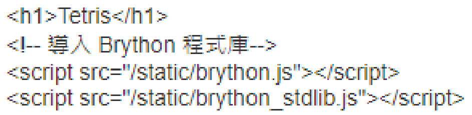
Tetris 線上遊戲加入計分系統報告

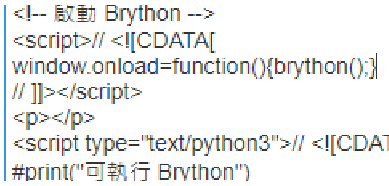
使用 cms 開啟近端，在 source code 中編輯 Brython 配置，把 Tetris 遊戲導入在個人網頁的任務中。

(1) 導入 Brython 程式庫。



```
<h1>Tetris</h1>
<!-- 導入 Brython 程式庫-->
<script src="/static/brython.js"></script>
<script src="/static/brython_stdlib.js"></script>
```

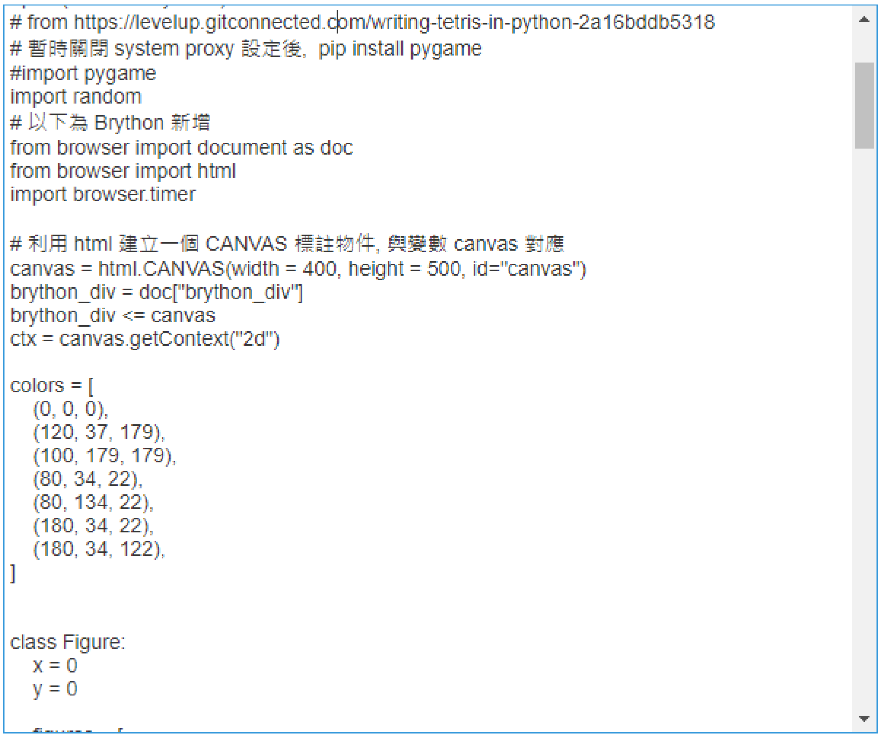
(2) 啟動 Brython。



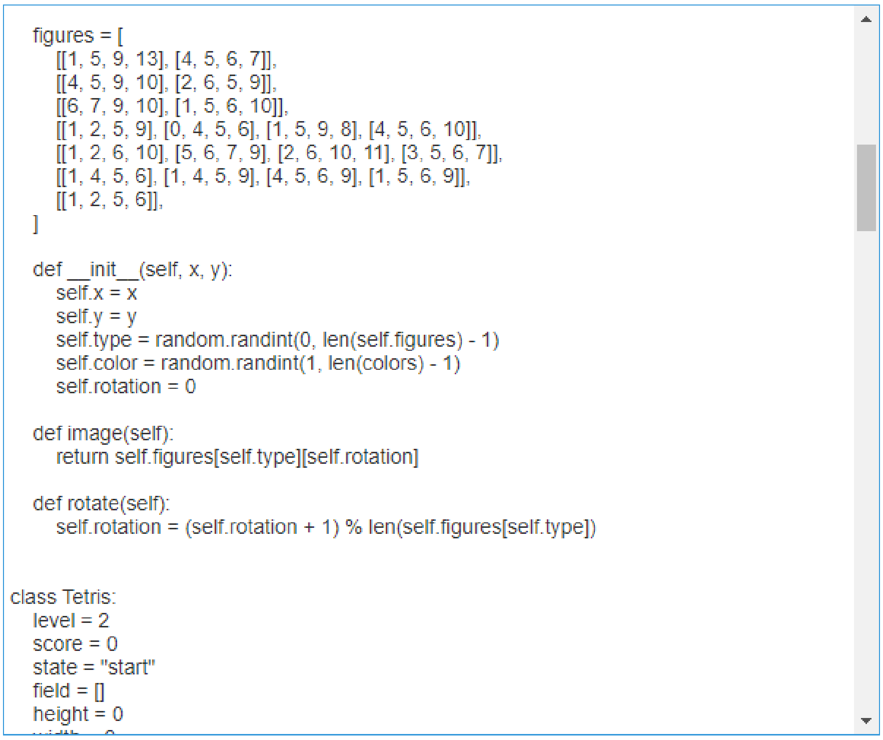
```
<!-- 啟動 Brython -->
<script>// <![CDATA[
window.onload=function(){brython();}
// ]]></script>
<p></p>
<script type="text/python3">// <![CDATA[
#print("可執行 Brython")
```

