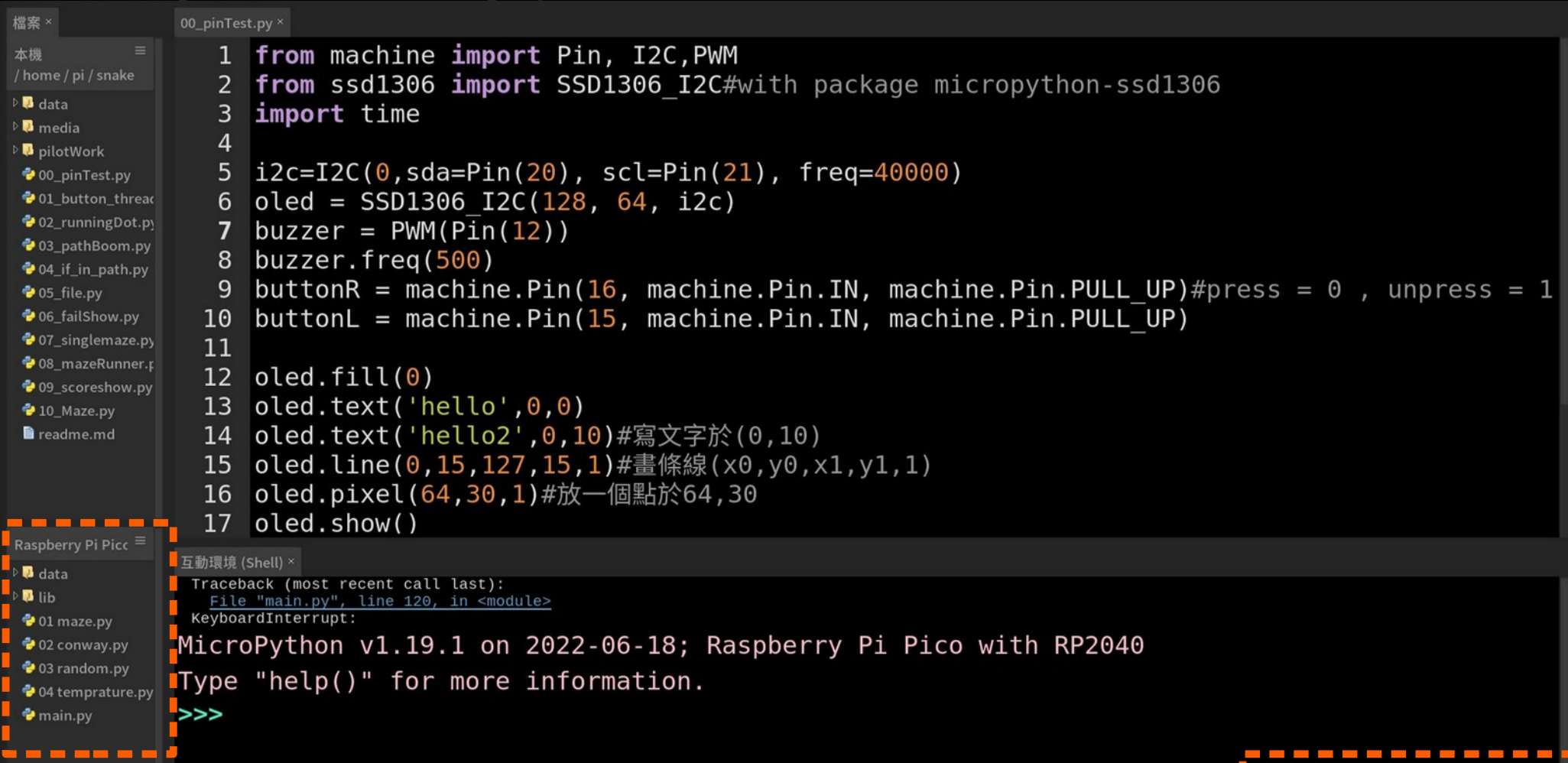


python 讀書會
資科專題迷你遊戲機
貪食蛇迷宮

chyijiunn



00_pinTest.py



```
1 from machine import Pin, I2C, PWM
2 from ssd1306 import SSD1306_I2C#with package micropython-ssd1306
3 import time
4
5 i2c=I2C(0,sda=Pin(20), scl=Pin(21), freq=40000)
6 oled = SSD1306_I2C(128, 64, i2c)
7 buzzer = PWM(Pin(12))
8 buzzer.freq(500)
9 buttonR = machine.Pin(16, machine.Pin.IN, machine.Pin.PULL_UP)#press = 0 , unpress = 1
10 buttonL = machine.Pin(15, machine.Pin.IN, machine.Pin.PULL_UP)
11
12 oled.fill(0)
13 oled.text('hello',0,0)
14 oled.text('hello2',0,10)#寫文字於(0,10)
15 oled.line(0,15,127,15,1)#畫條線(x0,y0,x1,y1,1)
16 oled.pixel(64,30,1)#放一個點於64,30
17 oled.show()
```

互動環境 (Shell) ×

```
Traceback (most recent call last):
  File "/dev/ttyACM0", line 120, in <module>
KeyboardInterrupt:

MicroPython v1.19.1 on 2022-06-18; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>>
```

Package - ssd1306

The screenshot shows the Raspberry Pi Pico IDE interface. At the top, a menu bar includes '檔案', '編輯', '檢視', '執行', '工具', and '說明'. Below the menu, a toolbar contains various icons. A file explorer on the left shows the project structure: '本機' (Local) with path '/home/pi/snake', and a 'pilotWork' directory containing files like '00_pinTest.p', '01_button_t', '02_runningD', '03_pathBoor', '04_if_in_pati', '05_file.py', '06_failShow.', '07_singlema', '08_mazeRun', '09_scoresho', '10_Maze.py', and 'readme.md'. The main editor window displays two lines of Python code:

```
1 from machine import Pin, I2C, PWM
2 from ssd1306 import SSD1306_I2C
```

. A tooltip '跳出 (Step out)' is visible over the code. A search bar at the top of the package manager window contains 'ssd1306' and a button '在 PyPI 中搜尋'. The package details for 'micropython-ssd1306' are shown, including the version '0.3', a summary 'ssd1306 module for MicroPython', the author 'Stefan Lehmann', and links to the GitHub repository and PyPI page. The package is listed as installed under the heading '<安裝>'. At the bottom, there are buttons for '安裝', '...', and '關閉'.

