

Tetris 線上遊戲專案報告

進去自己的 cms 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 的程式庫

Source code ×

```
<h1>Assignment</h1>
<p>Assignment1</p>
<script src="/static/brython.js"> // 
// ]]&gt;&lt;/script&gt;
&lt;script src="/static/brython_stdlib.js"&gt;&lt;/script&gt;</pre></div><div data-bbox="115 417 397 435" data-label="Text"><p>可將 CDATA 註解拿掉，如下圖</p></div><div data-bbox="115 454 878 567" data-label="Code-Block"><div><b>Source code</b> ×</div><pre>&lt;h1&gt;Assignment&lt;/h1&gt;
&lt;p&gt;Assignment1&lt;/p&gt;
&lt;script src="/static/brython.js"&gt;&lt;/script&gt;
&lt;script src="/static/brython_stdlib.js"&gt;&lt;/script&gt;</pre></div><div data-bbox="115 588 261 608" data-label="Section-Header"><h2>2.啟動 Brython</h2></div><div data-bbox="115 622 878 805" data-label="Code-Block"><div><b>Source code</b> ×</div><pre>&lt;h1&gt;Assignment&lt;/h1&gt;
&lt;p&gt;Assignment1&lt;/p&gt;
&lt;script src="/static/brython.js"&gt;
&lt;script src="/static/brython_stdlib.js"&gt;&lt;/script&gt;
&lt;script&gt; // <![CDATA[
window.onload=function(){
brython({debug:1, pythonpath:['/static/', '../downloads/py/']});
}
// ]]&gt;&lt;/script&gt;</pre></div>
```

一樣可將 CDATA 的註解拿掉，brython 的 debug 和路徑目前用不掉也可先拿掉

Source code



```
<h1>Assignment</h1>
<p>Assignment1</p>
<script src="/static/brython.js">
<script src="/static/brython_stdlib.js"></script>
<script>window.onload=function(){brython();}</script>
```

4. 導入 tetris python 程式(這裡使用 SciTE 文字編輯器較好編輯)

