5.Rocker control

Learning goals: Read and process the rocker event by modifying the direction angle of the RGB matrix to achieve an arrow pattern indicating multiple directions.

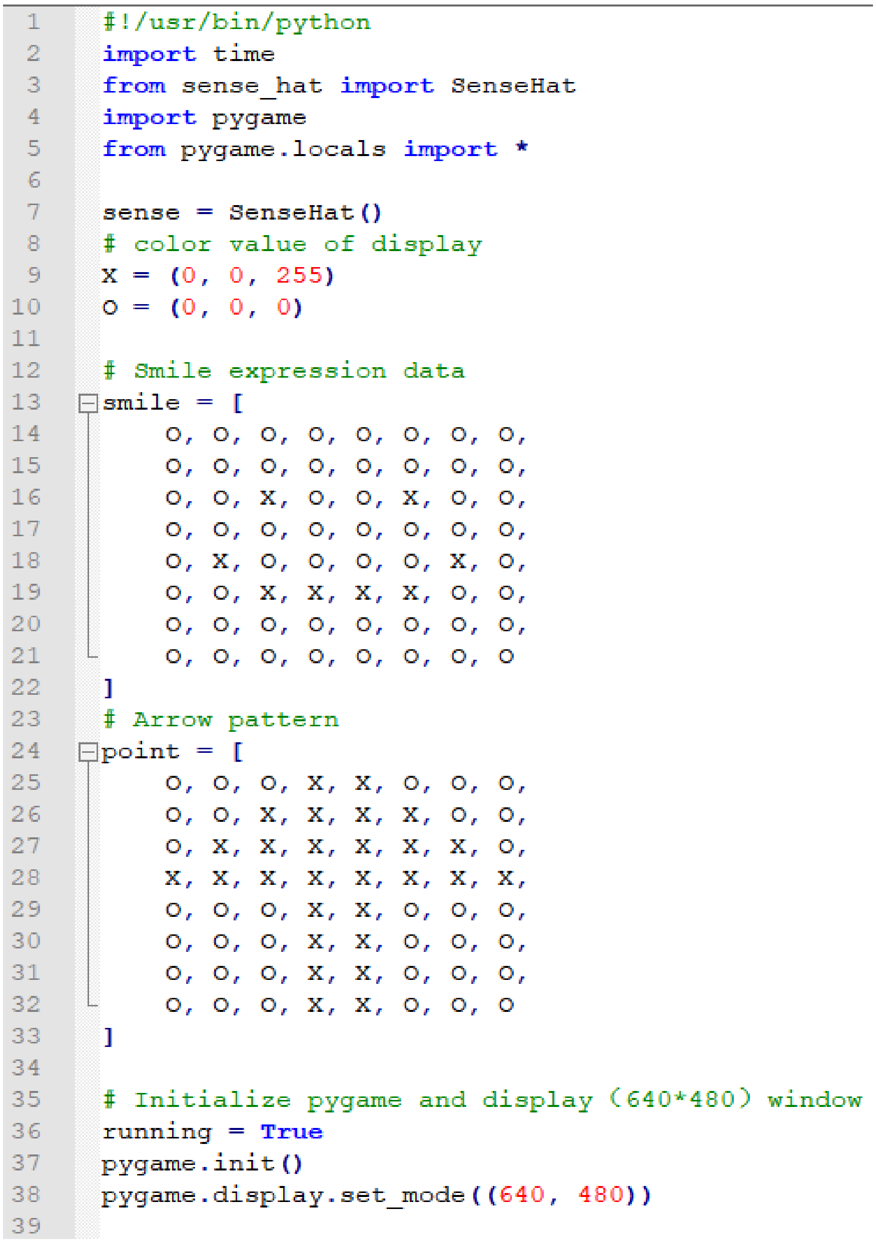
Experimental phenomena: After running the program. The black window interface of pygame pops up on the system desktop. This interface needs to be in the active state.
when the rocker is up, the up arrow is displayed,
when the rocker is down, the down arrow is displayed,
when the rocker is left, the left arrow is displayed,
when the rocker is right, the right arrow is displayed,
When the rocker is pressed, the smile is displayed.
Otherwise, the content is not displayed on the RGB dot matrix.