檔案 編輯 檢視 執行 工具 說明

跳出 (Step out)

檔案 ×

本機
/home/pi/snake

data
media
pilotWork
00_pinTest.p
01_button_t
02_runningD
03_pathBoor
04_if_in_pati
05_file.py
06_failShow.
07_singlema
08_mazeRun
09_scoresho
10_Maze.py
readme.md

Raspberry Pi Pico

data
lib
01 maze.py
02 conway.p
03 random.p
04 temperatu

為 Raspberry Pi Pico @ /dev/ttyACM0 管理套件

ssd1306 在 PyPI 中搜尋

<安裝>

micropython-ssd1306

最新穩定版本: 0.3
摘要: ssd1306 module for MicroPython
作者: Stefan Lehmann
網站主頁: <https://github.com/stlehmann/micropython-ssd1306>
PyPI 頁面: <https://pypi.org/project/micropython-ssd1306/>

安裝 ... 關閉

貪食蛇需求

- 吃東西會變長
- 一直跑
- 可轉彎
- 撞到自己 **gameOver**

滿足貪食蛇課程需求

- 吃東西會變長
 - 一直跑
 - 可轉彎
 - 撞到自己 **gameOver**
- 不只變長、還得縮短
 - 新增像素點
 - 搭配按鈕，至少兩顆
 - 紀錄座標

滿足貪食蛇課程需求

- 吃東西會變長
- 跑：新增像素點
- 轉彎：兩顆按鈕
- 撞到：紀錄座標

- **oled.pixel(x,y,1)**
oled.show()

滿足跑需求

- 跑：新增像素點

- **While True:**

```
oled.pixel(x,y,1)  
oled.show()
```

滿足跑需求

- 跑：新增像素點
- 不會跑

- **While True:**

```
oled.pixel(x,y,1)  
oled.show()
```


滿足跑 - 方向需求

- 跑：新增像素點
- 會跑、跑去哪？

- **While True:**

```
oled.pixel(x,y,1)  
oled.show()  
x+1  
y+1
```

設計的直覺 **vs** 使用的直覺

- **Direction** ← **button**

- **While True:**

oled.pixel(x,y,1)

oled.show()

if button1 == 0 AND button2 == 0 : x = x+1

滿足跑 - 方向需求

- 跑：新增像素點
- 會跑、跑去哪？
 - **Direction** ← **button**

- **While True:**

```
oled.pixel(x,y,1)
oled.show()
if button1 == 0:x+1
if button2==0:y+1
```

滿足跑 - 方向需求

- 跑：新增像素點
- 不會跑、跑去哪？
- **Direction \leftarrow button**

- **While True:**
 button1 == 0 : direction += 1
 button2 == 0 : direction += 1
- **While True:**
 oled.pixel(x,y,1)
 oled.show()
 if direction == ? : x+1
 if direction == ? : y+1

滿足跑 - 方向需求

- 跑：新增像素點
- 不會跑、跑去哪？

- **Direction ← button**

- **Derection == 0**

1

2

3

- **While True:**

- button1 == 0 : direction +=1**

- button2 == 0 : direction -= 1**

- **While True:**

- oled.pixel(x,y,1)**

- oled.show()**

- if direction == ? : x+1**

- if direction == ? : y+1**

滿足跑 - 方向需求

- 跑：新增像素點
- 不會跑、跑去哪？

- **Direction ← button**

- **Direction == 0**

1

2

3

- **While True:**

- button1 == 0 : direction += 1**

- button2 == 0 : direction -= 1**

- **While True:**

- oled.pixel(x,y,1)**

- oled.show()**

- if direction % 4 == 0 : x+1**

- if direction % 4 == 1 : y+1**

- then ?**

02_runningDot.py

```
15 direction = 0
16
17 def button_thread():
18     global direction
19     while True:
20         if buttonR.value() == 0: direction = direction + 1
21         if buttonL.value() == 0: direction = direction - 1
22         sleep(0.14)
23
24 _thread.start_new_thread(button_thread, ())
25
26 while True:
27     oled.fill(0)
28     if direction % 4 == 0: x += 1
29     if direction % 4 == 1: y += 1
30     #另外兩個怎麼設計呢？
31     oled.pixel(x,y,1)
32
33     if buttonR.value() == 0 and buttonL.value() == 0: break
34     oled.show()
```

滿足轉彎需求

- 跑：新增像素點
- 轉彎：兩顆按鈕
- 撞到：紀錄座標

- **Oled.fill(x,y,1)**
- **If button ==0:turn()**
- **Path = []**
if in path:sys.exit

滿足紀錄需求

- 跑：新增像素點
- 轉彎：兩顆按鈕
- 撞到：紀錄座標

- **Oled.fill(x,y,1)**
- **If button == 0:turn()**
- **Path = []**
if in path:sys.exit

滿足紀錄需求

- 撞到：紀錄座標

- **Path = []**
- **Path.append()**
- **if in path:sys.exit**