新北市立板橋高級中學

學生自主學習計畫

**研究成果報告書**

Python程式設計實作—俄羅斯方塊

研 究 生：三年十三班 王承琳

中 華 民 國 一 百 一 十 一 年 三 月

**摘要**

本研究使用程式設計製作俄羅斯方塊遊戲，透過研究Python不同函式的指令跟運算來訓練邏輯思考及對於Python語言的運用能力，並學習如何找出問題、分析問題進而解決問題、完成專案。

**研究心得**

　　研究初期時，我對於研究步驟沒有很清楚的規劃，導致我像隻無頭蒼蠅般，寫出來的程式雜亂無章，邏輯錯亂，所以我花了一部份時間進行步驟規劃以及程式的重整，將我所想要的遊戲效果或功能列點並分為不同系統，在根據此去依序進行。當遊戲的基礎功能都完成後，我會請同學試玩來測試遊戲是否有錯誤並進行修改，而在Debug時，我發現備註的重要性，若沒有在打程式時善用備註紀錄思考方向，在後續修改時會花很多時間來理解當初的想法以及思路，因此我在之後的程式碼都有打備註，並將之前的備註補上，且我也發現寫程式真的需要非常多的耐心以及細心，因為每個系統都是環環相扣的，一步錯，步步錯，所以在修改時須耐心地不斷檢查各個部分錯誤的地方，有時候粗心少打一個空格都會造成程式的錯誤。在後續重新整理程式碼時，我注意到我許多太過於繁複且沒必要的步驟，雖然最後有達到想要的效果，但卻是轉了一個彎才達成，也就是說我邏輯不夠直線，常造成程式太過冗長，就像寫作文中的贅字一樣，邏輯夠不夠清楚可以間接影響專案的完成速度以及精簡度，所以我將程式碼全部重整過，盡量修改成最精簡的樣子，刪去不必要的變數或迴圈等，並重新順一次思路，建立邏輯思考能力的基礎。

**目錄**

摘要 1

研究心得 1

[目錄](#_Toc22234775) 2

[一、研究背景與動機](#_Toc22234776) 3

[1.1 研究背景與動機](#_Toc22234777) 3

[1.2 研究期許](#_Toc22234777) 3

[二、緒論](#_Toc22234776) 4

[2.1 俄羅斯方塊起源](#_Toc22234777) 4

[2.2 俄羅斯方塊基本圖形與規則](#_Toc22234777) 4

[三、研究方法](#_Toc22234776) 5

[3.1 研究流程](#_Toc22234777) 5

[3.2 研究工具](#_Toc22234777) 5

[四、程式設計實作](#_Toc22234776) 6

[4.1 初始化](#_Toc22234777) 6

[4.2 座標系統](#_Toc22234777) 8

[4.3 移動系統](#_Toc22234777) 11

[4.4 旋轉系統](#_Toc22234777) 25

[4.5 消除系統](#_Toc22234777) 26

[4.6 畫面顯示](#_Toc22234777) 30

[4.7 音效](#_Toc22234777) 34

[4.8 總體設計概念](#_Toc22234777) 35

[五、結論與建議](#_Toc22234776) 36

5[.1 結論](#_Toc22234777) 36

[5.2 建議](#_Toc22234777) 37

[資料來源](#_Toc22234776) 38

第一章 研究背景與動機

1. 研究背景與動機

在我國中小時，Facebook裡有一款遊戲叫做Tetris Battle，當時風靡全校，幾乎只要有使用Facebook的同學都有在玩這款遊戲，每天放學後的娛樂之一就是與好友進行對戰。而在我國三時，因為即將到會考的日子了，所以我沒有再玩過這款遊戲，直到升上高中後突然憶起這款經典遊戲時，它已經結束營運了。

我想要嘗試著將我腦海中的那個經典重現，故我選擇了此主題，使用程式語言Python製作出一款俄羅斯方塊遊戲，除了可以回味小時候的感動，更可以精進程式設計的能力。

1. 研究期許

我希望我能夠透過此研究訓練邏輯思考能力及如何針對問題做出改善，利用Python的功能來達到我所想要的效果。此外，就獨立思考而言，我希望我能夠透過設計這款遊戲訓練解決問題的能力，就學術研究而言，我希望我的研究能夠幫助到其他對程式設計同樣有興趣的學生們，就專業領域而言，我希望我能夠精進程式設計的運用能力，將課堂所學化為己用，比起課堂上練習的題目，自己獨立完成一款遊戲的難度與挑戰性都高了許多，能進步的幅度也大了許多。

第二章 緒論

1. 俄羅斯方塊起源

此遊戲是由一位名為阿列克謝·帕基特諾夫的蘇聯人所設計，約於1984年首次發佈。此遊戲的名稱「Tetris」是由希臘語「四」的前綴「tetra-」（因所有落下方塊皆由四塊組成）和「tennis」（作者最喜歡的運動）拼接而成，而華語地區則因為作者是俄羅斯人所以將其稱為「俄羅斯方塊」。

1. 俄羅斯方塊基本圖形與規則

一、基本圖形：最基本的俄羅斯方塊圖形由四個正方形所組成，共有七種組合。

一張含有 文字, 時鐘, 手錶, 美工圖案 的圖片

自動產生的描述一張含有 文字, 時鐘, 手錶, 美工圖案 的圖片

自動產生的描述一張含有 文字, 時鐘, 手錶, 美工圖案 的圖片

自動產生的描述

T

O

S

L

一張含有 文字, 時鐘, 手錶, 美工圖案 的圖片

自動產生的描述

一張含有 文字, 時鐘, 手錶, 美工圖案 的圖片

自動產生的描述一張含有 文字, 時鐘, 手錶, 美工圖案 的圖片

自動產生的描述一張含有 文字, 時鐘, 手錶, 美工圖案 的圖片

自動產生的描述

I

Z

J

圖2.1、俄羅斯方塊的七種圖形

(資料來源：維基百科－俄羅斯方塊)

二、規則

(一) 玩家操作

圖2.2、玩家操作的三種類型

(二) 得分方式

當區域中某一橫列格子全部被方塊填滿時，則玩家得分並**消除該層方塊。**

當方塊堆疊到遊戲區域頂端而無法繼續消除層數時，則**判定遊戲結束。**

分數若越高下落速度就越快，且每消除一層方塊的得分也越多。

第三章 研究方法

1. 研究流程

研究者先訂定所需完成的功能，並分為5個部分來進行。

圖3.1、研究流程圖

(資料來源：研究者自行繪製)

1. 研究工具

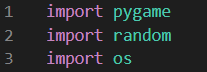
研究工具為Visual Studio Code，使用的程式語言為Python 3.9.6及Python內鍵模組Pygame 2.1.2，使用Pygame模組前需先引入（圖3.2），圖中random及os為後續所需模組。