1.Create python file

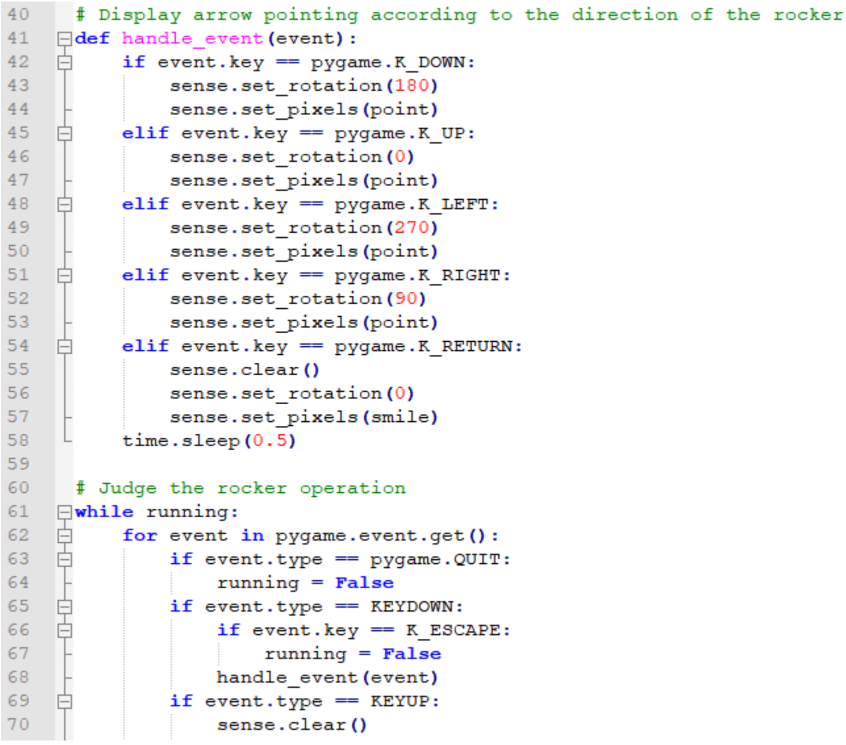
```
nano joystick.py
```

We need to input content as shown below:

```
#!/usr/bin/python
import time
from sense_hat import SenseHat
import pygame
from pygame.locals import *

sense = SenseHat()

# color value of display
X = (0, 0, 255)
O = (0, 0, 0)

# Smile expression data
smile = [
    O, O, O, O, O, O, O, O,
    O, O, O, O, O, O, O, O,
    O, O, X, O, O, X, O, O,
    O, O, O, O, O, O, O, O,
    O, X, O, O, O, O, X, O,
    O, O, X, X, X, X, O, O,
    O, O, O, O, O, O, O, O,
    O, O, O, O, O, O, O, O
]
# Arrow pattern
point = [
    O, O, O, X, X, O, O, O,
    O, O, X, X, X, X, O, O,
    O, X, X, X, X, X, X, O,
    X, X, X, X, X, X, X, X,
    O, O, O, X, X, O, O, O,
    O, O, O, X, X, O, O, O,
    O, O, O, X, X, O, O, O,
    O, O, O, X, X, O, O, O
]
# Initialize pygame and display（640*480）window
running = True
pygame.init()
pygame.display.set_mode((640, 480))

# Display arrow pointing according to the direction of the rocker
def handle_event(event):
    if event.key == pygame.K_DOWN:
        sense.set_rotation(180)
        sense.set_pixels(point)
    elif event.key == pygame.K_UP:
        sense.set_rotation(0)
        sense.set_pixels(point)
    elif event.key == pygame.K_LEFT:
        sense.set_rotation(270)
        sense.set_pixels(point)
    elif event.key == pygame.K_RIGHT:
        sense.set_rotation(90)
        sense.set_pixels(point)
    elif event.key == pygame.K_RETURN:
        sense.clear()
        sense.set_rotation(0)
        sense.set_pixels(smile)
    time.sleep(0.5)

# Judge the rocker operation
while running:
    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)
        if event.type == KEYUP:
            sense.clear()
```



```
#!/usr/bin/python
import time
from sense_hat import SenseHat
import pygame
from pygame.locals import *

sense = SenseHat()
# color value of display
X = (0, 0, 255)
O = (0, 0, 0)

# Smile expression data
smile = [
    O, O, O, O, O, O, O, O,
    O, O, O, O, O, O, O, O,
    O, O, X, O, O, X, O, O,
    O, O, O, O, O, O, O, O,
    O, X, O, O, O, O, X, O,
    O, O, X, X, X, X, O, O,
    O, O, O, O, O, O, O, O,
    O, O, O, O, O, O, O, O
]
# Arrow pattern
point = [
    O, O, O, X, X, O, O, O,
    O, O, X, X, X, X, O, O,
    O, X, X, X, X, X, X, O,
    X, X, X, X, X, X, X, X,
    O, O, O, X, X, O, O, O,
    O, O, O, X, X, O, O, O,
    O, O, O, X, X, O, O, O,
    O, O, O, X, X, O, O, O
]

# Initialize pygame and display (640*480) window
running = True
pygame.init()
pygame.display.set_mode((640, 480))

