

這裡要介紹書中出現的兩位解說員，以及必須先瞭解的內容，包括接下來要製作的遊戲、下載檔案的網頁用法、學習方式等。

》》 登場人物簡介

本書由在《Python 遊戲開發講座入門篇 | 基礎知識與 RPG 遊戲》也負責過協助工作的「水鳥川堇」以及「白川彩華」擔任解說員。這兩位 Python 專家現在負責以下工作。



水鳥川堇

IT 公司的老闆，在公司成立電腦遊戲事業部，開始開發教育用遊戲軟體，同時也在母校慶王大學擔任程式設計教學的客座副教授。

白川彩華

在慶王大學學習過程式設計的理科女。因強大的程式設計技術實力而被延攔至水鳥川堇的公司，從事遊戲開發工作。這些遊戲軟體是教育機構或教科書出版社委託製作，目的是讓孩童們能愉快地學習程式設計，他對自己的工作感到自豪，每天都孜孜不倦地設計程式。



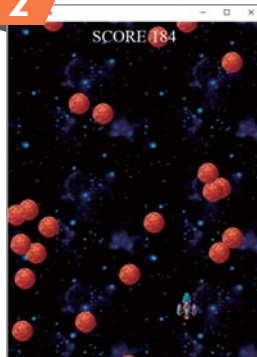
》》 本書製作的遊戲

第 1～2 章要學習開發遊戲的基本技術，之後進行演練。

本書將 步 步詳細解說動作遊戲、彈幕射擊遊戲、3D 賽車遊戲等熱門遊戲類型是如何設計程式？

請反覆練習，磨練程式設計的能力！

Chapter 2 小遊戲



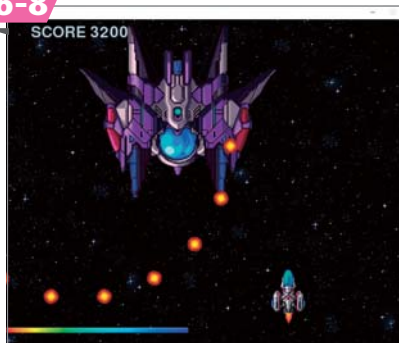
《METEOR》(左)與《流浪貓》(右)是躲避敵人的小遊戲。這是最適合學習設計遊戲程式基本技術的題材。

Chapter 3-4 動作遊戲



《提心吊膽企鵝迷宮》是取得迷宮內所有糖果就過關的吃點數遊戲。

Chapter 6-8 彈幕射擊遊戲



《Galaxy Lancer》是在擊落敵機的同時，不斷前進的垂直捲動式彈幕射擊遊戲。

Chapter 9-11 3D 賽車遊戲



《Python Racer》是大量運用 3D 模擬技術，呈現出逼真立體感、彎道、賽道起伏的賽車遊戲。

你注意到隱藏角色了嗎？（答案請見 P.314）



特典 遊戲啟動器及掉落物拼圖遊戲



也能啟動原創遊戲！



《Game Center 208X》（左）是可以選擇、啟動遊戲的遊戲啟動器。《Animal》（右）是相同顏色的方塊排成一排即可消除的標準掉落物拼圖遊戲。

Lesson

1-3

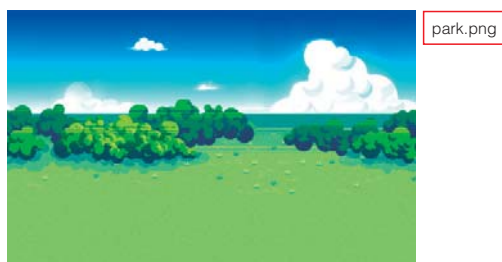
角色的動畫

遊戲中出現的角色會跑、跳，做出各種動作。2D 遊戲是使用多張影像準備角色的動作，再依序顯示這些影像，變成動畫。以下將說明這個方法。

》》 描繪影像

tkinter 會把影像或圖形繪製在畫布元件上。以下將先確認顯示一張影像檔案的程式。這個程式使用了下面的影像檔案，你可以透過本書提供的網址下載這個檔案。

圖 1-3-1 這次使用的影像檔案



請輸入以下程式，另存新檔之後，再執行程式。

程式 ▶ list0103_1.py

```

1  import tkinter
2
3  root = tkinter.Tk()
4  root.title("在Canvas繪製影像")
5  canvas = tkinter.Canvas(width=480, height=300)
6  canvas.pack()
7  img_bg = tkinter.PhotoImage(file="park.png")
8  canvas.create_image(240, 150, image=img_bg)
9  root.mainloop()

