichiers pour en comprendre le fonctionnement.
  2. Complétez la procédure affichage pour qu'elle affiche à l'écran le labyrinthe stocké dans le tableau 1. Ce tableau représente un plan du labyrinthe vu de dessus. La valeur vrai dans une case représente un mur (qu'on affichera avec le symbole #) et la valeur faux un couloir (qu'on affichera avec un espace).
  3. Copiez la procédure déplacement écrite précédemment dans ce programme. Appelez cette procédure après avoir appelé la procédure affichage. Pour le moment le déplacement du symbole aura pour effet d' "effacer" les cases du labyrinthe.
  4. Modifiez les paramètres d'entrée de la procédure déplacement pour que les bornes minimale et maximale du déplacement ne soient plus 2 et 5 mais soient les dimensions du labyrinthe.
  5. Passez aussi le tableau représentant le labyrinthe en paramètre d'entrée de la procédure déplacement. Faites-en sorte que le déplacement du symbole respecte la configuration du labyrinthe, c'est-à-dire qu'il est impossible de se déplacer dans un mur.
- 

## 3 Exercices facultatifs

Vous pouvez ensuite implémenter les améliorations ci-dessous :

---

### *Exercices*

---

1. Affichez des symboles autour du labyrinthe (par exemple le caractère 'I' à gauche et à droite et '-' en haut et en bas), pour mieux délimiter ses contours.
  2. Écrivez d'autres fichiers texte représentant des labyrinthes utilisables par votre programme.
  3. Comptez et affichez au dessus ou en dessous du labyrinthe le nombre de déplacements effectués.
  4. Insérez le symbole 'S' dans les fichiers textes des labyrinthes pour représenter la case de sortie. Le programme s'arrêtera désormais quand la case de sortie a été atteinte.
-