

Python : Un premier jeu

1. Une première interface :

Les commandes suivantes permettent de créer une fenêtre pour afficher notre jeu :

```
from tkinter import *  
fenetre=Tk()  
fenetre.title("Jeu Python")  
n=15  
c=50  
can=Canvas(fenetre,width=n*c,height=n*c, bg="black")  
can.pack()
```

1. Comment changer la couleur du fond ?

2. Comment changer la taille de la fenêtre ?

2. Le personnage :

Pour se faciliter la tâche, on représente notre personnage par une forme géométrique simple,

La fonction suivante permet de créer un rectangle :

```
can.create_rectangle(0,0,50,50,fill="red")
```

Ici le coin supérieur gauche du rectangle est aux coordonnées 0,0 et le coin inférieur droit aux coordonnées 50,50. De même pour un cercle :

```
can.create_oval(50,50,100,100,fill="red")
```

Nous pouvons aussi afficher du texte avec la commande :

```
can.create_text(100,100,text="Bonsoir",fill="blue")
```

Et enfin des images :

- Au début du fichier :

```
img = PhotoImage(file= 'image.gif')
```

- Dans notre code :

```
can.create_image(300, 300, image=img)
```

3. Le déplacement :

La commande suivante permet de lancer la fonction Test si on appuie sur la touche « d » :

```
fenetre.bind("<d>", Test)
```

Les fonctions Test, la fonction rondIn et la ligne ci-dessous permettent de gérer le déplacement de notre personnage vers la droite.

```
def Test(event = None):  
    global pos  
    pos[0]+=50  
    rondIn()
```

```
def rondIn():  
    global c,posx,posy,Mt  
    can.delete(ALL)  
    rond = can.create_oval(pos[0],pos[1],pos[0]+c,pos[1]+c,fill='#fff')
```

1. recopiez ces deux fonctions en bas de votre code ainsi que la commande fenetre.bind ci-dessus et l'instruction suivante en haut de votre code :

```
pos=[0,0]
```

2. Maintenant que vous savez gérer le déplacement vers la droite, faites de même pour la gauche, le haut, et le bas.

Les matrices

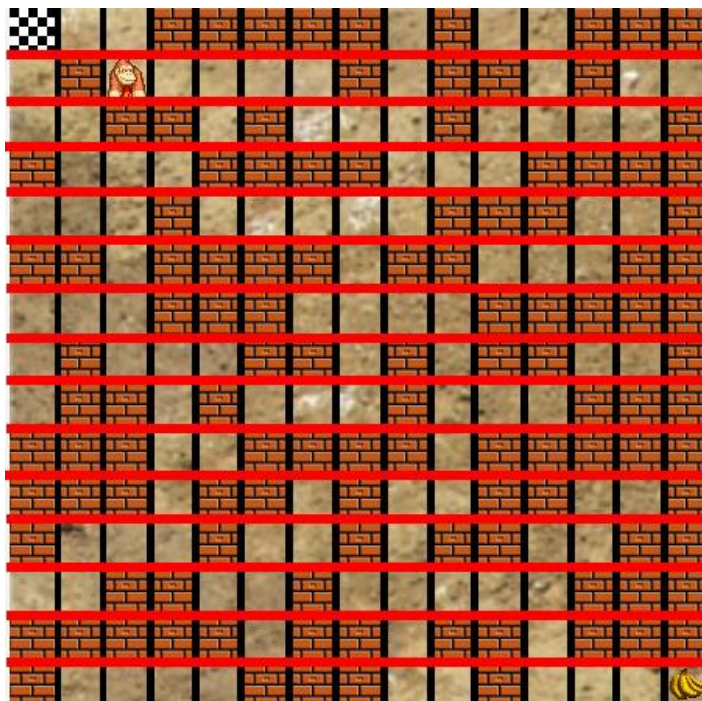
En Python, une manière de représenter les données d'une surface plane divisée en « petit carré » se fait à l'aide d'un objet qu'on appelle « matrice ».

Pour saisir un peu plus précisément l'idée...

Voici un plateau de jeu représentant un labyrinthe :