(Untitled) * SciTE

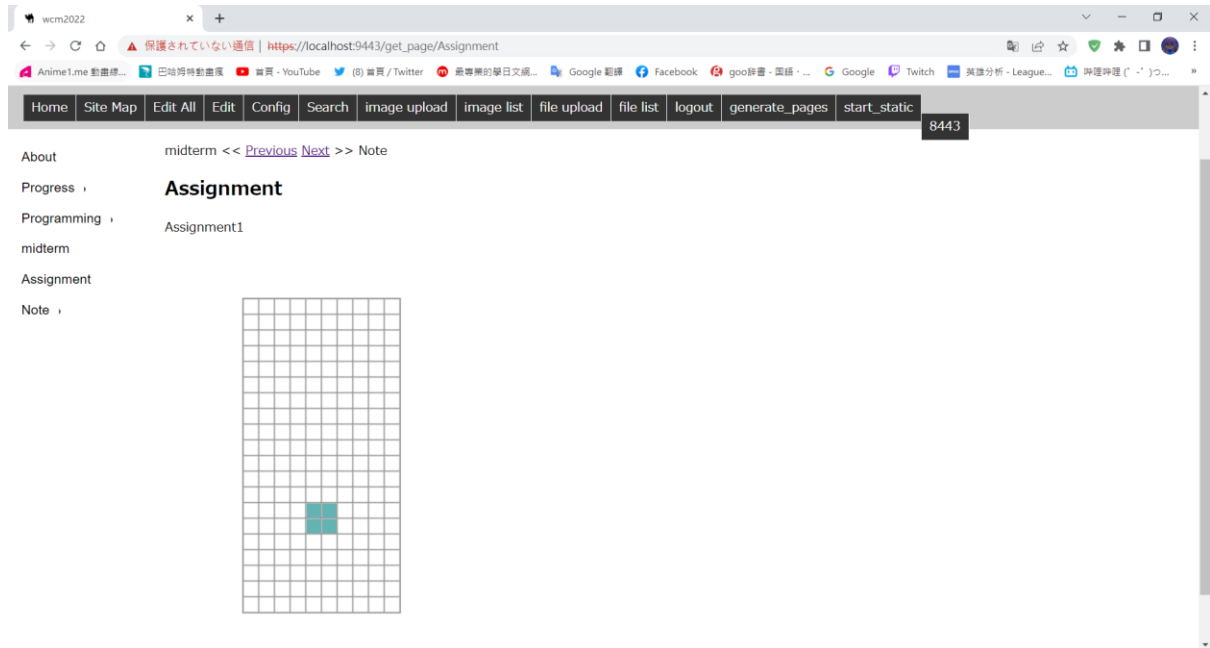
File Edit Search View Tools Options Language Buffers Help

```
1 Untitled *
1  <h1>Assignment</h1>
2  <p>Assignment1</p>
3  <script src="/static/brython.js"></script>
4  <script src="/static/brython_stdlib.js"></script>
5  - <script>// <![CDATA[
6    window.onload=function(){brython();}
7    // ]]></script>
8    <script type="text/python3">// <![CDATA[
9      # from https://levelup.gitconnected.com/writing-tetris-in-python-2a16bddb5318
10     # 暫時關閉 system proxy 設定後, pip install pygame
11     #import pygame
12     import random
13     # 以下為 Brython 新增
14     from browser import document as doc
15     from browser import html
16     import browser.timer
17
18     # 利用 html 建立一個 CANVAS 標註物件, 與變數 canvas 對應
19     canvas = html.CANVAS(width = 400, height = 500, id="canvas")
20     brython_div = doc["brython_div"]
21     brython_div <= canvas
22     ctx = canvas.getContext("2d")
23
24     colors = [
25         (0, 0, 0),
26         (120, 37, 179),
27         (100, 179, 179),
28         (80, 34, 22),
29         (80, 134, 22),
30         (180, 34, 22),
31         (180, 34, 122),
32     ]
33
34
35     class Figure:
36         x = 0
37         y = 0
38
39         figures = [
40             [[1, 5, 9, 13], [4, 5, 6, 7]],
41             [[4, 5, 9, 10], [2, 6, 5, 9]],
42             [[6, 7, 9, 10], [1, 5, 6, 10]],
43             [[1, 2, 5, 9], [0, 4, 5, 6], [1, 5, 9, 8], [4, 5, 6, 10]],
```

