

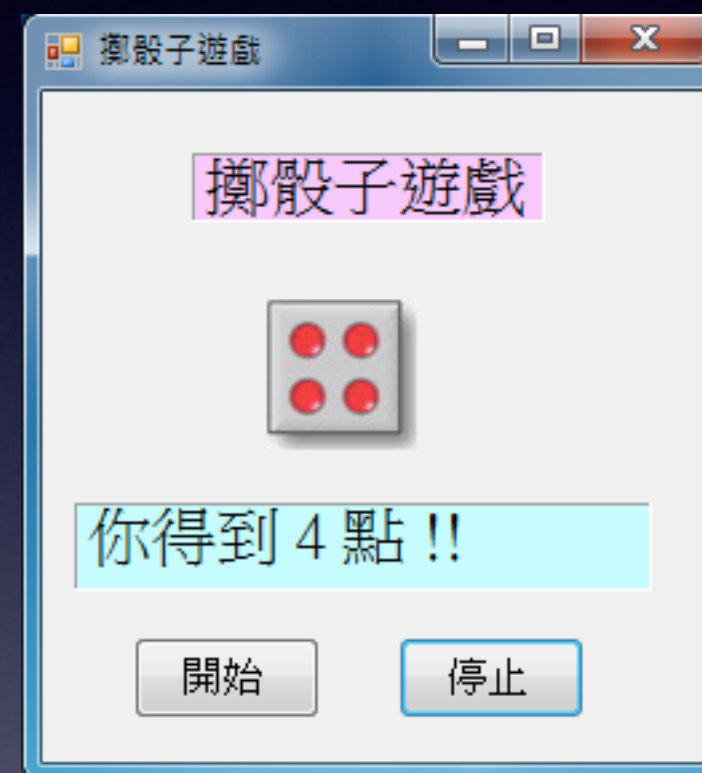
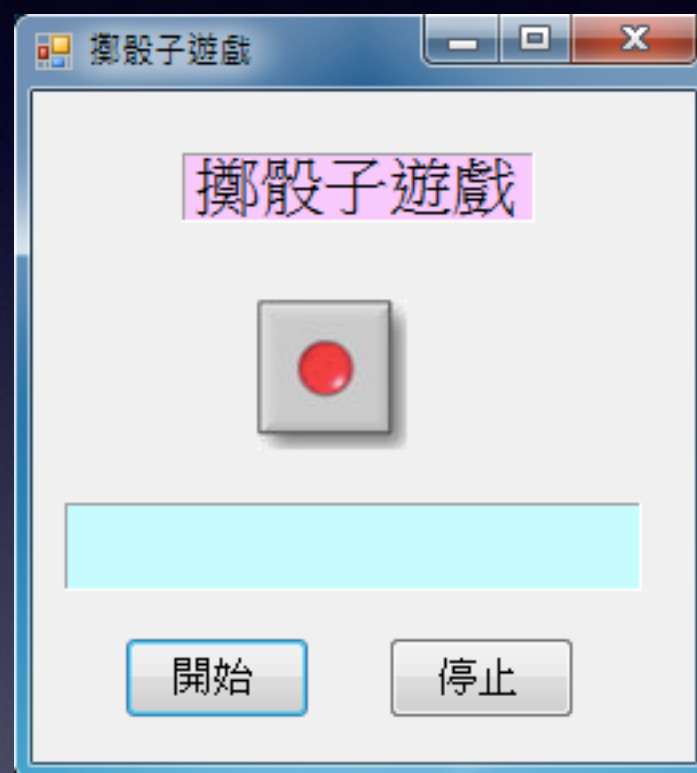
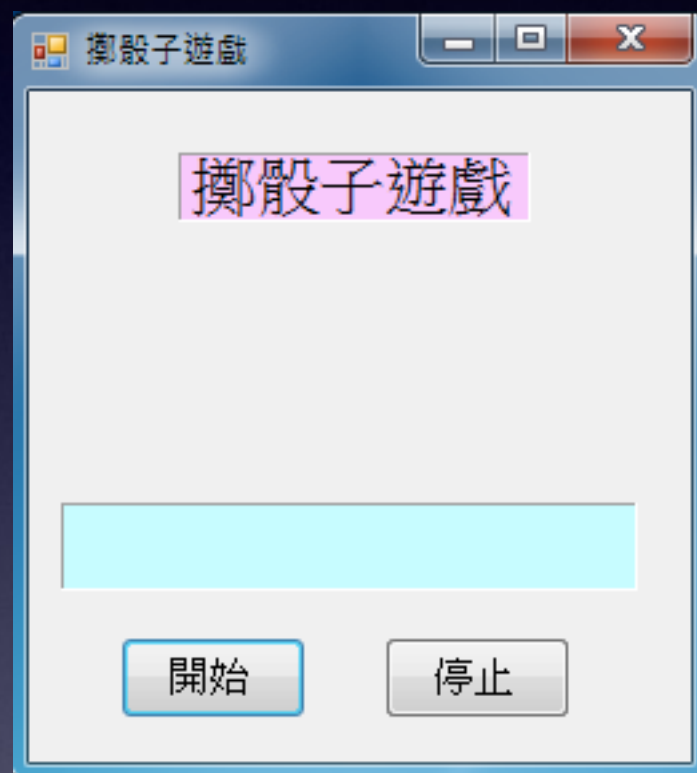
# 視窗程式設計

