

TABLE TENNIS GAME

WITH PYTHON



B04703068 李荐濂

B04703089 郭昕

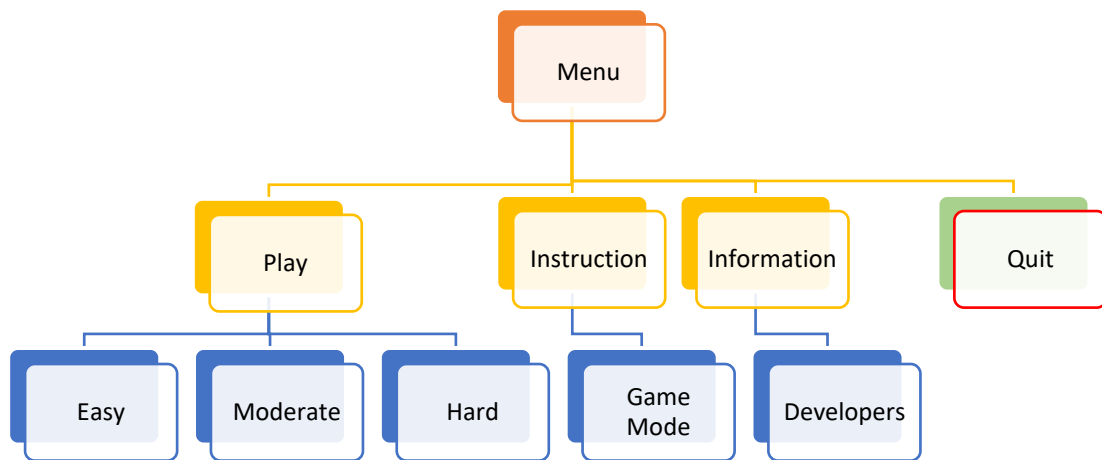
B04703108 劉庭華

B04703121 王聖揚

● 前言

這份期末報告，我們這組由四位財金系學生組成，選擇題目時曾討論過做財金相關的分析工具，既實用又能結合其他課學習到的知識，然而在尋找資料和相關程式碼的過程中，我們發現大部份我們所知的財務模型都早已有建立模組，因此能夠發揮的空間受到很大的限制，多次討論後，我們決定製作和自己興趣相關的題目-桌球遊戲，組員當中，有桌球校隊以及系隊的成員，此題目大家很了解並感到興趣。對於只會處理字串、list 以及寫簡單迴圈的初學者來說，要完成這支程式相當困難，但我們仍然決定接受這個挑戰，完成了以下的遊戲。