```

匯入 tkinter 模組

建立視窗元件
設定視窗標題
建立畫布元件
置入畫布
在變數 img_bg 載入影像
在畫布繪製影像
顯示視窗

執行這個程式，會在視窗內顯示影像（圖 1-3-2）。

畫布是用 `Canvas()` 命令設定寬度與高度，如第 5 行所示。用第 6 行的 `pack()` 命令在視窗內置入畫布。`pack()` 命令會在視窗內置入 GUI，這次的用法是根據畫布大小來決定視窗尺寸。

圓形的碰撞偵測

接著要說明圓形的碰撞偵測。其實碰撞偵測的方法有很多種，不過只要記住「判斷矩形是否重疊」以及「判斷圓形是否重疊」這兩種方法，就可以運用在各種情況。

》》》 用圓形進行碰撞偵測

以下要確認判斷兩個圓形是否重疊的程式，說明確認動作後的判斷方法。請輸入以下程式，另存新檔之後，再執行程式。

程式 ▶ list0202_1.py

```

1  import tkinter
2  import math
3
4  def hit check circle():
5      dis = math.sqrt((x1 x2)*(x1 x2) + (y1 y2)
6      *(y1 y2))
7      if dis <= r1 + r2:
8          return True
9      return False
10
11 def mouse move(e):
12     global x1, y1
13     x1 = e.x
14     y1 = e.y
15     col = "green"
16     if hit check circle() == True:
17         col = "lime"
18         canvas.delete("CIR1")
19         canvas.create oval(x1 r1, y1 r1, x1+r1, y1+r1,
20         fill=col, tag="CIR1")
21
22 root = tkinter.Tk()
23 root.title("用圓形進行碰撞偵測")
24 canvas = tkinter.Canvas(width=600, height=400,
25         bg="white")
26 canvas.pack()
27 canvas.bind("<Motion>", mouse move)
28
29 x1 = 50
30 y1 = 50
31 r1 = 40
32 canvas.create oval(x1 r1, y1 r1, x1+r1, y1+r1,
33         fill="green", tag="CIR1")
34
35 x2 = 300
36 y2 = 200
37 r2 = 80
38 canvas.create oval(x2 r2, y2 r2, x2+r2, y2+r2,
```