## 擲骰子遊戲

王昱景 Brian Wang  
[brian.wang@hkmci.com](mailto:brian.wang@hkmci.com)

# 功能需求

- 在表單上按“開始”鈕啟動計時器
- 計時器會以每0.05秒速度並以亂數的方式快速切換1.jpg~6.jpg的骰子圖
- 當按“停止”鈕後會將得到的點數及對應骰子圖1.jpg~6.jpg顯示出來，並告知“你得到幾點”



# 實作步驟

- 新增專案並以”dice1”為新專案名稱
- 由輸出要求必須在表單建立以下控制項
  - 一個圖片方塊控制項用來存放骰子圖，並將SizeMode屬性設為AutoSize
  - 兩個標籤控制項，一個用來顯示抬頭另一個顯示”你得到點數”



- 兩個按鈕控制項，分別為“開始”、“停止”鈕
- 一個計時器控制項，用來產生亂數 1.jpg~6.jpg 的骰子圖，該控制項的 Enabled 屬性預設值為 false

AutoScaleMode: Font  
Font: 11.25pt  
Size: 264, 289  
Text: 擲骰子遊戲

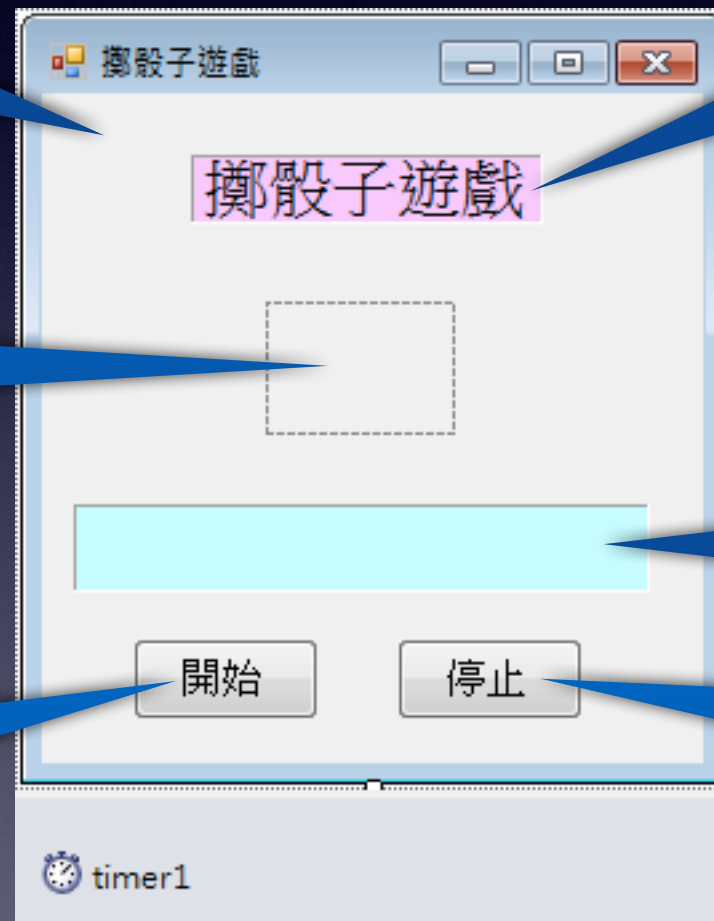
AutoSize: True  
BackColor: 255, 192, 255  
BorderStyle: Fixed3D  
Font: 18pt  
X,Y,W,L: 56, 23, 132, 26  
Text: 擲骰子遊戲

Name: pic1  
X,Y,W,L: 84, 78, 71, 50  
SizeMode: AutoSize

Name: labMsg  
BackColor: 192, 255, 255  
BorderStyle: Fixed3D  
Font: 18pt  
X,Y,W,L: 15, 157, 216, 33