● 遊戲架構

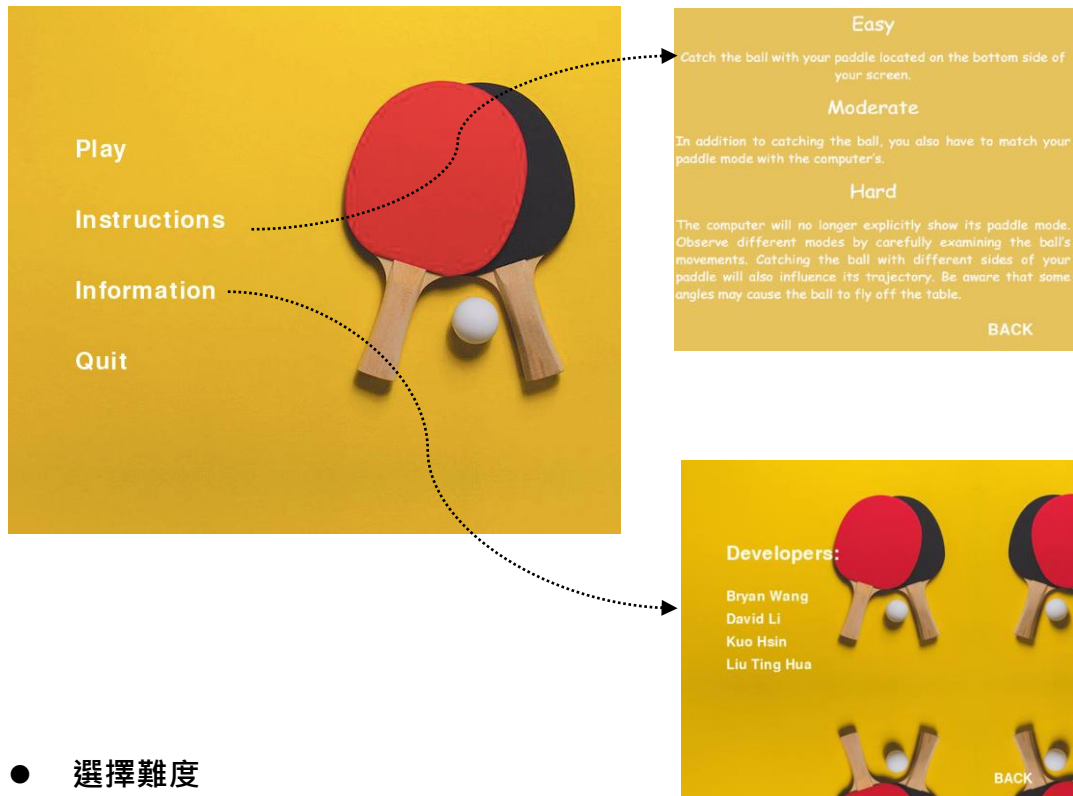


* 空白建為暫停功能

● 模組

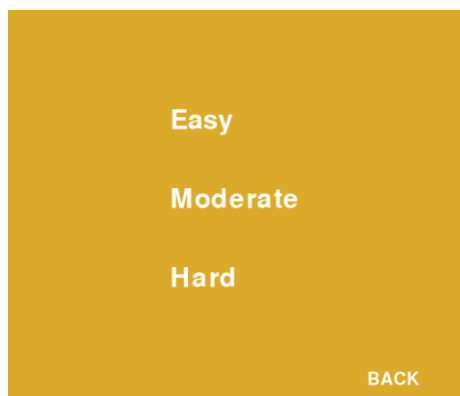
我們使用的模組有 pygame、os、random

● 目錄



● 選擇難度

點選「Play」之後，會出現以下畫面



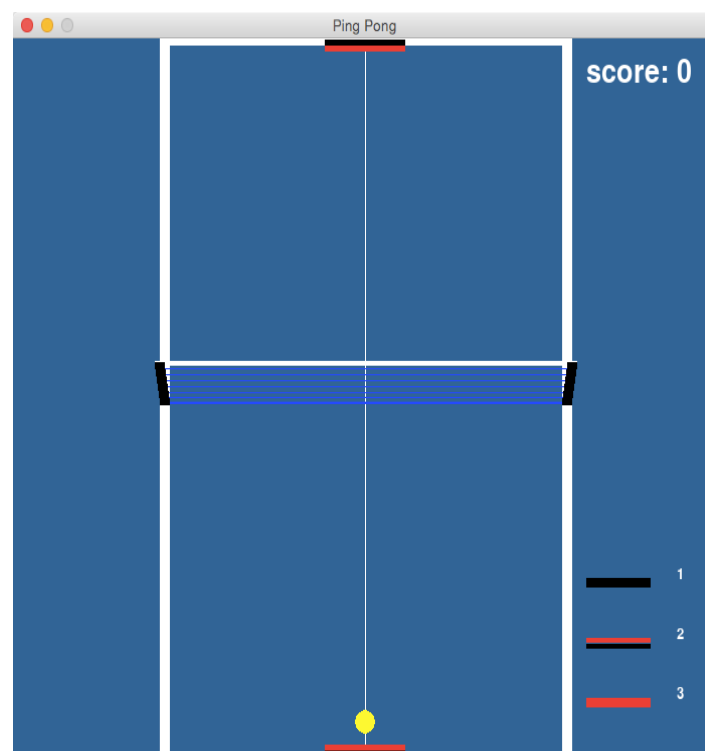
點選了不同的難度，就進入不同難度的遊戲，以下是「進入 easy 版」的程式碼

```
if 250 <= x <= 200 + text1.get_width() and 150 <= y <= 150 + text1.get_height():
    screen.blit(text1_1, (250, 150))
    screen.blit(text2, (250, 270))
    screen.blit(text3, (250, 390))
    screen.blit(text4, (550, 550))
    difficulty = 0
    if pygame.mouse.get_pressed()[0]:
        while game_end:
            while replay:
                run()
```

