

視窗程式設計

撲克牌比大小遊戲

王昱景 Brian Wang
brian.wang@hkmci.com

功能需求

- 先按“洗牌”鈕電腦亂數洗牌，會由紅桃1~13中挑出兩張牌，置入表單中的圖片方塊上面
- 此時撲克牌是蓋牌狀態，待你在任一張撲克牌上按一下選取，電腦會比較你所選的牌是否比位選取的牌大？

Form1

撲克牌比大小遊戲



請按洗牌鈕遊戲

Form1

撲克牌比大小遊戲



你贏了！按洗牌

Form1

撲克牌比大小遊戲



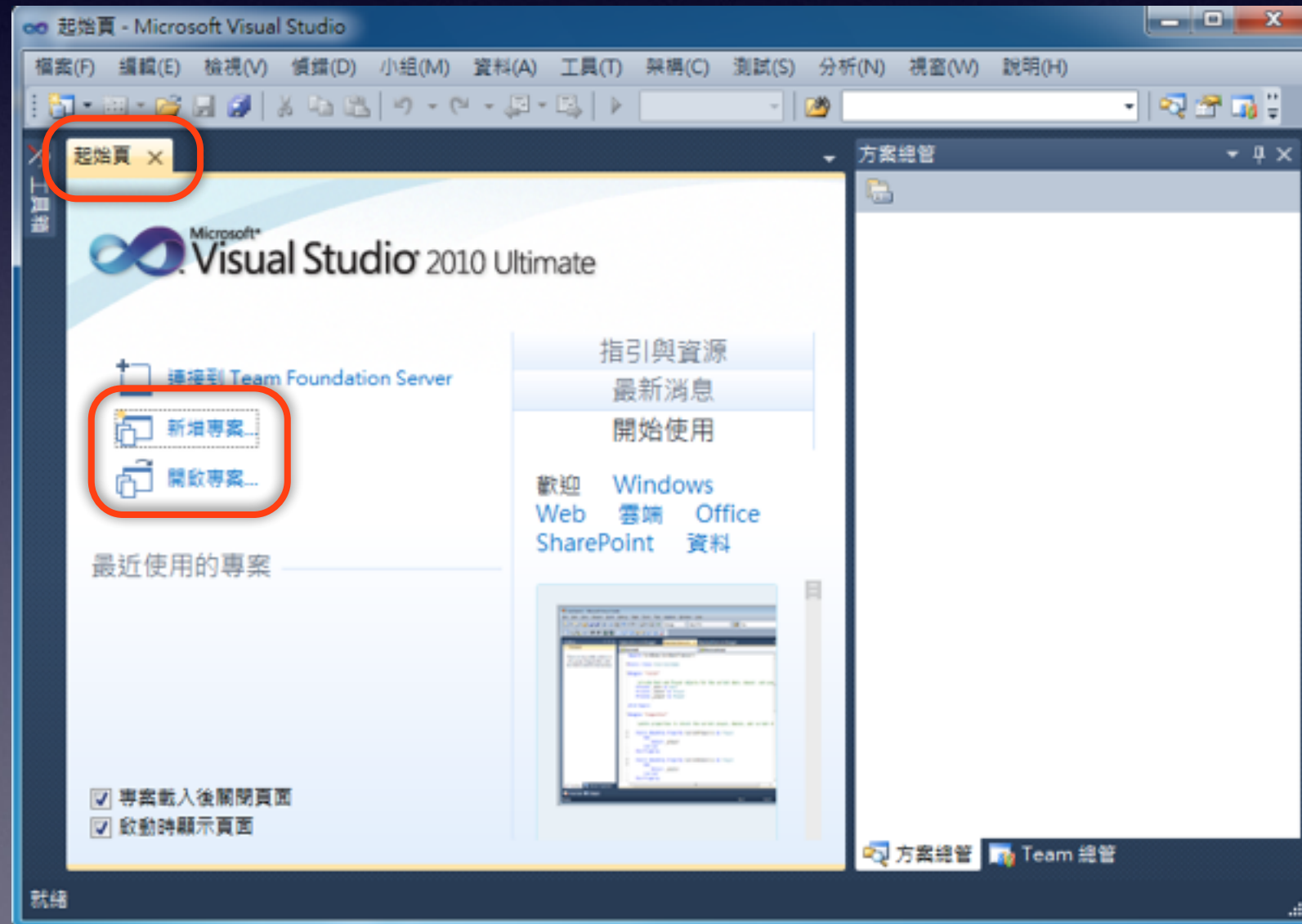
你輸了！按洗牌

- 假設撲克牌大小依序：
 $2 < 3 \dots 9 < 10 < J < Q < K < A$
- 若所選取的牌大，會顯示“你贏了！按洗牌鈕重玩...”
- 若所選取的牌小，會顯示“你輸了！按洗牌鈕重玩...”
- 若所選兩張牌大小相等，會顯示“平手！按洗牌鈕重玩...”

