



## Mini-projet : Labyrinthe



**Les programmes d'exemples sont dans le fichier labyrinthe.zip**

On désire créer un jeu d'aventure style [Dungeon Master](#) .

Le but est de trouver la sortie du labyrinthe

Le moteur du jeu est fourni. Il faudra l'améliorer.



La variable « d » correspond à l'orientation du joueur dans le labyrinthe.

d=2 #0 nord, 1 est ,2 sud,3 ouest

Les variables « x » et « y » sont les coordonnées de la position du joueur dans le labyrinthe.

x=1 #depart

y=7

**Défi N°1 : Exécuter la source fournie : ultra facile**

**Décompresser** le fichier zip fourni. Exécuter le jeu.

**Défi N°2 : Agrandir le niveau : facile**

Le but est d'augmenter la surface du labyrinthe. Pour cela il faudra modifier :

- La taille de la fenêtre de pygame ;
- Le fichier csv ;
- Les coordonnées d'affichage de la carte : positionX et positionY.

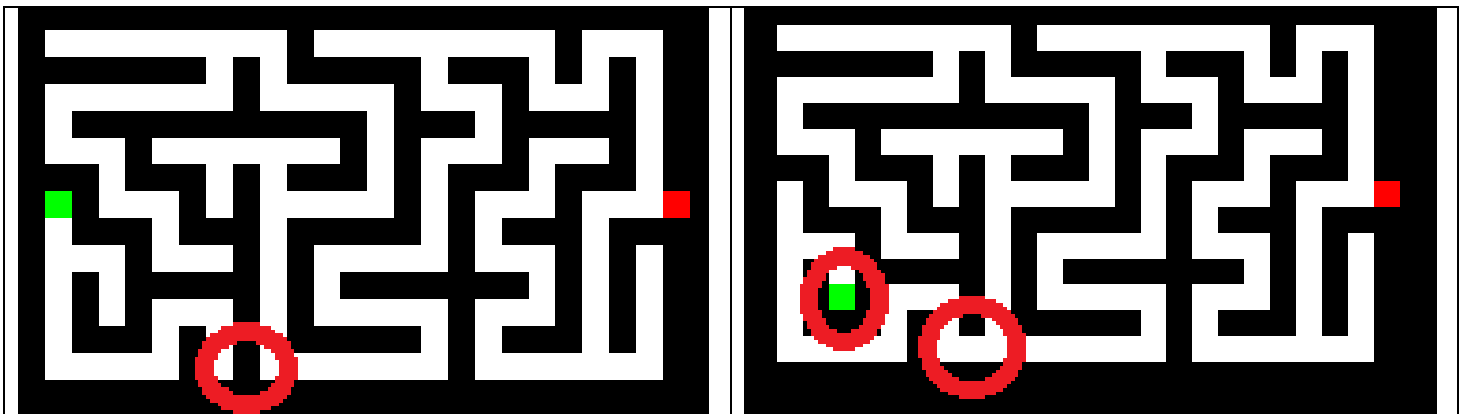
Rappel

0 : mur

1 : couloir ou passage

**Défi N°3 : Bloquer le labyrinthe : Moyen**

Un mur bloque le labyrinthe, pour le faire disparaître il faut trouver un emplacement précis.



Exemple :

Ajoute un mur aux coordonnées  $y = 13$  et  $x = 8$

```
tableau[13][8]='0'
```

Supprime le mur aux coordonnées  $y = 13$  et  $x = 8$

```
tableau[13][8]='1'
```

**Défi N°4 : Bloquer le labyrinthe avec temporisation : difficile**

Même principe que le précédent, sauf que le joueur n'a que quelques secondes pour passer avant que le mur ne réapparaisse.

Aide : utiliser la bibliothèque time

```
import time
```

et la fonction

```
time.clock() #retourne nombre de secondes depuis le démarrage du programme
```

**Défi N°5 : Enchaîner sur 3 niveaux de labyrinthe : difficile**

Une fois un labyrinthe terminé, on enchaîne sur un niveau plus difficile

**Bonus :**

Les modifications de votre choix comme par exemple des murs avec un graphique spécifique.