原本我們寫了三個不同的 `run()` 分別給三種難度，但我們發現因為基本的遊戲規則一樣，所以這三個 `run()` 有很多地方是一樣的，因此我們想了一個辦法模組化：加入 `difficulty`，在讀到不同 `difficulty` 的時候，決定哪些功能要運作。這使得我們的程式碼精簡很多。

● 進入遊戲

1. Easy



Return the ball as many times as possible.

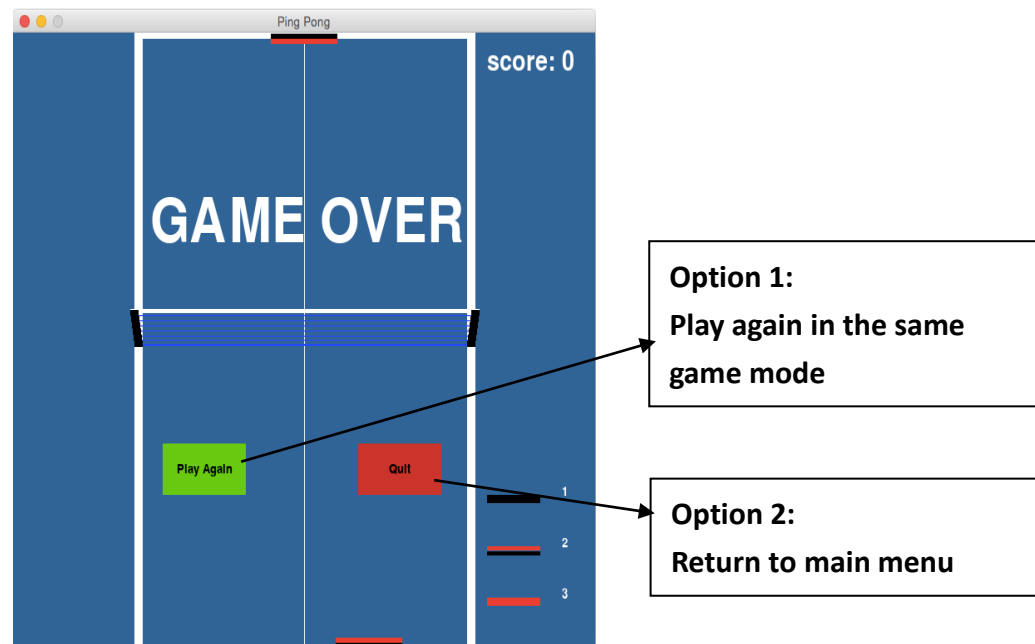
Direction of the hit will be random, but will always be inbound.

Speed Increases with every 5 hits.