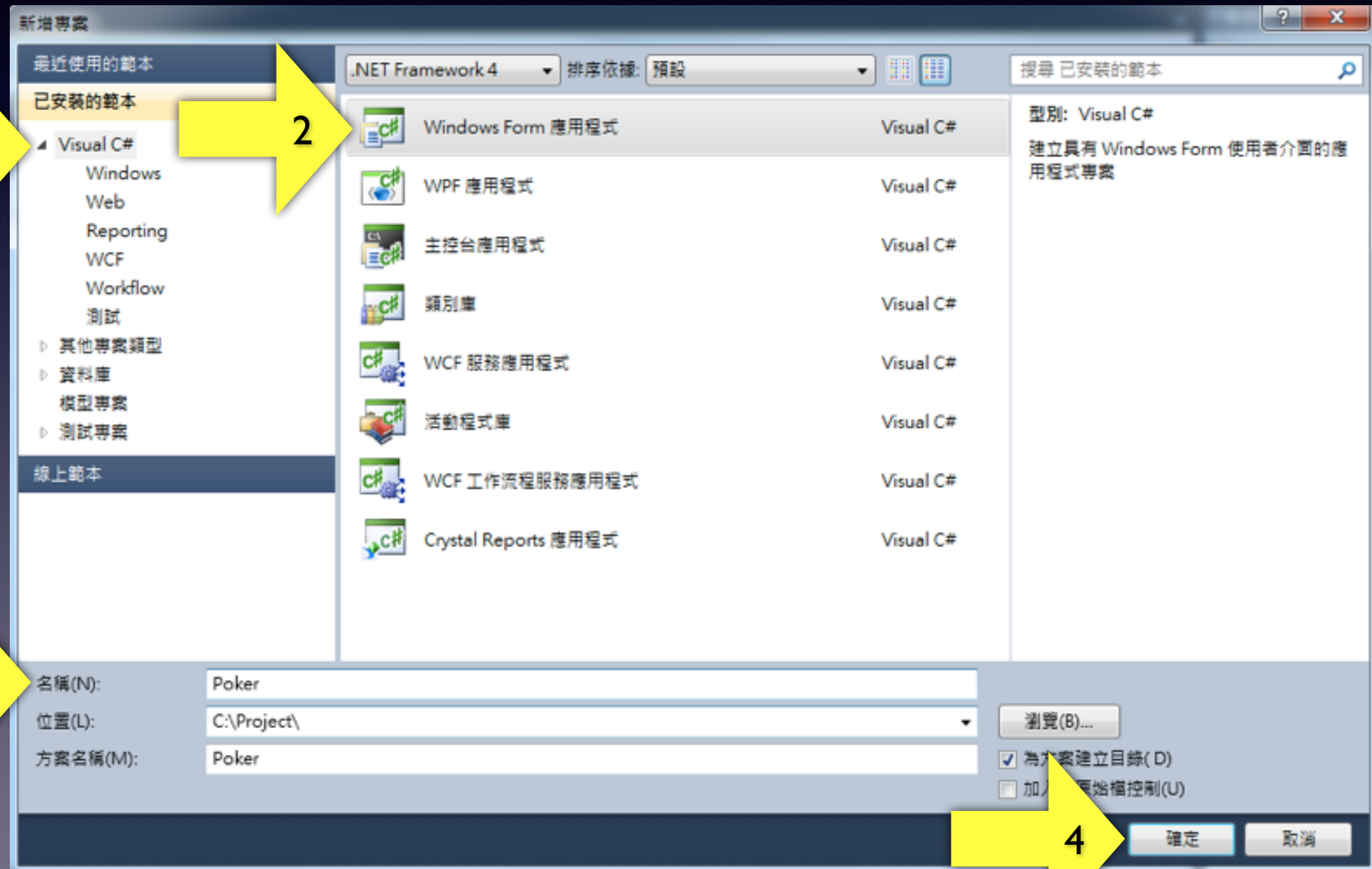
實作步驟

- 啟動Microsoft Visual Studio 2010，在工作列上執行【開始>所有程式>Microsoft Visual Studio 2010>Microsoft Visual Studio 2010】指令

