麗山高中資訊專題研究

**Python拈遊戲**

**學生：20418呂可名**

**指導老師:黃履峰**

目錄

[壹、序論 3](#_Toc454452982)

[**一、** **研究動機** 3](#_Toc454452983)

[**二、** **研究目標** 3](#_Toc454452984)

[**貳、** **相關研究** 4](#_Toc454452985)

[**一、** **拈遊戲簡介** 4](#_Toc454452986)

[**二、** **拈遊戲必勝原理研究** 4](#_Toc454452987)

[**i.** **異或(exclusive or)** 4](#_Toc454452988)

[**ii.** **拈運算** 4](#_Toc454452989)

[**三、** **MVC程式架構** 5](#_Toc454452990)

[**四、** **Turtle、Tk、Pygame模組** 6](#_Toc454452991)

[**i.** **Turtle** 6](#_Toc454452992)

[**ii.** **Tk** 6](#_Toc454452993)

[**iii.** **Pygame** 6](#_Toc454452994)

[**參、** **遊戲實作** 7](#_Toc454452995)

[**一、** **演算法** 7](#_Toc454452996)

[**i.** **程式碼** 7](#_Toc454452997)

[**ii.** **程式解說** 9](#_Toc454452998)

[**二、** **使用者介面** 9](#_Toc454452999)

[**i.** **主要類別(class)** 9](#_Toc454453000)

[**ii.** **MVC主要功能** 10](#_Toc454453001)

[**iii.** **其它類別功能** 15](#_Toc454453002)

[**肆、** **結論與未來展望** 26](#_Toc454453003)

[**一、** **結論** 26](#_Toc454453004)

[**二、** **未來展望** 26](#_Toc454453005)

[**伍、** **參考資料** 27](#_Toc454453006)

[**一、** **網路資訊** 27](#_Toc454453007)

[**二、** **參考書籍** 27](#_Toc454453008)

**圖表目錄**

[表 1 異或真質表 5](#_Toc454454876)

[表 2 實際計算流程 6](#_Toc454454877)

[圖 1 MVC架構 7](#_Toc454454878)

[圖 2 TK視窗 10](#_Toc454454879)

[圖 3 文字訊息 15](#_Toc454454880)

[圖 4 兩種不同型態的Level 18](#_Toc454454881)

[圖 5 Hint按鈕(點擊前) 19](#_Toc454454882)

[圖 6 Hint按鈕(點擊後) 20](#_Toc454454883)

[圖 7 雙人模式 21](#_Toc454454884)

[圖 8 更改排數 22](#_Toc454454885)

[圖 9 不同的背景顏色與圖案 24](#_Toc454454886)

# 壹、序論

* 1. **研究動機**

第一次接觸到這個遊戲是在高一上學期的數學研方老師所提到。當下就對它非常有興趣，也很想將它做成一個遊戲。對於開發這個遊戲，除了編寫出此程式的演算法之外，我想進一步美化此程式的使用者介面，設計不同版本及規則的遊戲。因此我選擇與 C++、Java相較之下比較好上手的 Python來完成此遊戲。

* 1. **研究目標**

學習Python的使用者介面程式架構，並用python程式語言設計出圖形介面的拈遊戲。

1. **相關研究**
   1. **拈遊戲簡介**

拈遊戲據說來自於中國，藉由被販賣到美洲的勞工外傳。其遊戲主要利用石頭或硬幣遊玩，最為常見的玩法為將硬幣分為三列，分別為三、四、五個硬幣，兩人輪流取硬幣，每次只能取一個或以上的硬幣在同列上，玩到最後，拿到最後一個硬幣的人就贏了。

* 1. **拈遊戲必勝原理研究**
     1. **異或(exclusive or)**

異或(也稱為互斥或)，是一種邏輯運算，它與一般的「或」不同的地方是，只有當兩個輸入不同的邏輯運算。通常以XOR、⊕、^表示。

|  |  |  |
| --- | --- | --- |
| XOR | True(1) | False(0) |
| True(1) | False(0) | True(1) |
| False(0) | True(1) | False(0) |

表 1 異或真質表

* + 1. **拈運算**

整個遊戲的必勝原理利用了異或進行了一連串的二進位運算，如下表：

|  |  |  |
| --- | --- | --- |
| 棒子數 | 二進位數 | 說明 |
| 3  4  5  +(xor) | = 011  = 100  = 101  = 010 不安全殘局 | 如為3,4,5個，則換成二進位進行異或運算三位無皆為0則為不安全殘局。 |
| 此時第一排拿走兩個 | | |
| 1  4  5  +(xor) | = 001  = 100  = 101  = 000 安全殘局 | 此時從第一列取走2個，變為1,4,5個，換成二進位進位進行異或運算三位皆為0則為安全殘局 |

