Sense HAT

Med Raspberry Pi

Windows User
TEC BALLERUP Telegrafvej 9

Contents

Sense HAT Installationsvejledning

Udstyrsliste

Hardware

- Computer (m. keyboard, mus & skærm)
- Raspberry Pi
- HDMI kabel
- SD kort + Kortlæser
- Strømforsyning
- Skærm til Pi
- Net kabel
- Sense HAT V1.0

Software

- Raspbian

  https://www.raspberrypi.org/downloads/raspbian/

- Rufus /andet program til at flashe OS

  https://rufus.ie/

- Python

  https://www.python.org/downloads/

Installation af Raspbian

- Installér Rufus.
- Forbind din kortlæser med SD kortet i til din computer.
- Åben Rufus og vælg din Raspbian ISO.
- Vælg det SD kort hvorpå du vil installere Raspbian.
- Bekræft installationen samt at filformateringen er i formatet *FAT32*.
- Tryk ”START”.

Opdatering af Raspbian

Opdatering af Raspbian kan gøres i terminalen med følgende kommando:

Sudo apt-get update

Pong

Sourcekode

from time import sleep #Imports the sleep library. This is used to set a difficulty later on.

from sense_hat import SenseHat #Imports the SenseHat Library - This allows the interaction with the Sense HAT

sense = SenseHat()

#Initial clear

sense.clear()

#Functions and colors

```
white = (255,255,255)
blue = (0, 0, 255)
red = (255, 0, 0)
clear_bat = (0,0,0) #Used to clear the ball when moving and between rounds.
bat_y = 4
ball_position = [3, 3]
ball_velocity = [1, 1]
#Method to draw the bat
def draw_bat(color):
    sense.set_pixel(0, bat_y, color)
    sense.set_pixel(0, bat_y+1, color)
    sense.set_pixel(0, bat_y-1, color)
#When the stick is moved upwards, the bat follows.
def move_up(event):
    global bat_y
    if event.action == 'pressed' and bat_y >1:
        draw_bat(clear_bat)
        bat_y -= 1
        draw_bat(white)
#When the stick is moved downwards, the bat follows
def move_down(event):
    global bat_y
    if event.action == 'pressed' and bat_y < 6:
        draw_bat(clear_bat)
        bat_y += 1
        draw_bat(white)
#We define the ball and its properties. We take in the colors defined above.
def draw_ball(color):
    sense.set_pixel(ball_position[0], ball_position[1], color) #We use the initial ball_position values and a color to draw the ball
    if ball_position[0] == 1:
```

```
        ball_velocity[0] = -ball_velocity [0]
draw_bat(white)#Initial draw of bat with the values defined above
#We tell the Pi what the sticks do when interacted with
sense.stick.direction_up = move_up
sense.stick.direction_down = move_down
lives = 3 #Gives the player 3 lives
speed = 0.5 #speed is used to change the timer on the sleep function.
points = 0 #Counts the players points throughout three lives.
i = 0 #Used for while loop. This value is never changed.
#Main program
while i == 0:
    draw_ball(blue)
    sleep(speed)
    draw_ball(clear_bat)
    #Movement + bounce on x axis.
    ball_position[0] += ball_velocity[0]
    if ball_position[0] == 7 or ball_position[0] == 0:
        ball_velocity[0] = -ball_velocity[0]
    ball_position[1] += ball_velocity[1]
    #Movement + bounce on y axis
    if ball_position[1] == 7 or ball_position[1] == 0:
        ball_velocity[1] = -ball_velocity[1]
#The ball hits the bat. This will increase the difficulty and your total amount of accrued points.
    if ball_position[0] == 1 and (bat_y -1) <= ball_position[1] <= (bat_y +1):
        ball_velocity[0] = -ball_velocity [0]
        points = points+1
        speed = speed*0.95 #The difficulty is ramped up multiplicatively
    if ball_position[0] == 0:
        lives = lives-1
        ball_position = [3, 3]
```

```
        ball_velocity = [1, 1]
        speed = 0.5
        draw_bat(clear_bat)
        bat_y = 4
        draw_bat(white)
        sleep(2)
    #If lives reaches 0, "you lose" will be displayed on the HAT and the points will be printed in the shell. The points are then returned to the default value.
        if lives == 0:
            sense.show_message("You lose")
            print(points)
            point = 0
```

Konklusion

Arbejdet med Raspberry Pi og dets GUI har jeg oplevet let og intuitivt. Grundet mine få interaktioner med kodesproget Python har jeg skulle tilegne mig kenskab til dennes syntax. Dette har dog ikke været et problem, da strukturen ellers ligner sprog jeg allerede har kenskab til.