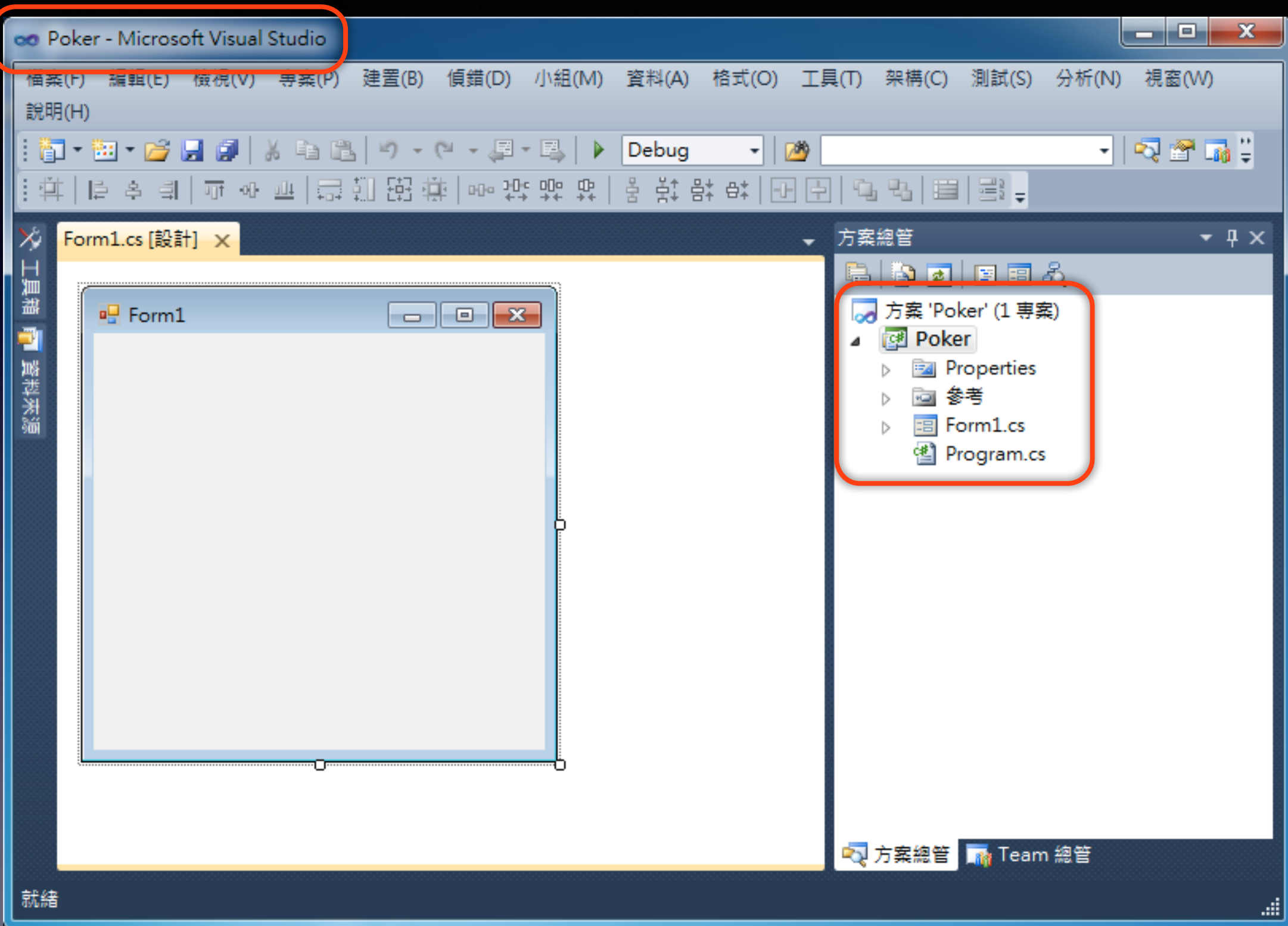
- 進入Microsoft Visual Studio 2010的整合開發環境前會先出現**起始頁**畫面
- 該畫面是不需使用功能表便可直接建立新的專案或開啟已經建好的舊專案