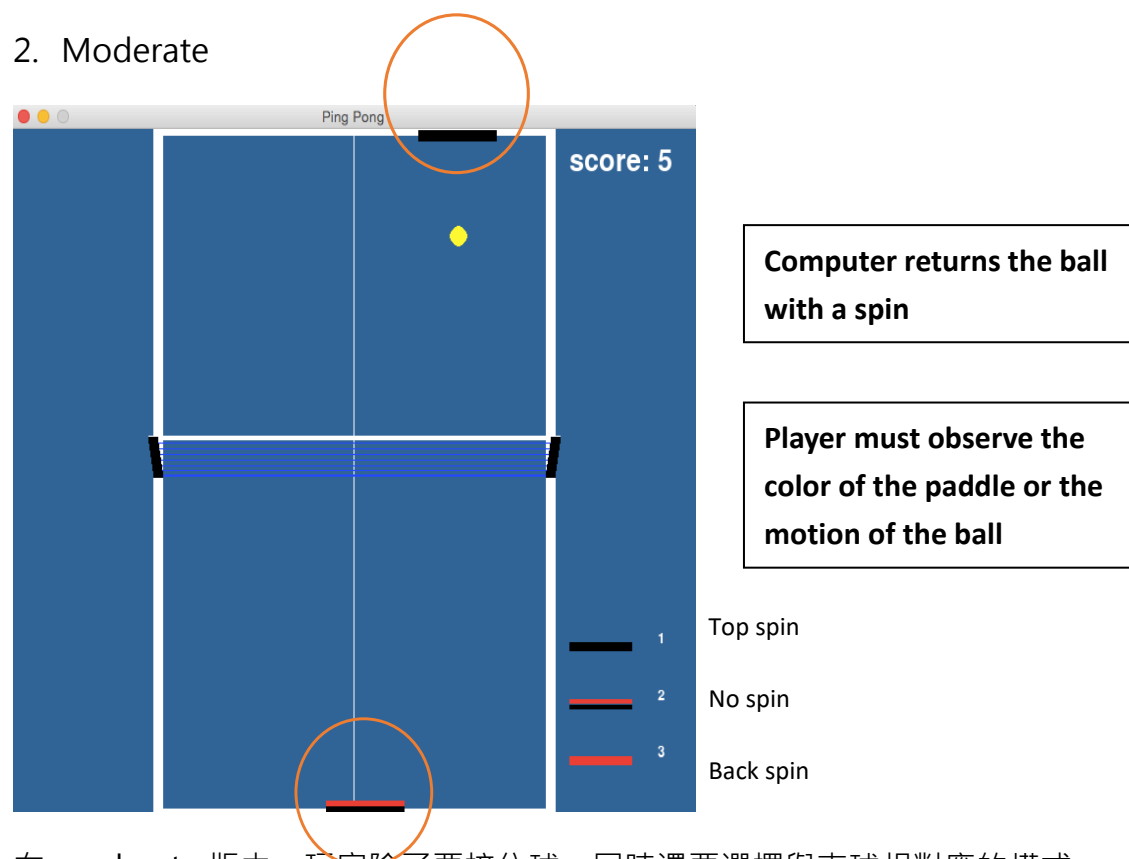
規則：

電腦方會隨機發出不同方向且不同類型的球。在 `easy` 版中，玩家僅需移動下方的板子使之對準來球，不需要考慮球的類型。若成功回擊，分數就會增加。另外，每成功回擊 5 次，球的速度就會加快。當玩家沒有成功回擊來球時，遊戲結束。並出現兩個選項：Play again 和 Quit。玩家可自行點選