```



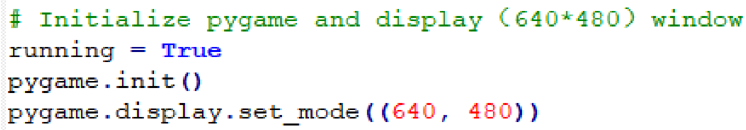
```
# Display arrow pointing according to the direction of the rocker
def handle_event(event):
    if event.key == pygame.K_DOWN:
        sense.set_rotation(180)
        sense.set_pixels(point)
    elif event.key == pygame.K_UP:
        sense.set_rotation(0)
        sense.set_pixels(point)
    elif event.key == pygame.K_LEFT:
        sense.set_rotation(270)
        sense.set_pixels(point)
    elif event.key == pygame.K_RIGHT:
        sense.set_rotation(90)
        sense.set_pixels(point)
    elif event.key == pygame.K_RETURN:
        sense.clear()
        sense.set_rotation(0)
        sense.set_pixels(smile)
    time.sleep(0.5)

# Judge the rocker operation
while running:
    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)
        if event.type == KEYUP:
            sense.clear()
```

Please press Ctrl+O to save, press Ctrl+X to quit.

The code of the experiment, please refer to **joystick.py** in the Python sample program folder.

2.Commonly function

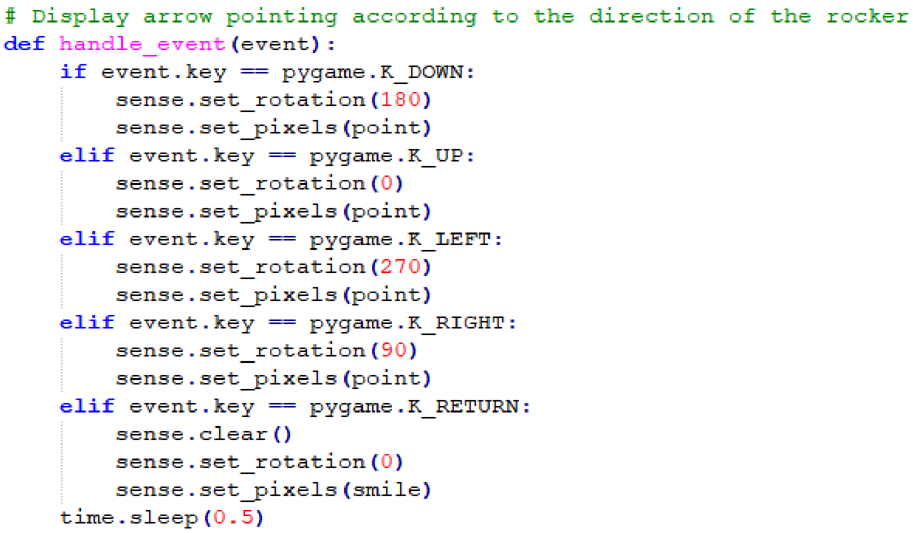
① Import pygame library, display pygame black window.



```
# Initialize pygame and display (640*480) window
running = True
pygame.init()
pygame.display.set_mode((640, 480))
```