Name: btnStart  
Font: 11.25pt  
X,Y,W,L: 34, 204, 70, 31  
Text: 開始

Name: btnStop  
Font: 11.25pt  
X,Y,W,L: 133, 204, 70, 31  
Text: 停止



- 將“**images**”資料夾下的**1.jpg~6.jpg**骰子圖檔複製到目前製作專案“**disc1\bin\Debug**”資料夾下
- 下列圖示為骰子的對應檔名



- 必須宣告一個`getpoint`整數變數用來存放得到的骰子點數

```
int getpoint; // 宣告getpoint用來存放得到的點數
```

- 此變數在計時器的`timer1_Tick`及“停止”鈕的`btnStop_Click`事件處理函式都會用到



- 每0.05秒執行timer1計時器控制項的timer1\_Tick事件處理函式一次
- 請在表單載入時執行的Form1\_Load事件處理函式內設定timer1的Interval屬性為50

```
// 表單載入時執行
private void Form1_Load(object sender, EventArgs e)
{
    timer1.Interval = 50; // 指定每50毫秒(即0.05秒)執行一次Timer1_Tick事件
}
```

- 在計時器的timer1\_Tick事件處理函式內設定以亂數方式產生1.jpg~6.jpg的骰子圖

```
// 每50豪秒(即0.05秒)執行一次timer1_Tick事件
private void timer1_Tick(object sender, EventArgs e)
{
    Random rnd = new Random();
    getpoint = rnd.Next(1, 7);
    pic1.Image = Image.FromFile(Convert.ToString(getpoint) + ".jpg");
}
```

- 在按下“開始”鈕執行的btnStart\_Click事件處理函式內將timer1的Enabled屬性值設為true，以便讓計時器啟動
- 此時即會每0.05秒執行timer1\_Tick事件處理函式一次，使表單上的pic1圖片方塊以亂數呈現1.jpg~6.jpg的骰子圖

```
// 按 [開始] 鈕執行
private void btnStart_Click(object sender, EventArgs e)
{
    timer1.Enabled = true; //啟動timer1計時器
}
```

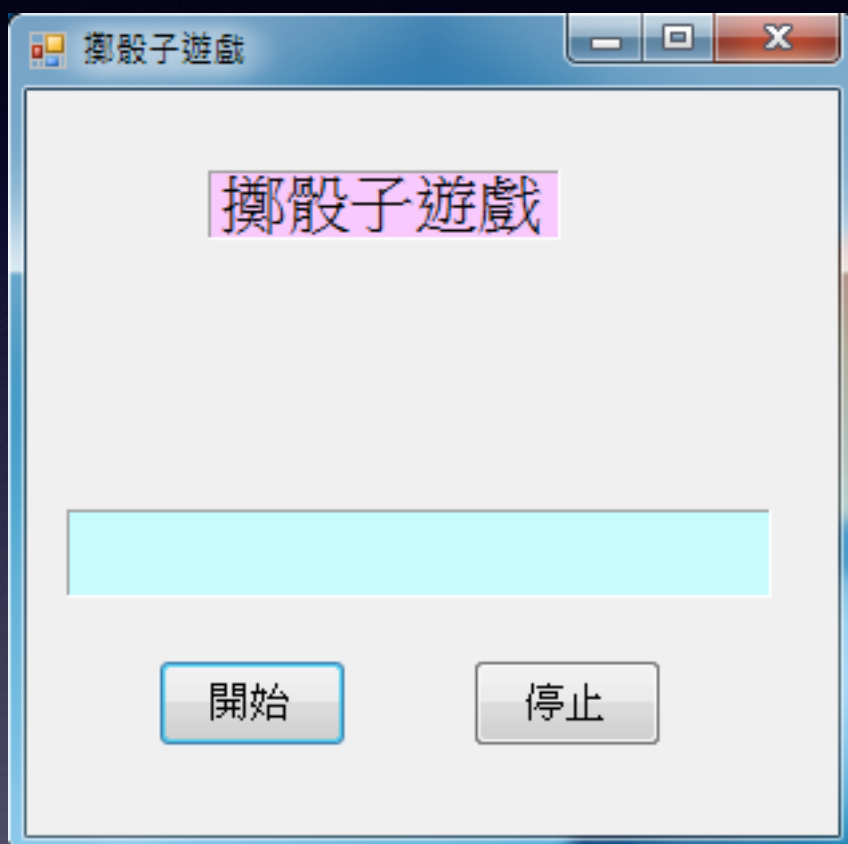
- 在按下“停止”鈕執行的btnStop\_Click事件處理函式內將timer1的Enabled屬性值設為false，讓計時器不啟動
- 並將所得到的骰子點數顯示在labMsg標籤上

```
// 按 [停止] 鈕執行
private void btnStop_Click(object sender, EventArgs e)
{
    timer1.Enabled = false; // 停止timer1計時器|
    labMsg.Text = "你得到 " + Convert.ToString(getpoint) + " 點 !!";
}
```



