

第十三章 繪圖與多媒體



13.1 座標和顏色的設定

13.4 圖檔的讀取和儲存

13.2 繪圖物件

13.5 音效與多媒體播放

13.3 常用的繪圖方法

備註：可依進度點選小節



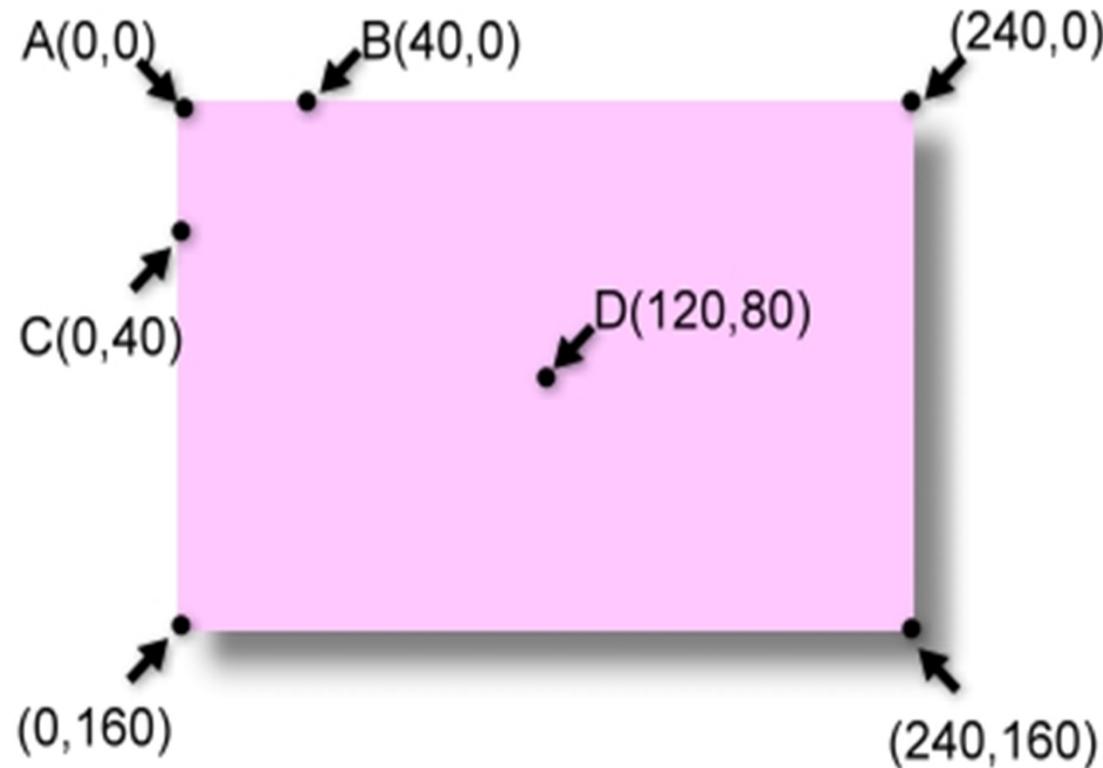
13.1 座標和顏色設定

13.1.1 座標

- 座標系統是以像素（Pixel）為單位。
- 像素是指螢幕上一個點，每個像素都有一個座標點與之對應。
- C# 將螢幕左上角的座標設為 $(0, 0)$ ，向右為正，向下為正。
- Visual C# 2008 除可在表單上繪圖，也可指定控制項當作畫布。
- 一般以 (x, y) 代表畫布上某個像素的座標點，水平以 X-座標值表示，垂直以 y-座標值表示。



下圖是寬度為 240、高度 160 畫布，畫布上 A(0,0) ,
B(40,0) , C(0,40) 、D(120,80) 為畫布四個頂點座標點
的相對位置。





13.1.2 顏色設定

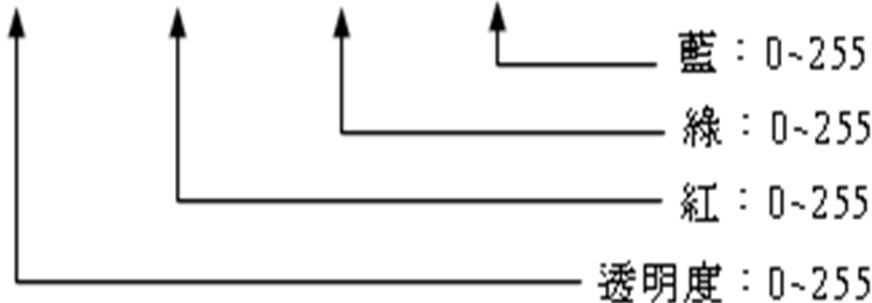
- 在 Visual C# 2008 中對於顏色設定的方法常用的有：
 - ① **FromArgb**
 - ② **FromKnownColor**
 - ③ **FromHtml**
 - ④ **Color.顏色常值**
- 每種方法都各有其優點。



1. 使用 FromArgb 設定顏色

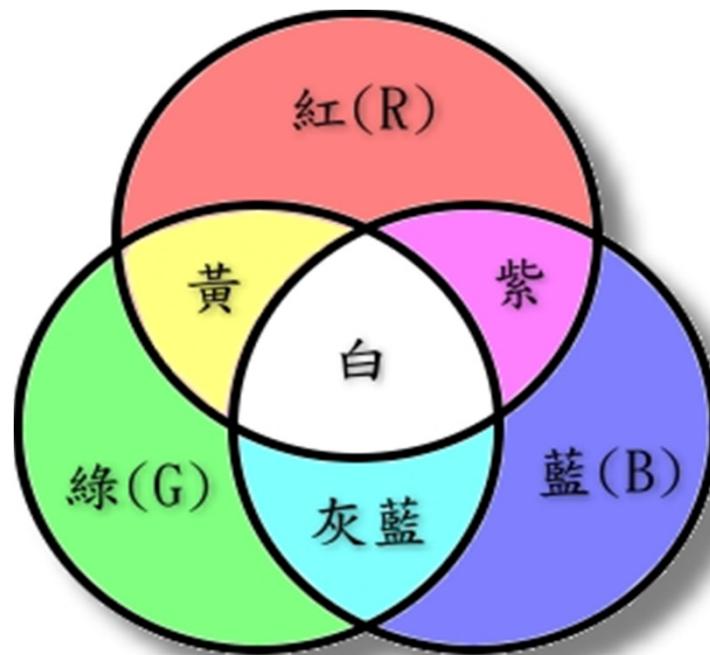
- FromArgb方法共有四個引數，分別代表：透明度(Alpha)紅（Red）、綠（Green）、藍（Blue）顏色光的強度。
- 每個值分別從 0~255 共分成 256 個強度。
- 數值越大表示該顏色光越強。
- Alpha 透明度引數可省略，值為 0 代表完全透明，255代表不透明。不同引數值調出各種顏色光。
- 語法：

語法：Color.FromArgb([Alpha,] Red, Green, Blue)





- [例] ① **Color.FromArgb (255,255,0,0);** // 為紅色不透明
② **Color.FromArgb (126,0,255,0);** // 為綠色半透明
③ **Color.FromArgb (255,0,0,255);** // 為藍色不透明
④ **Color.FromArgb (255,255,0,255);** // 為紫色不透明(紅+藍)