(1)(2)可以將註解或是路徑先刪掉。

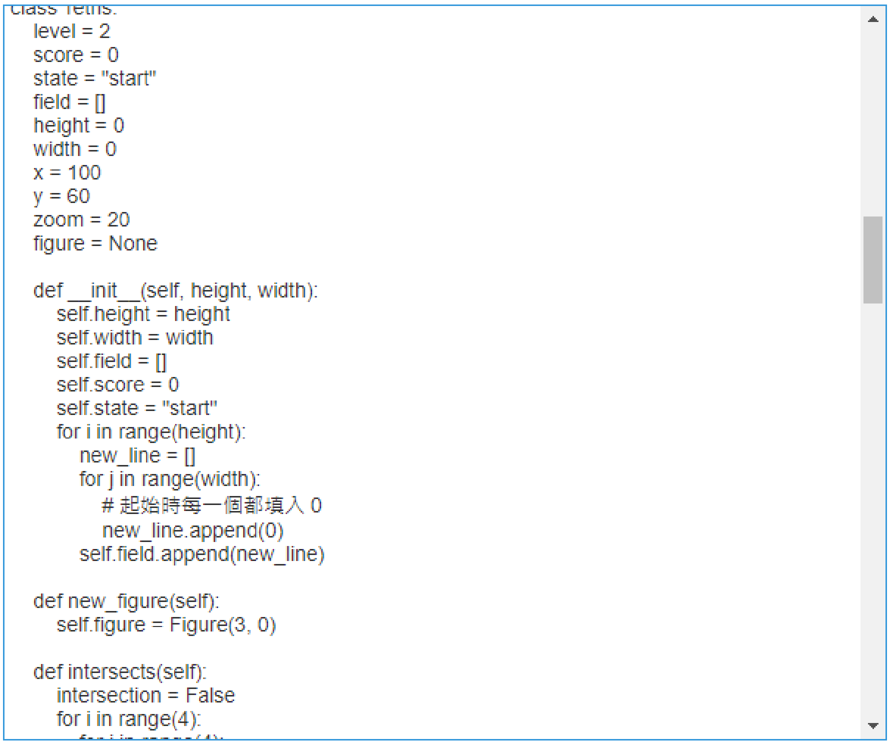
(3)將 Tetris 的程式導入到 source code 中。



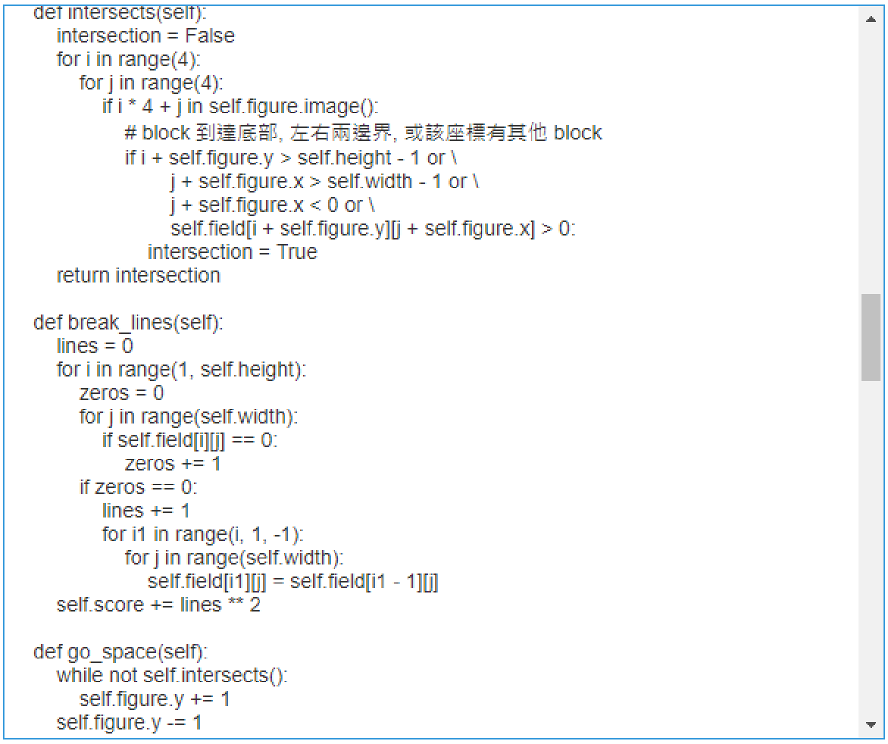
```
# from https://levelup.gitconnected.com/writing-tetris-in-python-2a16bddb5318
# 暫時關閉 system proxy 設定後, pip install pygame
#import pygame
import random
# 以下為 Brython 新增
from browser import document as doc
from browser import html
import browser.timer

# 利用 html 建立一個 CANVAS 標註物件, 與變數 canvas 對應
canvas = html.CANVAS(width = 400, height = 500, id="canvas")
brython_div = doc["brython_div"]
brython_div <= canvas
ctx = canvas.getContext("2d")

colors = [
    (0, 0, 0),
    (120, 37, 179),
    (100, 179, 179),
    (80, 34, 22),
    (80, 134, 22),
    (180, 34, 22),
    (180, 34, 122),
]


class Figure:
    x = 0
    y = 0
```



