

投稿類別：工程技術類

篇名：

battle pong 遊戲製作與探討

作者：

李秉鴻。台北市立松山工農。電子科三年級智班

蘇駿杰。台北市立松山工農。電子科三年級智班

指導老師：

郭盈顯老師

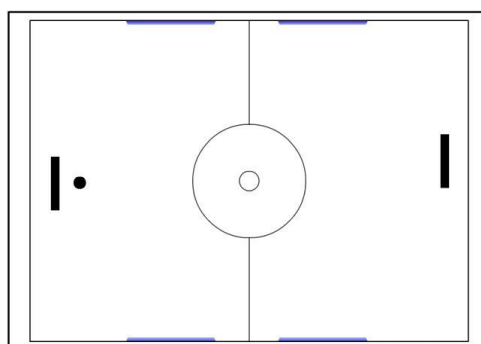
## 壹●前言

到了三年級，製作專題是每個高職學生所要面對的，而我們也對此苦惱著。我們開始回想過去所使用過和所學到的，想自己做出一個有意義的專題。到最後，想到的就是製作一個遊戲，因為遊戲帶給了許多人很多歡樂、刺激感、成就感等不同的反應，而遊戲也是從小到大陪伴著我們。以前是我們玩別人所製作的遊戲，這次，是我們想以自己目前所學來製作一個遊戲給別人玩，也是考驗自己對自身所學的有多深。以下是此專題的研究目的：

- 一、研究程式碼的亂數 (Rnd)、If 和 Select Case 的應用。
- 二、研究如何用鍵盤控制板子移動。
- 三、研究板子和球如何移動。
- 四、研究板子和上下牆壁在碰到球的時候如何反彈球。

## 貳●正文

我們想以 Visual Basic 製作遊戲 battle pong。而所謂的 battle pong 是指控制板子方向並擋住球，若球觸碰到板子後方牆壁則對方得一分，就和桌上曲棍球類似，如圖一。

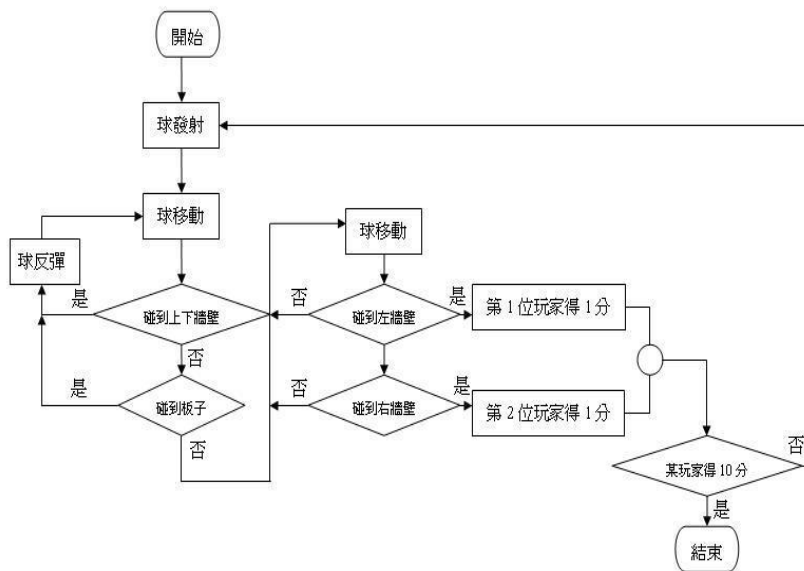


圖一 遊戲面板圖 (註一)

### 一、遊戲玩法

遊戲開始時球會在場中央隨機朝一方向發射出去，雙方玩家要用鍵盤控制自己的板子，當球碰到上下的牆壁或者是玩家的板子，就會反彈回去。如果第一位玩家 (左方玩家) 沒有用板子接到球而讓球碰到左邊牆壁

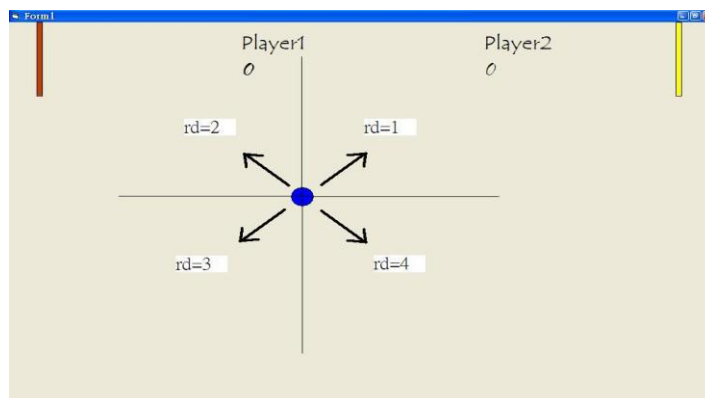
時，第二位玩家會得一分，反之亦然，得分後球會再從中央隨機發射出去。當其中一位玩家的分數達到 10 分時就獲勝，遊戲也結束了。如下圖為程式的流程圖：