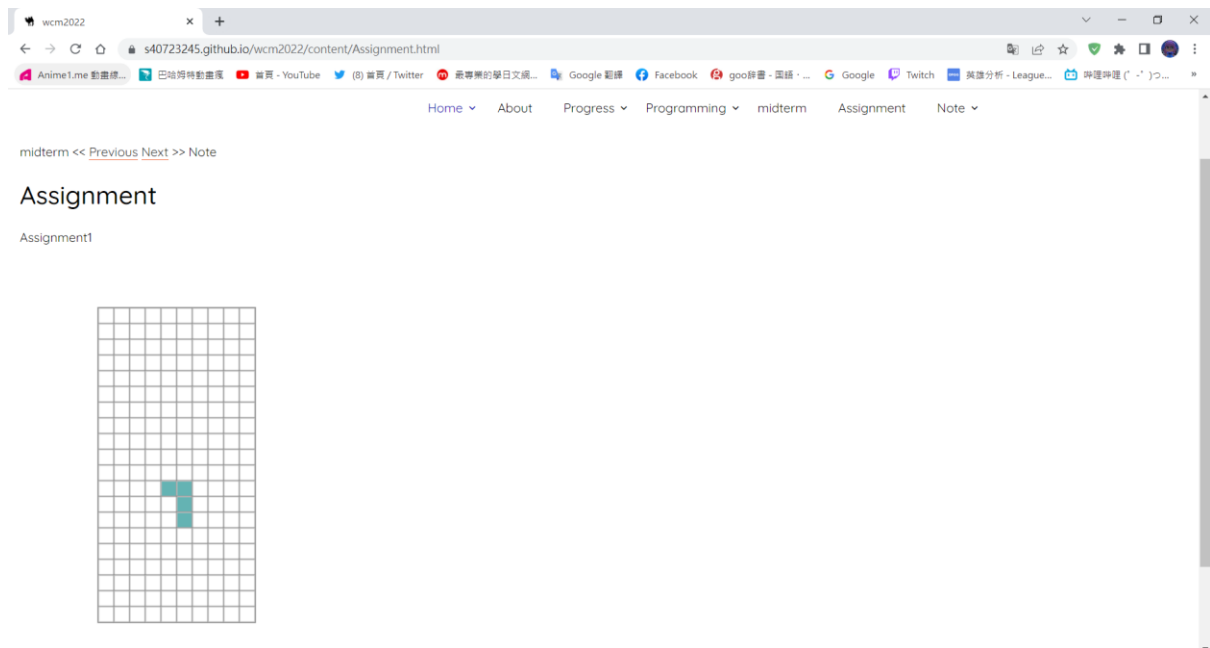
5.導入 brython_div

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

配置完成，儲存起來即可在近端網頁上呈現



遠端網頁



5. Tetris 程式說明

程式放在 gist 底下：

<https://gist.githubusercontent.com/s40723245/9a2abb77c3e7b064d400399b58>

[52503f/raw/61ce009ef1d79175627ea02fb2a874844bd06b28/pygame_to_brython](https://gist.githubusercontent.com/s40723245/52503f/raw/61ce009ef1d79175627ea02fb2a874844bd06b28/pygame_to_brython)

[on_tetris.py](#)

```
Tetris.py - SciTE
File Edit Search View Tools Options Language Buffers Help
1 Tetris.py
1  # from https://levelup.gitconnected.com/writing-tetris-in-python-2a16bddd5318
2  # 暫時關閉 system proxy 設定後, pip install pygame
3  #import pygame
4  import random # 導入 random 模組
5  # 以下為 Brython 新增
6  # 從 Brython程式庫中的 browser 模組 導入document 簡寫為 doc
7  from browser import document as doc
8  # 從 browser 導入 html 類別, 主要用於建立 CANVAS 標註物件, 並插入頁面中
9  from browser import html
10 # 導入 browser.timer, 用於定時執行特定函數
11 import browser.timer
12
13 # 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
14 canvas = html.CANVAS(width = 400, height = 500, id="canvas")
15
16 # 將 document 中 id 為 "brython_div" 的標註, 設為與 brython_div 變數對應
17 brython_div = doc["brython_div"]
18 # 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
19 brython_div <= canvas
20 # 將canvas的 2d 繪圖 context 命名為 ctx
21 ctx = canvas.getContext("2d")
22
23 # 設定RGB, 7種方塊的顏色
24 colors = [
25     (0, 0, 0),
26     (120, 37, 179),
27     (100, 179, 179),
28     (80, 34, 22),
29     (80, 134, 22),
30     (180, 34, 22),
31     (180, 34, 122),
32 ]
33
34 # 新增 Figure 類別, 7種方塊旋轉後的各種狀態, 宣告x、y為o
35 class Figure:
36     x = 0
37     y = 0
38
39     figures = [
40         [[1, 5, 9, 13], [4, 5, 6, 7]],
41         [[4, 5, 9, 10], [2, 6, 5, 9]],
42         [[6, 7, 9, 10], [1, 5, 6, 10]],
43         [[1, 2, 5, 9], [0, 4, 5, 6], [1, 5, 9, 8], [4, 5, 6, 10]],
44         [[1, 2, 6, 10], [5, 6, 7, 9], [2, 6, 10, 11], [3, 5, 6, 7]],
45         [[1, 4, 5, 6], [1, 4, 5, 9], [4, 5, 6, 9], [1, 5, 6, 9]],
46         [[1, 2, 5, 6]],
47     ]
48
```