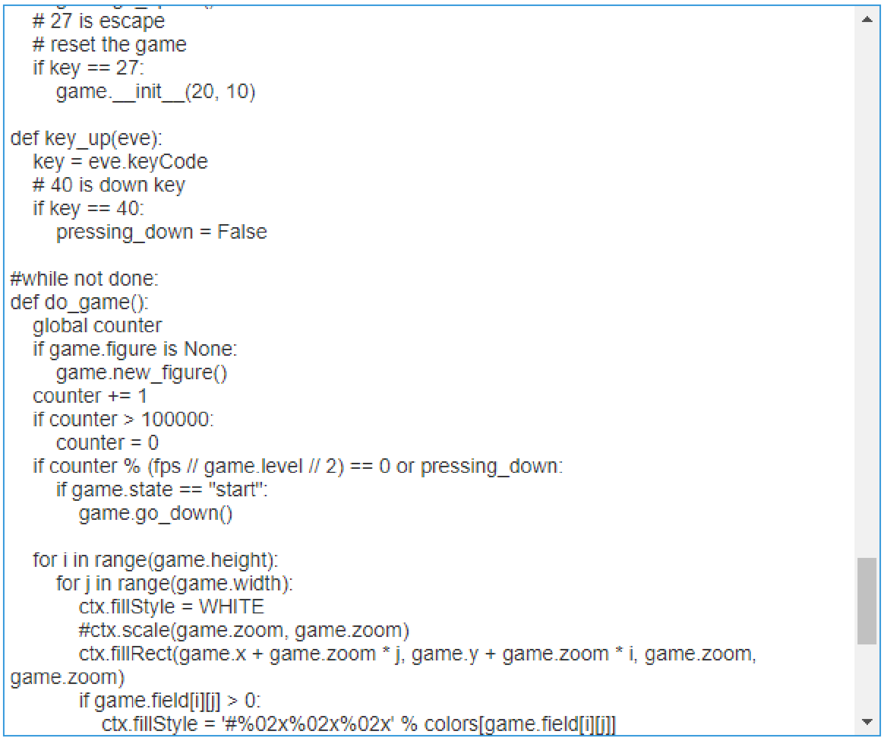
```
    figures = [
        [[1, 5, 9, 13], [4, 5, 6, 7]],
        [[4, 5, 9, 10], [2, 6, 5, 9]],
        [[6, 7, 9, 10], [1, 5, 6, 10]],
        [[1, 2, 5, 9], [0, 4, 5, 6], [1, 5, 9, 8], [4, 5, 6, 10]],
        [[1, 2, 6, 10], [5, 6, 7, 9], [2, 6, 10, 11], [3, 5, 6, 7]],
        [[1, 4, 5, 6], [1, 4, 5, 9], [4, 5, 6, 9], [1, 5, 6, 9]],
        [[1, 2, 5, 6]],
    ]

    def __init__(self, x, y):
        self.x = x
        self.y = y
        self.type = random.randint(0, len(self.figures) - 1)
        self.color = random.randint(1, len(colors) - 1)
        self.rotation = 0

    def image(self):
        return self.figures[self.type][self.rotation]

    def rotate(self):
        self.rotation = (self.rotation + 1) % len(self.figures[self.type])


class Tetris:
    level = 2
    score = 0
    state = "start"
    field = []
    height = 0
    width = 0
```



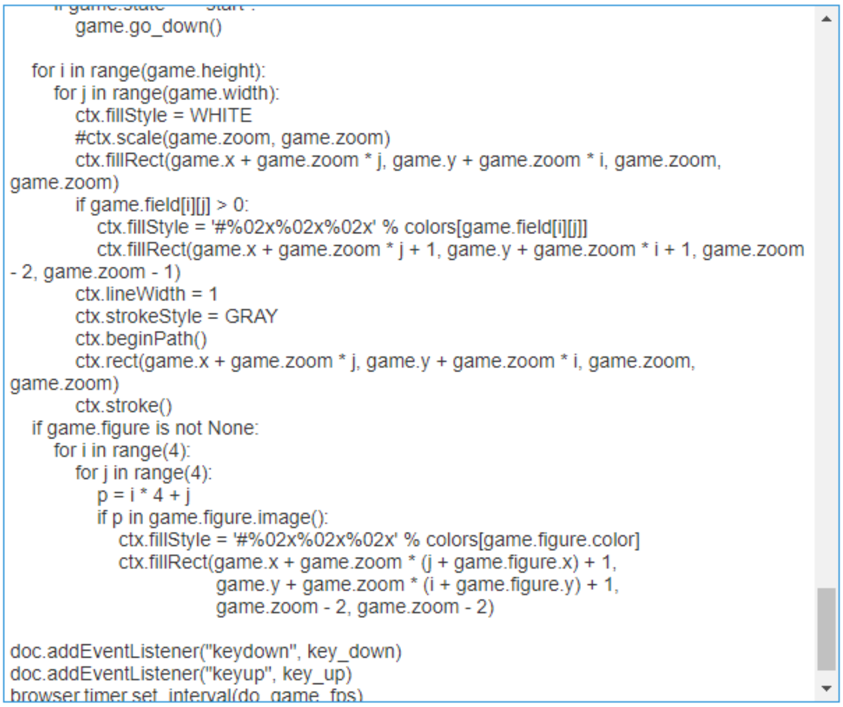
```
class Tetris:
    level = 2
    score = 0
    state = "start"
    field = []
    height = 0
    width = 0
    x = 100
    y = 60
    zoom = 20
    figure = None

    def __init__(self, height, width):
        self.height = height
        self.width = width
        self.field = []
        self.score = 0
        self.state = "start"
        for i in range(height):
            new_line = []
            for j in range(width):
                # 起始時每一個都填入 0
                new_line.append(0)
            self.field.append(new_line)

    def new_figure(self):
        self.figure = Figure(3, 0)

    def intersects(self):
        intersection = False
        for i in range(4):
            for j in range(4):
```



```
    def intersects(self):
        intersection = False
        for i in range(4):
            for j in range(4):
                if i * 4 + j in self.figure.image():
                    # block 到達底部, 左右兩邊界, 或該座標有其他 block
                    if i + self.figure.y > self.height - 1 or \
                            j + self.figure.x > self.width - 1 or \
                            j + self.figure.x < 0 or \
                            self.field[i + self.figure.y][j + self.figure.x] > 0:
                        intersection = True
        return intersection

    def break_lines(self):
        lines = 0
        for i in range(1, self.height):
            zeros = 0
            for j in range(self.width):
                if self.field[i][j] == 0:
                    zeros += 1
            if zeros == 0:
                lines += 1
                for i1 in range(i, 1, -1):
                    for j in range(self.width):
                        self.field[i1][j] = self.field[i1 - 1][j]
        self.score += lines ** 2

    def go_space(self):
        while not self.intersects():
            self.figure.y += 1
        self.figure.y -= 1
```