紅、綠、藍三色光合成圖



2. 使用 FromKnownColor 方法設定顏色

- 是屬於 Color 結構的方法。從指定預先定義的色彩來建立 Color 結構。並且由 KnownColor 加列舉常數來表示顏色。
- KnownColor 用來指定系統顏色，如：作用視窗標題欄的名稱為 ActionCaption、控制項預設顏色為 Control 等。但也可以是顏色列舉常數，如：Green、Control等。語法如下：

Color.FromKnownColor(KnownColor.列舉常數)

- [例] 將表單(視窗)標題欄顏色當做(指定給) button1 按鈕的背景色。寫法：

button1.BackColor=Color.FromKnownColor(KnownColor.ActionCaption);



3. 使用顏色列舉常數名稱

- 在C# 2008中，定義許多常用顏色，直接引用就可以。
- 撰寫程式時只要在這些顏色名稱前面輸入「Color.」時，自動出現顏色清單列出所有顏色名稱常值，點選需要的顏色名稱即可

Color.顏色列舉常數名稱

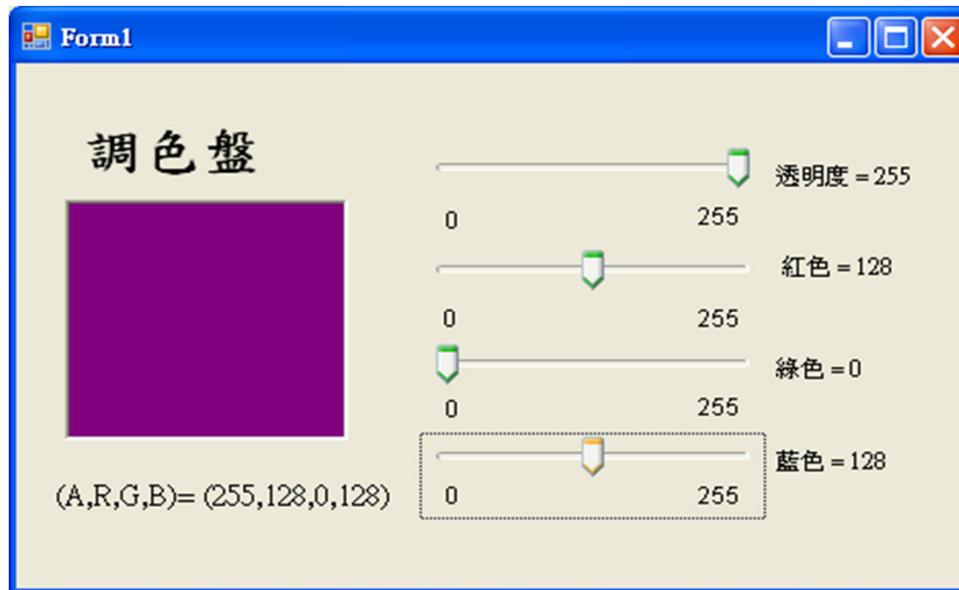
- [例] 將button1按鈕控制項的前景色設為紅色，寫法：

button1.ForeColor = Color.Red;



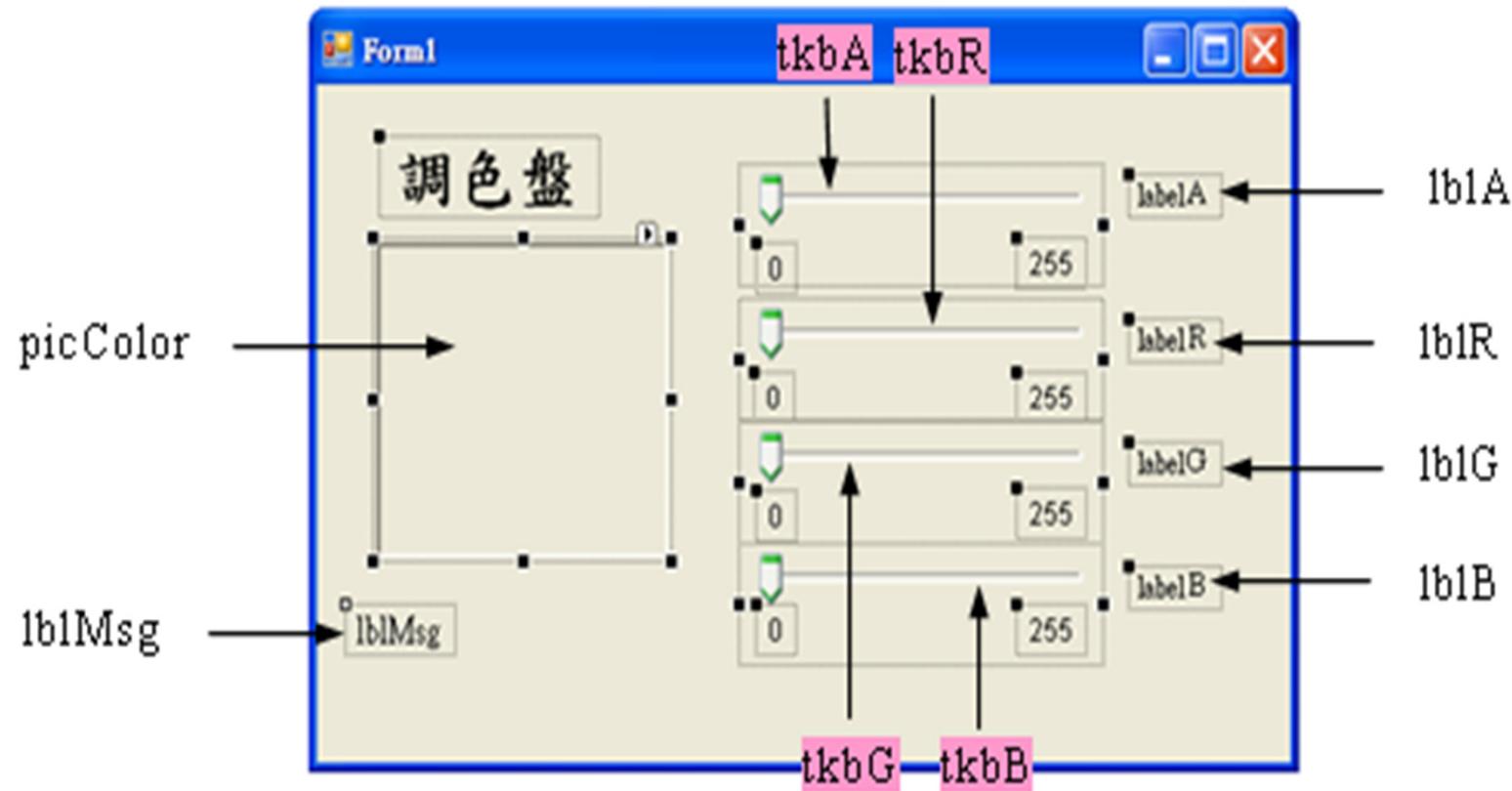
範例演練

利用四個滑動桿控制項，分別來調整色塊的透明度、紅色、綠色和藍色的色階值，設定值一有改變同時將四個設定值分別顯示在表單上，且將紅、綠、藍、透明度混出的色光當作圖片控制項的背景色。





