

產生地圖、通過關卡實作

授課教師：楊凱翔

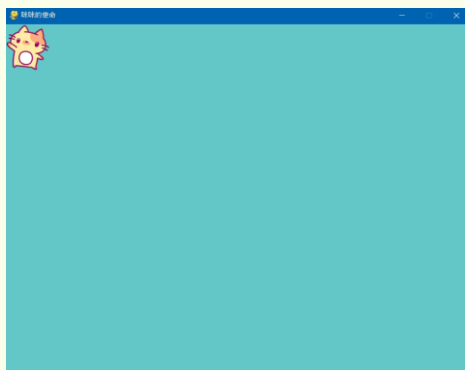
課程回顧

[illegible]

[影片連結](#)

課程回顧

可以用方向鍵控制貓咪在視窗中走動

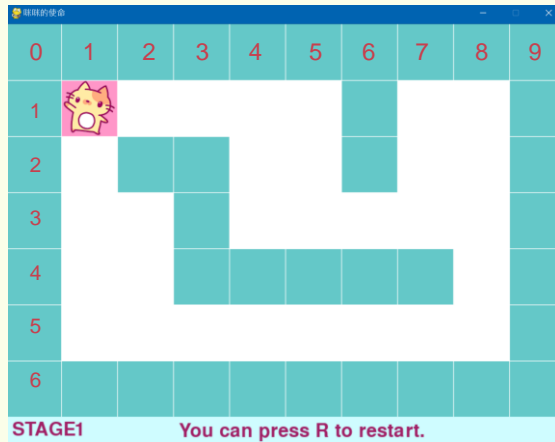


產生地圖

1

產生地圖 分析遊戲畫面

- 視窗大小800x600
- 一格80x80
- 白色可以走
- 藍色不能走
- 走過變粉色



產生地圖 利用串列管理地圖

- 白色可以走→0
- 藍色不能走→1
- 走過變粉色→2

```

27 def main():
28     pygame.init() # 初始化pygame模組
29     pygame.display.set_caption("咪咪的使命") # 設定視窗名稱
30     screen = pygame.display.set_mode((width, height)) # 設定視窗大小
31     img_cat = pygame.image.load('img/mimi.png') # 載入貓咪圖片
32     cat_x, cat_y = 0,0 # 初始化貓咪位置
33
34     maze = [
35         [1,1,1,1,1,1,1,1,1,1],
36         [1,0,0,0,0,0,1,0,0,1],
37         [1,0,1,1,0,0,1,0,0,1],
38         [1,0,0,1,0,0,0,0,0,1],
39         [1,0,0,1,1,1,1,1,0,1],
40         [1,0,0,0,0,0,0,0,0,1],
41         [1,1,1,1,1,1,1,1,1,1]
42     ]

```



產生地圖 利用函數繪製地圖



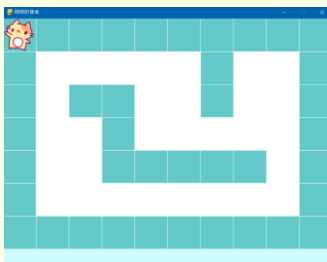
```
9 # 畫迷宮
10 def draw_maze(screen,maze):
11     # 畫格子顏色:0是白色沒走過、1是藍色障礙物、2是粉色已走過
12     for y in range(7):
13         for x in range(10):
14             if maze[y][x] == 0:
15                 # 畫布 顏色 x座標 y座標 寬 長
16                 pygame.draw.rect(screen, WHITE, [x*80, y*80, 80, 80])
17             elif maze[y][x] == 1:
18                 pygame.draw.rect(screen, SKYBLUE, [x*80, y*80, 80, 80])
19             elif maze[y][x] == 2:
20                 pygame.draw.rect(screen, PINK, [x*80, y*80, 80, 80])
21
22     # 畫格子線
23     if x != 0: # 橫的
24         # 畫布 顏色 起始座標 終點座標 寬
25         pygame.draw.line(screen, WHITE, [x*80, 0], [x*80, height-41], 1)
26     if y != 0: # 真的
27         pygame.draw.line(screen, WHITE, [0, y*80], [width, y*80], 1)
28
29 def main():
30     pygame.init() # 初始化pygame模組
31     pygame.display.set_caption("咪咪的使命") # 設定視窗名稱
```