表 2 實際計算流程

* 安全與不安全殘局
  + 安全殘局：所有排數進行異或判斷的值為0
  + 不安全殘局：所有排數進行異或判斷的值不為0

若處於安全殘局，則對方不管怎麼拿都會處於不安全殘局。若處於不安全殘局，則需要在某些排取適當的數量才可成為安全殘局。只要一直維持在安全殘局即可獲得勝利。

* 1. **MVC程式架構**

為模型(Model)、視圖(View)、控制器(Controller) 三個部分所構成。

三個部分彼此環環相扣，為實作一個動態的程式設計

1. 模型(Model)：程式的核心，存放資料及一些基本演算法，並且連結整個遊戲。
2. 視圖(View)：負責呈現圖形介面的部分。
3. 控制器(Controller)：接收使用者所進行的操作，包括鍵盤輸入、滑鼠點擊等。

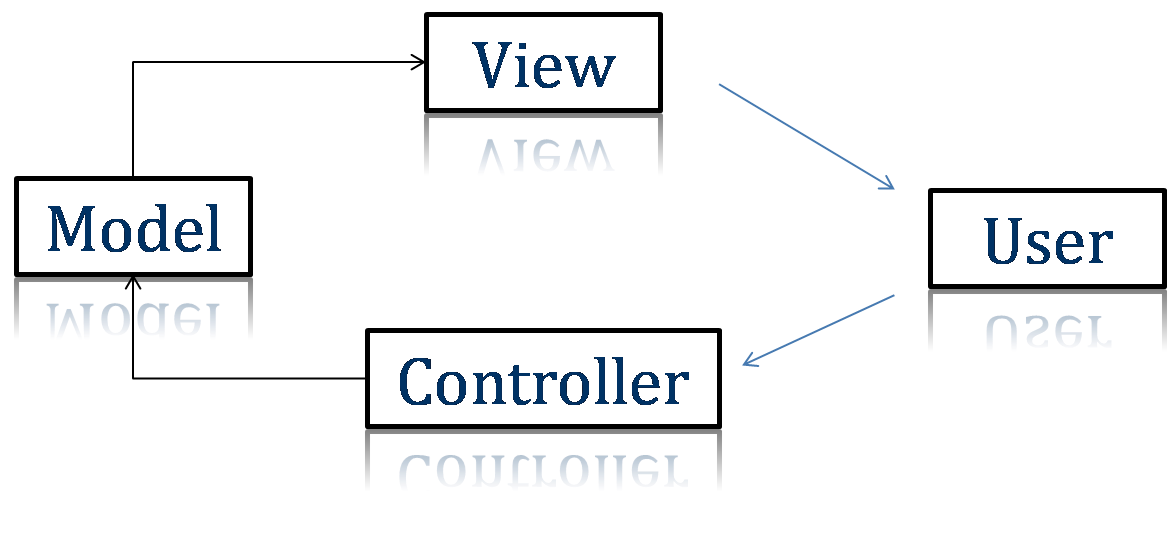


圖 1 MVC架構

* 1. **Turtle、Tk、Pygame模組**

在整個遊戲主要運用到了Python的三個模組，分別是Turtle、Tk 以及 Pygame。

* + 1. **Turtle**

Turtle模組是一個處理圖形介面的模組。其中包含了許多簡單又好用的函數，讓一般人能夠很快上手。

* + 1. **Tk**

是圖形介面的始祖，許多跟圖形介面有關的模組都繼承Tk，包括Turtle。而他能處理一些更底層的部分。

* + 1. **Pygame**

Python用來製作遊戲的模組，可以處理圖形介面、聲音等等多媒體的效果。在此遊戲內用來播放背景音樂。

1. **遊戲實作**
   1. **演算法**

程式的核心必勝演算法，就如同上面所說的利用異或判斷以及二進位運算來實現。

* + 1. **程式碼**

**def** computerzug**(**state**):**

nRows**=** len**(**state**)** #指定排數

xored**=** 0

**for** s **in** state**:**

#每一排進行判斷

xored **^=** s

move**=** randommove**(**state**)** # 不經考慮，隨機出手。

**if** xored **!=** 0**:**

# 此情形，我方有機會勝利，

# 做一個動作，使得 接下來的 xored ==0，則敵方必敗。

**for** z **in** range**(**nRows**):**

s **=** state**[**z**]** **^** xored

**if** s **<=** state**[**z**]:**

move**=** **(**z**,** s**)**

**break**

**return** move

* + 1. **程式解說**

xored**=** 0

**for** s **in** state**:**

#每一排進行判斷

xored **^=** s

利用迴圈進行每一排的異或判斷

**if** xored **!=** 0**:**

# 此情形，我方有機會勝利，

# 做一個動作，使得 接下來的 xored ==0，則敵方必敗。

**for** z **in** range**(**nRows**):**

s **=** state**[**z**]** **^** xored

**if** s **<=** state**[**z**]:**

move**=** **(**z**,** s**)**

**break**

利用異或判斷，決定該在第幾排拿掉幾根棒子。

* 1. **使用者介面**
     1. **主要類別(class)**

1. **NimGame**

整個遊戲主要的類，包含了執行此遊戲的函數，主要實例化MVC及其他的類，將整個程式連結起來，

1. **NimModel**

為資料及演算法存放處，以及處理遊戲的一些基本運作，包括移動棒子、判斷輸贏、以及切換模式等等。

1. **NimView**

管理遊戲中所有畫面的顯現，包含背景、棒子、選單、音樂、文字等等，通常接收來自NimModel的訊息，

1. **NimController**

接收來自玩家使用鍵盤、滑鼠的訊息，進而將此訊息傳送到NimModel處理。

* + 1. **MVC主要功能**

1. **建立視窗**

NimGame內部的兩行指令，分別是利用Turtle、Tk模組建立的視窗，再將這兩個視窗合併為一個視窗

self**.**screen **=** turtle**.**Screen**()**

self**.**tk **=** self**.**screen**.**getcanvas**().**master

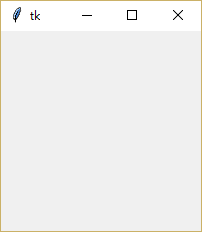


圖 2 TK視窗

1. **製作棒子**

