

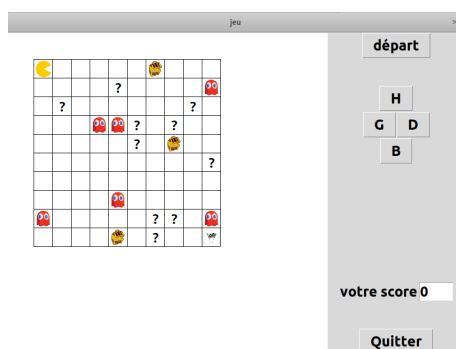
Initiation à la programmation – TP 12 du 17 novembre

Objectif : Programmer un jeu en mode console sur une grille (sera prolongé en version graphique au semestre 2).

Exercice : Jeu sur une grille

Il s'agit de faire en mode console un petit jeu : l'utilisateur doit programmer des déplacements dans une grille pour aller chercher des trésors en évitant des obstacles (fantômes).

Une version avec interface graphique pourrait donner :



On travaillera avec une grille carrée de taille n et une grille sera représentée par une liste de listes.

1. Écrire la fonction `creeGrille(n,val)` qui renvoie une grille carrée de taille n où toutes les « cases » contiennent `val`.
2. Écrire la fonction `affiche(grille)` qui affiche en mode console la grille. Il ne doit plus rester de `[]` dans l'affichage (voir exemple ci-dessous).
3. On veut placer aléatoirement un certain nombre t de trésors (représentés par des 1) et un certain nombre f de fantômes (représentés par des 0) dans la grille. On mettra des "_" dans les cases qui ne contiennent rien.

Écrire la fonction `aleagrille(n,t,f)` qui crée une grille carrée de taille n avec des "_" dans toutes les cases, puis choisit t cases pour mettre un 1 dedans et f cases pour mettre un 0 dedans. La grille ainsi obtenue est alors renvoyée.

```
>>> affiche(aleagrille(8, 4,6))
-- 1 -- -- --
-- -- -- -- --
-- -- 0 -- --
-- -- -- -- 1
```

```

  0 0
1 1 - 0 - 0 -
- - - - 0 -
- - - - 0 -
- - - - - -

```

- On indiquera un déplacement avec l'un des caractères 'H' (haut) 'B' (bas) 'D' (droite) ou 'G' (gauche). Écrire la fonction `deplacement(grille, dep, i, j)` qui effectue le déplacement `dep` à partir de la case `i,j` de grille : la fonction doit vérifier que le déplacement ne fait pas sortir de la grille et elle doit modifier la grille (en changeant de case le symbole '*' qui représente la position du joueur). A la fin la fonction doit renvoyer la position du joueur après ce déplacement (nouvelle ou inchangée s'il a tenté de sortir). Dans un premier temps on ne se préoccupe pas des contenus des cases sur lesquelles le joueur arrive.
- On va maintenant modifier la fonction précédente pour gérer des scores. Au départ l'utilisateur a 0 point. S'il passe sur un trésor (cases 0), il gagne 5 points (une seul fois le trésor une fois pris disparaît), s'il passe sur un fantôme, le fantôme « disparaît » et il perd 10 points et s'il essaye un déplacement qui le ferait sortir de la grille, il perd 2 points. On fera un affichage au joueur au fur et à mesure quand il perd ou gagne des points et la fonction renverra à la fois la nouvelle position et le nombre de points du déplacement sous la forme d'une liste.

```

>>> affiche(ex)
- - - 0 1
- - - - 1
- - - - -
- * 1 - -
0 0 1 - 0
>>> deplacement(ex, "D", 3,1)
fantome !!! vous perdez 10 points
[3, 2, -10]

>>> affiche(ex)
- - - 0 1
- - - - 1
- - - - -
- * - - -
0 0 1 - 0

```

- faire une fonction pour une partie : cette fonction aura deux paramètres : la taille de la grille et le nombre d'objets. Dans ce programme l'ordinateur doit générer une grille, positionner le joueur (donc un caractère "*") dans la case (0,0), une arrivée (donc un symbole "A") dans la case (n-1,n-1), afficher la grille et tant que le jeu n'est pas fini, il demande à l'utilisateur le déplacement suivant, le gère (changement éventuel de position ou du nombre de points) avec un affichage de la grille suite à chaque déplacement. Le jeu s'arrête quand l'utilisateur est parvenu à la case (n-1,n-1) ; il gagne s'il a un nombre de points positif qu'on affichera.

```

>>> jeu(4, 10)
fantomes=1 tresor = 0

* - 0 -
0 1 - -
1 - 1 0
0 1 1 $
votre deplacement? B
tresor !!! vous gagnez 5 points
- - 0 -
* 1 - -

```

```

1  -  1  0
0  1  1  $
votre déplacement?  H
*  -  0  -
-  1  -  -
1  -  1  0
0  1  1  $
votre déplacement?  D
-  *  0  -
-  1  -  -
1  -  1  0
0  1  1  $
votre déplacement?  D
tresor !!! vous gagnez 5 points
-  -  *  -
-  1  -  -
1  -  1  0
0  1  1  $
votre déplacement?  B
-  -  -  -
-  1  *  -
1  -  1  0
0  1  1  $
votre déplacement?  D
-  -  -  -
-  1  -  *
1  -  1  0
0  1  1  $
votre déplacement?  D
sortie !!! vous perdez 2 points
-  -  -  -
-  1  -  *
1  -  1  0
0  1  1  $
votre déplacement?  B
tresor !!! vous gagnez 5 points
-  -  -  -
-  1  -  -
1  -  1  *
0  1  1  $
votre déplacement?  B
-  -  -  -
-  1  -  -
1  -  1  -
0  1  1  *
vous etes sortis avec 13 points
bravo

```

7. Améliorations :

- faire choisir la taille et le nombre d'objets à l'utilisateur
- regarder s'il reste des trésors à la fin du parcours et l'indiquer à l'utilisateur
- remplacer certains objets à l'affichage par des ? ; quand l'utilisateur arrive sur la case le contenu est dévoilé.
- permettre la saisie en une seule fois de plusieurs déplacements successifs comme BBDDDDBBB et les gérer ensuite un par un.
- permettre d'utiliser les flèches de direction ou d'autres touches du clavier