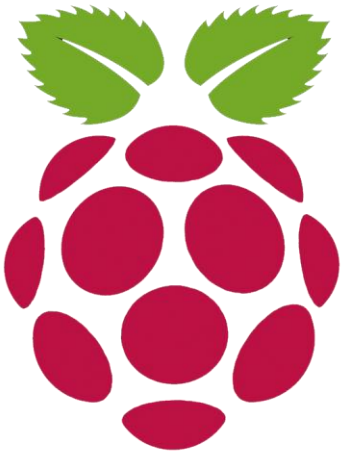
Province de Hainaut
Institut **P**rovincial d'**E**nseignement **S**econdaire
Rue Paul Pastur, 11
7800 ATH

TPT : Raspberry N°6 : Programme „Portal“



Par:
Rucquoy Edouard

Pour: M. Murez

Date d’expérimentation
11/05/2016

Date de remise du rapport
25/05/2016

2

Prérequis pour faire fonctionner le jeu :

Installer python, les librairies pygame et sense-hat

Le jeu :

Voici le code d'initiation du jeu, avec le placement des portails et du joueur. Il y a également l'importation des librairies utilisées au début. Et les 4 couleurs utilisées tout au long du jeu.

```
1  # Portal game with SenseHat on Raspberry Pi - Rucquoy Edouard
2  from sense_hat import SenseHat
3  import os
4  import time
5  import pygame
6  from pygame.locals import *
7  from random import randint
8
9
10 sense = SenseHat() # Init SenseHat
11 sense.clear() # Clear matrix
12 pygame.init() # Init pygame
13
14 # Declaration de l'affichage
15 # (obligatoire pour gestion des touches/joystick)
16 pygame.display.set_mode((1, 1))
17
18
19 # Random place for Blue & Orange portals
20 x_blue = randint(1,6)
21 y_blue = randint(1,6)
22
23 x_red = randint(1,6)
24 y_red = randint(1,6)
25
26 # Init white player
27 x_white = 0;
28 y_white = 0;
29
30 # Init colors
31 red = (255, 128, 0)
32 blue = (0, 0, 255)
33 white = (255, 255, 255)
34 black = (0, 0, 0)
35
36 # Init game
37 sense.set_pixel(x_blue, y_blue, blue)
38 sense.set_pixel(x_red, y_red, red)
39 sense.set_pixel(x_white, y_white, white)
```

3

Après l'initialisation, voici la partie la plus importante : la gestion des déplacements et la téléportation.

Partie 1 : Déclaration des variables globales et gestion de la touche BAS et HAUT (joystick).

```
40  # Fonction de gestion des actions en fonction des touches
41  def handle_event(event):
42      global y_white
43      global x_white
44      global y_blue
45      global x_blue
46      global y_red
47      global x_red
48      global black
49      global white
50      if event.key == pygame.K_DOWN:
51          if y_white < 7:
52              if (y_blue == y_white+1) and (x_blue == x_white):
53                  sense.set_pixel(x_white, y_white, black)
54                  y_white = y_red+1
55                  x_white = x_red
56                  sense.set_pixel(x_white, y_white, white)
57              elif (y_red == y_white+1) and (x_red == x_white):
58                  sense.set_pixel(x_white, y_white, black)
59                  y_white = y_blue+1
60                  x_white = x_blue
61                  sense.set_pixel(x_white, y_white, white)
62              else:
63                  sense.set_pixel(x_white, y_white, black)
64              y_white = y_white + 1
65              sense.set_pixel(x_white, y_white, white)
66
67      elif event.key == pygame.K_UP:
68          if y_white > 0:
69              if (y_blue == y_white-1) and (x_blue == x_white):
70                  sense.set_pixel(x_white, y_white, black)
71                  y_white = y_red-1
72                  x_white = x_red
73                  sense.set_pixel(x_white, y_white, white)
74              elif (y_red == y_white-1) and (x_red == x_white):
75                  sense.set_pixel(x_white, y_white, black)
76                  y_white = y_blue-1
77                  x_white = x_blue
78                  sense.set_pixel(x_white, y_white, white)
79              else:
80                  sense.set_pixel(x_white, y_white, black)
81              y_white = y_white - 1
82              sense.set_pixel(x_white, y_white, white)
```

4

Partie 2 : Gestion de la touche GAUCHE et DROITE (joystick)

```
    elif event.key == pygame.K_LEFT:
        if x_white < 7:
            if (x_blue == x_white-1) and (y_white == y_blue):
                sense.set_pixel(x_white, y_white, black)
                x_white = x_red-1
                y_white = y_red
                sense.set_pixel(x_white, y_white, white)
            elif (x_red == x_white-1) and (y_white == y_red):
                sense.set_pixel(x_white, y_white, black)
                x_white = x_blue-1
                y_white = y_blue
                sense.set_pixel(x_white, y_white, white)
            else:
                sense.set_pixel(x_white, y_white, black)
                x_white = x_white - 1
                sense.set_pixel(x_white, y_white, white)
    elif event.key == pygame.K_RIGHT:
        if x_white > 0:
            if (x_blue == x_white+1) and (y_white == y_blue):
                sense.set_pixel(x_white, y_white, black)
                x_white = x_red+1
                y_white = y_red
                sense.set_pixel(x_white, y_white, white)
            elif (x_red == x_white+1) and (y_white == y_red):
                sense.set_pixel(x_white, y_white, black)
                x_white = x_blue+1
                y_white = y_blue
                sense.set_pixel(x_white, y_white, white)
            else:
                sense.set_pixel(x_white, y_white, black)
                x_white = x_white + 1
                sense.set_pixel(x_white, y_white, white)
    elif event.key == pygame.K_RETURN:
        pass
```

Et la fin du code, contenant la boucle pour ne pas que le jeu se coupe après un mouvement.

```
# Variable (jeu en cours de fonctionnement ou stoppé)
running = True

# Boucle du jeu
while running:
    pygame.event.pump()
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                    running = False
            if event.type == KEYDOWN:
                    if event.key == K_ESCAPE:
                        running = False
                    handle_event(event)
```