在NimView內先指定好排數，再利用迴圈製作，並且顯現在螢幕上。

self**.**nRows**=** game**.**nRows

**for** row **in** range**(**self**.**nRows**):**

**for** col **in** range**(**MAXSTICKS**):**

self**.**sticks**[(**row**,** col**)]** **=** Stick**(**row**,** col**,** game

1. **電腦移動棒子、判斷輸贏**

控制電腦選取棒子的主要函數，能夠切換以計算或者是隨機的方式選取，並且判斷輸贏。

**def** move**(**self**,** row**,** col**):**

maxspalte **=** self**.**sticks**[**row**]**

self**.**sticks**[**row**]** **=** col

**if** self**.**game**.**mode **==** 0**:**

self**.**game**.**view**.**notify\_move**(**row**,** col**,**maxspalte**,** self**.**player**)**

**if** self**.**game\_over**():**

self**.**game**.**state **=** NimGame**.**OVER

self**.**winner **=** self**.**player

self**.**game**.**view**.**notify\_over**()**

#遊戲結束按鈕隱藏

self**.**game**.**hint**.**ht**()**

self**.**game**.**level**.**ht**()**

**elif** self**.**player **==** 0**:**

self**.**player **=** 1

#偵測智商

**if** self**.**game**.**level**.**value **==** **True:**

row**,** col **=** computerzug**(**self**.**sticks**)**

**else:**

row**,** col **=** randommove**(**self**.**sticks**)**

self**.**move**(**row**,** col**)**

self**.**player **=** 0

**elif** self**.**game**.**mode **==** 1**:**

self**.**game**.**view**.**player\_move**(**row**,** col**,** maxspalte**,** self**.**player**)**

**if** self**.**game\_over**():**

self**.**game**.**state **=** NimGame**.**OVER

self**.**winner **=** self**.**player self**.**game**.**view**.**player\_over**()**

#遊戲結束按鈕隱藏

self**.**game**.**hint**.**ht**()**

self**.**game**.**level**.**ht**()**

**if** self**.**player **==** 0**:**

self**.**player **=** 1

**else:**

self**.**player **=** 0

1. **滑鼠點擊移動棒子**

在NimController內利用一個迴圈進行判斷每一次玩家所點擊的棒子是否被移動過，藉此來移動棒子。

**for** stick **in** self**.**sticks**.**values**():**

**if** self**.**game**.**mode **==** 0**:**

stick**.**onclick**(**stick**.**makemove**)**

**elif** self**.**game**.**mode **==** 1**:**

stick**.**onclick**(**stick**.**playermove

1. **鍵盤控制**

在NimController內部接收鍵盤的訊息，再將此訊息傳至其他類別進行處理。

# 用 "space" 來 開始遊戲

self**.**game**.**screen**.**onkey**(**self**.**game**.**model**.**setup**,** "space"**)**