產生地圖 利用函數繪製地圖



- 修改填入顏色，讓底部呈淺藍
- 呼叫draw_maze函數以繪製地圖



```
46 while True:
47     screen.fill((207,252,255))
48     for event in pygame.event.get(): # 迴圈處理pygame事件
49         if event.type == pygame.QUIT: # 當視窗的x被按下
50             pygame.quit() # 斷掉初始化pygame模組
51             sys.exit() # 結束程式
52         if event.type == pygame.KEYDOWN: # 當按下按鍵時
53             if event.key == pygame.K_UP: # 當按鍵為上方向鍵
54                 if cat_y > 0:
55                     cat_y -= 80
56
57             if event.key == pygame.K_DOWN: # 當按鍵為下方向鍵
58                 if cat_y < 560-80:
59                     cat_y += 80
60
61             if event.key == pygame.K_LEFT: # 當按鍵為左方向鍵
62                 if cat_x > 0:
63                     cat_x -= 80
64
65             if event.key == pygame.K_RIGHT: # 當按鍵為右方向鍵
66                 if cat_x < 800-80:
67                     cat_x += 80
68
69     draw_maze(screen, maze)
70     screen.blit(img_cat, [cat_x, cat_y])
71     pygame.display.update()
```



產生地圖 改良角色位置表示方式

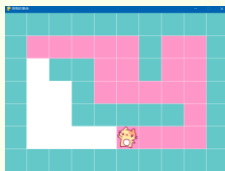
- 以在maze串列中的第幾列(maze_y)、第幾行(maze_x)表示貓咪位置

```
29 def main():
30     pygame.init() # 初始化pygame模組
31     pygame.display.set_caption("咪咪的使命") # 設定視窗名稱
32     screen = pygame.display.set_mode((width, height)) # 設定視窗大小
33     img_cat = pygame.image.load('img/mimi.png') # 載入貓咪圖片
34     maze_x, maze_y = 1, 1 # 初始化貓咪位置
35
36     maze = [
37         [1,1,1,1,1,1,1,1,1],
38         [1,0,0,0,0,0,1,0,1],
39         [1,0,1,1,0,0,1,0,1],
40         [1,0,0,1,0,0,0,0,1],
41         [1,0,0,1,1,1,1,1,1],
42         [1,0,0,0,0,0,0,0,1],
43         [1,1,1,1,1,1,1,1,1]
44     ]
```



產生地圖 改良角色位置表示方式

- 將[cat_x, cat_y]改成[maze_x*80, maze_y*80]
- 若貓咪現在位置為白色，則改成**粉色**
- 當按下**上方向鍵**，若貓咪現在位置上方是白色，將貓咪上移，以此類推



```
46 while True:
47     screen.fill((207,252,255))
48     if maze[maze_y][maze_x] == 0: # 將貓咪現在位置改色
49         maze[maze_y][maze_x] = 2
50     for event in pygame.event.get(): # 迴圈處理pygame事件
51         if event.type == pygame.QUIT: # 當視窗的x被按下
52             pygame.quit() # 解除初始化pygame模組
53             sys.exit() # 結束程式
54         if event.type == pygame.KEYDOWN: # 當按下按鍵時
55             if event.key == pygame.K_UP: # 當按鍵為上方向鍵
56                 if maze[maze_y-1][maze_x] == 0: #
57                     maze_y = maze_y-1 # 更新角色位置
58
59             if event.key == pygame.K_DOWN: # 當按鍵為下方向鍵
60                 if maze[maze_y+1][maze_x] == 0:
61                     maze_y = maze_y+1
62
63             if event.key == pygame.K_LEFT: # 當按鍵為左方向鍵
64                 if maze[maze_y][maze_x-1] == 0:
65                     maze_x = maze_x-1
66
67             if event.key == pygame.K_RIGHT: # 當按鍵為右方向鍵
68                 if maze[maze_y][maze_x+1] == 0:
69                     maze_x = maze_x+1
70
71     draw_maze(screen, maze)
72     screen.blit(img_cat, [maze_x*80, maze_y*80])
73     pygame.display.update()
```