# 功能需求

- 延續上例，試以圖片方塊控制項陣列來存放骰子圖
- 當按”開始”鈕會以每0.05秒速度並以亂數的方式由1.jpg~6.jpg的骰子途中選取3張圖至於3個圖片方塊上
- 當按”停止”鈕後會將得到的3個骰子點數進行加總，最後告知”你得到幾點”



# 實作步驟

- 新增專案並以”**disc2**”為新專案名稱
- 由輸出要求必須在表單建立以下控制項
  - 使用標籤控制項顯示”**擲骰子遊戲**”標題
  - 建立**3個圖片方塊**控制項**pic1~pic3**用來顯示骰子的點數，並將**SizeMode**屬性設為**AutoSize**

- 建立名稱為**labMsg**標籤控制項用來顯示骰子的總點數
- 兩個按鈕控制項，分別為“**開始**”、“**停止**”鈕
- 一個**計時器**控制項，用來產生亂數1.jpg~6.jpg的骰子圖，該控制項的**Enabled**屬性預設值為**false**



AutoScaleMode: Font  
Font: 11.25pt  
Size: 320, 317  
Text: 擲骰子遊戲

Name: pic1  
X,Y,W,L: 24, 80, 71, 50  
SizeMode: AutoSize

Name: pic2  
X,Y,W,L: 113, 80, 71, 50  
SizeMode: AutoSize

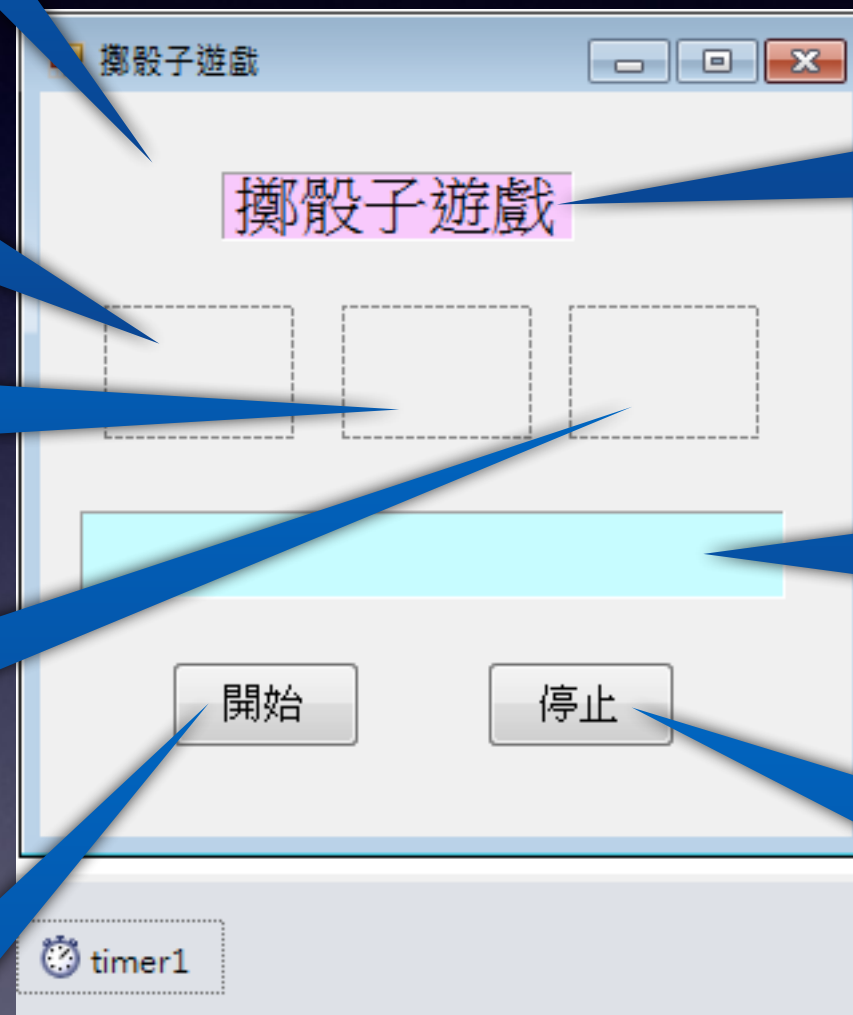
Name: pic3  
X,Y,W,L: 198, 80, 71, 50  
SizeMode: AutoSize

