

Tetris 線上遊戲專案報告

進入自己的 cmsimde 近端網站網頁，打開 Source code 進行 Brython 配置並將 Tetris 小遊戲導入個人網頁中

1.導入 Brython 程式庫

Source code ×

```
<!-- 導入 brython 程式庫 -->
<script src="/static/brython.js"></script>
<script src="/static/brython_stdlib.js"></script>
```

2.啟動 Brython

```
<!-- 啟動 Brython -->
<script> // <![CDATA[
window.onload=function(){
  brython({debug:1, pythonpath:['/static','../downloads/py/']});
// ]]></script>
```

3.導入 tetris python 程式

```
<!-- 導入 tetris python 程式 -->
<script type="text/python"> // <![CDATA[
```

5.導入 brython_div

```
<div id="brython_div"></div>
```

配置完成，儲存起來即可在近端網頁上呈現

40823132_wcm2022

[Home](#)[Site Map](#)[Edit All](#)[Edit](#)[Config](#)[Search](#)[image upload](#)[image list](#)[file upload](#)[file list](#)[logout](#)[generate_pages](#)[start_static](#)[8443](#)

About

H1 midterm << [Previous](#) [Next](#) >> online

H1 midterm

Tetris 線上遊戲專案報告

Tetris 線上遊戲專案報告

Tetris

online

H1 midterm << [Previous](#) [Next](#) >> online

遠端網頁

40823132_github.io/wcm2022/content/Tetris%20線上遊戲專案報告.html

Chrome 線上應用... 校曆 TAB e!t info 馬格Pretty Derby 網路速度測試 | Fast... Facebook 巴哈姆特動畫瘋 料理科學 C... HATSUNE MIKU E... 電子是小學生活... War Thunder - Ne... 0609

40823132_wcm2022

Search

[Home](#) [About](#) [H1 midterm](#) [Tetris 線上遊戲專案報告](#) [online](#)

H1 midterm << [Previous](#) [Next](#) >> online

Tetris 線上遊戲專案報告

Tetris

H1 midterm << [Previous](#) [Next](#) >> online

Copyright ©2022 All rights reserved | This template is made with ♥ by Colorlib

5. Tetris 程式說明

程式放在 gist 底下：

https://gist.githubusercontent.com/40823132/92d5c526421624be951ba74cb276dfa5/raw/a5c8147de66e767ba30f5e6e865478cac36a0a97/brython_tetris.py

```
Tetris.py - SciTE
File Edit Search View Tools Options Language Buffers Help
1 Tetris.py
12 # from https://levelup.gitconnected.com/writing-tetris-in-python-2a16bddd5318
13 # 暫時關閉 system proxy 設定後, pip install pygame
14 #import pygame
15 import random # 導入 random 模組
16 # 以下為 Brython 新增
17 # 從 Brython程式庫中的 browser 模組 導入document 簡寫為 doc
18 from browser import document as doc
19 # 從 browser 導入 html 類別, 主要用於建立 CANVAS 標註物件, 並插入頁面中
20 from browser import html
21 # 導入 browser.timer, 用於定時執行特定函數
22 import browser.timer
23
24 # 利用 html 建立一個 CANVAS 標註物件, 與變數 canvas 對應, 並設定畫布的長寬
25 canvas = html.CANVAS(width = 400, height = 500, id="canvas")
26
27 # 將 document 中 id 為 "brython_div" 的標註, 設為與 brython_div 變數對應
28 brython_div = doc["brython_div"]
29 # 將 canvas 標註放入 brython_div 所在位置, 頁面中原本就已經放入 <div id = "brython_div"> 標註
30 brython_div <= canvas
31 # 將canvas 的 2d 繪圖 context 命名為 ctx
32 ctx = canvas.getContext("2d")
33
34 # 設定RGB, 7種方塊的顏色
35 - colors = [
36     (0, 0, 0),
37     (120, 37, 179),
38     (100, 179, 179),
39     (80, 34, 22),
40     (80, 134, 22),
41     (180, 34, 22),
42     (180, 34, 122),
43 ]
44
45 # 新增 Figure 類別, 7種方塊旋轉後的各種狀態, 宣告x、y為o
46 - class Figure:
47     x = 0
48     y = 0
49
```