Step1 設計輸出入介面





Step2 控制項屬性設定

1. **tkbA**、**tkbR**、**tkbG**、**tkbB** 四個滑動桿控制項的屬性設定：

- ① **Maximum** 屬性設為 255
- ② **Minimum** 屬性設為 0
- ③ **TickStyle** 屬性設為 **None**

2. **picColor** 圖片控制項的 **BorderStyle** 屬性設為 **Fixed3D(立體)**。

3. 每個滑動桿控制項正下方的0和 255都使用標籤控制項的 **Text** 屬性來設定，總共需要八個標籤控制項。

4. 使用四個標籤控制項 (**lblA**、**lblR**、**lblG**、**lblB**)分別顯示各滑動桿鈕目前的設定值。

5. 使用一個標籤控制項顯示 (A,R,G,B) 的值。



Step3 撰寫程式碼 FileName : argb.sln

```
01 private void Form1_Load(object sender, EventArgs e)
02 {
03     tkbA.Value = 255;
04     myScroll(sender, e);
05     tkbA.Scroll += new EventHandler(myScroll);
06     tkbR.Scroll += new EventHandler(myScroll);
07     tkbG.Scroll += new EventHandler(myScroll);
08     tkbB.Scroll += new EventHandler(myScroll);
09 }
10 private void myScroll(object sender, EventArgs e)
11 {
12     lbIA.Text = "透明度 = " + tkbA.Value.ToString();
13     lbIR.Text = "紅色 = " + tkbR.Value.ToString();
14     lbIG.Text = "綠色 = " + tkbG.Value.ToString();
15     lbIB.Text = "藍色 = " + tkbB.Value.ToString();
16     picColor.BackColor = Color.FromArgb(tkbA.Value, tkbR.Value, tkbG.Value, tkbB.Value);
17     lbIMsg.Text = "(A,R,G,B)= " + "(" + tkbA.Value.ToString() + "," +
18                   tkbR.Value.ToString() + "," + tkbG.Value.ToString() + "," +
19                   tkbB.Value.ToString() + ")";
20 }
```



13.2 繪圖物件

13.2.1 Graphics畫布

- **Graphics**是畫布物件，像是繪圖時所使用的畫紙一樣，可讓繪圖方法在其中任意作畫。
- 將表單、圖片方塊、面板、按鈕...等控制項當作一個畫布，圖形就會顯示在該控制項上。
- 宣告方式：

Graphics 畫布物件變數；

畫布物件變數 = 物件名稱.**CreateGraphics()**;



[例] 要在 **button1** 按鈕控制項上面，新增一個名稱為 **g** 的繪圖物件，寫法：

```
Graphics g ;  
g = button1.CreateGraphics();
```

■ 可合併成一行：

```
Graphics g = button1.CreateGraphics();
```

■ 清畫布物件內的圖形，可用 **Clear** 方法將畫布內容清成指定的顏色，語法：

```
畫布物件.Clear(Color.顏色名稱);
```

[例] 將畫布g清成白色，即成為白色畫布：

```
g.Clear(Color.White);
```



- 使用 **Clear** 方法只清除畫面。
- 若要將 **Graphics** 畫布物件從記憶體中移除，要用 **Dispose**方法，語法：

畫布物件.Dispose();

[例] 將畫布g由記憶體中移除。

g.Dispose();



13.2.2 Pen 畫筆

- Pen 是畫筆物件，像是一支繪圖時所使用的畫筆，可在 **Graphics** 所建立的 畫布物件g 上使用繪圖方法畫出直線或曲線。
- 建立畫筆物件時，除先定義顏色還要指定畫筆粗細
- 語法：

Pen 畫筆物件變數= new Pen(Color.顏色名稱[,粗細]);

- [例] 宣告 p 是屬於畫筆物件，建立畫筆的寬度為 4 pixels 及畫筆的顏色設為藍色：

Pen p = new Pen(Color.Blue,4);



- 建立 Pen 畫筆物件後，若要再度改變畫筆顏色或粗細，可再用下面語法重新設定：

畫筆物件.**Color** = Color.顏色名稱；

畫筆物件.**Width** = 粗細；

- [例] 重新設定畫筆p的寬度為6 pixels 及畫筆的顏色為紅色，寫法：

p.Color = Color.Red ;

p.Width = 6 ;



13.2.3 Brushes 筆刷

- **Brushes** 是純色的筆刷，用法和 **Color** 類似
- 可供繪圖指令指定繪圖的顏色，來填滿一個區域。
- 宣告方式：

Brushes.顏色名稱;



13.2.4 Brush筆刷

- **Brush**筆刷可建立下面所介紹各種樣式的筆刷，
在畫布物件上用繪圖方法畫出一個填滿的區塊
(如矩形、橢圓、多邊形...)。
- 有部分物件是包含在 **Drawing2D** 的命名空間。
- 例 **HatchBrush** (花紋筆刷)。
- 要先使用下列敘述將 **Drawing2D** 含入，含入位置
必須寫在程式碼最前面，寫法：

using System.Drawing.Drawing2D;



1. SolidBrush 筆刷

■ SolidBrush 筆刷可建立一個純色的筆刷，語法

SolidBrush筆刷物件變數 = new **SolidBrush(color.顏色名稱);**

[例] 建立sb 為一支暗紅色的筆刷，寫法：

SolidBrush sb = new **SolidBrush(Color.DrakRed);**



2. HatchBrush筆刷

可建立一個花紋筆刷，語法：

HatchBrush筆刷物件變數 =

```
new HatchBrush(hatchStyle , foreColor , backColor);
```

[例] 建立 hb 為前景紅色、背景藍色的波浪花紋筆刷，
寫法：

HatchBrush hb =

```
new HatchBrush(HatchStyle.Wave,Color.Red ,Color.Blue);
```



3. Rectangle矩形類別

- 可建立一個矩形物件，供繪圖時使用。
- 建立時要定義矩形的起點座標和矩形寬度和高度。
- 語法：

Rectangle矩形物件變數 = new Rectangle(x,y,width,height);

[例] 要建立r1為左上角座標(30,40)，寬度100 和高度為 60 的矩形，寫法：

Rectangle r1 = new Rectangle(30,40,100,60);



4. LinearGradientBrush 筆刷

- 可建立一個漸層的筆刷
- 語法：

LinearGradientBrush 筆刷物件變數 =

```
new LinearGradientBrush(Rectangle, color1, color2,  
angle);
```

[例] 建立 glb 為由紅色到黃色，方向為水平漸層筆刷，
寫法：

```
Rectangle r1 = new Rectangle(10,20,120,80);
```

```
LinearGradientBrush glb =  
new LinearGradientBrush(r1,Color.Red ,Color. Yellow,90);
```



5. Bitmap 圖形類別

- 下面 **TextureBrush** 筆刷建立時，要用到 **Bitmap** 類別，
Bitmap 可建立一個圖形物件，供繪圖使用。
- 建立時要指定檔名和路徑。圖形檔格式可是 **BMP, GIF, JPEG...** 等。語法：

Bitmap 圖形物件變數 = new **Bitmap**("filename");

[例1] **Bitmap** **bmp** = new **Bitmap**("C:\\vb.bmp");

[例2] **Bitmap** **bmp** = new **Bitmap**("..\\vb.jpg");



6. TextureBrush 筆刷

- 是一支以圖形物件當作圖案的筆刷，引數中圖形物件要用 **Bitmap** 來建立。語法：

TextureBrush 筆刷物件變數 = **new TextureBrush(圖形物件);**

- [例] 建立 p1 以 “vb.bmp” 圖案為筆刷材質，寬度為 40 的畫筆，寫法：

Bitmap bmp = new Bitmap("vb.bmp");