圖3.2、引入模組

第四章 程式設計實作

1. 初始化
   1. Pygame初始化及視窗建立

起初先進行Pygame初始化才能進行後續動作。

一張含有 文字 的圖片

自動產生的描述

圖4.1、遊戲初始化及基本設定

* 1. 遊戲迴圈

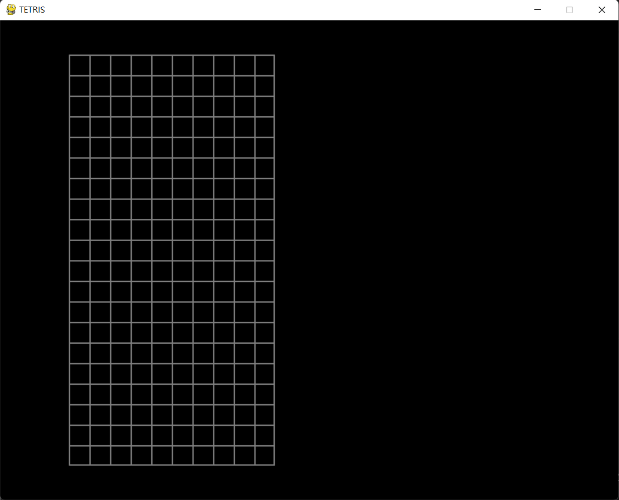
遊戲要一直持續進行，所以建立一個while迴圈，作為遊戲持續運行迴圈（遊戲主迴圈）。

一張含有 文字 的圖片

自動產生的描述

圖4.2、遊戲主迴圈

|  |
| --- |
| **程式說明**  [256-258]若點擊關閉視窗，則退出迴圈。  [261-269]將視窗填滿黑色，繪製遊戲區域並更新畫面顯示（圖4.3）。 |



**每格為30\*30像素，組成10\*20格的長方形遊戲區域。**

圖4.3、遊戲畫面繪製範例圖

|  |
| --- |
| **設計說明**  現今大部分俄羅斯方塊的遊戲區域皆設定為長方形，在繪製遊戲畫面時，我有考慮過使用其他圖形，例如正方形抑或是星型等特殊圖形，但顧及到以下幾點，最後我還是選了長方形來做為遊戲區域。   1. 須預留足夠的高度使方塊不要太快落地 2. 若左右太寬，玩家需要花較久的時間才能達到消除條件 3. 若使用特殊圖形作為遊戲區域，將無法用方塊完全填滿 |

1. 訂定座標系統

一、定位方式

為了達成七種不同圖形皆以同種方式定位，透過觀察各圖形的共通點為起點，俄羅斯方塊各圖形皆由四個小正方形組成，由於當中旋轉後掃過佔最大面積的為I型方塊（圖4.4），故以此為標準，分為16個格子，標為0-15。

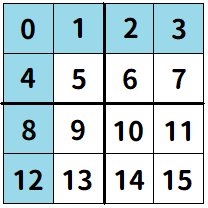
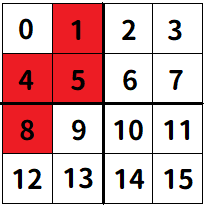
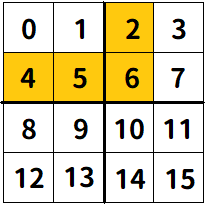


圖4.6、L型方塊編號圖

圖4.5、Z型方塊編號圖

圖4.4、I型方塊編號圖

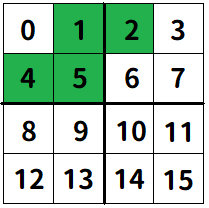
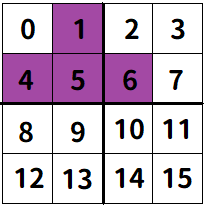
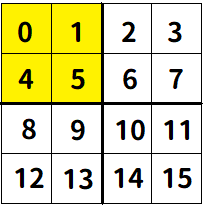


圖4.9、T型方塊編號圖

圖4.8、O型方塊編號圖

圖4.7、S型方塊編號圖

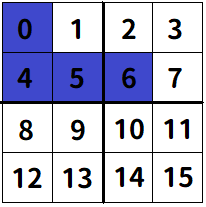


圖4.10、J型方塊編號圖

(資料來源：研究者自行繪製)

二、訂定座標

依此標準我們可以將這7種圖形及其旋轉後的圖形定義為4個編號。

一張含有 文字 的圖片

自動產生的描述

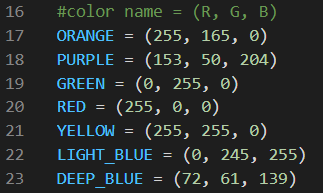


圖4.11、方塊編號序列

|  |
| --- |
| **程式說明**  [56]建立新序列命名為all\_objects並將七種方塊的序列存入。  →方便後續隨機選擇初始方塊及旋轉方塊（各方塊序列按照順時針旋轉的順序排列）。  [57] 建立新序列命名為all\_colors並將七種方塊的對應顏色存入。 |

定義完方塊編號後，再將各編號的初始座標存入字典中。

一張含有 文字 的圖片

自動產生的描述

圖4.12、方塊編號的初始座標字典

|  |
| --- |
| **程式說明**  [63]建立類別Position存放座標系統的屬性及函式。  [64-74]建立初始函數並存入各編號方塊出現的初始座標（all\_spaces）及其在4\*4格子中的[列, 行]（mn\_dic）（圖4.13）→方便後續繪製圖形  [228]將變數position加入Position類別 |

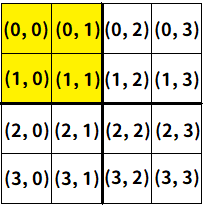


圖4.14、[列, 行]標示圖

1. 移動系統
   1. 直線落下

建立一個Move類別存放所有移動系統的屬性及函式，可讓程式碼更清楚明瞭。







圖4.15、重置遊戲初始值函式

|  |
| --- |
| **程式說明**  [77-78]建立init函式存放遊戲初始值（後續會再放入其他數值）。  [230-231]將變數move加入Move類別並執行init函式重置遊戲初始值。 |

每個初始方塊初始值（座標、顏色等）不同，故建立函式來進行數值重設。



一張含有 文字 的圖片

自動產生的描述

圖4.16、重置方塊初始值函式

|  |
| --- |
| **程式說明**  [126-135]建立draw\_init函數存放每個初始方塊的初始值  [128]隨機選擇圖形種類　[129]隨機選擇圖形旋轉態。  [130]獲取圖形對應的顏色　[132]定位原點座標[列, 行](遊戲區域的格數)。  [134]方塊座標y值增加的次數　[135]方塊座標y值增加的總值。 |