# 用 "Esc" 來 清除 畫面

self**.**game**.**screen**.**onkey**(**self**.**game**.**view**.**clear**,** "Escape"**)**

1. **文字訊息**

遊戲下方所顯現的文字，使用了一些Turtle內建的文字版面設定。

**def** display**(**self**,** msg1**,** msg2**=None):**

self**.**screen**.**tracer**(False)**

self**.**writer**.**clear**()**

**if** msg2 **is** **not** **None:**

self**.**writer**.**goto**(**0**,** **-** SCREENHEIGHT **//** 2 **+** 48**)**

self**.**writer**.**pencolor**(**"red"**)**

self**.**writer**.**write**(**msg2**,** align**=**"center"**,** font**=(**"Courier"**,**18**,**"bold"**))**

self**.**writer**.**goto**(**0**,** **-** SCREENHEIGHT **//** 2 **+** 20**)**

遊戲剛開始時的文字說明。這部分是在此函數內創建了一個類(屬於Turtle類)，專門用來書寫文字。

**def** set\_msg**(**self**):**

msg **=** '''

Python 拈遊戲 (NimGame)

以下是遊戲規則:

1. 玩家與電腦輪流撿棒子，撿到最後一個的就贏了。

2. 點擊任一棒子則玩家將會撿走該棒子右側的所有棒子。

3. 點擊左下角的方塊會給予提示。

4. 點擊右下角的烏龜可選擇難度(黃色代表難，橘色代表簡單)

5. 遊戲開始前請選擇排數

'''

self**.**t **=** Turtle**()**

self**.**t**.**pu**()**

self**.**t**.**ht**()**

self**.**t**.**goto**(**0**,-**50**)**

self**.**t**.**write**(**msg**,** align**=**"center"**,** font**=(**'Arial'**,** 15**,** 'bold'**))**