TextureBrush tb = new TextureBrush(bmp);

Pen p1 = new Pen(tb,40);



13.3 常用的繪圖方法

■ 繪製圖形的步驟：

- ① 宣告一個畫布物件
- ② 將畫布物件指定給一個控制項
- ③ 在該控制項上面繪製圖形。

- 若要畫線條建立 Pen 畫筆物件。
- 要畫填滿區塊建立 Brush 筆刷物件。
- 最後用各種繪圖方法，畫出需要圖形。



13.3.1 繪製直線

- 要先宣告一個畫布物件
- 再將畫布物件指定給一個控制項。
- 建立 Pen 畫筆物件，指定畫筆的顏色和粗細。
- 最後用繪圖方法，畫出需要的圖形。



1. DrawLine方法

- 可用畫筆物件，畫出一條直線。建立時要定義直線的起點座標和終點座標。
- 直線顏色和粗細，由畫筆物件來決定。
- 語法：

畫布物件.**DrawLine**(畫筆物件, x1, y1, x2, y2);

[例] 在畫布g 上的 (10,10)~(120,120) 兩座標點間繪製一條直線，畫筆顏色和寬度由 p1 畫筆物件決定，寫法：

g.DrawLine(p1,10,10,120,120);



2. DrawRectangle 方法

- 可用畫筆物件，畫出一個矩形框。
- 建立時要定義矩形左上角座標和右下角座標。
- 外框顏色和粗細由畫筆物件決定。
- 語法：

畫布物件.DrawRectangle(畫筆物件,x1, y1, x2, y2);

[例] 在(10,10) ~ (120,120) 間畫一個矩型框，外框顏色
和寬度由p1畫筆物件決定，寫法下：

g.DrawRectangle(p1,10,10,120,120);



3. DrawPolygon 方法

- 可用畫筆物件畫出一個封閉的空心多邊形。
- 建立時除畫筆物件外，還要以 Point 類別指定多邊形的頂點座標。
- 此方法會依照 Point 類別內點的順序來繪製直線，繪完最後一點，該點會和起點相連成一個封閉的多邊形。
- 語法：

畫布物件.DrawPolygon(畫筆物件,Point[]);



4. Point 類別

- 用來建立記錄每一點的X、Y座標值，其中可為一個點，也可包含多個點。
- 語法：

Point 變數 = new Point(x1, y1);

[例] Point[] p1 = new Point(0,0);

Point[] 變數= new Point[]

{new Point(x1,y1), new Point(x2,y2), new Point (x3,y3),.....};

[例] Point[] pts = new Point[]

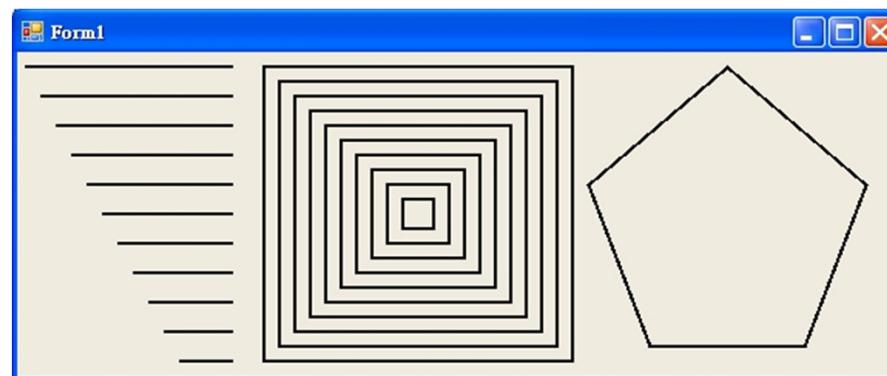
{new Point(0,0), new Point(10,10), new Point(20,20)};



範例演練

試在表單大小 580×255 (寬x高) 上分別畫出三個圖形，線條粗細都設為2。

- ① 圖1 第1條線由 $(5,10)-(140,10)$ ，第2條線由 $(15,30)-(140, 30)$ ，其餘位置以此類推，共有11條線。
- ② 圖2 第1個正方形起點為 $(160,10)$ 、邊長為200，第2個正方形起點為 $(150,20)$ 、邊長為180，其餘正方形以此類推，共有10個正方形。
- ③ 圖3 多邊形端點分別在 $(460,10)、(550,90)、(510,200)、(410,200)、(370,90)$ 。





FileName : linedemo.sln

```
01 private void Form1_Load(object sender, EventArgs e)
02 {
03     this.Size = new Size(580, 255);
04 }
05 private void Form1_Paint(object sender, PaintEventArgs e)
06 {
07     Graphics g = this.CreateGraphics();
08     Pen p = new Pen(Color.Black, 2);
09     int i;
10     for (i = 1; i <= 11; i++)
11     {
12         g.DrawLine(p, 5 + (i - 1) * 10, 10 + (i - 1) * 20, 140, 10 + (i - 1) * 20);
13     }
14     for (i = 1; i <= 10; i++)
15     {
16         g.DrawRectangle(p, 160 + (i - 1) * 10, 10 + (i - 1) * 10, 200 - (i - 1) * 20, 200 -
17             (i - 1) * 20);
18     }
19     Point[] p_arr = { new Point(460, 10), new Point(550, 90), new Point(510, 200),
20                     new Point(410, 200), new Point(370, 90) };
21     g.DrawPolygon(p, p_arr);
22 }
```



13.3.2 繪製曲線

1. DrawEllipse 方法

- 可用畫筆物件畫出一個橢圓框。
- 建立時要定義橢圓形左上角座標、橢圓寬和高度
- 當 寬=高 即為正圓。
- 語法：

畫布物件.DrawEllipse(畫筆物件, x, y, width, height);

畫布物件.DrawEllipse(畫筆物件, Rectangle);

[例1] g.DrawEllipse(p,0,0,120,80);

[例2] g.DrawEllipse(p,rec1);



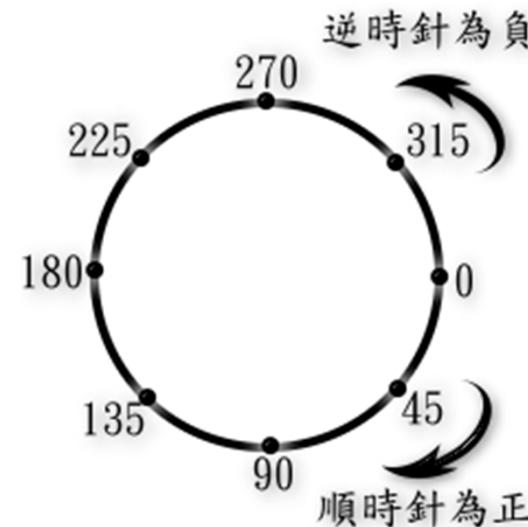
2. DrawArc 方法

- 可用畫筆物件畫出圓弧（橢圓的一部分）。
- 建立時要定義橢圓形左上角座標、橢圓寬和高度，圓弧起點角度和畫弧角度。
- 若畫弧角度為負值，表是逆時針畫圓弧。
- 語法：

畫布物件.DrawArc(畫筆物件, x, y, width, height, startAngle, sweepAngle);

[例] g.DrawArc(p,0,0,50,30,45,90);