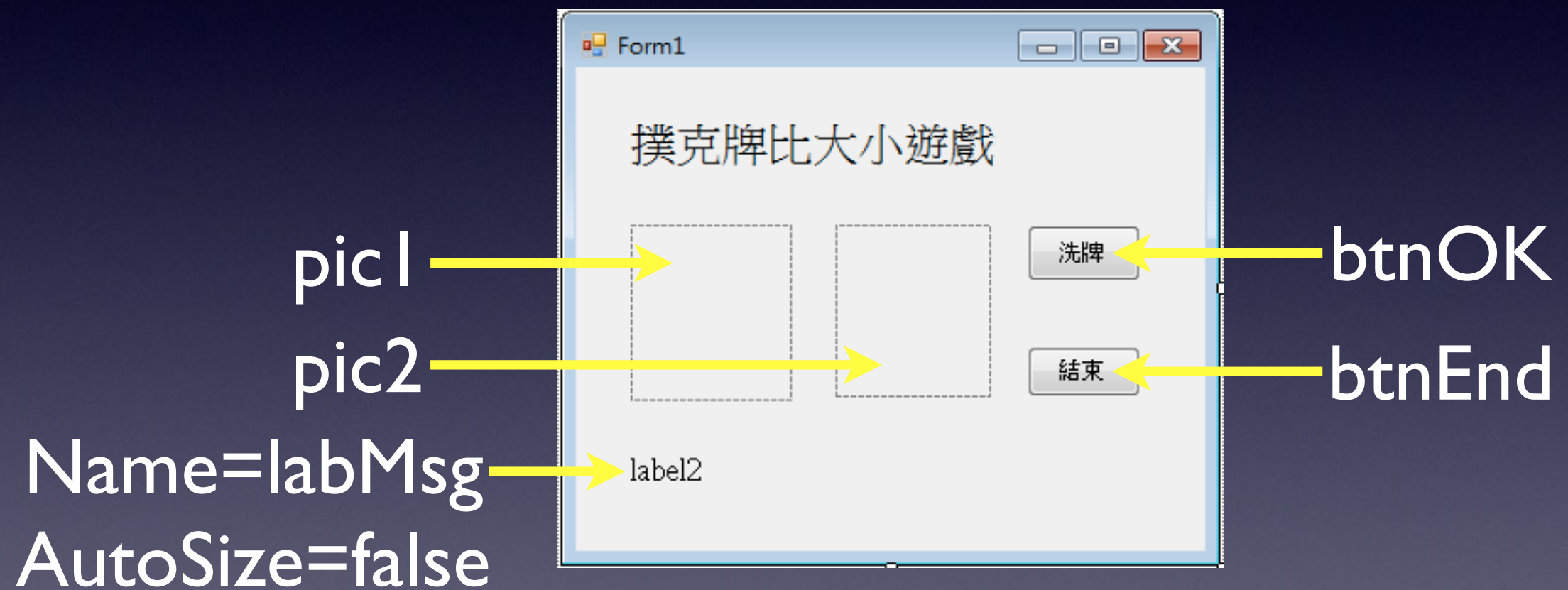
- 點選左方的“新增專案..”
- 開啟下圖“新增專案”對話方塊



- 選取“Visual C#”圖示
- 選取“Windows Form應用程式”圖示
- 在“名稱(N)”欄輸入“Poker”專案名稱
- 在“位置(L)”欄選擇“C:\Project”路徑
- 按確定鈕，進入Microsoft Visual Studio 2010的整合開發環境