圖二 程式流程圖

## 二、演算法

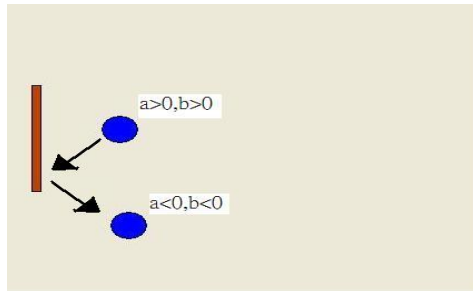
球會隨機向任一方向發射這點，我們想到用三個變數  $a$ 、 $b$ 、 $rd$ 。其中  $a$ 、 $b$  變數為球移動上下左右時的距離，再用  $rd$  決定發射方向，如圖三。



圖三 球發射方向示意圖

板和上下牆壁在碰到球的時候如何反彈球這點，只要變動  $a$ 、 $b$  變

數的正負值即可。假設球往左邊板子發射，原本 a 和 b 是正值，當碰到板子的時候，a 跟 b 變數變成負值，就會產生反彈的效果，如圖四。



圖四 板子反彈球示意圖

### 三、重要程式說明

(一) 球的移動方向和距離：

```

Randomize           //給 Rnd 不一樣的亂
                    數值
a = Int(Rnd * 50) + 50 //球的左右移動距離
b = Int(Rnd * 70) + 30 //球的上下移動距離
rd = Int(Rnd * 4)    //球的方向
Select Case rd
    Case 1          b = -b
    Case 2          a = -a
    Case 3          a = -a 、 b = -b
End Select
// 1 為右上、2 為左下、3 為左上
Shape1.Left = Shape1.Left + a
    
```

其中，a、b變數是球發射速度；rd變數是球的發射方向shape1是球；Int是取整數值；Rnd\*x是產生一個在0到x之間的亂數值。

Select Case rd這段就是當我的rd在下列哪個值的時候會執行什麼動作，以Case 1為例，當rd是1的時候我的b就變成負值。

Shape1.Left是球的x座標，Shape1.Top則是球的y座標。而

Shape1.Left = Shape1.Left + a的意思就是把球往右方移動a的值；  
Shape1.Top = Shape1.Top + b的意思就是把球往下方移動b的值，這樣球  
就會往各種方向移動了。（註二）

（二）板子移動（以第一位玩家為例）：

```
If KeyCode = 83 Then Shape3.Top =  
    Shape3.Top + 1600 //按 W 板子往上  
If KeyCode = 87 Then Shape3.Top =  
    Shape3.Top - 1600 //按 S 板子往下
```

其中，KeyCode = 83為鍵盤的W鍵；KeyCode= 87為鍵盤的S鍵；  
Shape3.Top為第一位玩家的板子(左邊板子)的上方位置座標。當第1  
位玩家按W之後板子就會往上移動，按S就會往下移動。

If動作1 Then動作2就好像英文一樣，如果我做了動作1，那程  
式就會執行動作2的功能。（註三）

（三）板子反彈（以第二位玩家為例）：

```
If Shape1.Left > 13900 And Shape1.Left < 14100 Then  
    If Shape1.Top - Shape2.Top > -460 And Shape1.Top -  
        Shape2.Top < 2000 Then    a = -a  
    end if  
end if  
//球碰到 2P 板子後反彈
```

其中，Shape1為球。Shape1.Left > 13900 And Shape1.Left < 14100  
和Shape1.Top - Shape2.Top > -460 And Shape1.Top -Shape2.Top < 2000  
這個是板子的判定區，當球碰到板子，也就是進入了板子的判定  
區，那球的左右方向就會相反，看起來就像反彈回去一樣。

（四）上下牆壁：

```
If Shape1.Top > 10600 Or Shape1.Top < 0 Then b = -b  
//球碰到上方或下方牆壁後反彈
```