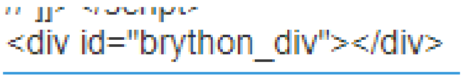


```
    # 27 is escape
    # reset the game
    if key == 27:
        game.__init__(20, 10)

def key_up(eve):
    key = eve.keyCode
    # 40 is down key
    if key == 40:
        pressing_down = False

#while not done:
def do_game():
    global counter
    if game.figure is None:
        game.new_figure()
    counter += 1
    if counter > 100000:
        counter = 0
    if counter % (fps // game.level // 2) == 0 or pressing_down:
        if game.state == "start":
            game.go_down()

    for i in range(game.height):
        for j in range(game.width):
            ctx.fillStyle = WHITE
            #ctx.scale(game.zoom, game.zoom)
            ctx.fillRect(game.x + game.zoom * j, game.y + game.zoom * i, game.zoom,
game.zoom)
            if game.field[i][j] > 0:
                ctx.fillStyle = '#%02x%02x%02x' % colors[game.field[i][j]]
```



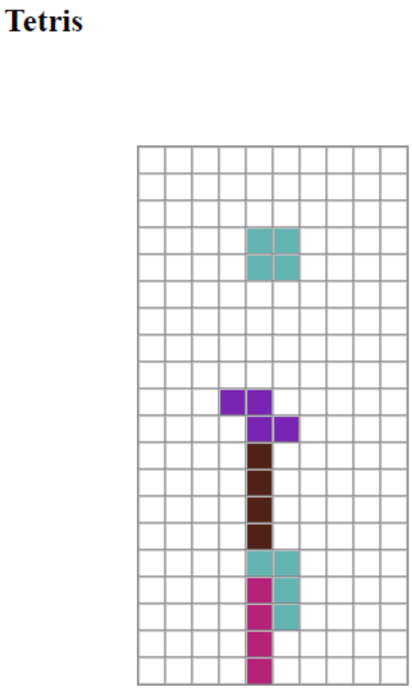
```
        if game.state == "start":
            game.go_down()

    for i in range(game.height):
        for j in range(game.width):
            ctx.fillStyle = WHITE
            #ctx.scale(game.zoom, game.zoom)
            ctx.fillRect(game.x + game.zoom * j, game.y + game.zoom * i, game.zoom,
game.zoom)
            if game.field[i][j] > 0:
                ctx.fillStyle = '#%02x%02x%02x' % colors[game.field[i][j]]
                ctx.fillRect(game.x + game.zoom * j + 1, game.y + game.zoom * i + 1, game.zoom
- 2, game.zoom - 1)
            ctx.lineWidth = 1
            ctx.strokeStyle = GRAY
            ctx.beginPath()
            ctx.rect(game.x + game.zoom * j, game.y + game.zoom * i, game.zoom,
game.zoom)
            ctx.stroke()
    if game.figure is not None:
        for i in range(4):
            for j in range(4):
                p = i * 4 + j
                if p in game.figure.image():
                    ctx.fillStyle = '#%02x%02x%02x' % colors[game.figure.color]
                    ctx.fillRect(game.x + game.zoom * (j + game.figure.x) + 1,
                                 game.y + game.zoom * (i + game.figure.y) + 1,
                                 game.zoom - 2, game.zoom - 2)

doc.addEventListener("keydown", key_down)
doc.addEventListener("keyup", key_up)
browser.timer.set_interval(do_game, fps)
```