2

通過關卡

通過關卡 通關判定

- 函數 stage_clear 將遍歷二維串列 maze 中的每一個元素
- 若未通關，也就是迷宮中還有白色格子 (maze中還有0)
 - 返回False
- 若通關，則
 - 返回True

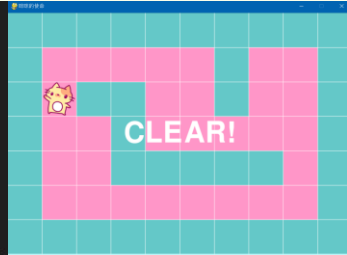
```

22     # 畫格子線
23     if x != 0: # 橫的
24         # 畫布 顏色 起始座標
25         pygame.draw.line(screen, WHITE, [x*80, 0], [x*
26     if y != 0: # 直的
27         pygame.draw.line(screen, WHITE, [0, y*80], [width,
28
29     # 通關判斷(若過關迷宮內只會剩1跟2)
30     def stage_clear(maze):
31         for row in maze:
32             for cell in row:
33                 if cell == 0: # 找到一個0
34                     return False
35         return True
36
37     def main():
38         pygame.init() # 初始化pygame模組
39         pygame.display.set_caption("咪咪的使命") # 設定視窗名稱
40         screen = pygame.display.set_mode((width, height)) # 設定視

```

通過關卡 顯示文字

```
29 # 過關判斷(若過關迷宮內只會有1跟2)
30 def stage_clear(maze):
31     for row in maze:
32         for cell in row:
33             if cell == 0: # 找到一個0
34                 return False
35     return True
36
37 # 繪製過關文字
38 def draw_clear_txt(screen):
39     font = pygame.font.Font(None, 100) # 建立字型物件
40     txt = font.render('CLEAR!', True, WHITE) # 繪製字串
41     txt_rect = txt.get_rect(center=(width // 2, (height-40) // 2)) # 計算文字矩形的位置
42     screen.blit(txt, txt_rect)
43
44 def main():
45     pygame.init() # 初始化pygame模組
46     pygame.display.set_caption("咪咪的使命") # 設定視窗名稱
```



通過關卡

- 在迴圈內呼叫函數

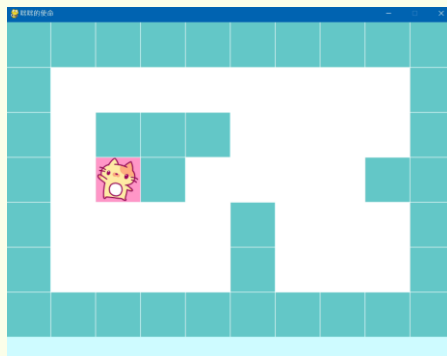
```
83         if event.key == pygame.K_LEFT: # 當按鍵為左方向鍵
84             if maze[maze_y][maze_x-1] == 0:
85                 maze_x = maze_x-1
86
87         if event.key == pygame.K_RIGHT: # 當按鍵為右方向鍵
88             if maze[maze_y][maze_x+1] == 0:
89                 maze_x = maze_x+1
90
91         draw_maze(screen, maze)
92         screen.blit(img_cat, [maze_x*80, maze_y*80])
93         if stage_clear(maze):
94             draw_clear_txt(screen)
95
96         pygame.display.update()
97
98 if __name__ == '__main__': # 呼叫main函數
99     main()
100
```

作業

作業1

- 請參考課堂程式碼與範例影片，完成以下要求

1. 生成一張「咪咪的使命」的遊戲地圖（地圖大小依然是10x7），可以改變貓咪的初始位置
2. 通關時請顯示文字或圖片（可自訂）
3. 請錄製通關該地圖的影片，並上傳連結（務必確認權限）或 .mp4 檔至教學魔法師，不必上傳程式碼



[影片連結](#)