MUHAMMED SARAC

Sense Hat

Praktik Centret

Indholdsfortegnelse

Forord

Vi har fået Sense Hat opgaven som vi skulle tilslutte en Raspberry pi og kode en snake spil i Python, kode maze spillet i Python, kode vaterpas i Python og få dem til funger det til funger i Sense Hat. Til dette øvelse har mig og Rasmus arbejdet sammen

Tidsplan

Fredag 18.08.2017

Downloade Rasbian, etcher.

Installer Rasbian.

Kode Snake spillet

Mandag 21.08.2017

Kode Maze spillet

Kode Vaterpas

Raporter

Process

Vi startede med, sætte Raspberry Pi op med installer Rasbian. Kode snake spillet i Python. Hvor vi så tilsluttede, Sense Hat til Raspberry.
Under processen har vi testede og fået den til funger med det koder som kan ses under overskrifted kode.
Efter dette har vi kodet Maze og vaterpas og testede til vi fik dem til virke koderne kan ses under kode.

Kode

Snake Spillet

```
from random import randint

from time import sleep

from sense_hat import SenseHat



sense = SenseHat()


temp = sense.get_temperature_from_pressure()

print("Temperature: %s C" % temp)



class SnakeGame():
    def __init__(self):
        self.width = 8
        self.height = 8
        self.food_color = (255,0,0)
        self.snake_color = (0,255,0)

    def start_game(self):
        self.position = [[0,0]]
        self.place_food()
        self.speed = [1,0]
        self.time_speed = 0.4

        self.playing = True
        while self.playing:
            sense.clear(0, 0, 0)
```

```
        self.inputs()
        if (not self.collsion_check()):
            self.move()
            self.draw()
            sleep(self.time_speed)
        else:
            sense.clear(0,0,0)
            sense.show_message(str(len(self.position)), text_colour=[255,0,0],back_colour=[255,255,255])
            self.start_game()
    pass

  def inputs(self):
    for event in sense.stick.get_events():
        if event.action == 'pressed' and event.direction == 'up':
            self.speed = [0,-1]
        if event.action == 'pressed' and event.direction == 'down':
            self.speed = [0,1]
        if event.action == 'pressed' and event.direction == 'right':
            self.speed = [1,0]
        if event.action == 'pressed' and event.direction == 'left':
            self.speed = [-1,0]

  def collsion_check(self):
    if(self.wall_check() or self.self_collision()):
        return True
    else:
        self.food_collsion()
        return False

  def wall_check(self):
```

```
    if(self.position[0][0] + self.speed[0] < self.width and self.position[0][0] + self.speed[0] >= 0 and
self.position[0][1] + self.speed[1] < self.height and self.position[0][1] + self.speed[1] >= 0):
        return False
    else:
        return True

  def self_collision(self):
    for i, e in list(enumerate(self.position)):
      if (i != 0):
        if (e[0] == self.position[0][0] and e[1] == self.position[0][1]):
          return True;
    return False

  def food_collsion(self):
    if (self.position[0][0] + self.speed[0] == self.food[0] and self.position[0][1] + self.speed[1] ==
self.food[1]):
      self.position.append([9,9])
      self.place_food()
      self.time_speed -= 0.01
    pass

  def move(self):
    self.move_tail()

    self.position[0][0] += self.speed[0]
    self.position[0][1] += self.speed[1]
```

```
    def move_tail(self):
        for i, e in reversed(list(enumerate(self.position))):
            if (i > 0):
                self.position[i][0] = self.position[i - 1][0]
                self.position[i][1] = self.position[i - 1][1]

    def place_food(self):

        self.food = [randint(0,7), randint(0,7)]

        for i, e in list(enumerate(self.position)):
            if (self.position[i][0] == self.food[0] and self.position[i][1] == self.food[1]):
                self.place_food()
                print("nono")
                break;

    def draw(self):
        for i in self.position:
            sense.set_pixel(i[0], i[1], self.snake_color)

        sense.set_pixel(self.food[0], self.food[1],self.food_color)

snake = SnakeGame()
snake.start_game()
```