跟上面板子反彈的原理很類似，Shape1.Top > 10600 Or Shape1.Top < 0是上下牆壁的判定線，如果球超過這條判定線，那上下方向就會相反，就像球被牆壁彈回去一樣。

#### 四、程式列表

```
初始設定：  
Randomize  
Timer1.Enabled = True  
Timer1.Interval = 10  
Shape1.Left = 7000  
Shape1.Top = 5200  
Shape2.Left = 14500  
Shape2.Top = 5200  
Shape3.Top = 5200  
Shape3.Left = 600  
Label1.Caption = 0  
Label2.Caption = "Player1"  
Label3.Caption = 0  
Label4.Caption = "Player2"  
Label5.Caption = ""  
Command1.Visible = False  
a = Int(Rnd * 50) + 50  
b = Int(Rnd * 70) + 30  
rd = Int(Rnd * 4)  
Select Case rd  
Case 1  
b = -b  
Case 2  
a = -a  
Case 3  
a = -a  
b = -b
```

```
End Select
```

```
c = 0
```

```
d = 0
```

```
f = 0
```

```
bg = 0
```

鍵盤移動板子：

```
If KeyCode = 87 Then Shape3.Top = Shape3.Top - 1600
```

```
If KeyCode = 83 Then Shape3.Top = Shape3.Top + 1600
```

```
If KeyCode = 38 Then Shape2.Top = Shape2.Top - 1600
```

```
If KeyCode = 40 Then Shape2.Top = Shape2.Top + 1600
```

上下牆壁反彈：

```
If Shape1.Top > 10600 Or Shape1.Top < 0 Then b = -b
```

玩家板子反彈(第 1 位玩家)：

```
If Shape1.Left > 400 And Shape1.Left < 600 Then
```

```
    If Shape1.Top - Shape2.Top > -460 And Shape1.Top - Shape2.Top < 2000 Then a = -a
```

```
    end if
```

```
end if
```

得分判定(第 1 位玩家)：

```
If Shape1.Left < -100 Then
```

```
a = 0
```

```
b = 0
```

```
c = 0
```

```
Select Case d
```

```
Case 150
```

```
Shape1.Left = 7000
```

```
Shape1.Top = 5200
```

```
Shape2.Top = 5200
```

```
Shape3.Top = 5200
```

```
a = Int(Rnd * 50) + 50
```

```
b = Int(Rnd * 70) + 30
```

```
rd = Int(Rnd * 4)
```

```
Select Case rd
```

```
Case 1
b = -b
Case 2
a = -a
Case 3
a = -a
b = -b
End Select
c = 0
d = 0
bg = 0
Label3.Caption = Label3.Caption + 1
End Select
d = d + 1
End If

勝利判定：
If Label1.Caption = 10 And f = 0 Then
Label5.Caption = "game over! player1 win!"
Command1.Visible = True
Shape1.Left = 7000
Shape1.Top = 5200
Shape2.Top = 5200
Shape3.Top = 5200
a = 0
b = 0
c = 0
bg = 0
f = 1
ElseIf Label3.Caption = 10 And f = 0 Then
Label5.Caption = "game over! player2 win!"
Command1.Visible = True
Shape1.Left = 7000
Shape1.Top = 5200
a = 0
b = 0
```

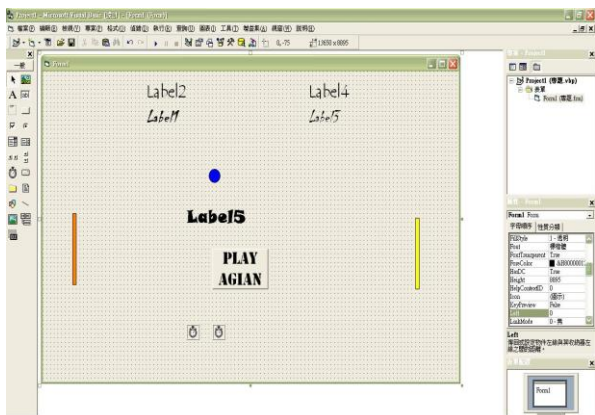


```
c = 0
bg = 0
f = 1
End If

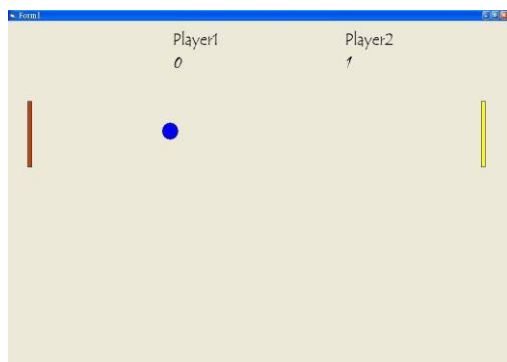
漸加速：
Select Case c
Case 10
If a > 0 Then
a = a + 1
ElseIf a = 0 Then
a = 0
Else
a = a - 1
End If
End Select
c = c + 1
If c > 10 Then c = 0

球移動：
Shape1.Left = Shape1.Left + a
Shape1.Top = Shape1.Top + b
```