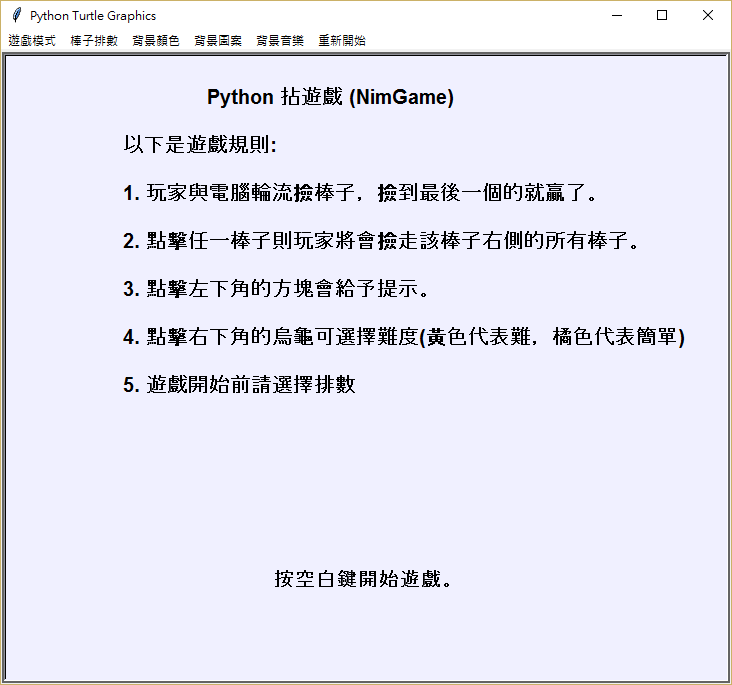


圖 3 文字訊息

1. **其它類別功能**
   * + 1. **Stick**

棒子的實體，實現棒子所有的屬性、功能，以及利用內部的一些函數讓自身屬性能夠於其他地方使用。

**class** **Stick(**Turtle**):**

**def** \_\_init\_\_**(**self**,** row**,** col**,** game**):**

Turtle**.**\_\_init\_\_**(**self**,** visible**=False)**

self**.**row**=** row

self**.**col**=** col

self**.**game**=** game

self**.**nRows**=** self**.**game**.**nRows

x**,** y**=** self**.**coords**(**row**,** col**)**

self**.**shape**(**"square"**)**

a **=** 25.0

self**.**shapesize**(**HUNIT**/**a**,** WUNIT**/**25.0**)**

self**.**speed**(**0**)**

self**.**pu**()**

self**.**goto**(**x**,**y**)**

self**.**color**(**'gray'**)**

**def** coords**(**self**,** row**,** col**):**

packet**,** remainder **=** divmod**(**col**,** 5**)**

x **=** **(**3 **+** 11 **\*** packet **+** 2 **\*** remainder**)** **\*** WUNIT

y **=** **(**row**)** **\*** HUNIT

x0**=** x **-**SCREENWIDTH**//**2 **+** WUNIT**//**2

y0**=** **-**y **+**SCREENHEIGHT**//**2 **-** HUNIT**//**2

**return** x0**,** y0

**def** makemove**(**self**,** x**,** y**):**

**if** self**.**game**.**state **!=** NimGame**.**RUNNING**:**

**return**

self**.**game**.**controller**.**notify\_move**(**self**.**row**,** self**.**col**)**

**def** playermove**(**self**,** x**,** y**):**

**if** self**.**game**.**state **!=** NimGame**.**RUNNING**:**

**return**

self**.**game**.**model**.**player\_move**(**self**.**row**,** self**.**col**)**

* + - 1. **Level**

為一按鈕，以滑鼠點擊的方式改變電腦的演算法處理方式，分為以必勝法則選取棒子，或是隨機選取。內部的屬性都以繼承Turtle的方式來呈現，並且建立一個函數changevalue來改變自身的值，讓其它地方能夠偵測到自身被點擊。

**class** **Level(**Turtle**):**

**def** \_\_init\_\_**(**self**,** game**=None,** x**=**0**,** y**=**0**):**

Turtle**.**\_\_init\_\_**(**self**,** visible**=False)**

self**.**value **=** **True**

self**.**shape**(**'turtle'**)**

self**.**shapesize**(**HUNIT**/**15**,** WUNIT**/**5**)**

self**.**speed**(**0**)**

self**.**left**(**90**)**

self**.**pu**()**

self**.**goto**(**x**,**y**)**

self**.**color**(**'yellow'**)**

self**.**onclick**(**self**.**changevalue**)**

**def** changevalue**(**self**,** x**,** y**):**

**if(**self**.**value**==True):**

self**.**value **=** **False**

self**.**color**(**'orange'**)**

**else:**

self**.**value **=** **True**

self**.**color**(**'yellow'**)**

**003.PNG**

圖 4 兩種不同型態的Level

* + - 1. **Hint**

為一按鈕，當其被點擊時，螢幕會出現提示，讓玩家能夠輕鬆打敗電腦。這裡採用點擊不放才會出現指令，因此需要兩個不同的函數分別偵測點擊與放開改變其值。

**class** **Hint(**Turtle**):**

**def** \_\_init\_\_**(**self**,** game**,** x**=**0**,** y**=**0**):**

Turtle**.**\_\_init\_\_**(**self**,** visible**=False)**

self**.**game **=** game

self**.**value **=** **True**

self**.**shape**(**'square'**)**

self**.**shapesize**(**HUNIT**/**15**,** WUNIT**/**5**)**

self**.**speed**(**0**)**

self**.**pu**()**

self**.**goto**(**x**,**y**)**

self**.**color**(**'blue'**)**

self**.**onclick**(**self**.**changevalue**)**

self**.**onrelease**(**self**.**changevalue02**)**

**def** changevalue**(**self**,** x**,** y**):**

**if(**self**.**value**==True):**

self**.**value **=** **False**

self**.**color**(**'violet'**)** self**.**game**.**view**.**writer**.**goto**(**0**,-**100**)**

move **=** computerzug**(**self**.**game**.**model**.**sticks**)**

z**=** '提示：第 {} 排，拿到剩下 {} 個。'**.**format**(**move**[**0**]+**1**,** move**[**1**])**self**.**game**.**view**.**writer**.**write**(**z**,**align**=**"center"**,** font**=(**"Courier"**,**14**,**"bold"**))**

**def** changevalue02**(**self**,** x**,** y**):**

**if(**self**.**value **==** **False):**

self**.**value **=** **True**

self**.**color**(**'blue'**)**

self**.**game**.**view**.**writer**.**undo**()**



圖 5 Hint按鈕(點擊前)