建立完重製方塊初始值的函式後，跟重置遊戲初始值的函式一起執行。





圖4.17、重置遊戲、方塊初始值

|  |
| --- |
| **程式說明**  [82]將draw\_init函式加入init函式中，則當遊戲重新開始後會重置初始方塊的數值。 |

建立完初始值後，將方塊繪製在畫面上。



一張含有 文字 的圖片

自動產生的描述

圖4.18、繪製方塊函式

|  |
| --- |
| **程式說明**  [103-113]建立draw函式繪製方塊。  [104-106]每運行一次此函式，方塊y值增加次數加1，當次數乘初始速度為30的倍數時，則方塊y值增加總值加上其值且定位原點的列數加1。  [107-112]透過迴圈進行方塊中四個小正方形（之後將其稱為「小方塊」）的繪製（[110]若超出遊戲邊界上方則不進行繪製）。 |

|  |
| --- |
| **設計說明**  因小方塊初始座標在迴圈內皆須重置，則y座標會回到初始位置，為了實現讓每個方塊一起下落（即y座標持續增加），故我將方塊y座標增加總值獨立為一個迴圈外的變數，則只要在方塊y座標重置後再加上增加總值，即可實現目的。且因此函式會在遊戲主迴圈內進行，而主迴圈有限制的運行效率（FPS），為了實現方塊下落時以一格為單位，所以方塊ｙ座標應一次增加30像素，但若直接設定每執行一次draw函式就將y座標增加30像素，下落速度會太快，因此我建立speed\_n變數存放方塊座標y值增加的次數，就可以使方塊以一格為單位下落且不會造成速度過快。 |

建立完繪製圖形函式後，在遊戲主迴圈內進行方塊繪製。



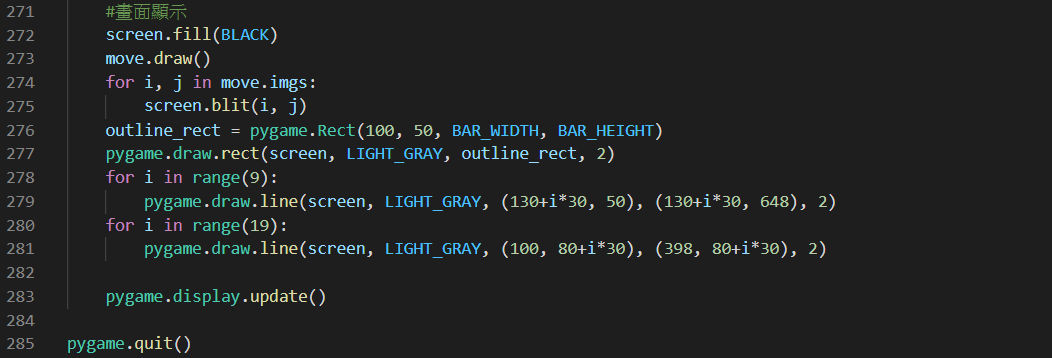


圖4.19、進行方塊繪製

|  |
| --- |
| **程式說明**  [273]在遊戲主迴圈內執行draw函式。 |

* 1. 左右移動

左右移動改變的是x座標，因此要建立方塊x座標增加數值的變數。



一張含有 文字 的圖片

自動產生的描述

圖4.20、方塊x座標增加數值的變數

|  |
| --- |
| **程式說明**  [142]在draw\_init函式中建立SPEEDx變數存放方塊座標x值增加數值的初始值。 |

建立一函式進行左右移動（改變方塊x座標）。





圖4.21、左右移動函式

|  |
| --- |
| **程式說明**  [160-166]建立move函式，輸入移動方向（左、右）。  [161-166]若方向輸入”R”，則圖形右移一格；反的，若方向輸入”L”， 則圖形左移一格。 |

若左右移動或是後續的旋轉方塊沒有限制邊界，可能會造成方塊超出遊戲區域的情況，故建立一函式來檢查方塊是否超出遊戲區域並將其修正到正確位置。



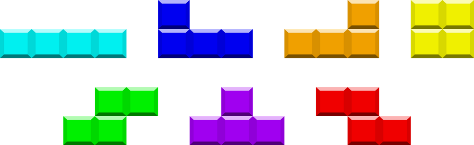
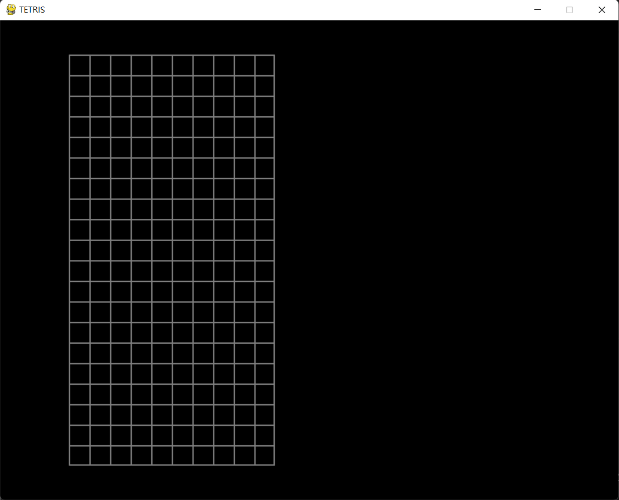
一張含有 文字 的圖片

自動產生的描述

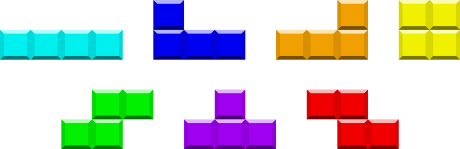
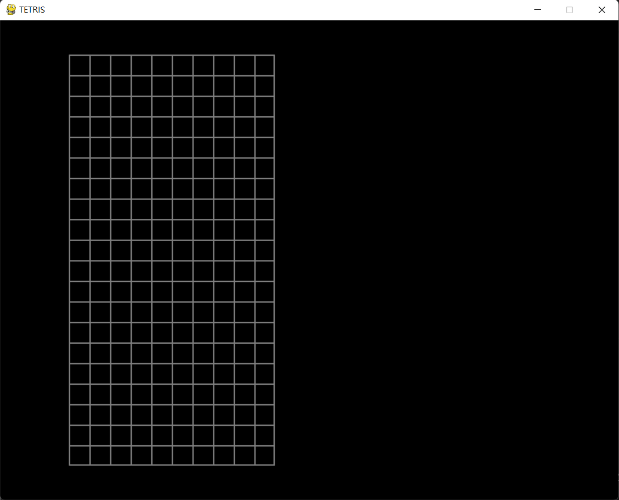
圖4.22、檢查方塊是否超出遊戲區域

|  |
| --- |
| **程式說明**  [130-141] 為了避免圖形左右移動或旋轉後超出遊戲邊界，故建立region\_judge函式判定是否超過遊戲邊界。  [131-133]將圖形中方塊目前所在行數加入序列，最大值為圖形最右邊的行數，最小值為圖形最左邊的行數。  [135-137]若圖形最右邊的行數大於9，即超出右邊界線，則圖形左移一格。  [139-141]若圖形最左邊的行數小於0，即超出左邊界線，則圖形右移一格。 |