圓弧由45度畫到135度





3. DrawPie方法

- 可用畫筆物件畫扇形框。
- 方法和 DrawArc 方法類似，只多兩條和橢圓中心連線。
- 語法：

畫布物件.DrawPie(畫筆物件, x, y, width, height, startAngle,sweepAngle);

[例] 圓弧由 45° 逆時針畫到 315°

`g.DrawPie(p,0,0,50,30,45,-90)`



4. DrawBezier 方法

- 可用畫筆物件依照四個指定的點座標畫出貝茲曲線。
- 語法：

畫布物件.**DrawBezier**(畫筆物件, p1, p2, p3, p4);

5. DrawCurve 方法

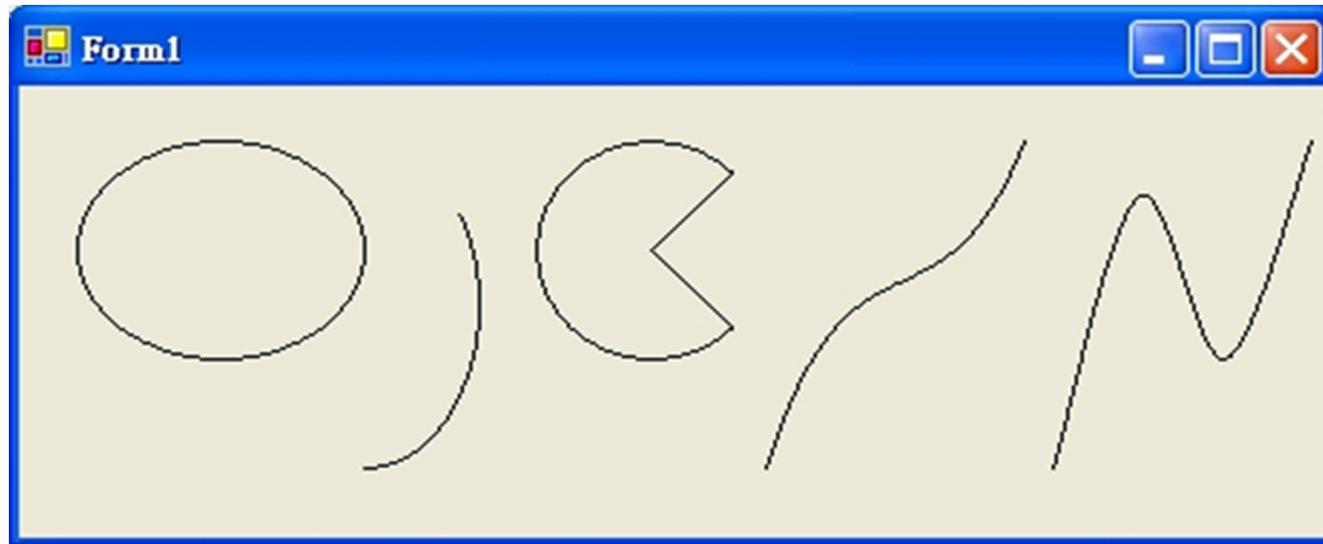
- 可用畫筆物件依照 Point 類別指定的點畫出一條曲線。
- DrawBezier 要指定四個點，但 DrawCurve 指定的點數有彈性。
- 語法：

畫布物件.**DrawCurve**(畫筆物件,Point[]);



範例演練

試分別使用上面五種畫曲線的方法，練習使用 DrawEllipse
畫圓、DrawArc 畫圓弧、DrawPie 畫扇形圖、DrawBezier
畫貝茲曲線和 DrawCurve 畫曲線。





FileName : drawarc.sln

```
01 private void Form1_Paint(object sender, PaintEventArgs e)
02 {
03     Graphics g = this.CreateGraphics();
04     Pen p = new Pen(Color.Black, 1);
05     this.Width = 480;
06     this.Height = 200;
07     g.DrawEllipse(p, 20, 20, 100, 80);
08     g.DrawArc(p, 80, 20, 80, 120, 90, -135);
09     g.DrawPie(p, 180, 20, 80, 80, 45, 270);
10     g.DrawBezier(p, new Point(260, 140),new Point(290, 40),
11                   new Point(320, 100), new Point(350, 20));
12     Point[] ps = new Point[] { new Point(360, 140), new Point(390, 40),
13                               new Point(420, 100), new Point(450, 20) };
14     g.DrawCurve(p, ps);
15 }
```



13.3.3 繪製區塊

1. FillRectangle方法

- 可用筆刷物件在畫布上繪製實心的矩形區塊。
- 語法：

畫布物件.FillRectangle(筆刷物件, x1, y1, x2, y2);

畫布物件.FillRectangle(筆刷物件, Rectangle);

2. FillPolygon方法

- 可用筆刷物件在畫布上繪製封閉的實心多邊形區塊。
- 語法：

畫布物件.FillPolygon(筆刷物件,Point[]);



3. FillEllipse 方法

- FillEllipse 方法可用筆刷物件，在畫布上繪製一個實心橢圓區塊。
- 語法：

畫布物件.FillEllipse(筆刷物件, x, y, width, height);

畫布物件.FillEllipse(筆刷物件, Rectangle);

4. FillPie 方法

- FillPie 方法可用筆刷物件，在畫布上繪製一個實心扇形區塊。
- 語法：

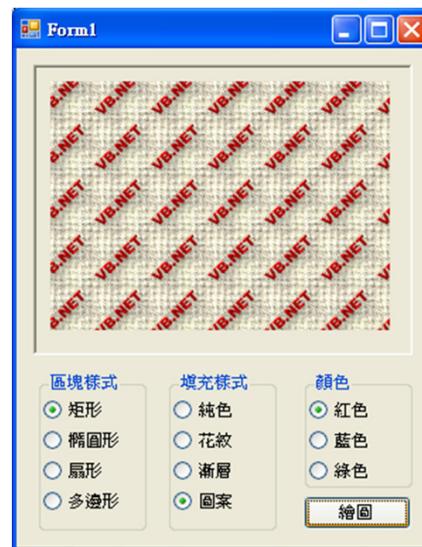
畫布物件.FillPie(筆刷物件,x,y,w,h,startAngle,sweepAngle);

畫布物件.FillPie(筆刷物件,Rectangle,startAngle,sweepAngle);



