俄羅斯方塊

程式範例:

# 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

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):

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

self.freeze()

def go\_down(self):

self.figure.y += 1

if self.intersects():

self.figure.y -= 1

self.freeze()

def freeze(self):

for i in range(4):

for j in range(4):

if i \* 4 + j in self.figure.image():

self.field[i + self.figure.y][j + self.figure.x] = self.figure.color

self.break\_lines()

self.new\_figure()

if self.intersects():

self.state = "gameover"

def go\_side(self, dx):

old\_x = self.figure.x

self.figure.x += dx

if self.intersects():

self.figure.x = old\_x

def rotate(self):

old\_rotation = self.figure.rotation

self.figure.rotate()

if self.intersects():

self.figure.rotation = old\_rotation

# Define some colors

# from https://stackoverflow.com/questions/3380726/converting-a-rgb-color-tuple-to-a-six-digit-code

BLACK = '#%02x%02x%02x' % (0, 0, 0)

WHITE = '#%02x%02x%02x' % (255, 255, 255)

GRAY = '#%02x%02x%02x' % (128, 128, 128)

done = False

fps = 25

game = Tetris(20, 10)

counter = 0

pressing\_down = False

def key\_down(eve):

key = eve.keyCode

#if event.type == pygame.QUIT:

# 32 is pause

if key == 32:

done = True

# 82 is r key to rotate

if key == 82:

game.rotate()

# 40 is down key

if key == 40:

pressing\_down = True

# 37 is left key

if key == 37:

game.go\_side(-1)

# 39 is right key

if key == 39:

game.go\_side(1)

# 68 is d key to move block to bottom

if key == 68:

game.go\_space()

# 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]]

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)

結果:

一張含有 文字, 拉門 的圖片

自動產生的描述