Game over



2. Moderate



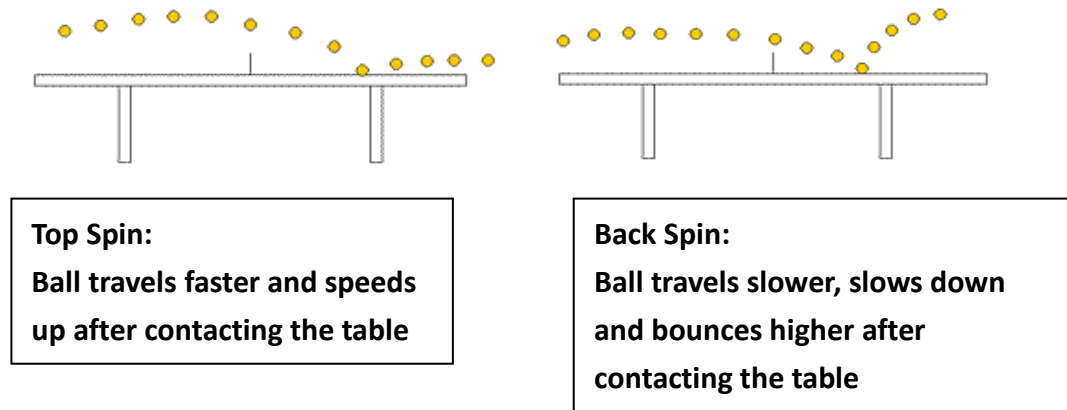
在 moderate 版中，玩家除了要接住球，同時還要選擇與來球相對應的模式。

在此版本中，玩家有兩個方法判斷來球模式：以球速變化判斷或者看電腦方提

供的提示。以下詳細介紹此兩種方法：

(1) 以球速判斷:

若球在第一次彈跳後加速，為上旋球；若減速，為下旋球；若不變，則為平球



(2) 看電腦提示: (較容易)

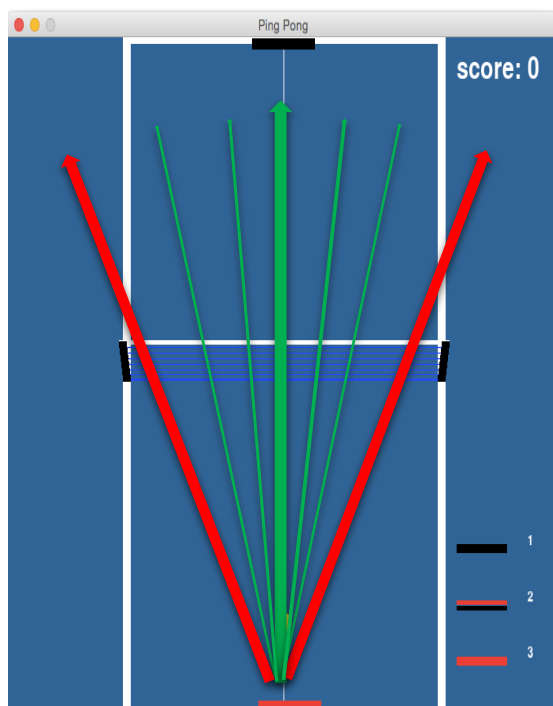
若為上旋球，電腦方的拍子會顯示黑色；若為下旋球，為紅色；若為平球，則為半黑半紅。

判斷好球的類型後，玩家要選擇回擊的模式: 上旋球按「1」、平球按「2」、下旋球按「3」。因此玩家如果是看電腦提示，只要選擇與電腦方球拍想同顏色的 mode，就能成功擊球。舉上圖為例，電腦方的拍面為黑色(上旋球)，引此玩家必須按「1」。若玩家如上圖選擇了錯誤的 mode，則遊戲結束。畫面上會顯示正確的 mode 和玩家選擇的錯誤 mode。(下頁圖)



Notice that the computer hits a top spin(1) but the player returns with no spin(2)