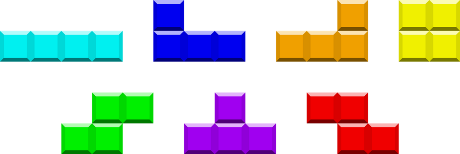
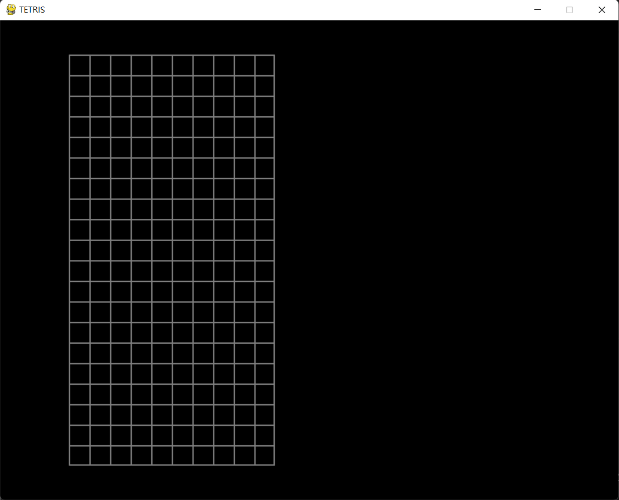
旋轉後



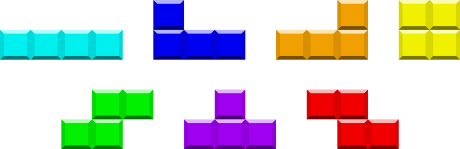
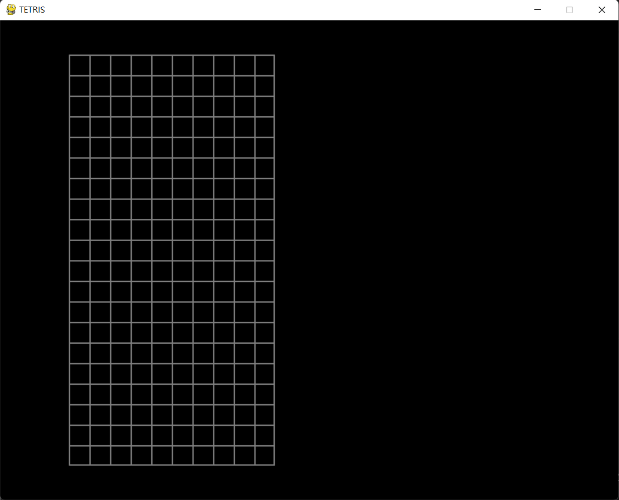
O



O



O



O

正確

錯誤

　圖4.23、方塊旋轉後正確及錯誤位置示意圖

資料來源：研究者自行繪製

　(資料來源：研究者自行繪製)

按下左右鍵時，執行左右移動函式。



一張含有 文字 的圖片

自動產生的描述

圖4.24、檢查是否按下左右鍵

|  |
| --- |
| **程式說明**  [274-279]在遊戲主迴圈內加入判斷式，若按下左、右鍵則執行move函式。 |

在繪製方塊之前先檢查方塊是否超出遊戲區域並修正，然後改變方塊x座標

，則繪製出來的方塊就會在正確位置了。



一張含有 文字 的圖片

自動產生的描述

圖4.25、檢查方塊是否超出遊戲區域並改變方塊x座標

|  |
| --- |
| **程式說明**  [105]在draw函式中加入region\_judge函式（加在迴圈外，即可移動整個圖形）。  [108] 方塊x座標加上需增加的總值。 |

* 1. 正確位置停下

為了判斷圖形應停下的位置，須將哪格有方塊填滿記錄下來。

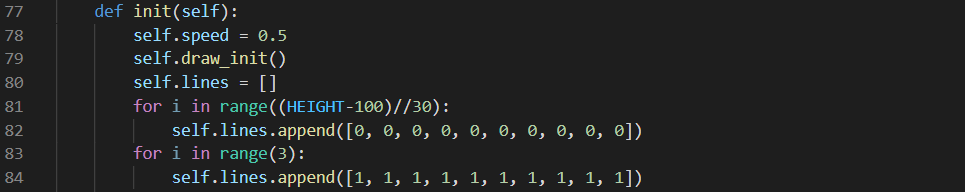
 

圖4.26、紀錄方塊是否被填滿

|  |
| --- |
| **程式說明**  [80-84]在init函式中建立一個二維序列存放每一格是否被填滿（0為未填滿；1為已填滿）。 |

|  |
| --- |
| **設計說明**  應我採取以4\*4(格)的範圍來定位，也是以4\*4(格)的範圍來檢查，故若我只建立20列（遊戲區域共20列），則當檢查到倒數第三列時，接下來的列數不足4列，會造成錯誤，故要再增加三列並設定為填滿狀況以便後續依列檢查。 |

紀錄完方塊填滿狀況後，建立judge\_touch函式判斷方塊應在何時停下。



一張含有 文字 的圖片

自動產生的描述

圖4.27、判斷方塊應在何時停下函式

|  |
| --- |
| **程式說明**  [138]建立judge\_list序列存放布林值。  [139-146]透過迴圈檢查在每一列是否可以作為圖形停下的定位線。 |

使方塊在定位線停下。



一張含有 文字 的圖片

自動產生的描述

圖4.28、判斷定位線

|  |
| --- |
| **程式說明**  [111]將judge\_touch函式加入draw函式。  [112-119]若從圖形定位原點所在的列數往下到某列檢查結果皆是True，則此列就為定位線。 |

|  |
| --- |
| **設計說明**  初始圖形定位原點會在遊戲區域外（－2），故檢查時若定位原點列數小於0，就須從0開始檢查（否則會從倒數兩列開始檢查），若大於0，則從定位原點所在列數開始，此設計可以判斷圖形下落到任何位置時，哪列可以做為定位線停下。  範例說明 以橘色L型方塊為例，紅色框框為第16列第4行的檢查範圍，第16列為定位線。  第16列→  第  4  行  ↓  圖4.29 、定位線檢查示意圖  (資料來源：研究者自行繪製) |

因Pygame的draw函式沒辦法畫一次就一直停留在畫面上，所以我另尋一個辦法，將已停下的方塊存放在一起，最後再繪製在畫面上。



一張含有 文字 的圖片

自動產生的描述

圖4.30、建立存放停下方塊的序列

|  |
| --- |
| **程式說明**  [85]在init函式中建立imgs序列存放所有方塊停下後的圖形及其座標。 |

建立一函式執行讓方塊到定位線時停下或直接下落，並在方塊停下時，將方塊及座標存入剛剛建立的序列。



一張含有 文字 的圖片

自動產生的描述

圖4.31、將方塊及座標加入序列

|  |
| --- |
| **程式說明**  [158-171]建立go\_down函式，使圖形到達定位線時停下，或是按下空白鍵後直接落下。  [162]獲取應停下的y座標值。  [163-168]建立畫布做為已停下的方塊以便後續顯示於畫面上，並將畫布及其座標加入imgs序列。  [169]將方塊新填滿的格子的位置從未填滿（0）更新為已填滿（1）。  [170-171]當四個方塊皆完成，則執行draw\_init函式，進行下一個初始方塊數值的重置。 |