1 Tetris.py

```
49 # 定義 __init__ 的功能, 隨機選擇一種類型和一種顏色
50 def __init__(self, x, y):
51     self.x = x
52     self.y = y
53     self.type = random.randint(0, len(self.figures) - 1)
54     self.color = random.randint(1, len(colors) - 1)
55     self.rotation = 0
56
57 # 定義亂數產生的方塊顏色及旋轉
58 def image(self):
59     return self.figures[self.type][self.rotation]
60
61 # 定義方塊做順時針旋轉
62 def rotate(self):
63     self.rotation = (self.rotation + 1) % len(self.figures[self.type])
64
65 # 定義方塊做逆時針旋轉
66 def rotate1(self):
67     self.rotation = (self.rotation - 1) % len(self.figures[self.type])
68
69 # 新增 Tetris 類別, 初始化遊戲裡的一些變數
70 class Tetris:
71     level = 2
72     score = 0
73     state = "start"
74     field = []
75     height = 0
76     width = 0
77     x = 100
78     y = 60
79     zoom = 20
80     figure = None
81
82 # 定義 __init__ 的功能
83 def __init__(self, height, width):
84     self.height = height
85     self.width = width
86     self.field = []
87     self.score = 0
88     self.state = "start"
89     for i in range(height):
90         new_line = []
91         for j in range(width):
92             # 起始時每一個都填入 0
93             new_line.append(0)
94         self.field.append(new_line)
95
```

1 Tetris.py

```

96     # 新增方塊並放在(3, 0)的位置
97     def new_figure(self):
98         self.figure = Figure(3, 0)
99
100    # 檢查當前正在下降的方塊是否與在場地上的方塊交錯
101    def intersects(self):
102        intersection = False
103        for i in range(4):
104            for j in range(4):
105                if i * 4 + j in self.figure.image():
106                    # block 到達底部, 左右兩邊界, 或該座標有其他 block
107                    if i + self.figure.y > self.height - 1 or \
108                        j + self.figure.x > self.width - 1 or \
109                        j + self.figure.x < 0 or \
110                        self.field[i + self.figure.y][j + self.figure.x] > 0:
111                        intersection = True
112        return intersection
113
114    # 如果有完整的水平線則消除掉
115    def break_lines(self):
116        lines = 0
117        for i in range(1, self.height):
118            zeros = 0
119            for j in range(self.width):
120                if self.field[i][j] == 0:
121                    zeros += 1
122            if zeros == 0:
123                lines += 1
124                for i1 in range(i, 1, -1):
125                    for j in range(self.width):
126                        self.field[i1][j] = self.field[i1 - 1][j]
127        self.score += lines ** 2
128
129    # 定義方塊直接下降到最底下
130    def go_space(self):
131        while not self.intersects():
132            self.figure.y += 1
133        self.figure.y -= 1
134        self.freeze()
135
136    # 定義方塊向下移動
137    def go_down(self):
138        self.figure.y += 1
139        if self.intersects():
140            self.figure.y -= 1
141            self.freeze()
142