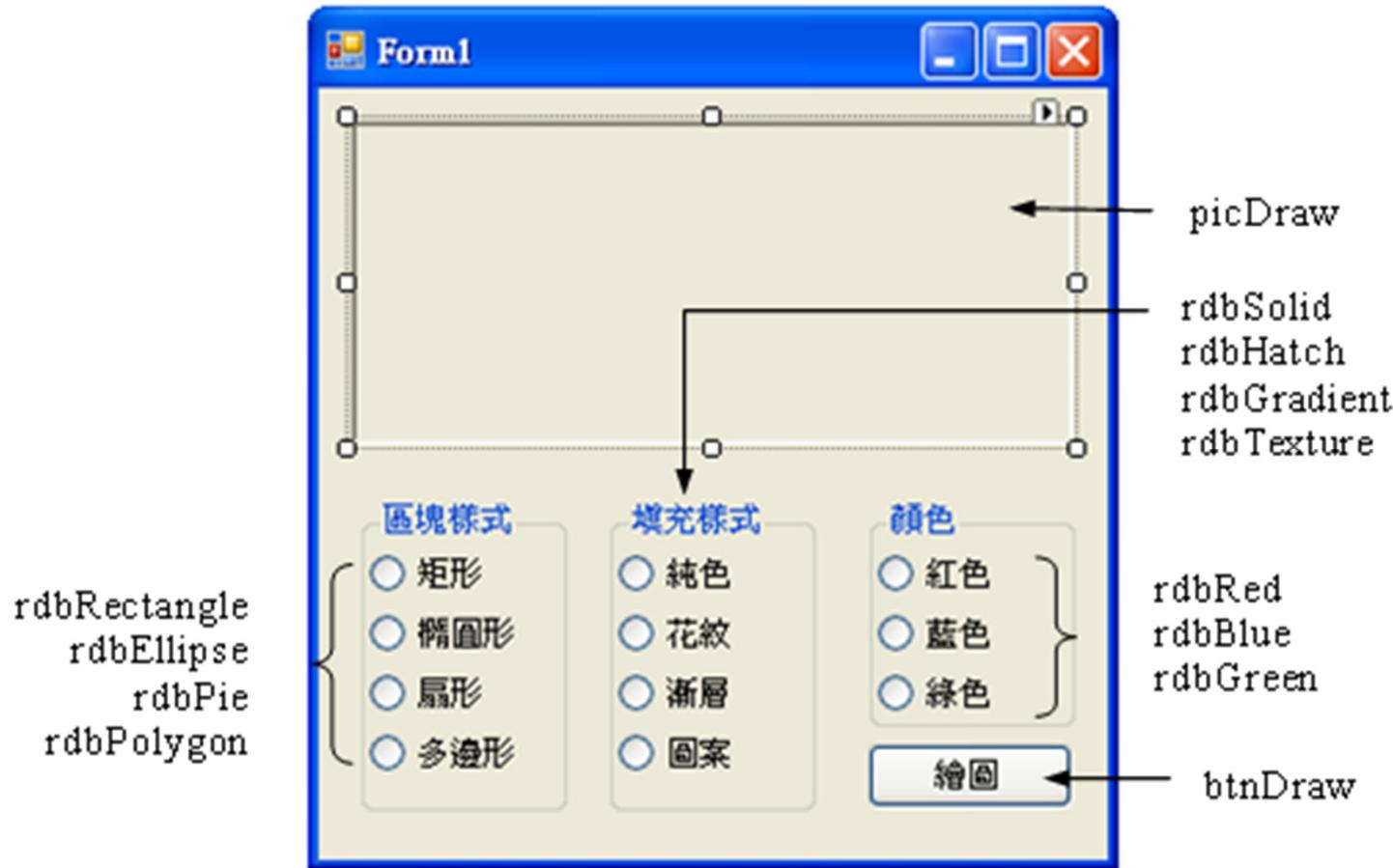
範例演練

視窗中有「區塊樣式」、「填充樣式」和「顏色」三個框架，使用者可以自行點選樣式組合，然後按「繪圖」鈕繪製指定的圖形樣式。





■ Step1 建立輸出入介面



rdbRectangle
rdbEllipse
rdbPie
rdbPolygon

區塊樣式
 矩形
 橢圓形
 扇形
 多邊形

填充樣式
 純色
 花紋
 漸層
 圖案

顏色
 紅色
 藍色
 綠色

← picDraw
rdbSolid
rdbHatch
rdbGradient
rdbTexture

rdbRed
rdbBlue
rdbGreen

← btnDraw



Step2 撰寫程式碼

FileName : fill.sln

```
01 using System.Drawing.Drawing2D;  
02  
03 namespace fill  
04 {  
05     public partial class Form1 : Form  
06     {  
07         public Form1()  
08         {  
09             InitializeComponent();  
10         }  
11  
12         private void Form1_Load(object sender, EventArgs e)  
13         {  
14             rdbRectangle.Checked = true;  
15             rdbSolid.Checked = true;  
16             rdbRed.Checked = true;  
17         }  
18     }
```



```
19  private void btnDraw_Click(object sender, EventArgs e)
20  {
21      Graphics g = picDraw.CreateGraphics();
22      g.Clear(Color.FromKnownColor(KnownColor.Control));
23      Color c;
24      if (rbRed.Checked)
25      {
26          c = Color.Red;
27      }
28      else if (rbBlue.Checked)
29      {
30          c = Color.Blue;
31      }
32      else
33      {
34          c = Color.Green;
35      }
36      Brush b;
37      if (rbSolid.Checked)
38      {
39          SolidBrush b1 = new SolidBrush(c);
40          b = b1;
41      }
42  }
```



```
43     else if (rbHatch.Checked)
44     {
45         HatchBrush b2 = new HatchBrush(HatchStyle.Wave, c, Color.White);
46         b = b2;
47     }
48     else if (rbGradient.Checked)
49     {
50         Rectangle ret = new Rectangle(0, 0, 280, 280);
51         LinearGradientBrush b3 = new LinearGradientBrush
52             (ret, c, Color.White, 90);
53     }
54     else
55     {
56         Bitmap img = new Bitmap("vb.bmp");
57         TextureBrush b4 = new TextureBrush(img);
58         b = b4;
59     }
```



```
61     if (rdbRectangle.Checked)
62     {
63         g.FillRectangle(b, 10, 10, 240, 180);
64     }
65     else if (rdbEllipse.Checked)
66     {
67         g.FillEllipse(b, 10, 10, 240, 180);
68     }
69     else if (rdbPie.Checked)
70     {
71         g.FillPie(b, 30, 10, 240, 180, 45, 270);
72     }
73     else
74     {
75         Point[] ps = new Point[] { new Point(130, 10), new Point(30, 200),
76                                     new Point(240, 70), new Point(10, 70), new Point(220, 200) };
77         g.FillPolygon(b, ps);
78     }
79 }
80 }
```



13.3.4 繪製文字

■ 使用 DrawString 方法前

- ① 先宣告一個畫布物件
- ② 將畫布物件指定給一個控制項。
- ③ 建立 Brushes 或 Brush 筆刷物件
- ④ 依照需求指定筆刷的種類。

■ 語法：

```
畫布物件.DrawString("字串", new Font("字型名稱",FontSize), Brushes, x, y);
```

[例1]

```
g.DrawString("VB.NET", new Font("Arial",24), Brushes.Black,20,50);
```

[例2]

```
HatchBrush b = new HatchBrush(HatchStyle.Wave,c1,c2);
g.DrawString("關懷",new Font("標楷體",48),b,20,50);
```



13.3.5 其它常用的繪製方法

1. TranslateTransform 方法

- 設定圖形位移的大小。

畫布物件.**TranslateTransform(x, y);**

[例] 希望以後繪製的圖形座標，都會向右移15點向下30點：

g.TranslateTransform(15,30);

2. ScaleTransform方法

- 設定圖形縮放的比例。引數中要定義寬度和高度的縮放比例。

畫布物件.**ScaleTransform(sw, sh);**

[例] 希望以後繪製圖形大小，寬度縮小一半，高度放大2倍，寫法：

g.ScaleTransform(0.5,2);



3. RotateTransform 方法

- 設定圖形旋轉的角度。引數中角度以 360 度為單位（資料型別為 float）。
- 語法：

畫布物件.**RotateTransform(angle);**

[例] 希望設定後繪製的圖形都順時針旋轉 45 度，寫法：

g.RotateTransform(45);

- 如執行兩次 RotateTransform 方法，兩次旋轉角度會相加，影響以後繪製圖形。例

g.RotateTransform(45)