Name: btnStart  
Font: 11.25pt  
X,Y,W,L: 49, 213, 71, 33  
Text: 開始

AutoSize: True  
BackColor: 255, 192, 255  
BorderStyle: Fixed3D  
Font: 18pt  
X,Y,W,L: 68, 30, 132, 26  
Text: 擲骰子遊戲

Name: labMsg  
BackColor: 192, 255, 255  
BorderStyle: Fixed3D  
Font: 18pt  
X,Y,W,L: 15, 157, 264, 33

Name: btnStop  
Font: 11.25pt  
X,Y,W,L: 167, 213, 71, 33  
Text: 停止



- 將“**images**”資料夾下的**1.jpg~6.jpg**骰子圖檔複製到目前製作專案“**disc2\bin\Debug**”資料夾下
- 下列圖示為骰子的對應檔名



- 需產生3次1~6間的亂數當骰子的變數，分別在3個圖片方塊控制項上顯示對應的骰子圖
- 若應用上例方式產生亂數和載入骰子圖到圖片方塊控制項的程式碼需要連寫三遍
- 徒增加程式碼長度不易維護，可改用控制項陣列來解決此問題

- 建立 `getpoint[1]~getpoint[3]` 用來存放3個骰子的點數

```
int[] getpoint = new int[4];
```

- 建立 `arrPic[1]~arrPic[3]` 來設定 `pic1~pic3` 三個圖片方塊控制項顯示的圖片

```
PictureBox[] arrPic = new PictureBox[4];
```



- 宣告 **total** 變數存放 `getpoint[1]~getpoint[3]` 三骰子點數的加總

```
int total = 0;
```

- 由於上述變數在所有事件處理函式皆有參用，因此必須將上述變數宣告於所有事件處理函式之外

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();

        // 宣告arrPic為圖片控制項陣列，陣列元素arrPic[0]~arrPic[3]，但arrPic[0]省略不用
        PictureBox[] arrPic = new PictureBox[4];
        // 宣告getPoint陣列用來存放亂數值，陣列元素getpoint[0]~getpoint[3]，但getpoint[0]省略不用
        int[] getpoint = new int[4];
        int total = 0; // total整數變數用來存放三個骰子的總點數
    }
}
```

- 表單載入的Form1\_Load事件處理函式設定timer1每0.05秒觸動Tick事件一次
- 將pic1~pic3圖片方塊控制項指定給arrPic[1]~arrPic[3]的圖片方塊控制項陣列元素

```
// 表單載入時執行
private void Form1_Load(object sender, EventArgs e)
{
    timer1.Interval = 50; // 每0.05秒執行timer1_Tick事件處理函式一次
    arrPic[1] = pic1; // 控制項陣列的第一個元素即為pic1圖片控制項
    arrPic[2] = pic2;
    arrPic[3] = pic3;
}
```

- 在timer計時器的timer1\_Tick事件處理函式內設定以亂數方式產生1.jpg~6.jpg的骰子圖
- 將對應的骰子圖顯示於pic1~pic3圖片方塊控制項上

```
// 當timer1啟動時，每0.05秒會執行timer1_Tick事件處理函式一次
private void timer1_Tick(object sender, EventArgs e)
{
    total = 0;
    Random rnd = new Random();

    // 控制產生亂數的次數
    for (int i = 1; i <= arrPic.GetUpperBound(0); i++)
    {
        getpoint[i] = rnd.Next(1, 7); // 產生1~6 間之亂數
        total += getpoint[i]; // 產生的點數累加到total變數

        // 將指定圖檔置入控制項陣列在表單對應的圖片控制項
        arrPic[i].Image = Image.FromFile(Convert.ToString(getpoint[i]) + ".jpg");
    }
}
```



- 在按下“開始”鈕執行的btnStart\_Click事件處理函式內將timer1的Enabled的屬性值設為true
- 每0.05秒執行timer1\_Tick事件處理函式一次，使表單上的pic1 圖片方塊以亂數呈現1.jpg~6.jpg的骰子圖

```
// 按 [開始] 鈕執行
private void btnStart_Click(object sender, EventArgs e)
{
    timer1.Enabled = true;
}
```

- 在按下”停止”鈕執行的btnStop\_Click事件處理函式內將timer1的Enabled屬性值設為false
- 並將所得到的total骰子總點數顯示在labMsg標籤上

```
// 按 [停止] 鈕執行
private void btnStop_Click(object sender, EventArgs e)
{
    timer1.Enabled = false;
    lblMsg.Text = "共得到 " + Convert.ToString(total) + " 點 !!";
}
```