1 Tetris.py

```
49
50 - figures = [
51     [[1, 5, 9, 13], [4, 5, 6, 7]],
52     [[4, 5, 9, 10], [2, 6, 5, 9]],
53     [[6, 7, 9, 10], [1, 5, 6, 10]],
54     [[1, 2, 5, 9], [0, 4, 5, 6], [1, 5, 9, 8], [4, 5, 6, 10]],
55     [[1, 2, 6, 10], [5, 6, 7, 9], [2, 6, 10, 11], [3, 5, 6, 7]],
56     [[1, 4, 5, 6], [1, 4, 5, 9], [4, 5, 6, 9], [1, 5, 6, 9]],
57     [[1, 2, 5, 6]],
58 ]
59
60 # 定義 __init__ 的功能，隨機選擇一種類型和一種顏色
61 - def __init__(self, x, y):
62     self.x = x
63     self.y = y
64     self.type = random.randint(0, len(self.figures) - 1)
65     self.color = random.randint(1, len(colors) - 1)
66     self.rotation = 0
67
68 # 定義亂數產生的方塊顏色及旋轉
69 - def image(self):
70     return self.figures[self.type][self.rotation]
71
72 # 定義方塊做順時針旋轉
73 - def rotate(self):
74     self.rotation = (self.rotation + 1) % len(self.figures[self.type])
75
76 # 定義方塊做逆時針旋轉
77 - def rotate1(self):
78     self.rotation = (self.rotation - 1) % len(self.figures[self.type])
79
80 # 新增 Tetris 類別，初始化遊戲裡的一些變數
81 - class Tetris:
82     level = 2
83     score = 0
84     state = "start"
85     field = []
86     height = 0
87     width = 0
88     x = 100
89     y = 60
```

1 Tetris.py

```
90     zoom = 20
91     figure = None
92
93     # 定義 __init__ 的功能
94     - def __init__(self, height, width):
95         self.height = height
96         self.width = width
97         self.field = []
98         self.score = 0
99         self.state = "start"
100     - for i in range(height):
101         new_line = []
102     -     for j in range(width):
103         # 起始時每一個都填入 0
104         new_line.append(0)
105         self.field.append(new_line)
106
107     # 新增方塊並放在(3, 0)的位置
108     - def new_figure(self):
109         self.figure = Figure(3, 0)
110
111     # 檢查當前正在下降的方塊是否與在場地上的方塊交錯
112     - def intersects(self):
113         intersection = False
114     -     for i in range(4):
115     -         for j in range(4):
116     -             if i * 4 + j in self.figure.image():
117     -                 # block 到達底部, 左右兩邊界, 或該座標有其他 block
118     -                 if i + self.figure.y > self.height - 1 or \
119     -                     j + self.figure.x > self.width - 1 or \
120     -                     j + self.figure.x < 0 or \
121     -                     self.field[i + self.figure.y][j + self.figure.x] > 0:
122     -                     intersection = True
123         return intersection
124
125     # 如果有完整的水平線則消除掉
126     - def break_lines(self):
127         lines = 0
128     -     for i in range(1, self.height):
129         zeros = 0
130     -         for j in range(self.width):
```

1 Tetris.py

```
130 -         for j in range(self.width):
131 -             if self.field[i][j] == 0:
132 -                 zeros += 1
133 -             if zeros == 0:
134 -                 lines += 1
135 -                 for i1 in range(i, 1, -1):
136 -                     for j in range(self.width):
137 -                         self.field[i1][j] = self.field[i1 - 1][j]
138 - self.score += lines ** 2
139
140 # 定義方塊直接下降到最底下
141 def go_space(self):
142     while not self.intersects():
143         self.figure.y += 1
144     self.figure.y -= 1
145     self.freeze()
146
147 # 定義方塊向下移動
148 def go_down(self):
149     self.figure.y += 1
150     if self.intersects():
151         self.figure.y -= 1
152         self.freeze()
153
154 # 判斷方塊是否與場地上的方塊交錯，如果是，則遊戲結束
155 def freeze(self):
156     for i in range(4):
157         for j in range(4):
158             if i * 4 + j in self.figure.image():
159                 self.field[i + self.figure.y][j + self.figure.x] = self.figure.color
160     self.break_lines()
161     self.new_figure()
162     if self.intersects():
163         self.state = "gameover"
164
165 # 定義方塊向左右移動的函式
166 def go_side(self, dx):
167     old_x = self.figure.x
168     self.figure.x += dx
169     if self.intersects():
170         self.figure.x = old_x
```