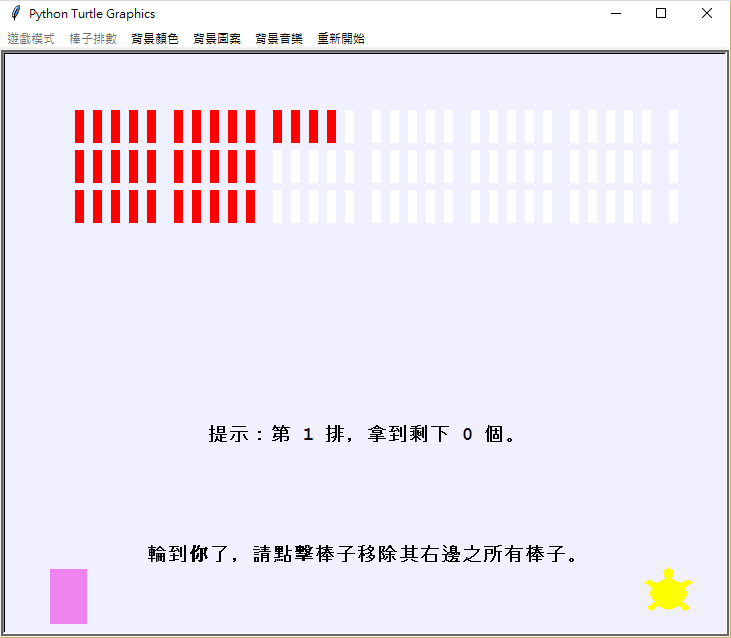
****

圖 6 Hint按鈕(點擊後)

* + - 1. **CvMenu**

遊戲上排的選單列，所有的選單屬性及功能都在此控制，這部分使用了TK模組來實現。

1. **雙人模式**

增加雙人模式的功能，當模式改變後，遊戲會重新開始，是因為這部分屬於底層的更改，許多部份都必須要重新再執行一遍。

bMenu **=** Menu**(**aMenu**,** tearoff**=**0**)**

#bLabel = ['單人模式', '雙人模式']

bLabel **=** **{**'單人模式'**:** 0**,**

'雙人模式'**:** 1**}**

keyL **=** sorted**(**bLabel**)**

**for** k **in** keyL**:**

**def** cmd**(**x**=**k**):**

nRows **=** game**.**nRows

mode **=** bLabel**[**x**]**

aNimGame**=** NimGame**(**nRows**,** mode**)**

bMenu**.**add\_command**(**label**=**k**,** command**=**cmd**)**

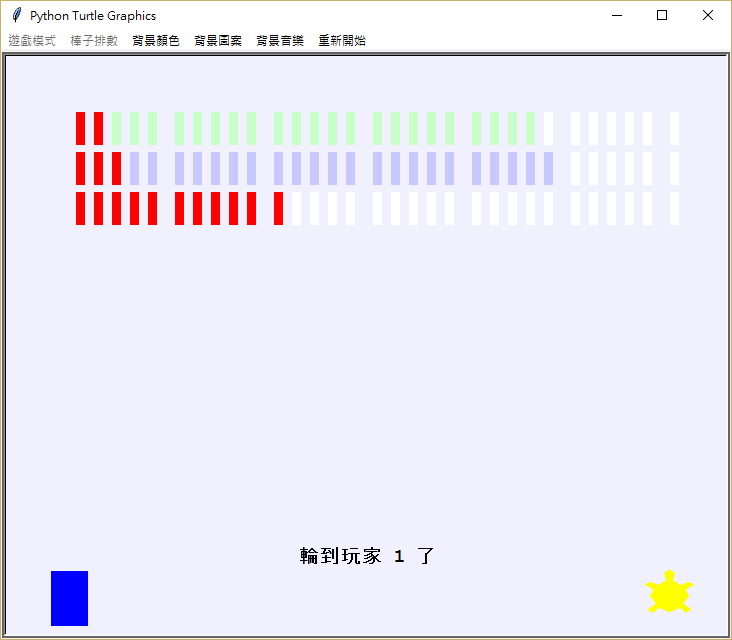


圖 7 雙人模式

1. **更改排數**

建立一個列表將排數分別以字串的形式輸入進選單列，同樣的更改後也會使遊戲重新開始。

cMenu **=** Menu**(**aMenu**,** tearoff**=**0**)**

cLabel **=** **[**'3'**,** '4'**,** '5'**,** '6'**,** '7'**,** '8'**,** '9'**,** '10'**]**