(4)再導入**"brython_div"**。



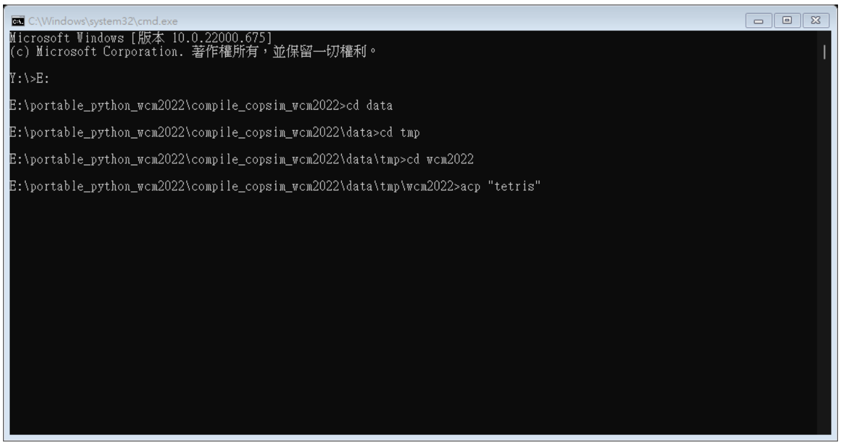
```
// ]]> </script>
<div id="brython_div"></div>
```

(5)完成編輯後 save，點擊 viewpage 即可呈現。



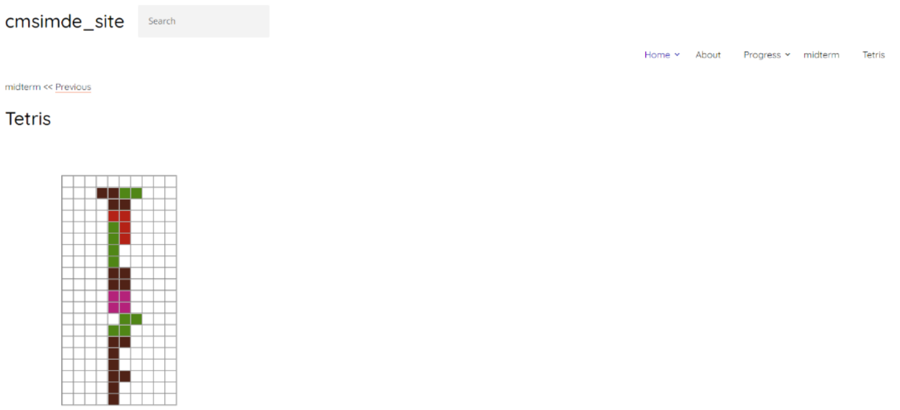
Tetris

(6)使用 acp 將資料推送上去。



```
Microsoft Windows [版本 10.0.22000.675]
(c) Microsoft Corporation. 著作權所有，並保留一切權利。

Y:\>E:

E:\portable_python_wcm2022\compile_copsim_wcm2022>cd data

E:\portable_python_wcm2022\compile_copsim_wcm2022\data>cd tmp

E:\portable_python_wcm2022\compile_copsim_wcm2022\data\tmp>cd wcm2022

E:\portable_python_wcm2022\compile_copsim_wcm2022\data\tmp\wcm2022>acp "tetris"
```

(7)在遠端頁面。



midterm << Previous

Tetris

Gist 連結:

https://gist.githubusercontent.com/s40723226/8b653e0ade658a75faf5439c5f8206ec/raw/32f25728965460b58b49a16d1ad54e98c1ffef0c/Tetris

個人網頁:

https://s40723226.github.io/wcm2022/content/index.html

個人倉儲:

https://github.com/s40723226/wcm2022