為了判斷玩家是否按下空白鍵（執行直接落下），建立變數存放布林值。



一張含有 文字 的圖片

自動產生的描述

圖4.32、是否按下空白鍵的變數

|  |
| --- |
| **程式說明**  [134]在draw\_init函式中加入Go\_Down變數，存放是否按下空白鍵的布林值。 |

判斷是否按下空白鍵，並改變其布林值。



一張含有 文字 的圖片

自動產生的描述

圖4.33、判斷是否按下空白鍵

|  |
| --- |
| **程式說明**  [279-280] 當按下空白鍵時，Go\_Down則會變為True。 |

除了按下空白鍵時，當方塊到達定位線時也要執行go\_down函式。



一張含有 文字 的圖片

自動產生的描述

圖4.34、使方塊在定位線停下

|  |
| --- |
| **程式說明**  [120-121]當方塊落下至定位線或按下空白鍵時，執行go\_down函式。 |

將停下的所有方塊繪製到畫面上。



一張含有 文字 的圖片

自動產生的描述

圖4.35、繪製已停下方塊

|  |
| --- |
| **程式說明**  [289-290]將imgs中已停下的方塊繪製到螢幕上。 |

四、落下預判框線

因自己和同學在測試是否有bug時皆發現若沒有預判落下位置的框線的話，很容易發生看錯行導致放錯位置的情形，故我增加了此功能。



一張含有 文字 的圖片

自動產生的描述

圖4.36、繪製落下預判框線

|  |
| --- |
| **程式說明**  [157-159]繪製白色框線在方塊應落下的位置。 |

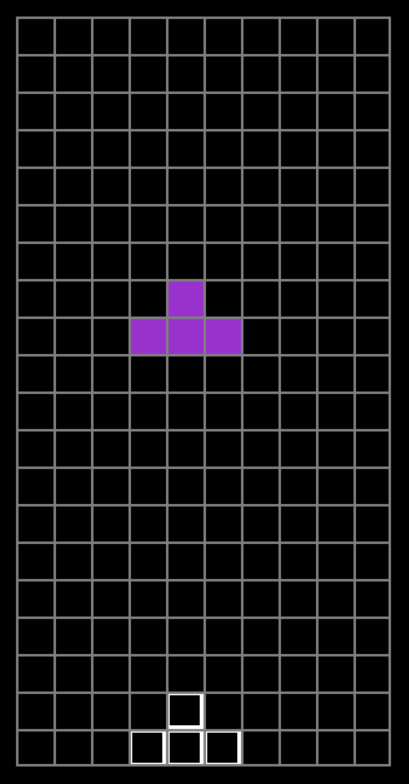


圖4.37、落下預判框線示意圖

五、判定何時gameover

為了判斷是否遊戲結束，建立變數存放布林值。



一張含有 文字, 電子用品 的圖片

自動產生的描述

圖4.38、是否遊戲結束的變數

|  |
| --- |
| 程式說明  [86]在init函式中加入gameover變數，存放是否遊戲結束的布林值。 |

判斷是否遊戲結束並改變其布林值。

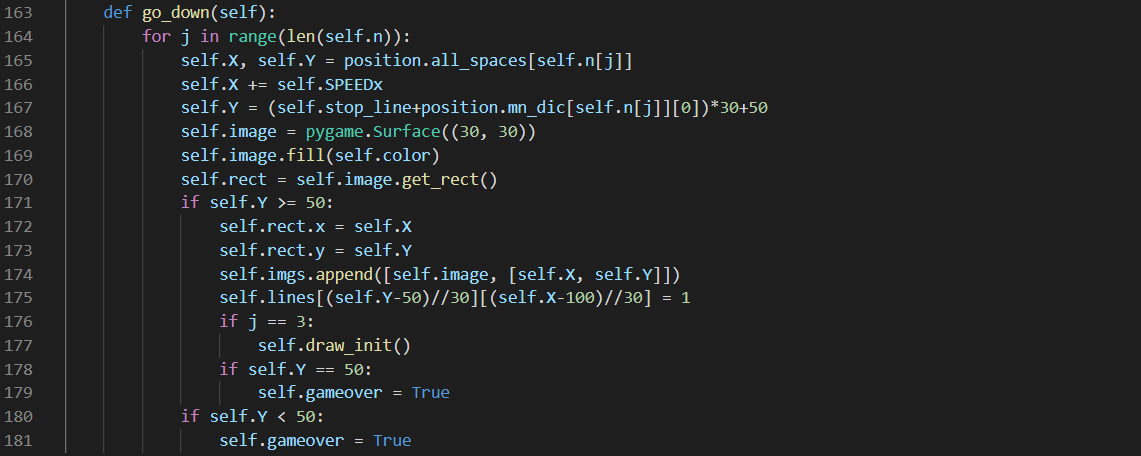
  


圖4.39、判斷是否遊戲結束

|  |
| --- |
| **程式說明**  [178-181]在go\_down函式中加入判斷式，若停下後的方塊碰到或超出遊戲邊界，則gameover變為True。而若方塊已超出遊戲邊界，則不繪製。 |

在測試時，發現在遊戲結束後，仍可進行左右移動及直接落下等操作，為了防止這個狀況，故設定為在遊戲結束後玩家不能再進行方塊的操作。



一張含有 文字 的圖片

自動產生的描述

圖4.40、遊戲結束時禁止執行方塊操作

|  |
| --- |
| **程式說明**  [275]在遊戲主迴圈取得輸入的判斷式加入新條件，則若遊戲結束，玩家無法再移動方塊。 |

1. 旋轉系統

建立一個Rotate類別存放所有旋轉系統的屬性及函式並建立一函式執行方塊旋轉。



一張含有 文字 的圖片

自動產生的描述



圖4.41、方塊旋轉函式

|  |
| --- |
| **程式說明**  [184-189]建立rotate函式，當按上鍵時，圖形向右轉。  [185]因各圖形旋轉態的序列皆按照向右轉的順序排列，故只要取下一個序列即可取得向右旋轉後的方塊編號。  [189]取得新的方塊編號後，重新繪製圖形。  [233]將變數rotate加入Rotate類別 |