- 建立下面控制項以及設定相關屬性

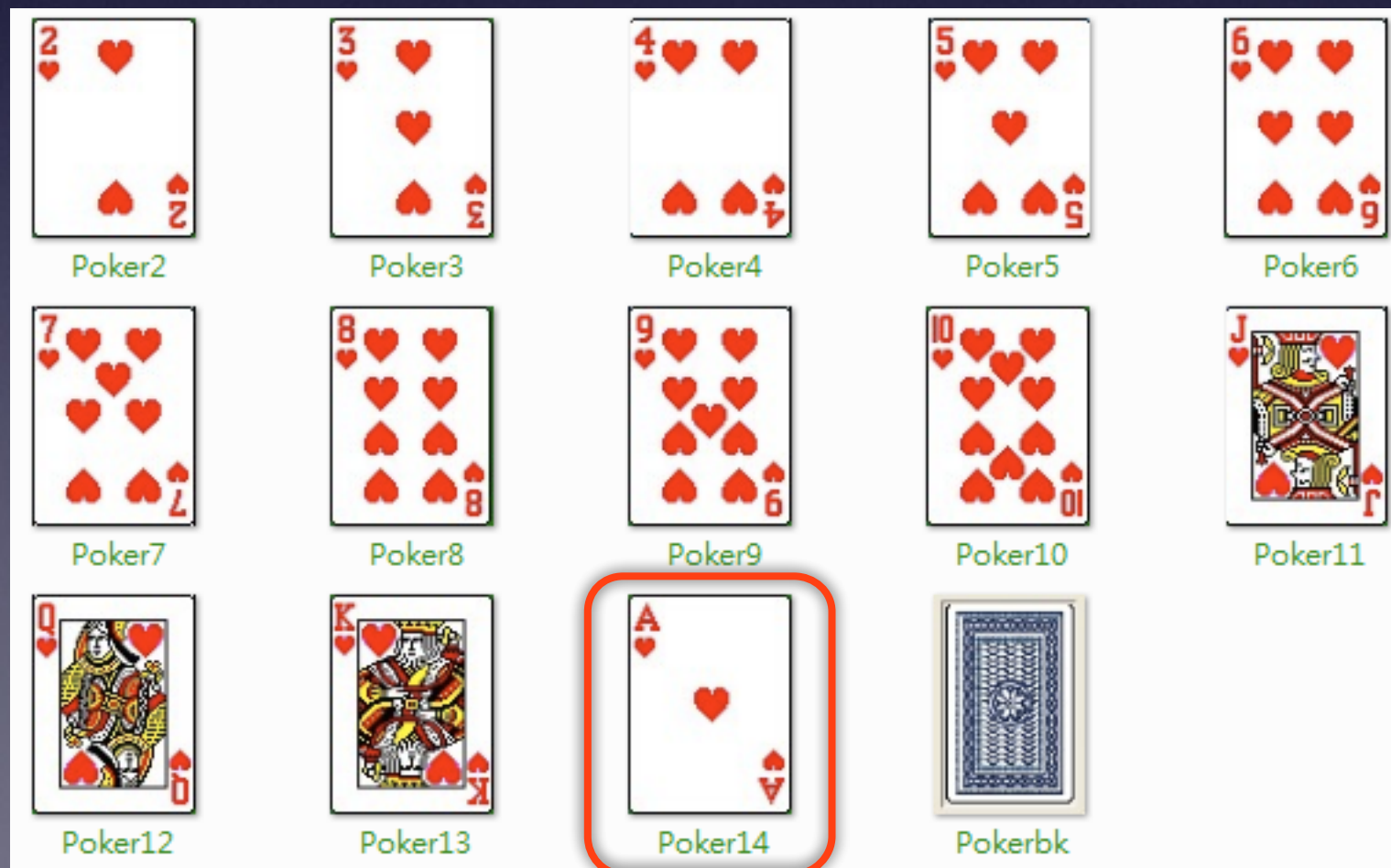


- image資料夾下有下列撲克牌紅桃1~13的圖檔，還有一張撲克牌背面圖檔 (Pokerbk.jpg)