**for** i **in** cLabel**:**

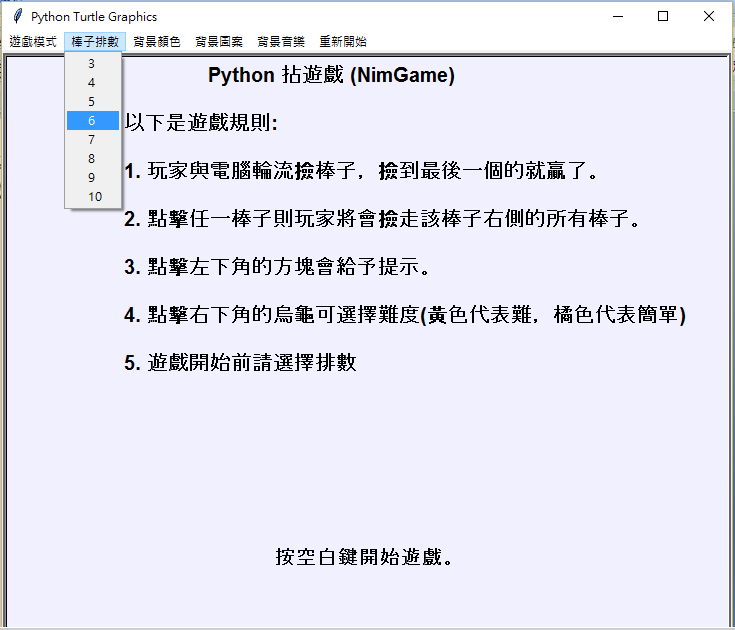
**def** cmd**(**x**=**i**):**

nRows **=** int**(**x**)**

mode **=** game**.**mode

aNimGame**=** NimGame**(**nRows**,** mode**)**

cMenu**.**add\_command**(**label**=**i**,** command**=**cmd**)**



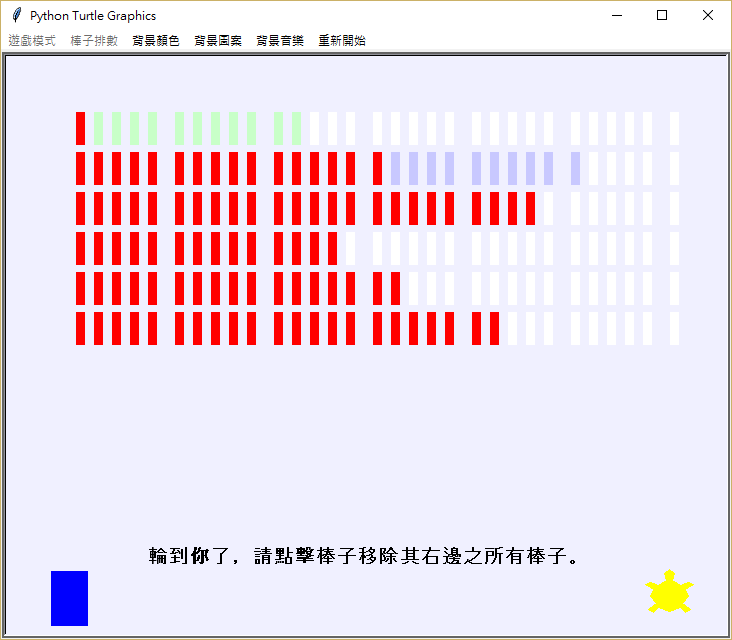


圖 8 更改排數

1. **背景顏色及圖案**

設定了四種不同的背景顏色及圖案，這部分使用了一種叫字典(dictionary)的資料處存型態，可使用文字與屬性對應。

dMenu **=** Menu**(**aMenu**,** tearoff**=**0**)**

dLabel**=** **{**'顏色 1'**:** '#F0F0FF'**,**

'顏色 2'**:** '#DDDDDD'**,**

'顏色 3'**:** '#FFFFBB'**,**

'顏色 4'**:** '#EEFFBB'**}**

keyL **=** sorted**(**dLabel**)**

**for** k **in** keyL**:**

**def** cmd**(**x**=**k**):**

game**.**screen**.**bgcolor**(**dLabel**[**x**])**

dMenu**.**add\_command**(**label**=** k**,** command**=** cmd**)**

#-------------------------------------------------

eMenu **=** Menu**(**aMenu**,** tearoff**=**0**)**

eLabel **=** **{**'無圖案'**:** 'nopic'**,**

'圖案 1'**:** 'pictures/background.png'**,**

'圖案 2'**:** 'pictures/dog01.png'**,**

'圖案 3'**:** 'pictures/calvin.png'**}**

keyL **=** sorted**(**eLabel**)**

**for** k **in** keyL**:**

**def** cmd**(**x**=**k**):**

game**.**screen**.**bgpic**(**eLabel**[**x**])**

eMenu**.**add\_command**(**label **=** k**,** command**=**cmd**)**

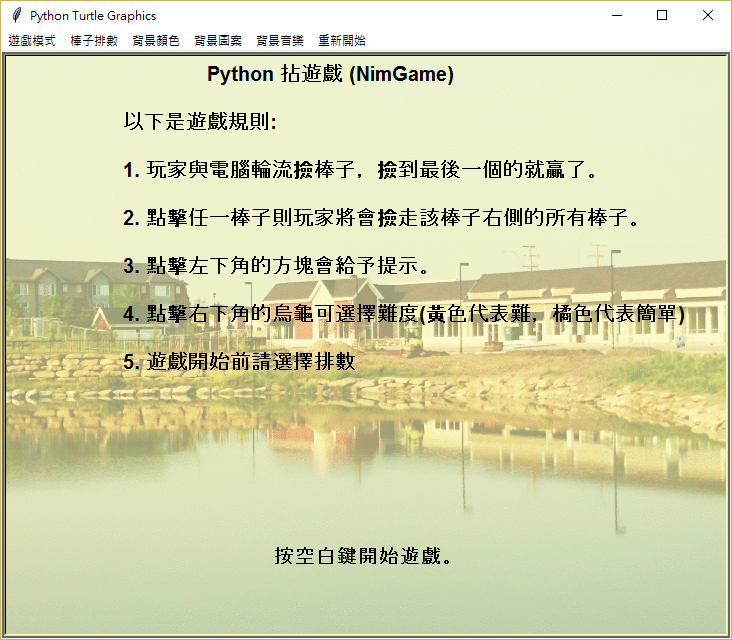


圖 9 不同的背景顏色與圖案

1. **背景音樂**

音樂的部分利用到了pygame模組，內部的mixer()函數用來控制音樂的開始、切換、以及停止。

fMenu **=** Menu**(**aMenu**,** tearoff**=**0**)**

fLabel**=** **{**'音樂停止'**:** '\_\_0\_\_.mid'**,**