判斷是否執行方塊旋轉函式（是否按下上鍵）。



一張含有 文字 的圖片

自動產生的描述

圖4.42、判斷是否按下上鍵

|  |
| --- |
| **程式說明**  [283-284]若按上鍵，則執行rotate函式，將圖形向右旋轉。 |

1. 消除系統
   1. 判斷可消除列

建立一個Break類別存放所有消除系統的屬性及函式並建立一函式檢查是否有層列全被方塊填滿。

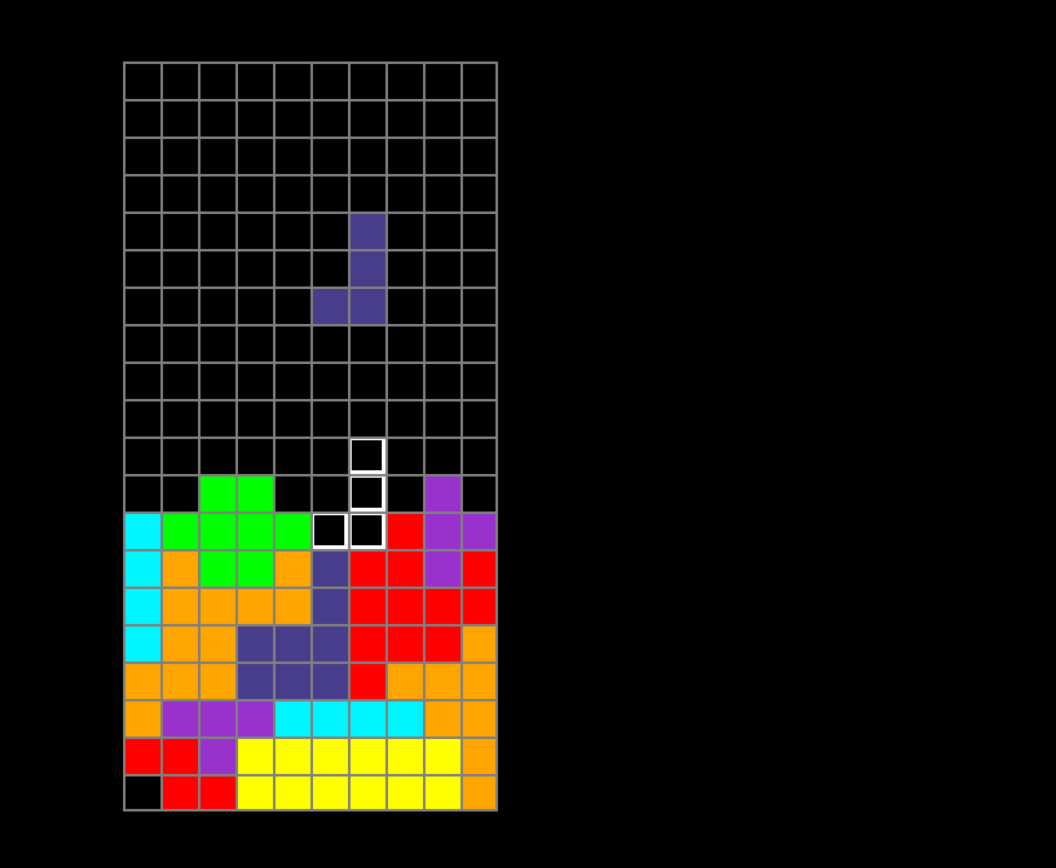
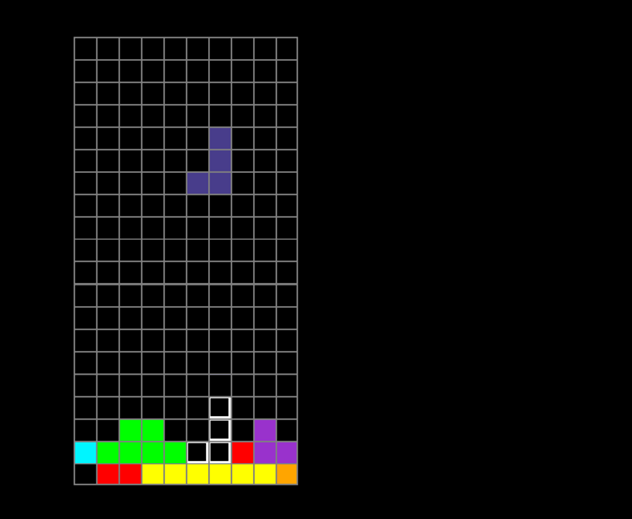


一張含有 文字 的圖片

自動產生的描述

圖4.43、檢查層列是否被方塊填滿

|  |
| --- |
| **程式說明**  [197-206]建立break\_judge函式判斷是否有可消除的列（皆被方塊填滿）。  [200-203]透過迴圈檢查每一列每一格是否皆被填滿（數值皆為1）。  [204-206]若10格皆被填滿，則將此列刪除並加入新的一列（設為皆未被填滿）。  [234]將變數break\_line加入Break類別。 |

可消除列

消除後

圖4.44、可消除列消除前後示意圖

* 1. 刪除可消除列並更新圖形及分數

建立分數初始值存放玩家分數。







圖4.45、建立分數初始值

|  |
| --- |
| **程式說明**  [192-193]建立init函式並存入分數初始值。  [236]執行init函式重置分數。 |

建立一函式來消除可消除列並增加分數。



一張含有 文字 的圖片

自動產生的描述

圖4.46、消除可消除列函式

|  |
| --- |
| **程式說明**  [210-219]建立break\_lines函式來刪除被消除的圖形並將上層圖形皆下降一格。  [211]建立imgs\_re序列存放需刪除的方塊　[212]每消除一列，就增加分數。  [214-215]將須刪除的方塊存入序列。  [216-217]將消除列上層方塊下降一格（座標加30）。  [218-219]將imgs內需消除方塊刪除。 |

執行消除可消除列函式。



一張含有 文字 的圖片

自動產生的描述

圖4.47、消除可消除列

|  |
| --- |
| **程式說明**  [206]在break\_judge函式中執行break\_lines函式，則每發現一次可消除列就刪除。 |

* 1. 等級提升

為了增加遊戲豐富度，方塊落下速度要逐漸增快，建立變數存放速度等級。





圖4.48、遊戲等級及升等時分數的變數

|  |
| --- |
| **程式說明**  [194-195]隨著分數增加，遊戲等級也會越來越高（速度越來越快），故在init函式中加入變數存放上一次升等時的分數及等級數。 |

當分數增加一個固定的值後，等級隨之提升。



一張含有 文字 的圖片

自動產生的描述

圖4.49、判斷分數增加值

|  |
| --- |
| **程式說明**  [293]在遊戲主迴圈內執行break\_judge函式來檢查是否有可消除列。  [294-297]每當分數增加80，則等級上升且速度加快。 |