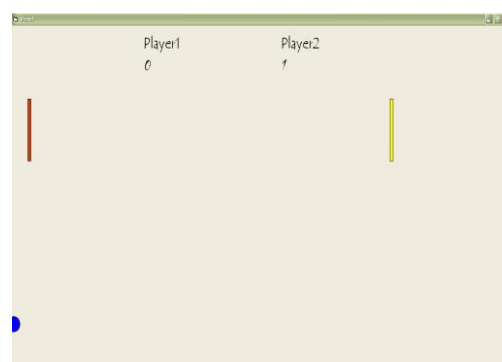
五、執行結果



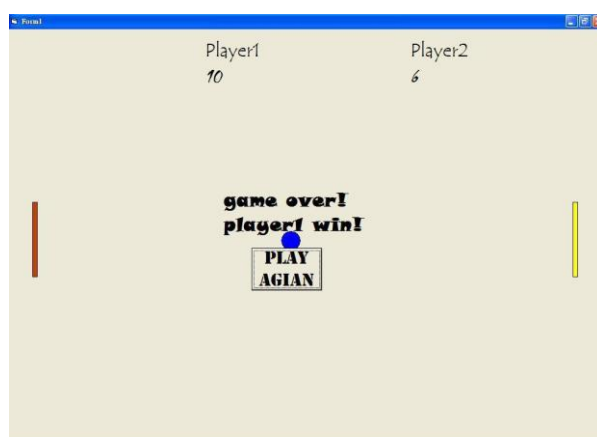
圖五 Form 配置圖



圖六 遊戲過程



圖七 得分



圖六 遊戲結束

## 參●結論

在做這個遊戲的時候，一開始以為應該都會跟自己想像的一樣，結果並沒有這麼順利。把球做好可以在視窗內移動之後，卻發現不知道怎麼讓球在碰到板子時反彈回去，甚至出現了因為程式碼順序顛倒的關係而產生的無敵狀態（不碰到板子也可以反彈）。

接下來就是球的速度和移動方向問題了。剛開始做只是想辦法讓球移動而已，卻沒有想到球的速度和方向，速度時快時慢，有時候還會出現瞬間得分的現象。在方向上，雖然做成了可以朝四面八方發射，但是有時候會出現球一直在場內上下移動而沒辦法讓板子擊球，後來請教老師之後，了解了變數的上下限，就改用兩個變數把方向跟速度都控制在正常範圍了。

上面兩個問題解決以後，基本上可以正常的玩遊戲了，不過仍有 BUG。像是球明明已經打到牆壁了，卻慢慢的回到場內，也就是「死後復活」。還有一個是球有被板子反彈，卻卡在板子內部出不去，這兩個 BUG 讓我們思考了很長的時間才找到解決辦法。「死後復活」的問題是因為速度的變數沒有歸零，導致球繼續移動；球卡在板子的問題則再設一個變數使球進入板子內時只反彈一次就行了。

雖然這遊戲看起來很簡單，但製作上卻超出自己的想像。拙劣的我們製作的遊戲還有很多很多沒有被開發出來的功能，像是摩擦力、球的增值、板子突然變長或變短等等。而此專題也可以做出無線踏板代替控制板子的上下鍵，這樣會增加遊戲時的趣味性。最後遊戲雖然仍不算完成，不過製作時的過程卻帶給我們十分有趣的經驗。

#### 肆●引註資料

- 一、遊戲 戰鬥桌上曲棍球 battle pong  
[http://mygamespocket.com/game\\_detail.php?game\\_index=867](http://mygamespocket.com/game_detail.php?game_index=867)
- 二、林信成主編(1999)。精通 Visual Basic 6.0 程式設計。台北市：第三波。
- 三、李春雄主編(2007)。Visual Basic 6.0 學習實務。台北縣：新文京。