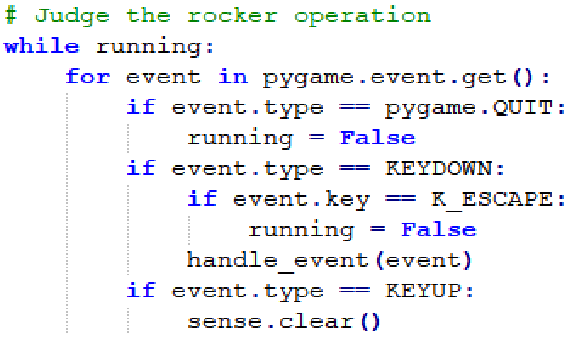
② Use the system clock **time library** to set the program delay: time.sleep(0.5)

③ Create handle_event function, analyze the direction of the joystick operation, display the corresponding arrow pattern



```
# Display arrow pointing according to the direction of the rocker
def handle_event(event):
    if event.key == pygame.K_DOWN:
        sense.set_rotation(180)
        sense.set_pixels(point)
    elif event.key == pygame.K_UP:
        sense.set_rotation(0)
        sense.set_pixels(point)
    elif event.key == pygame.K_LEFT:
        sense.set_rotation(270)
        sense.set_pixels(point)
    elif event.key == pygame.K_RIGHT:
        sense.set_rotation(90)
        sense.set_pixels(point)
    elif event.key == pygame.K_RETURN:
        sense.clear()
        sense.set_rotation(0)
        sense.set_pixels(smile)
    time.sleep(0.5)
```

④ Create while loop and for loop to detect button presses



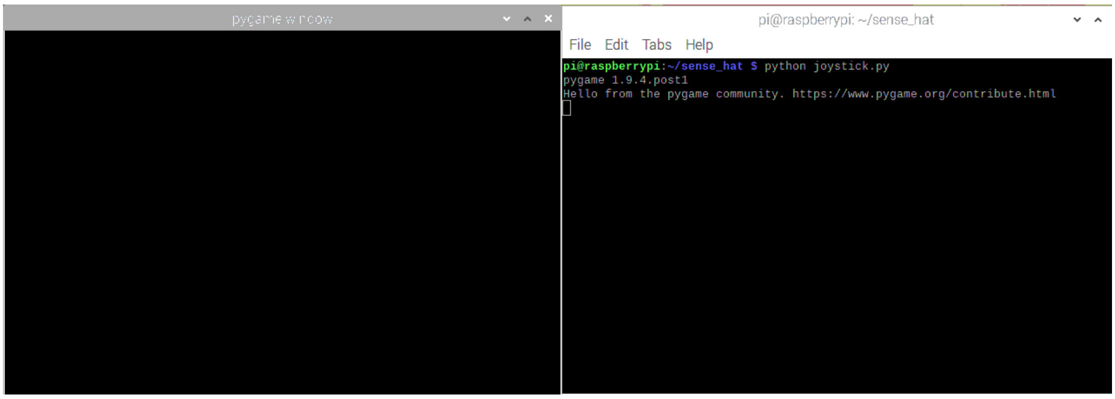
```
# Judge the rocker operation
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        if event.type == KEYDOWN:
            if event.key == K_ESCAPE:
                running = False
            handle_event(event)
        if event.type == KEYUP:
            sense.clear()
```

3.Running program

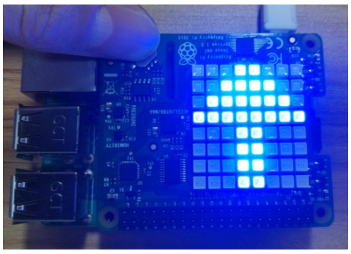
Input the following command to running:

python joystick.py

After running the program. The black window interface of pygame pops up on the system desktop. This interface needs to be in the active state.
when the rocker is up, the up arrow is displayed,
when the rocker is down, the down arrow is displayed,
when the rocker is left, the left arrow is displayed,
when the rocker is right, the right arrow is displayed,
When the rocker is pressed, the smile is displayed.
Otherwise, the content is not displayed on the RGB dot matrix.
As shown below:



```
pi@raspberrypi:~/sense_hat $ python joystick.py
pygame 1.9.4.post1
Hello from the pygame community. https://www.pygame.org/contribute.html
```



! Note: This program cannot be started by remote login tool such as PuTTY. We need to use the desktop or VNC Viewer to run the program. The desktop will pop up a black window, and the black window needs to be active.