1. 畫面顯示
   1. 初始畫面

初始畫面須顯示遊戲遊玩方法，為了判斷是否為第一次開始遊戲，建立一變數存放是否為第一次遊戲的布林值。



圖4.50、是否為第一次遊戲的變數

|  |
| --- |
| 程式說明  [61]建立init變數存放是否為第一次開始遊戲。 |

建立一函式進行文字繪製。

一張含有 文字 的圖片

自動產生的描述

圖4.51、繪製文字函式

|  |
| --- |
| **程式說明**  [224-233]建立draw\_text函式以執行文字繪製。 |

繪製遊戲初始畫面。



一張含有 文字 的圖片

自動產生的描述

圖4.52、繪製遊戲初始畫面

|  |
| --- |
| **程式說明**  [244-256]當為第一次開始遊戲時，進行初始畫面繪製。  [245-248]執行draw\_text函式以繪製遊戲遊玩方式的說明文字並更新顯示畫面。  [249-256]當按下關閉視窗時則退出此迴圈並將遊戲主迴圈關閉，若按下任意鍵則退出此迴圈（繼續遊戲主迴圈）。 |

一張含有 文字 的圖片

自動產生的描述

圖4.53、遊戲初始畫面

* 1. 遊戲結束畫面

繪製遊戲結束畫面。



一張含有 文字 的圖片

自動產生的描述

圖4.54、繪製遊戲結束畫面

|  |
| --- |
| **程式說明**  [239-240]將遊戲結束時要顯示的圖片載入。  [267-279]當遊戲結束時，進行遊戲結束畫面繪製。  [268-271]進行說明文字及圖片繪製並更新遊戲畫面。  [272-279] 當按下關閉視窗時則退出此迴圈並將遊戲主迴圈關閉，若按下任意鍵則執行移動系統及消行系統的init函式重置初始值並退出此迴圈（繼續遊戲主迴圈）。 |



圖4.55、遊戲結束畫面

* 1. 分數及等級

繪製分數及等級。





圖4.56、繪製分數及等級

|  |
| --- |
| **程式說明**  [299-300]顯示分數和等級的文字及數值。 |

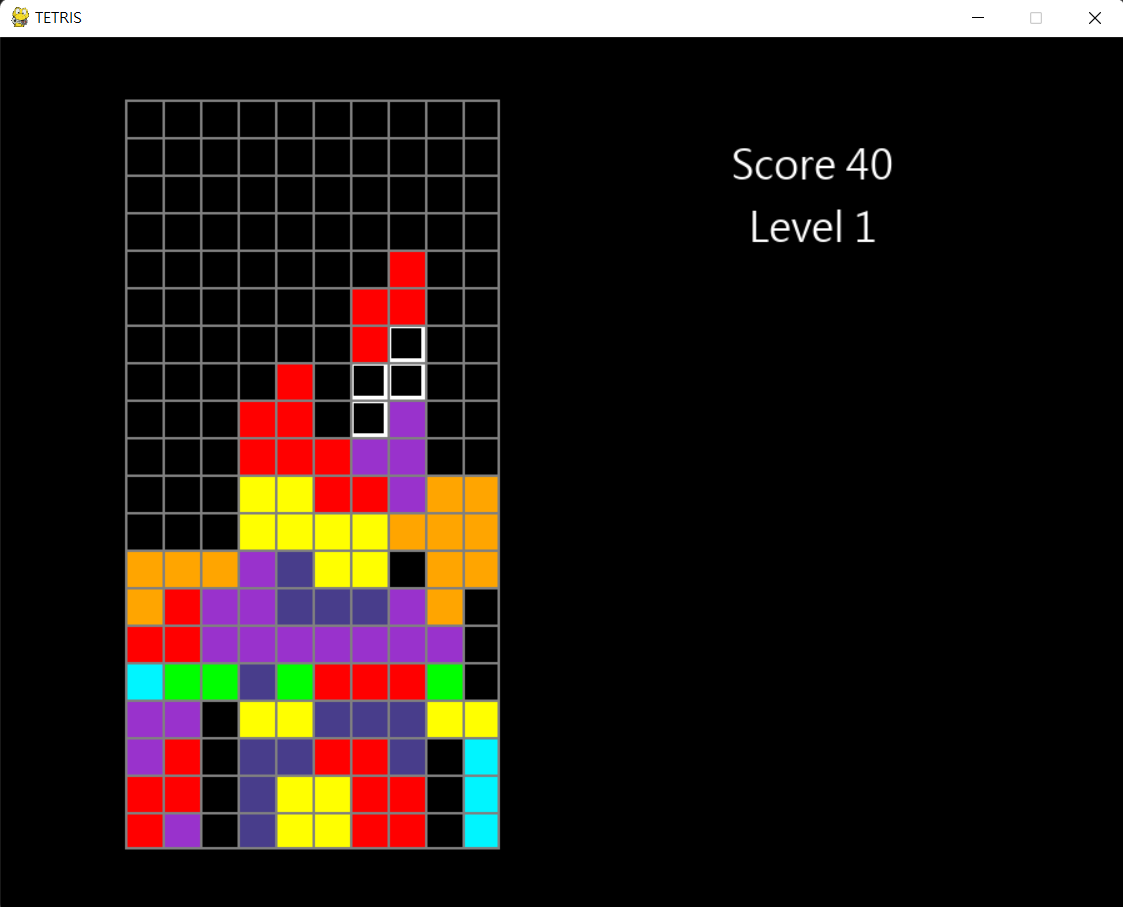


圖4.57、分數及等級顯示

1. 音效
   1. 混音器初始化

在使用音效之前，須先進行混音器初始化。



圖4.58、混音器初始化

* 1. 背景音樂

加入遊戲背景音樂。

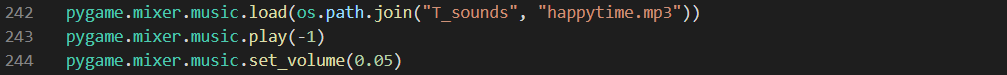


圖4.59、遊戲背景音樂

|  |
| --- |
| **程式說明**  [242]載入背景音樂檔案。  [243]將播放音樂函式參數值設為－1（循環播放）。  [244]設定背景音樂的音量大小。 |

三、得分音效

加入得分音效。





一張含有 文字 的圖片

自動產生的描述

圖4.60、得分音效

|  |
| --- |
| **程式說明**  [246]載入得分音效檔案。  [211]得分時播放得分音效。 |

1. 總體設計概念

我將想完成的遊戲功能分為4個系統，起初我並沒有使用類別來分類不同系統的函式，而造成的結果就是程式碼雜亂無章，每次要修改時都要花很多時間找出問題所在處。因此我後來將這4個系統所須函式分別存入不同類別，使用類別的好處有以下兩點：

* 1. 程式碼更方便閱讀

以往過去的程式設計經驗都僅限於上課練習題目或競賽內容，而這類型的程式碼幾乎都較簡短，不像一個遊戲，需要很多環節一起配合。若是沒使用類別，將會造成程式碼混雜，不只修改不易，也會讓其他人看不懂。尤其是多人一起完成的專案，分類清楚的程式碼是非常重要的，不同系統分成不同類別就可以讓人一眼就清楚明瞭這些函式的功用。