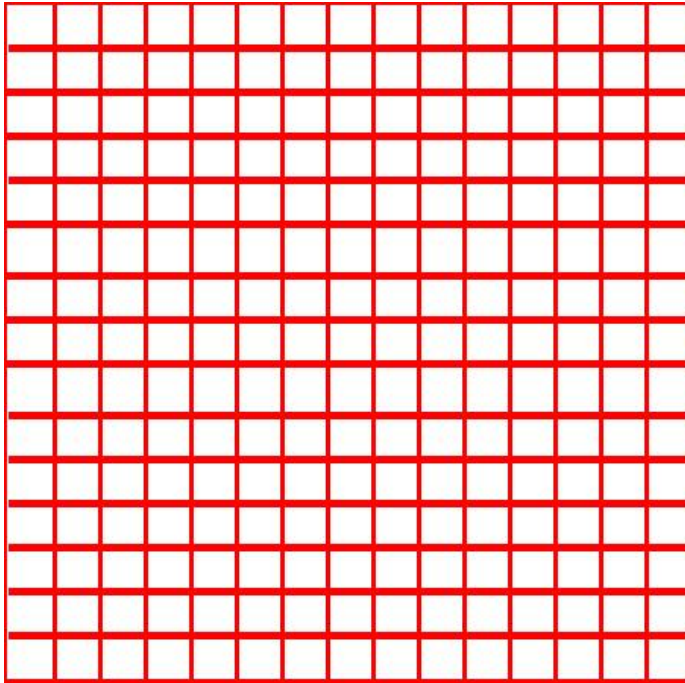


On peut voir que le plateau de jeu peut être divisé en plusieurs petits carrés de même taille, formant une grille de 15 carrés par 15 carrés.



Mais qu'est ce qui se cache derrière l'interface graphique ?

En réalité, il s'agit d'un tableau à deux dimensions, comme suit :



On appelle ça une **matrice**.

Une matrice en Python est représentée par une **liste de listes d'éléments**.

Une liste s'écrit `nom_de_la_liste = [élément , élément , élément]`

Par exemple, la liste 1 2 3 s'écrit `exemple_liste = [1,2,3]`

Ainsi, on peut écrire la matrice `exemple_matrice = [exemple_liste , exemple_liste]` qui correspondra à `[[1 , 2 , 3] , [1 , 2 , 3]]`

Par exemple, une matrice de 3 cases sur 3 cases est une liste contenant 3 listes de 3 éléments.

`Matrice = [[0,0,0] , [1,1,1] , [2,2,2]]` n'est autre que :

1	1	1
2	2	2
3	3	3

À faire :

- Ouvrir le fichier ex-matrice.py avec Python IDLE. Celui-ci contiendra une fonction afficher à **ne pas effacer**.
- Lancer le fichier en appuyant sur la touche F5. Quel résultat observe-t-on dans la console ?

- Créer une variable T2 qui sera une matrice de 5 cases sur 5 cases, représentée par l'image suivante, et l'afficher :

T2 =

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

L'accès à un élément d'une matrice se fait de la manière suivante :

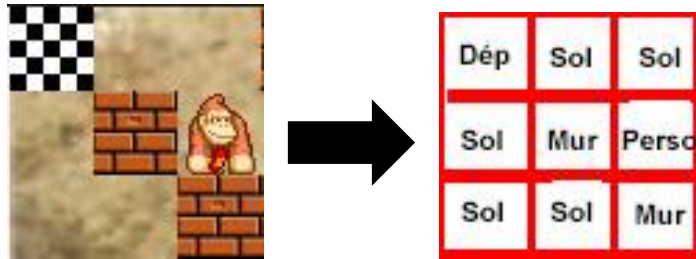
Nom_de_la_matrice [indice_de_ligne] [indice_de_colonne]

Par exemple, sur la matrice T2, si je veux afficher la valeur qui se trouve dans la 2^{ème} ligne, 3^{ème} colonne, il faut que j'effectue la commande :

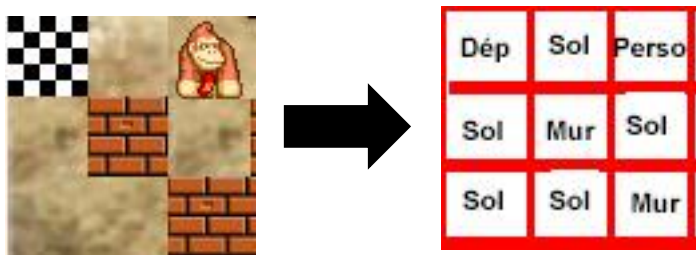
print(T2[1,2])

- Mais pourquoi T2[1,2] et pas T2[2,3] ?
Essayez **print(T2[0,0])** et expliquez-le.
- Quelle commande pour afficher le chiffre 23 dans la matrice ?
- Pour afficher le chiffre 15 ?
- **Quel est l'indice maximum pour les lignes ? les colonnes ?**

Après avoir compris ces principes de manipulations, on peut adapter cette structure à notre plateau de jeu. Le jeu n'est autre qu'une matrice, où chaque case prend une valeur différente suivant ce qu'elle représente sur l'interface graphique. Petite explication :



Si on déplace le personnage vers le haut, cela donnerait :



Ainsi, lorsque l'on appuie sur une touche pour faire monter le personnage d'une case, on met à jour **simultanément** :

- L'interface graphique
- La matrice

Mais surtout, on se sert de la matrice pour représenter **l'état du jeu**, c'est-à-dire :

- Où se trouve mon personnage
- Quelles sont les cases libres
- Quelles sont les cases que je ne peux pas franchir
- Etc...

A faire :

- Définir dans le fichier `ex-matrice.py` la matrice représentant l'état du jeu avant de déplacer le personnage et l'afficher. Dans la matrice, les éléments peuvent être n'importe quel type d'élément (chiffre, texte,...) **CEPENDANT** du texte doit être défini entre guillemets

Exemple : ["a", "b", "c"]