'音樂01'**:** 'musics/background.mid'**,**

'音樂02'**:** 'musics/canon.mid'**,**

'音樂03'**:** 'musics/John\_Cena.mp3'**}**

keyL **=** sorted**(**fLabel**)**

**for** k **in** keyL**:**

**def** cmd**(**x**=**k**):**

musicFile**=** fLabel**[**x**]**

**if** musicFile **!=** '\_\_0\_\_.mid'**:**

pygame**.**mixer**.**music**.**load**(**musicFile**)**

pygame**.**mixer**.**music**.**play**(-**1**,** 0.0**)**

**else:**

pygame**.**mixer**.**music**.**stop**()**

fMenu**.**add\_command**(**label**=** k**,** command**=** cmd**)**

1. **遊戲重新開始**

使遊戲重新開始。

**def** cmd**():**

aNimGame**=** NimGame**(**game**.**nRows**,** game**.**mode**)**

aMenu**.**add\_command**(**label**=** '重新開始'**,** command**=** cmd**)**

game**.**tk**.**config**(**menu**=** aMenu**)**

1. **結論與未來展望**
   1. **結論**

經過了一年的專題研究課程，相信自己已經成長了不少，許多能力包括寫程式、查資料、做簡報以及寫小論文都有所提升。在一開始，尋找自己的主題以及研究方向困擾了我很長一段時間，這遠比開始著手進行寫程式還要困難許多。然而到了最後，看到自己所創作出來的作品，就覺得這一切的努力都值得了。

* 1. **未來展望**
     + 增加更多玩家模式
     + 新的遊戲規則設計
     + 繼續修改程式錯誤
     + 美化遊戲介面

1. **參考資料**
   1. **網路資訊**

Nim - Wikipedia, the free encyclopedia

<https://en.wikipedia.org/wiki/Nim>

Exclusive or - Wikipedia, the free encyclopedia

<https://en.wikipedia.org/wiki/Exclusive_or>

拈及其各種變形遊戲

<http://episte.math.ntu.edu.tw/articles/mm/mm_03_2_02/>

rykids

<http://rykids.blogspot.tw/>

* 1. **參考書籍**

Chris Roffey(2014). *Python: Next Steps*

Sakis Kasamapalis(2015.08). *Python設計模式深入探討*

Brett Slatkin(2015.08). *Effective Python*

張凱慶(2014.08). *Python入門指南*