匯入 tkinter 模組

匯入 math 模組

定義函數

計算二點之間的距離再代入 dis

如果 dis 的值小於兩個圓形半徑的加總
就傳回 True

如果大於，則傳回 False

定義函數

變成全域變數

綠色圓形的 X 座標成為游標的座標

綠色圓形的 Y 座標成為游標的座標

把 green 字串代入 col

兩個圓形接觸之後

把 lime 字串代入 col

暫時刪除綠色圓形

繪製綠色圓形

建立視窗元件

設定視窗標題

建立畫布元件

置入畫布

移動滑鼠游標時執行的函數

綠色圓形中心的 X 座標

綠色圓形中心的 Y 座標

綠色圓形的半徑

在畫布繪製綠色圓形

橘色圓形中心的 X 座標

橘色圓形中心的 Y 座標

橘色圓形的半徑

在畫布繪製橘色圓形

```
35 fill="orange")
36 root.mainloop()
```

顯示視窗

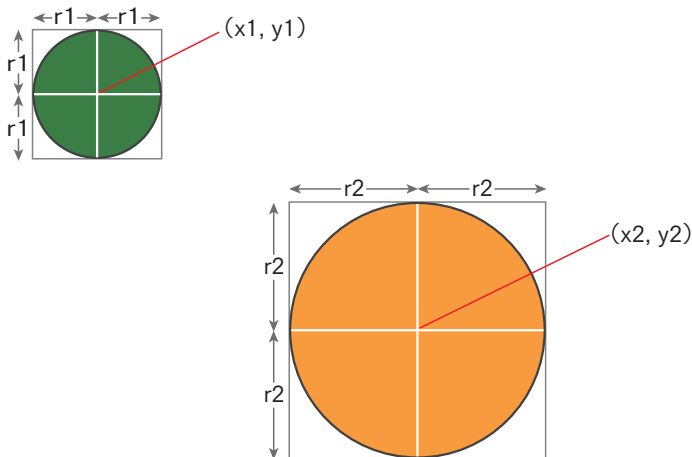
執行這個程式，會顯示綠色及橘色圓形，如下圖所示。在視窗上移動滑鼠游標，就能移動綠色圓形。綠色圓形接觸到橘色圓形會變成亮綠色。請從各種方向接觸橘色圓形，確認碰撞偵測的執行狀態。

圖 2-2-1 確認碰撞偵測



圓形的碰撞偵測是利用兩個圓心中心之間的距離（點數）來進行。在這個程式中，綠色圓形的中心座標是 (x_1, y_1) ，半徑是 r_1 ，橘色圓形的中心座標是 (x_2, y_2) 、半徑是 r_2 。

圖 2-2-2 兩個圓形中心的距離



兩個圓心的中心距離可以用數學計算兩點間距離的公式算出來，請見以下使用了根號的公式。

吃點數遊戲

動作遊戲有各種類型，在開發之前，將先說明動作遊戲及本章要製作的吃點數遊戲。

》》 何謂動作遊戲

動作遊戲是指即時操作角色，順利達成遊戲目的的遊戲類型。所謂的目的包括

- 抵達目標
- 打倒大魔王
- 取得全部的道具

等等，這些目的也會隨著遊戲而改變。遊戲內容很廣泛，例如也有邊解謎，邊前進的動作遊戲。

動作遊戲包括了稱作**吃點數**類型的遊戲。

你聽過吃點數遊戲嗎？

南夢宮公司的《小精靈》遊戲就是 一款知名的吃點數遊戲。這個遊戲是操作嘴巴會開闔的黃色角色，邊躲避怪獸敵人，邊吃掉（消除）路上的圓點。小精靈在 1980 年推出商用卡帶遊戲，十分暢銷。在美國受歡迎的程度遠勝日本，甚至將其製作成電視動畫。當時許多遊戲公司開發了商用吃點數遊戲，並在遊樂場設置了各種吃點數遊戲。

》》 適合用來學習製作遊戲

吃點數遊戲適合遊戲開發初學者用來學習如何製作遊戲。原因在於

- 透過用二維列表定義迷宮，學會與列表有關的重要知識
- 可學會判斷地面與牆壁，以及移動角色的基本處理技巧
- 學習建立敵人角色行為類型的演算法基礎
- 用固定畫面檢視整個遊戲，比較容易思考如何改良才能變有趣

這本書要邊製作吃點數類型的動作遊戲，邊學習這些內容。

》》 接下來要製作的遊戲內容

開發新遊戲時，最好用概略的草圖思考畫面結構，這點在《Python 遊戲開發講座入門篇 | 基礎知識與 RPG 遊戲》已經說明過。讓腦中的創意具體成型，就能清楚該如何寫程式。此外，條列出你認為必要的處理，即可明白該從何處著手。

當你在製作原創遊戲時，建議採取這種方法。不過接下來製作的遊戲並不是從創意發想開始，而是確認畫面及規則，並將其完成來展開學習。

遊戲的標題是《提心吊膽企鵝迷宮》。

以下要說明故事、畫面、規則。

■ 故事

我的名字是 penpen。
是生活在靠近南極大陸小島上的企鵝。

penko 很喜歡吃甜食。
所以我今天也要在冰雪地窖收集糖果，
把糖果當作禮物送給她。
你應該知道這是什麼意思吧？
當然是為了跨過戀人未滿的門檻啊！

她是我朋友以上
戀人未滿的
玩伴囉！

對了，情敵 red 的目標
penko。 一定要注意這
壞事的傢伙。

主角的名字是「penpen」，而敵人的名字是「red」。遊戲畫面如下所示。

圖 3-1-1 遊戲畫面



■ 遊戲規則

- 1 用方向鍵上下左右移動 **penpen**
- 2 收集到所有糖果就過關
- 3 碰到敵人會被攻擊
- 4 被攻擊後，**penpen** 的剩餘命數減 1，變成 0 之後遊戲結束
- 5 進入下一個關卡後，會出現新的敵人「**kumagon**」
- 6 全部過關後，就完成遊戲