兩次，以後繪圖的都旋轉 90 度，也就是說由上次旋轉的角度起再旋轉角度。



4. ResetTransform 方法

- **TranslateTransform**、**ScaleTransform**、**RotateTransform**方法，都以上次執行結果繼續變形
- 若要還原成原始設定，使用 **ResetTransform** 方法。
- 語法：

畫布物件.**ResetTransform()**;



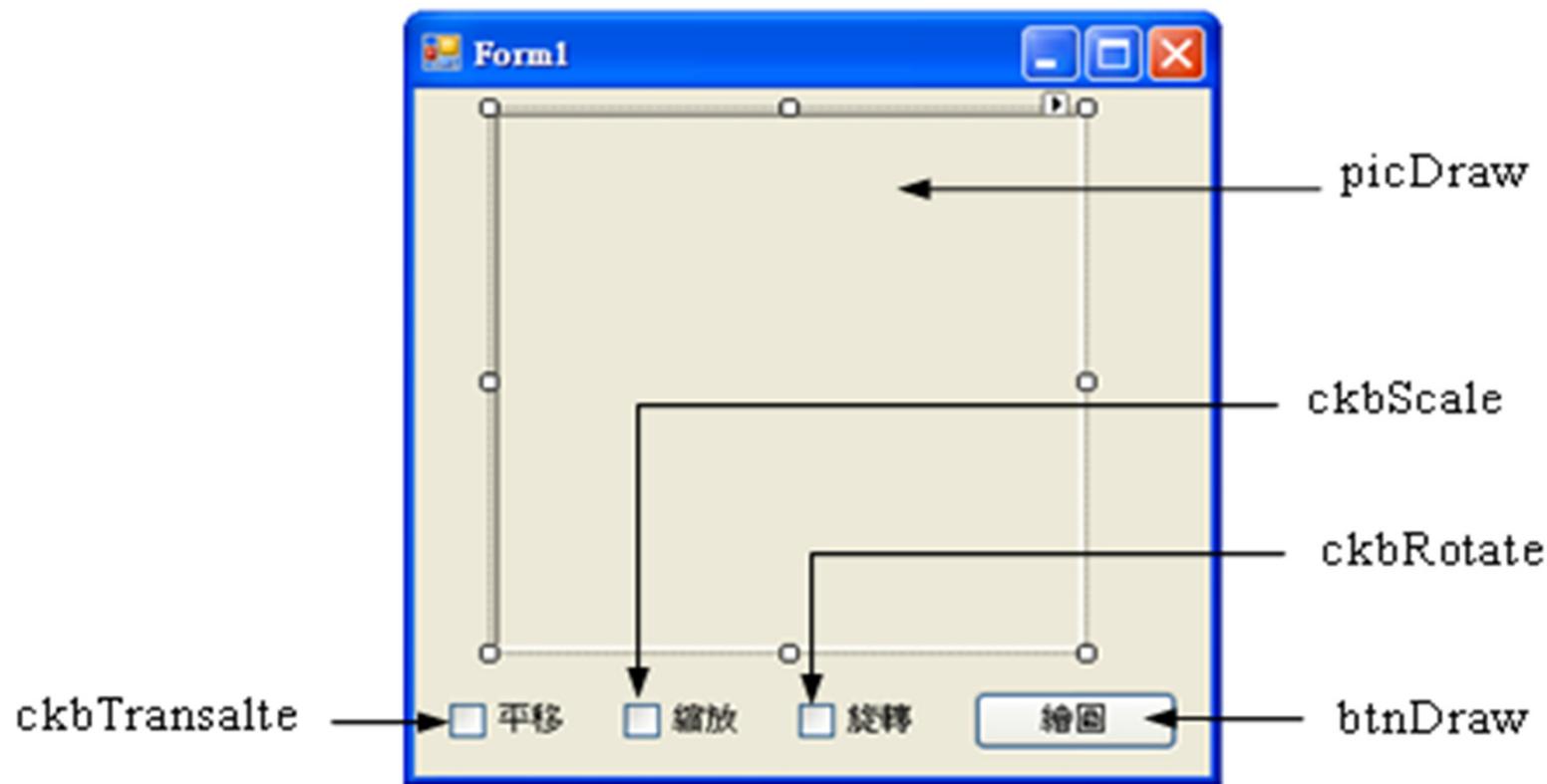
範例演練

在視窗下有 平移 、 放大 、 旋轉 三個核取方塊，核取後按 **繪圖** 鈕就會畫出指定文字樣式。





■ Step1 設計輸出入介面





Step2 撰寫程式碼

FileName : DrawString.sln

```
01 using System.Drawing.Drawing2D;  
02  
03 namespace DrawString  
04 {  
05     public partial class Form1 : Form  
06     {  
07         public Form1()  
08         {  
09             InitializeComponent();  
10         }
```



```
12 private void btnDraw_Click(object sender, EventArgs e)
13 {
14     Graphics g = picDraw.CreateGraphics();
15     g.Clear(Color.FromKnownColor(KnownColor.Control));
16     Rectangle rec = new Rectangle(0, 0, 220, 30);
17     int i;
18     for (i = 1; i <= 10; i++)
19     {
20         LinearGradientBrush b = new LinearGradientBrush
21             (rec, ColorTranslator.FromHtml ("#00FFFF"), Color.White, 10);
22         g.DrawString("Visual Basic 2008",new Font("Arial Black", 16), b, 0, 0);
23         if (ckbTranslate.Checked) g.TranslateTransform(10, 10);
24         if (ckbScale.Checked) g.ScaleTransform((float)1.1, (float)1.1);
25         if (ckbRotate.Checked) g.RotateTransform(10);
26     }
27 }
```



13.4 圖檔的讀取和儲存

1. Bitmap 圖形類別

- 除13.2 節介紹建立時指定載入圖檔方式，也可指定寬度和高度，會在記憶體保留空間：

Bitmap 圖形物件變數 = **new Bitmap(width, height);**

[例] **Bitmap bmp = new Bitmap(160,120);**

2. FromImage 方法

- 宣告一個畫布物件，其來源為圖形物件。

畫布物件變數 = **Graphics.FromImage(Bitmap);**

[例] **Bitmap bmp = new Bitmap(160,120);**
g = Graphics.FromImage(bmp);



3. Save方法

- 可將 **Bitmap** 物件圖形存成檔案，引數除檔名外，要包含路徑。

Bitmap.Save("FileName");