- 為方便程式中能載入對應的圖檔，先將這些圖檔拷貝到目前專案資料夾下的“bin\Debug”資料夾，讓所有的圖檔與專案的執行檔都放在相同路徑下

- 由於是13張紅桃中，紅桃A是最大，所以產生的亂數2表紅桃2、亂數3表紅桃3、...、亂數13表紅桃K、亂數14表紅桃A
- 亂數比大小時，紅桃A才能大於紅桃K
- 所以必須將紅桃A的圖檔設為Poker14.jpg



- 宣告num1及num2整數變數用來表示所選擇的撲克牌

```
int num1, num2; // num1為我方的數值、num2為電腦的數值
```

- 當選擇第一張牌，即代表選擇的是num1，電腦是num2；當選擇第二張牌，即代表選擇的是num2，電腦是num1
- 由於num1與num2在程式中有多處參用到，因此必須宣告在所有事件處理函式外面，以方便事件處理函式彼此參用

- 表單載入時會觸動Form1_Load事件處理函式，必須在此處做一些屬性的初值設定
- 為避免未洗牌前撲克牌被點選，而觸動Click事件，表單載入時，先將pic1和pic2圖片方塊控制項設成無法選取

```
pic1.Enabled = false;  
pic2.Enabled = false;
```

- 將圖片自動調整成和圖片方塊控制項一樣大小，以免超出表單，將pic1及pic2的 **SizeMode** 屬性設成 **StretchImage**

```
pic1.SizeMode = PictureBoxSizeMode.StretchImage;  
pic2.SizeMode = PictureBoxSizeMode.StretchImage;
```

- 開始時需蓋牌，所以先將pic1及pic2圖片方塊的 **Image** 屬性設為 **Pokerbk.jpg**，使pic1及pic2圖片方塊呈現Pokerbk.jpg

```
pic1.Image = new Bitmap("Pokerbk.jpg");  
pic2.Image = new Bitmap("Pokerbk.jpg");
```

- 在**labMsg**標籤控制項顯示”請按洗牌鈕遊戲開始...”

```
labMsg.Text = "請按洗牌鈕遊戲開始..."
```

```
// 表單載入時執行
private void Form1_Load(object sender, EventArgs e)
{
    pic1.Enabled = false;
    pic2.Enabled = false;
    pic1.SizeMode = PictureBoxSizeMode.StretchImage;
    pic2.SizeMode = PictureBoxSizeMode.StretchImage;
    pic1.Image = new Bitmap("Pokerbk.jpg");
    pic2.Image = new Bitmap("Pokerbk.jpg");
    labMsg.Text = "請按洗牌鈕遊戲開始..."
}
```


- 當按洗牌(btnOK)鈕即執行btnOK_Click事件處理函式，在此函式需完成下列動作
- 依程式要求，設定洗牌按鈕失效，設定pic1及pic2圖片方塊控制項允許被選取，表示允許觸發該控制項的Click事件

```
pic1.Enabled = true;  
pic2.Enabled = true;  
btnOK.Enabled = false;
```

- 此時未選牌前必須蓋牌，必須在pic1及pic2圖片方塊控制項顯示撲克牌蓋牌圖檔(Pokerbk.jpg)

```
pic1.Image = new Bitmap("Pokerbk.jpg");  
pic2.Image = new Bitmap("Pokerbk.jpg");
```

- 使用Random類別產生2~14間的亂數分別置入num1整數變數(代表第一張牌數字)和num2整數變數(代表第二張牌數字)

```
Random r = new Random();  
num1 = r.Next(2, 15); // 給第一張牌產生2~14的正整數  
num2 = r.Next(2, 15); // 給第二張牌產生2~14的正整數
```

- 使labMsg標籤控制項顯示“請在撲克牌上按一下選取...”

```
labMsg.Text = "請在撲克牌上按一下選取...";
```

```
// 按洗牌鈕執行
```

```
private void btnOK_Click(object sender, EventArgs e)
{
    pic1.Enabled = true;
    pic2.Enabled = true;
    btnOK.Enabled = false;
    pic1.Image = new Bitmap("Pokerbk.jpg");
    pic2.Image = new Bitmap("Pokerbk.jpg");
    Random r = new Random();
    num1 = r.Next(2, 15); // 給第一張牌產生2~14的正整數
    num2 = r.Next(2, 15); // 給第二張牌產生2~14的正整數
    labMsg.Text = "請在撲克牌上按一下選取...";
}
```