本章要製作 ❶ 到 ❸，學會如何大致完成「標題畫面→玩遊戲→遊戲結束」的流程。

以下要說明製作賽車遊戲時，必須具備的知識。

》》 何謂賽車遊戲

遊戲產業自 1970 年代誕生至今，製作出各種賽車遊戲。1980 年代遊戲產業的黎明期，出現了垂直捲動及全方向捲動的 2D 遊戲，後來模擬 3D 影像來表現車道的賽車遊戲逐漸成為主流。自 1990 年代起，具有 3DCG 繪圖功能的硬體普及之後，幾乎所有的賽車遊戲都是用 3DCG 繪製。

》》 遊戲的內容

本書要製作用模擬 3D 畫面的技法來繪製車道的賽車遊戲。遊戲的標題為《Python Racer》。以下先確認最後完成的遊戲畫面。

圖 9-1-1 3D 賽車遊戲《Python Racer》



運用擬 3D 技巧繪製道路 | 使用矩形

本章要學習使用 tkinter，以模擬 3D 的方式繪製道路的技法。

》》 使用 tkinter 的理由

本章使用 tkinter 是因為可以簡單描述程式，適合基礎學習。從下一章開始將使用 Pygame。



本章沒有使用影像，所以不需要建立 image 資料夾。

》》 讓板子的寬度產生變化

接下來要製作呈現深度空間的程式。首先要試著繪製距離愈遠寬度愈窄的板子。請輸入以下程式，另存新檔之後，再執行該檔案。

程式 ▶ list0905_1.py

```

1  import tkinter
2
3  root = tkinter.Tk()
4  root.title("繪製道路")
5  canvas = tkinter.Canvas(width=800, height=600,
6  bg="blue")
7  canvas.pack()
8
9  canvas.create_rectangle(0, 300, 800, 600, fill=
10 "green")
11
12 BORD COL = ["white", "silver", "gray"]
13 for i in range(1, 25):
14     w = i*33
15     h = 12
16     x = 400 - w/2
17     y = 288 + i*h
18     col = BORD COL[i%3]
19     canvas.create_rectangle(x, y, x+w, y+h,
20 fill=col)
21
22 root.mainloop()

```

匯入 tkinter 模組

建立視窗元件
設定視窗標題
建立畫布元件

置入畫布

在畫布下半部分繪製綠色矩形

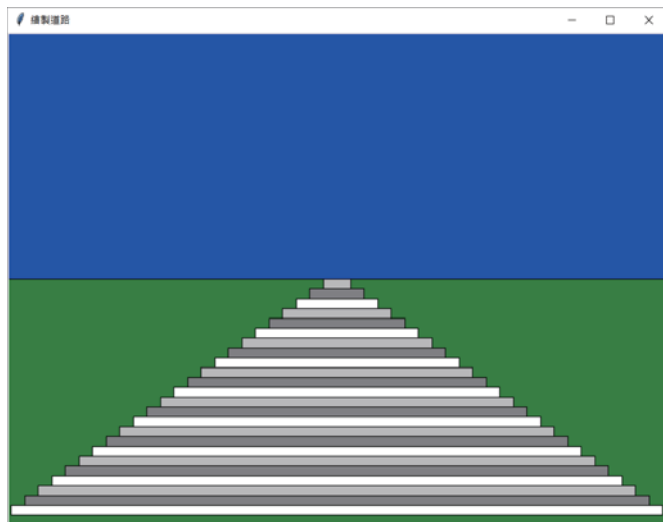
定義板子顏色的列表

i 重複從 1 開始持續加 1 到 25
把板子的寬度代入變數 w
把板子的高度代入變數 h
把板子的 X 座標代入變數 x
把繪製板子的 Y 座標代入變數 y
把板子的顏色代入變數 col
在 (x,y) 的位置繪製寬度 w、高度 h 的板子