3. Hard

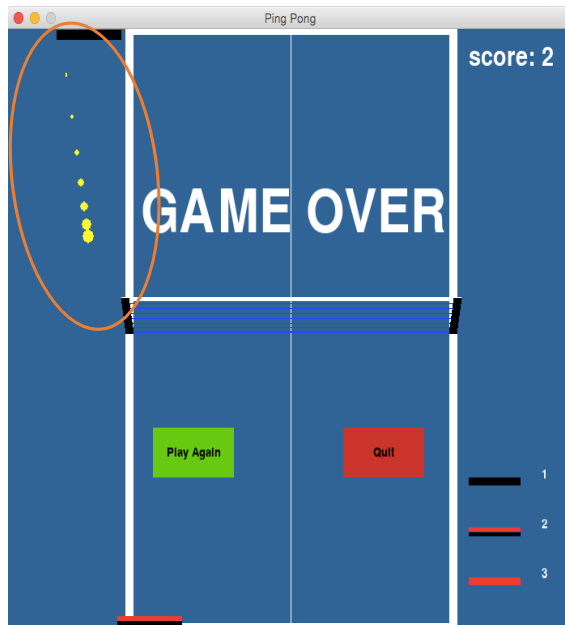


Computer no longer shows its spin mode with its paddle

Player must also decide the direction of the return

在 Hard 中，電腦方不再提供提示，因此玩家只能用球速變化來判斷球的類型。另外，玩家還須控制擊球方向，方法為：若球接觸到球拍的左方，球會往左彈；反之若接觸到右方，球會往右彈。因此若來球偏左，必須以球拍右方擊球，以確保打回去的球是在球桌範圍內。若回球超出球桌範圍，則遊戲結束，

畫面如下圖左:



Game over 畫面

```
def playerdxdy(ball, limit):
    segment = ball.x - limit
    if 0 <= segment and segment <= 10:
        dx = -3
    elif 10 <= segment and segment <= 20:
        dx = -2
    elif 20 <= segment and segment <= 30:
        dx = -1
    elif 30 <= segment and segment <= 50:
        dx = 0
    elif 50 <= segment and segment <= 60:
        dx = 1
    elif 60 <= segment and segment <= 70:
        dx = 2
    else:
        dx = 3
    dy = -6
    return dx, dy
```

以擊球部位控制方向程式碼

● 心得

郭昕

一開始對於整個遊戲該怎麼寫都沒什麼概念，從自己慢慢找資料開始到第一次做出球桌、會動的球跟球拍的時候真的很感動，最後功能逐步增加才變成現在這個版本，整個過程雖然很辛苦但也收穫很多，謝謝老師~~^^

王聖揚

這次的報告，程式我主要是調整遊戲中桌球桌的背景和處理球拍得部分，並製作口頭報告的簡報。寫程式的過程中，經常遇到繁雜的問題，例如在不同遊戲模式中球與球拍呈現出的畫面不夠真實，而我們會使用的工具不多，很多變數都需要手動反覆做調整，同時也使用了很多迴圈以及 if else statement，但在

和組員討論的過程中，發現當用不同邏輯去解決一個問題，能省下相當多的迴圈，讓程式更有效率也更方便閱讀。為了完成這支程式我們犧牲了很多時間和睡眠，但能夠靠自己所學的程式寫出一個複雜的遊戲帶給我很大的成就感，同時也讓自己的邏輯思考變得更清楚。期待未來能有機會能發揮這門課所學的程式技巧，並在履歷表中，程式語言能力-python 旁寫下 Excellent 這個字。

劉庭華:

在上這門課以前，我真的完完全全沒有接觸過程式設計，本身甚至還是一個思考不太縝密的人。因此在修課初期，經歷了很多身心靈的煎熬，常常作業一寫就花上了一整天，有時還寫不出來，時常覺得很沮喪。很慶幸我努力撐過了那段撞牆期，漸漸熟悉了寫程式的邏輯以後，寫程式變得順手很多(當然還是沒有很厲害啦)。而成就感最多的，莫過於這份期末報告，我們寫出了獨一無二的遊戲！

李荅濂

我認為這次期末報告讓我更深入了解到所謂 “ object-oriented programming” 的含義。這學期的課程無論是在寫作業或是考試，我們幾乎都是使用 procedural programming。雖然有在期末的課程中簡要介紹 class 的應用以及 object-oriented programming，但是是透過期末專案寫的 pygame 小遊戲我才真正了解到 OOP 的強大方便之處，也讓我熟悉一種截然不同的 coding 方式。