```

1 Tetris.py

```
143     # 判斷方塊是否與場地上的方塊交錯，如果是，則遊戲結束
144     — def freeze(self):
145     —     for i in range(4):
146     —         for j in range(4):
147     —             if i * 4 + j in self.figure.image():
148     —                 self.field[i + self.figure.y][j + self.figure.x] = self.figure.color
149     —                 self.break_lines()
150     —                 self.new_figure()
151     —             if self.intersects():
152     —                 self.state = "gameover"
153
154     # 定義方塊向左右移動的函式
155     — def go_side(self, dx):
156     —     old_x = self.figure.x
157     —     self.figure.x += dx
158     —     if self.intersects():
159     —         self.figure.x = old_x
160
161     # 定義方塊向逆時針旋轉的函式
162     — def rotate(self):
163     —     old_rotation = self.figure.rotation
164     —     self.figure.rotate()
165     —     if self.intersects():
166     —         self.figure.rotation = old_rotation
167
168     # 定義方塊向順時針旋轉的函式
169     — def rotate1(self):
170     —     old_rotation = self.figure.rotation
171     —     self.figure.rotate1()
172     —     if self.intersects():
173     —         self.figure.rotation = old_rotation
174
175     # Define some colors
176     # from https://stackoverflow.com/questions/3380726/converting-a-rgb-color-tu
177     # 宣告紅白灰的RGB值
178     BLACK = '#%02x%02x%02x' % (0, 0, 0)
179     WHITE = '#%02x%02x%02x' % (255, 255, 255)
180     GRAY = '#%02x%02x%02x' % (128, 128, 128)
181
182     done = False
183     fps = 60
184     game = Tetris(20, 10)
185     counter = 0
186
187     pressing_down = False
188
```

1 Tetris.py

```
189     # 定義按鍵設定
190     - def key_down(eve):
191         key = eve.keyCode
192         #if event.type == pygame.QUIT:
193         # 32 is pause
194         - if key == 32:
195             done = True
196         # 88 is x key to clockwise rotate
197         - if key == 88:
198             game.rotate1()
199         # 90 is z key to anticlockwise rotate
200         - if key == 90:
201             game.rotate()
202         # 67 is c key to drop -1 on the y-axis
203         - if key == 67:
204             game.go_down()
205         # 40 is down key
206         - if key == 40:
207             pressing_down = True
208         # 37 is left key
209         - if key == 37:
210             game.go_side(-1)
211         # 39 is right key
212         - if key == 39:
213             game.go_side(1)
214         # 32 is space key to move block to bottom
215         - if key == 32:
216             game.go_space()
217         # 27 is escape
218         # reset the game
219         - if key == 27:
220             game.__init__(20, 10)
221
222     - def key_up(eve):
223         key = eve.keyCode
224         # 40 is down key
225         - if key == 40:
226             pressing_down = False
227
```



```

228 #while not done:
229 # 定義開始遊戲後的設定
230 - def do_game():
231     global counter
232     - if game.figure is None:
233         game.new_figure()
234         counter += 1
235     - if counter > 100000:
236         counter = 0
237     - if counter % (fps // game.level // 2) == 0 or pressing_down:
238         - if game.state == "start":
239             game.go_down()
240         - for i in range(game.height):
241             for j in range(game.width):
242                 ctx.fillStyle = WHITE
243                 #ctx.scale(game.zoom, game.zoom)
244                 ctx.fillRect(game.x + game.zoom * j, game.y + game.zoom * i, game.zoom, game.zoom)
245             - if game.field[i][j] > 0:
246                 ctx.fillStyle = '#%02x%02x%02x' % colors[game.field[i][j]]
247                 ctx.fillRect(game.x + game.zoom * j + 1, game.y + game.zoom * i + 1, game.zoom - 2, game.zoom - 1)
248                 ctx.lineWidth = 1
249                 ctx.strokeStyle = GRAY
250                 ctx.beginPath()
251                 ctx.rect(game.x + game.zoom * j, game.y + game.zoom * i, game.zoom, game.zoom)
252                 ctx.stroke()
253     - if game.figure is not None:
254         for i in range(4):
255             for j in range(4):
256                 p = i * 4 + j
257             - if p in game.figure.image():
258                 ctx.fillStyle = '#%02x%02x%02x' % colors[game.figure.color]
259             - ctx.fillRect(game.x + game.zoom * (j + game.figure.x) + 1,
260                           game.y + game.zoom * (i + game.figure.y) + 1,
261                           game.zoom - 2, game.zoom - 2)
262
263
264 doc.addEventListener("keydown", key_down)
265 doc.addEventListener("keyup", key_up)
266 browser.timer.set_interval(do_game, fps)

```

參考資料

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

<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>

<https://levelup.gitconnected.com/writing-tetris-in-python-2a16bddb5318>