顯示視窗

執行這個程式後，就會顯示愈遠、寬度愈短的板子，共畫出 24 片板子。

圖 9-5-1 list0905_1.py 的執行結果



`create_rectangle()` 命令若沒有設定框線粗細或顏色，就會畫出粗細為 1 點的黑色框線。我們可以使用參數 `width=` 設定框線的粗細，用參數 `outline=` 設定顏色。



第 12 行的 `w = i*33` 是板子的寬度，第 13 行的 `h = 12` 是板子的高度。在第 14 ~ 15 行計算板子的 (x, y) 座標，用 `create_rectangle()` 命令繪製板子（矩形）。板子的顏色先用列表定義成 `col`，依序代入 `col`，變成 `col = BORD_COL[i%3]`，藉此設定顏色。

雖然繪製了長短不同的板子，但是形狀很奇怪，稱不上是道路。這是因為每個板子的高度都是 12 所致。3D 空間中的物體愈遠看起來愈小，所以不僅遠方板子的寬度要縮短，高度也要變小才對。接著要改良這個程式，營造出遠近感。

》》》 讓高度也產生變化

改變算式，讓板子的寬度與高度愈遠愈短。請輸入以下程式，另存新檔之後，再執行程式。

讓電腦控制的賽車 在賽道上行駛

讓電腦計算多台賽車的動作，並在道路上行駛。

》》》 本章的資料夾結構

由於進入不同章節，請在「Chapter11」資料夾內也建立「image_pr」資料夾，並在該資料夾內放入《Python Racer》要用的檔案。這個單元除了上一章用過的影像，還使用了藍、黃兩種車種的影像。

接下來還要加入聲音，所以請建立「sound_pr」資料夾，把聲音檔案放入該資料夾。這些素材可以透過本書提供的網址下載。

圖 11-1-1 「Chapter11」資料夾的結構

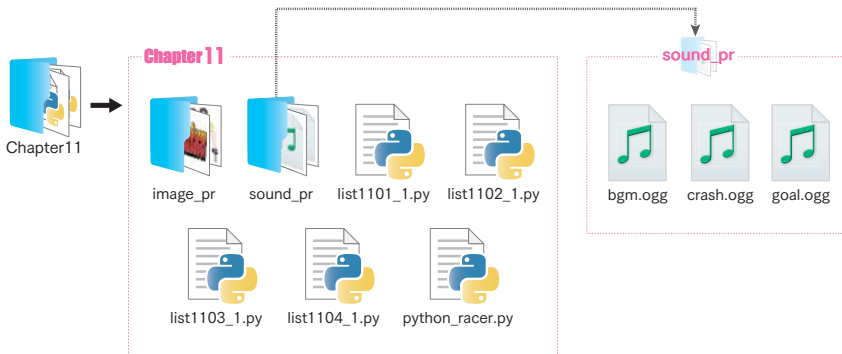


圖 11-1-2 這次使用的影像檔案

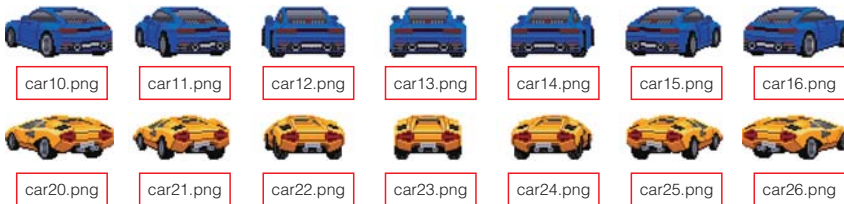


表 11-1-1 這次使用的聲音檔案

檔案名稱	內容
bgm.ogg	賽車時的 BGM
crash.ogg	碰撞時的 SE
goal.ogg	抵達終點時的片尾音樂