* 1. 可以快速、大量生產

在之後如果碰到需要類似的程式碼的時候，若我想另開一個檔案製作俄羅斯方塊進階版，能夠使用類別的繼承，就不須在新檔案上貼重複的內容，可以快速且大量的產出類似的程式碼。

第五章 結論與建議

1. 結論

我成功利用Python設計出俄羅斯方塊，並且邀請同學進行遊戲的測試，修正之後，完成了此次的成果，未來我也會再加入一些我想新增的功能使遊戲更完整及好玩。

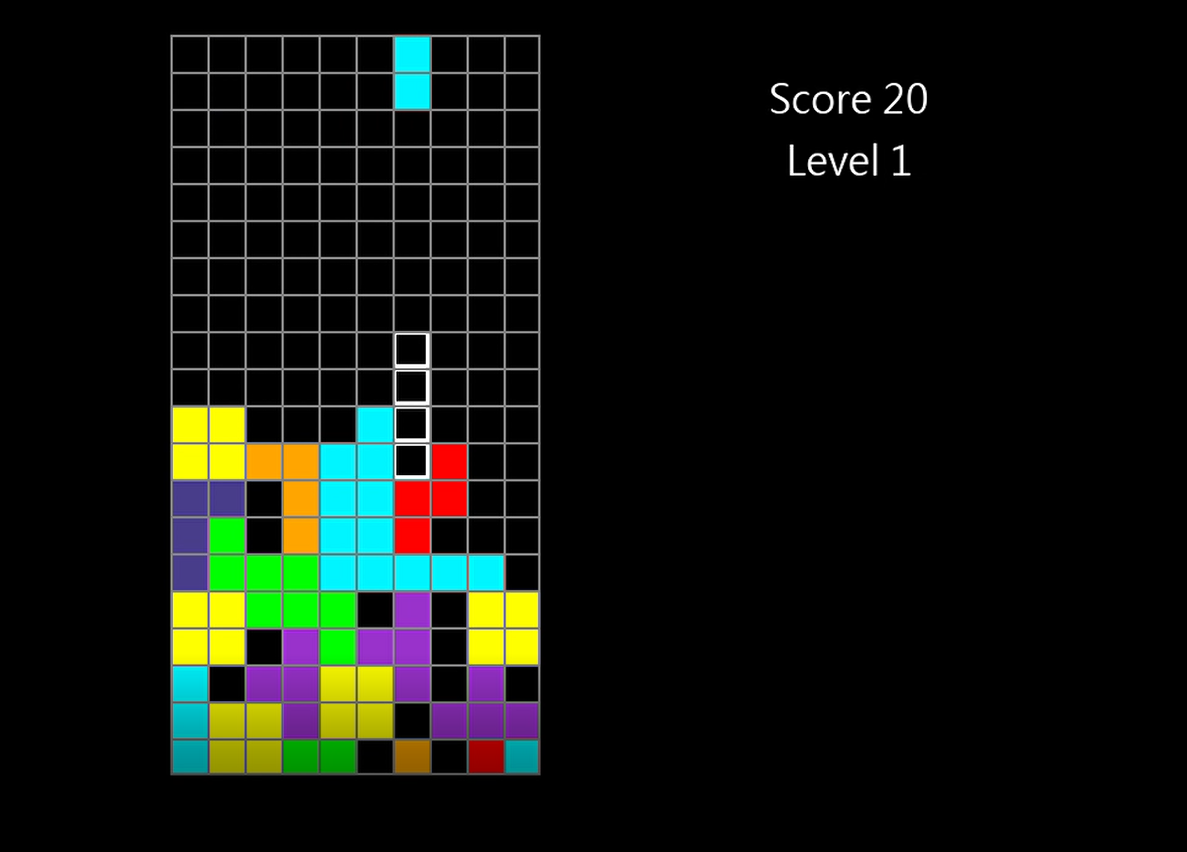


圖5.1、程式實際運行影片截圖（00:00:35）

影片網址：<https://youtu.be/hz0Yj-E1VoQ>

1. 建議

一、增加遊戲豐富度

(一) 遊戲區域形狀改變

將遊戲區域形狀改成特殊形狀（星型等）並將方塊改成對應的圖形。

更改方向 定位方法須重新設定，且方式會比目前複雜。

(二) 方塊種類增加

將方塊改成由五塊小正方形甚至更多塊所組成。

更改方向 可延續使用目前的定位方法，且方塊圖型會更多種。

(三) 增加道具

消除層列時有機率獲得道具。

更改方向 須新增道具動畫以及道具效果的函式。

(四) 增加炸彈

消除層列時有機率出現炸彈（取代某格方塊），若放置方塊時觸碰炸彈則

爆炸且一次可消較多層方塊。

更改方向 須新增爆炸動畫以及隨機出現的函式並配合消除系統。

(五) 對戰功能

雙人單機對戰（非線上）。

更改方向 須分成兩個座標系統，其他系統只須將兩個玩家的操作分隔開

來，舉例而言，若玩家A按了右鍵，則就只能玩家A區塊的方塊執行右

移函式。

二、Sprite模組

Pygame中有Sprite模組，其中有可以偵測sprite物件是否碰撞的函式，之後可以嘗試使用這個功能來判定方塊是否落地，程式碼或許會比現在的消除系統來得精簡許多，因為我所寫的判斷是否落地的函式是使用偏土法煉鋼的方法，用迴圈一格一格檢查，所以過程較為繁複。

資料來源

1. CSDN博客-專業IT技術發表平台－Pygame入門教程（一）初始化和主循環

<https://blog.csdn.net/baidu_36499789/article/details/113462828#:~:text=%E5%88%9D%E5%A7%8B%E5%8C%96pygame%E5%BA%93%20pygame.init%20%28%29%20%E6%98%AF%E5%9C%A8%E4%BD%BF%E7%94%A8%20pygame%20%E5%BA%93%E4%B9%8B%E5%89%8D%E5%88%9D%E5%A7%8B%E5%8C%96%20pygame,%E5%BA%93%E3%80%82%20init%20%E8%BF%99%E4%B8%AA%E5%8D%95%E8%AF%8D%E7%BC%A9%E5%86%99%E8%A6%81%E8%AE%B0%E5%A5%BD%E4%BA%86%EF%BC%8C%E5%AE%83%E6%98%AF%20initialize%20%28%E5%88%9D%E5%A7%8B%E5%8C%96%29%20%E7%9A%84%E7%BC%A9%E5%86%99%EF%BC%8C%E5%9C%A8%E7%BC%96%E7%A8%8B%E4%B8%AD%E7%BB%8F%E5%B8%B8%E5%87%BA%E7%8E%B0%E3%80%82%202.>

1. CSDN博客-專業IT技術發表平台－Python Pygame常用操作

<https://blog.csdn.net/doasmaster/article/details/104364993?spm=1001.2101.3001.6650.1&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1.pc_relevant_default&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1.pc_relevant_default&utm_relevant_index=2>