

## DK

```
In [ ]: import pygame
        from pygame.locals import *
```

```
In [ ]: laby_DK = [[2, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1],
                  [0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1],
                  [0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1],
                  [1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1],
                  [0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1],
                  [1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1],
                  [0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1],
                  [0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1],
                  [0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1],
                  [1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1],
                  [1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1],
                  [1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1],
                  [0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1],
                  [1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1],
                  [1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 3]]
```

```
In [ ]: largeur, hauteur = len(laby_DK[0]), len(laby_DK)
```

```
In [ ]: def init_fond():
        fenetre.blit(fond, (0,0))
        for x in range(largeur):
            for y in range(hauteur):
                if laby_DK[y][x] == 1:
                    fenetre.blit(mur, Rect(30*x, 30*y, 30, 30))
                elif laby_DK[y][x] == 2:
                    fenetre.blit(entree, Rect(30*x, 30*y, 30, 30))
                elif laby_DK[y][x] == 3:
                    fenetre.blit(sortie, Rect(30*x, 30*y, 30, 30))
```

```
In [ ]: def move_right(pos):
        global perso
        perso = dk_droite
        x_pos, y_pos = pos[0]//30, pos[1]//30
        if x_pos < largeur-1 and laby_DK[y_pos][x_pos+1] != 1:
            return pos.move(30, 0)
        else:
            return pos
```

```
In [ ]: def move_left(pos):
        global perso
```

Loading [MathJax]/extensions/Safe.js

```

perso = dk_gauche
x_pos, y_pos = pos[0]//30, pos[1]//30
if x_pos > 0 and laby_DK[y_pos][x_pos-1] != 1:
    return pos.move(-30, 0)
else:
    return pos

```

```

In [ ]: def move_up(pos):
        global perso
        perso = dk_haut
        x_pos, y_pos = pos[0]//30, pos[1]//30
        if y_pos > 0 and laby_DK[y_pos-1][x_pos] != 1:
            return pos.move(0, -30)
        else:
            return pos

```

```

In [ ]: def move_down(pos):
        global perso
        perso = dk_bas
        x_pos, y_pos = pos[0]//30, pos[1]//30
        if y_pos < hauteur - 1 and laby_DK[y_pos+1][x_pos] != 1:
            return pos.move(0, 30)
        else:
            return pos

```

```

In [ ]: pygame.init()

```

```

In [ ]: #Ouverture de la fenêtre Pygame
fenetre = pygame.display.set_mode((450, 450))
pygame.display.set_caption("DK Labyrinthe")

```

```

In [ ]: #Texte victoire
myfont = pygame.font.SysFont("Deja Vu Sans MS", 80)
texte = "Gagné !"
label_victoire = myfont.render(texte, True, (255, 0, 0),)

```

```

In [ ]:

```

```

In [ ]: #Chargement et collage du fond
fond =
pygame.image.load("/home/cedric/Travail/AlgoInfo/CodesPython/PyGame/DK/fond.jpg")

fenetre.blit(fond, (0,0))

```

```

In [ ]: #Chargement des images
dk_bas =
pygame.image.load("/home/cedric/Travail/AlgoInfo/CodesPython/PyGame/DK/dk_bas.p

```

```

dk_haut =
pygame.image.load("/home/cedric/Travail/AlgoInfo/CodesPython/PyGame/DK/dk_haut.

dk_gauche =
pygame.image.load("/home/cedric/Travail/AlgoInfo/CodesPython/PyGame/DK/dk_gauch

dk_droite =
pygame.image.load("/home/cedric/Travail/AlgoInfo/CodesPython/PyGame/DK/dk_droit

```

```
In [ ]: perso = dk_droite
```

```

In [ ]: position_perso = perso.get_rect()
mur =
pygame.image.load("/home/cedric/Travail/AlgoInfo/CodesPython/PyGame/DK/mur.png"

entree =
pygame.image.load("/home/cedric/Travail/AlgoInfo/CodesPython/PyGame/DK/depart.p

sortie =
pygame.image.load("/home/cedric/Travail/AlgoInfo/CodesPython/PyGame/DK/arrivee.

fenetre.blit(perso, position_perso)

```

```

In [ ]: #Rafraîchissement de l'écran
pygame.display.flip()
pygame.key.set_repeat(400, 30)

```

```

In [ ]: #BOUCLE INFINIE
continuer = True
while continuer:
    for event in pygame.event.get():    #Attente des événements
        if event.type == QUIT:
            continuer = False
        if event.type == KEYDOWN:
            if event.key == K_DOWN:
                position_perso = move_down(position_perso)
            if event.key == K_UP:
                position_perso = move_up(position_perso)
            if event.key == K_RIGHT:
                position_perso = move_right(position_perso)
            if event.key == K_LEFT:
                position_perso = move_left(position_perso)

    #Re-collage
    # fenetre.blit(fond, (0,0))
    init_fond()
    fenetre.blit(perso, position_perso)
    #Rafraichissement
    pygame.display.flip()
    if laby_DK[position_perso[1]//30][position_perso[0]//30] == 3:
        continuer = False

```

Loading [MathJax]/extensions/Safe.js

```
fenetre.blit(label_victoire, ((450-myfont.size(texte)[0])//2, (450-  
myfont.size(texte)[1])//2))  
pygame.display.flip()  
pygame.time.delay(2000)
```

In [ ]: `pygame.quit()`