- 當在表單上點選其中一張牌時，程式如何知道點選哪張牌？
- 如何判斷你點選的牌較大、較小或相同？

- 當點選第一張牌(若num1為6即表示紅桃6)，其對應的圖檔為Poker6.jpg
- 會執行pic1_Click事件處理函式，電腦認定選第一張牌，在此事件處理函式內將對應撲克牌圖檔顯示在圖片方塊上，並對兩張撲克牌(num1與num2)的大小做比較

- num1 設為第一張牌的大小即點選的撲克牌
- 先以字串合併方式取出num1對應的撲克牌圖檔，接著使用new Bitmap將它載入pic1 圖片方塊上面顯示

```
pic1.Image = new Bitmap("Poker" + Convert.ToString(num1) + ".jpg");
```

- 同樣方式，以num2所對應的撲克牌圖檔在入到pic2圖片方塊控制項上面

```
pic2.Image = new Bitmap("Poker" + Convert.ToString(num2) + ".jpg");
```

- 使用巢狀if敘述判斷num1是否大於、小於或等於num2
 - 若 $\text{num1} > \text{num2}$ ，lblMsg控制項顯示”你贏了！按洗牌鈕重玩...”
 - 若 $\text{num1} = \text{num2}$ ，lblMsg控制項顯示”平手！按洗牌鈕重玩...”
 - 若 $\text{num1} < \text{num2}$ ，lblMsg控制項顯示”你輸了！按洗牌鈕重玩...”

```
if (num1 > num2)
{
    labMsg.Text = "你贏了！按洗牌鈕重玩...";
}
else
{
    if (num1 == num2)
    {
        labMsg.Text = "平手！按洗牌鈕重玩...";
    }
    else
    {
        labMsg.Text = "你輸了！按洗牌鈕重玩...";
    }
}
```


- 最後將**pic1** 和**pic2**兩個圖片方塊控制項的**Enabled**屬性設為false(失效)，**btnOK**洗牌鈕的Enabled屬性設為true(有效)

```
pic1.Enabled = false;  
pic2.Enabled = false;  
btnOK.Enabled = true;
```

```
// 按pic1執行該圖片方塊的Click事件處理函式
private void pic1_Click(object sender, EventArgs e)
{
    pic1.Image = new Bitmap("Poker" + Convert.ToString(num1) + ".jpg");
    pic2.Image = new Bitmap("Poker" + Convert.ToString(num2) + ".jpg");

    if (num1 > num2)
    {
        labMsg.Text = "你贏了！按洗牌鈕重玩...";
    }
    else
    {
        if (num1 == num2)
        {
            labMsg.Text = "平手！按洗牌鈕重玩...";
        }
        else
        {
            labMsg.Text = "你輸了！按洗牌鈕重玩...";
        }
    }

    pic1.Enabled = false;
    pic2.Enabled = false;
    btnOK.Enabled = true;
}
```

- 若點選第二張牌(即pic2圖片方塊控制項)會觸動pic2_Click事件處理函式
- 電腦便認為選第二張牌，在此事件處理函式內做兩張撲克牌大小的比較
- 和選第一張牌的作法步驟相同，只是num2變成所選的牌，比較時是以num2為準

```
// 按pic2執行該圖片方塊的Click事件處理函式
private void pic2_Click(object sender, EventArgs e)
{
    pic1.Image = new Bitmap("Poker" + Convert.ToString(num1) + ".jpg");
    pic2.Image = new Bitmap("Poker" + Convert.ToString(num2) + ".jpg");

    if (num2 > num1)
    {
        labMsg.Text = "你贏了！按洗牌鈕重玩...";
    }
    else
    {
        if (num2 == num1)
        {
            labMsg.Text = "平手！按洗牌鈕重玩...";
        }
        else
        {
            labMsg.Text = "你輸了！按洗牌鈕重玩...";
        }
    }

    pic1.Enabled = false;
    pic2.Enabled = false;
    btnOK.Enabled = true;
}
```



```
// 按結束鈕執行  
private void btnEnd_Click(object sender, EventArgs e)  
{  
    Application.Exit();  
}
```