Maze Spillet

```
import math
from time import sleep
from random import randint
from sense_hat import SenseHat
```

```
sense = SenseHat()

x = (255,255,255)
h = (0,0,255)
f = (0,255,0)
p = (0,0,0)

maps = [[x,x,x,x,x,x,x,x,
        x,p,x,f,p,p,p,x,
        x,p,x,x,x,x,p,x,
        x,p,x,x,x,x,p,x,
        x,p,x,x,x,x,p,x,
        x,p,x,x,x,x,p,x,
        x,p,p,p,p,p,p,x,
        x,x,x,x,x,x,x,x],

       [x,x,x,x,x,x,x,x,
        x,p,x,x,x,x,x,x,
        x,p,x,x,x,x,x,x,
        x,p,x,x,x,x,x,x,
        x,p,x,x,x,x,x,x,
        x,p,x,x,x,x,x,x,
        x,f,x,x,x,x,x,x,
        x,x,x,x,x,x,x,x],

       [x,x,x,x,x,x,x,x,
        x,p,x,p,p,p,h,x,
        x,p,x,f,x,p,p,x,
```

```
        x,p,x,x,x,x,p,x,
        x,p,x,p,p,x,p,x,
        x,p,p,p,p,p,p,x,
        x,h,x,h,x,x,h,x,
        x,x,x,x,x,x,x,x]]

class Maze:
    def __init__(self):
        self.width = 8
        self.height = 8
        self.ball_pos = [1.0,1.0]
        self.ball_vel = [0,0]

        self.ball_color = (255,0,0)
        self.wall_color = (255,0,0)

        self.maze_map = maps[randint(0, len(maps) - 1)]

    def get_input(self):
        orientation = sense.get_orientation_radians()

        self.ball_vel[0] = (orientation["pitch"] * -1)
        self.ball_vel[1] = orientation["roll"]

    def check_collision(self):
        x_pos = math.floor(self.ball_pos[0] + self.ball_vel[0]) + math.floor(self.ball_pos[1]) * 8
        y_pos = math.floor(self.ball_pos[0]) + math.floor(self.ball_pos[1] + self.ball_vel[1]) * 8

        if (self.maze_map[x_pos] == x):
            self.ball_vel[0] = 0
```

```
    if (self.maze_map[y_pos] == x):
        self.ball_vel[1] = 0

    pos = math.floor(self.ball_pos[0] + self.ball_vel[0]) + math.floor(self.ball_pos[1] + self.ball_vel[1]) * 8

    if(self.maze_map[pos] == h):
        self.ball_pos = [1.0, 1.0]
    elif(self.maze_map[pos] == f):
        print("hell")
        for i in range(0,30):
            for varx in range(0, 8):
                for vary in range(0, 8):
                    sense.set_pixel(varx, vary, randint(0,255), randint(0,255), randint(0,255))
            sleep(0.1)




        self.maze_map = maps[randint(0, len(maps) - 1)]
        self.ball_pos = [1.0, 1.0]

        return False
    return True


def move(self):
    self.ball_pos[0] += self.ball_vel[0]
    self.ball_pos[1] += self.ball_vel[1]

def draw(self):
```

```
        for i in range(len(self.maze_map)):
            x_pos = i % 8
            y_pos = math.floor(i / 8)

            sense.set_pixel(x_pos, y_pos, self.maze_map[i])

        sense.set_pixel(math.floor(self.ball_pos[0]), math.floor(self.ball_pos[1]), self.ball_color)

m = Maze()

while (True):
    m.get_input()
    if (m.check_collision()):
        m.move()
        m.draw()
```

Vaterpas

```
from sense_hat import SenseHat
# navn giver sensehaten
sense = SenseHat()
#giver farve til vaterpasset
line_color = (4,50,200)

#Laver et while for den køre hele tiden.
running = True
while running:
    #giver variable til orietnering
    orientation = sense.get_orientation_radians()
    sense.clear(0,0,0)
```

```
#orienter x og y position via orientering
x_pos = 3.5 + orientation["pitch"] * 6
y_pos = 4
# lyser på pixelne
sense.set_pixel(x_pos, y_pos, line_color)
sense.set_pixel(x_pos + 1, y_pos, line_color)
```

Kilder

- https://github.com/martinohanlon/AstroPiSnake
- https://pythonhosted.org/sense-hat/api/
- https://pythonhosted.org/sense-hat/api/

Hardware

- Raspberry Pi
- Sense HAT

Software

- Etcher
- Rasbian
- Python

Konklusion

Vi har i dette øvelse kodett et snake spil, maze (labyrent) spil og en vaterpas i Python til et Sense HAT tilsluttede raspberry pi.