1 Tetris.py

```
171
172     # 定義方塊向逆時針旋轉的函式
173 -   def rotate(self):
174       old_rotation = self.figure.rotation
175       self.figure.rotate()
176 -   if self.intersects():
177       self.figure.rotation = old_rotation
178
179     # 定義方塊向順時針旋轉的函式
180 -   def rotate1(self):
181       old_rotation = self.figure.rotation
182       self.figure.rotate1()
183 -   if self.intersects():
184       self.figure.rotation = old_rotation
185
186     # Define some colors
187     # from https://stackoverflow.com/questions/3380726/convert-a-rgb-color-tuple-to-a-six-digit-code
188     # 宣告紅白灰的RGB值
189     BLACK = '#%02x%02x%02x' % (0, 0, 0)
190     WHITE = '#%02x%02x%02x' % (255, 255, 255)
191     GRAY = '#%02x%02x%02x' % (128, 128, 128)
192
193     done = False
194     fps = 60
195     game = Tetris(20, 10)
196     counter = 0
197
198     pressing_down = False
199
200     # 定義按鍵設定
201 -   def key_down(eve):
202       key = eve.keyCode
203       #if event.type == pygame.QUIT:
204       # 32 is pause
205 -   if key == 32:
206       done = True
207       # 88 is x key to clockwise rotate
208 -   if key == 88:
209       game.rotate1()
210       # 90 is z key to anticlockwise rotate
211 -   if key == 90:
```

```
1 Tetris.py
211 - if key == 90:
212     game.rotate()
213     # 67 is c key to drop -1 on the y-axis
214 - if key == 67:
215     game.go_down()
216     # 40 is down key
217 - if key == 40:
218     pressing_down = True
219     # 37 is left key
220 - if key == 37:
221     game.go_side(-1)
222     # 39 is right key
223 - if key == 39:
224     game.go_side(1)
225     # 32 is space key to move block to bottom
226 - if key == 32:
227     game.go_space()
228     # 27 is escape
229     # reset the game
230 - if key == 27:
231     game.__init__(20, 10)
232
233 - def key_up(eve):
234     key = eve.keyCode
235     # 40 is down key
236 - if key == 40:
237     pressing_down = False
238
239     #while not done:
240     # 定義開始遊戲後的設定
241 - def do_game():
242     global counter
243 - if game.figure is None:
244     game.new_figure()
245     counter += 1
246 - if counter > 100000:
247     counter = 0
248 - if counter % (fps // game.level // 2) == 0 or pressing_down:
249 - if game.state == "start":
250     game.go_down()
251 - for i in range(game.height):
```



```
Tetris.py - SciTE
File Edit Search View Tools Options Language Buffers Help
1 Tetris.py
252 -     for j in range(game.width):
253         ctx.fillStyle = WHITE
254         #ctx.scale(game.zoom, game.zoom)
255         ctx.fillRect(game.x + game.zoom * j, game.y + game.zoom * i, game.zoom, game.zoom)
256 -     if game.field[i][j] > 0:
257         ctx.fillStyle = '#%02x%02x%02x' % colors[game.field[i][j]]
258         ctx.fillRect(game.x + game.zoom * j + 1, game.y + game.zoom * i + 1, game.zoom - 2, game.zoom - 1)
259         ctx.lineWidth = 1
260         ctx.strokeStyle = GRAY
261         ctx.beginPath()
262         ctx.rect(game.x + game.zoom * j, game.y + game.zoom * i, game.zoom, game.zoom)
263         ctx.stroke()
264 -     if game.figure is not None:
265         for i in range(4):
266             for j in range(4):
267                 p = i * 4 + j
268                 if p in game.figure.image():
269                     ctx.fillStyle = '#%02x%02x%02x' % colors[game.figure.color]
270                     ctx.fillRect(game.x + game.zoom * (j + game.figure.x) + 1,
271                                 game.y + game.zoom * (i + game.figure.y) + 1,
272                                 game.zoom - 2, game.zoom - 2)
273
274
275 doc.addEventListener("keydown", key_down)
276 doc.addEventListener("keyup", key_up)
277 browser.timer.set_interval(do_game, fps)
```

參考資料

[How to write Tetris in Python. Step by step guide to writing Tetris in... | by Timur Bakibayev | Level Up Coding \(gitconnected.com\)](#)

<https://uupgrade.medium.com/python->

[%E9%82%A3%E4%BA%9B%E5%B9%B4%E6%88%91%E5%80%91%E4%B8%80%E8%B5%B7%E7%8E%A9%E9%81%8E%E7%9A%84%E9%81%8A%E6%88%B2-%E4%BA%8C-%E4%BF%84%E7%BE%85%E6%96%AF%E6%96%B9%E5%A1%8A-2250e08b72a6](#)