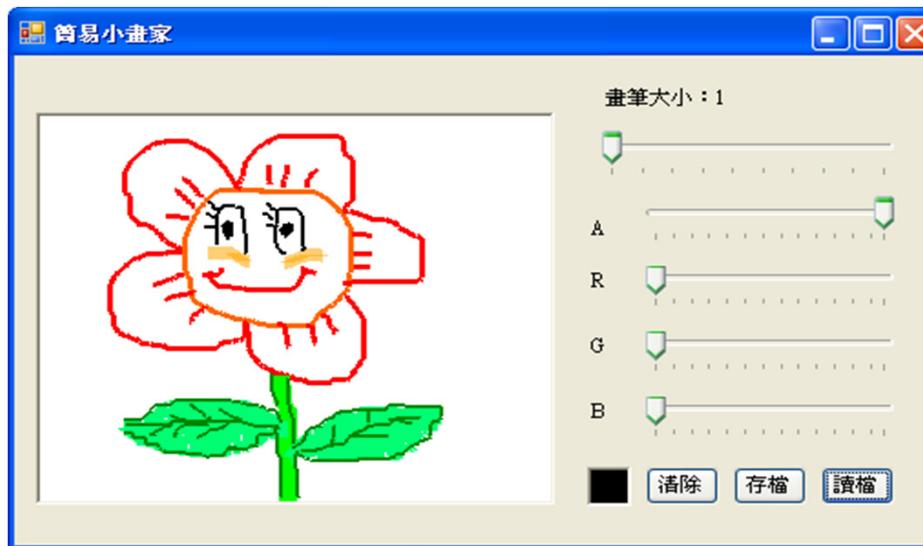
[例] **bmp.Save("..\\test.bmp");**



範例演練

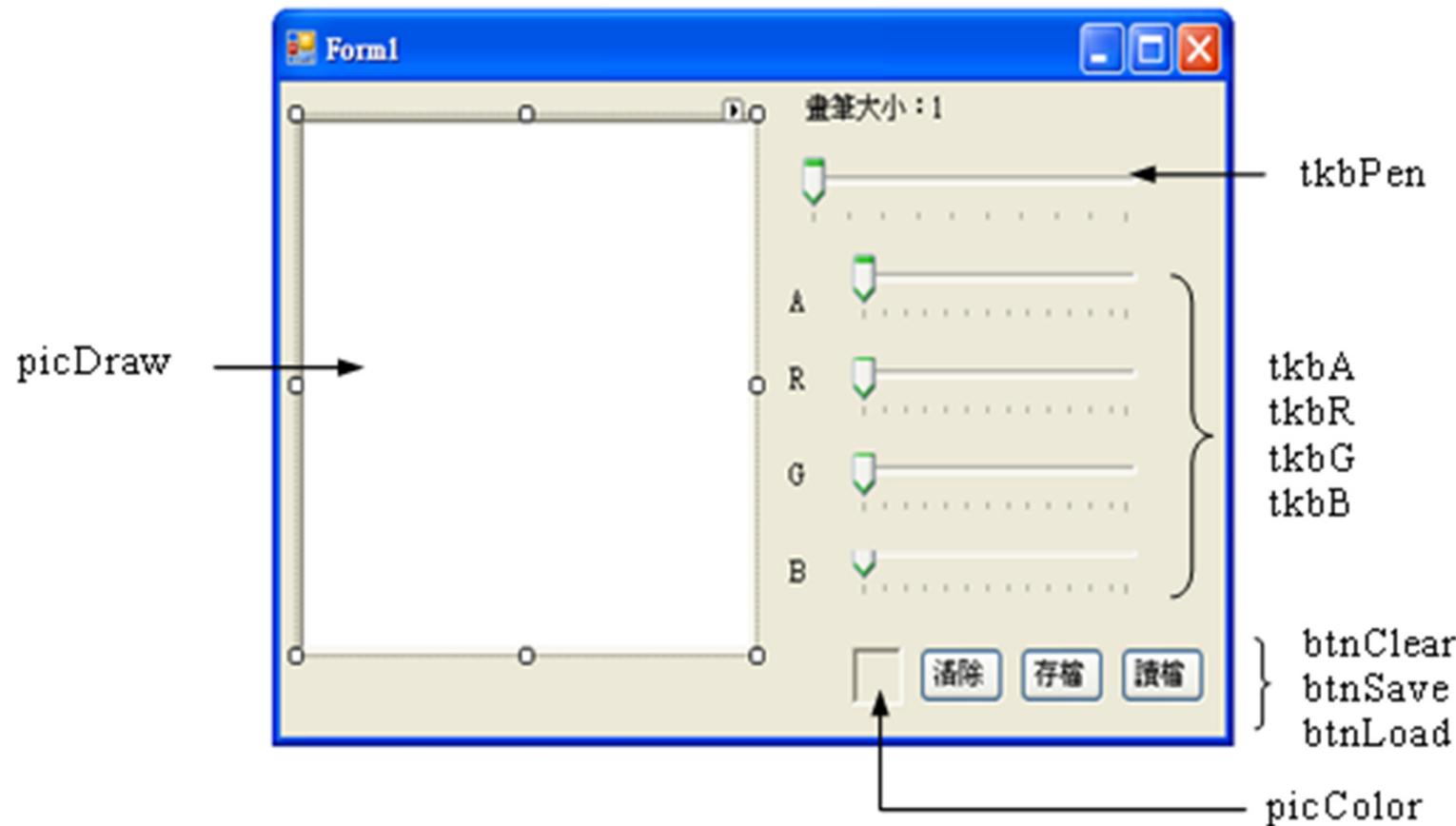
■ 在視窗中有一個白色方塊，使用者可用滑鼠任意畫圖。
畫筆大小和顏色，可以用滑動桿自行調整。

- ① 按 **清除** 鈕時，方塊清成白色。
- ② 按 **存檔** 鈕時，方塊內圖案以「**test1.jpg**」存檔。
- ③ 按 **讀檔** 鈕時，方塊內會顯示原來存檔的。





■ Step1 設計輸出入介面





Step2 撰寫程式碼

FileName :savedraw.sln

```
01 Bitmap bmp = new Bitmap(280, 240);
02 int p_w = 1;
03 Color p_c = Color.Black;
04 Pen p = new Pen(Color.Black, 1);
05 int x1, y1;
07 private void Form1_Load(object sender, EventArgs e)
08 {
09   tka.Value = 255;
10   tka.Scroll += new EventHandler(myScroll);
11   tkr.Scroll += new EventHandler(myScroll);
12   tkg.Scroll += new EventHandler(myScroll);
13   tkb.Scroll += new EventHandler(myScroll);
14   myScroll(sender, e);
15 }
16
```



```
17 private void myScroll(object sender, EventArgs e)
18 {
19     p_c = Color.FromArgb(tkbA.Value, tkbR.Value, tkbG.Value, tkbB.Value);
20     picColor.BackColor = p_c;
21     p = new Pen(p_c, p_w);
22 }
24 private void tkbPen_Scroll(object sender, EventArgs e)
25 {
26     p_w = tkbPen.Value;
27     p = new Pen(p_c, p_w);
28     lblPen.Text = "畫筆大小：" + p_w;
29 }
31 private void picDraw_MouseDown(object sender, MouseEventArgs e)
32 {
33     x1 = e.X;
34     y1 = e.Y;
35 }
```



```
37 private void picDraw_MouseMove(object sender, MouseEventArgs e)
38 {
39     Graphics g = Graphics.FromImage(bmp);
40     if (e.Button == MouseButtons.Left)
41     {
42         g.DrawLine(p, x1, y1, e.X, e.Y);
43         picDraw.Image = bmp;
44         x1 = e.X;
45         y1 = e.Y;
46     }
47     g.Dispose();
48 }
49 private void btnClear_Click(object sender, EventArgs e)
50 {
51     Graphics g = Graphics.FromImage(bmp);
52     g.Clear(Color.White);
53     picDraw.Image = bmp;
54 }
```



```
56 private void btnSave_Click(object sender, EventArgs e)
57 {
58     bmp.Save("test1.jpg");
59 }
60
61 private void btnLoad_Click(object sender, EventArgs e)
62 {
63     Graphics g = Graphics.FromImage(bmp);
64     System.IO.FileStream fs = new System.IO.FileStream("test1.jpg",System.IO.
65         FileMode.Open);
66     bmp = new Bitmap(fs);
67     picDraw.Image = bmp;
68 }
```

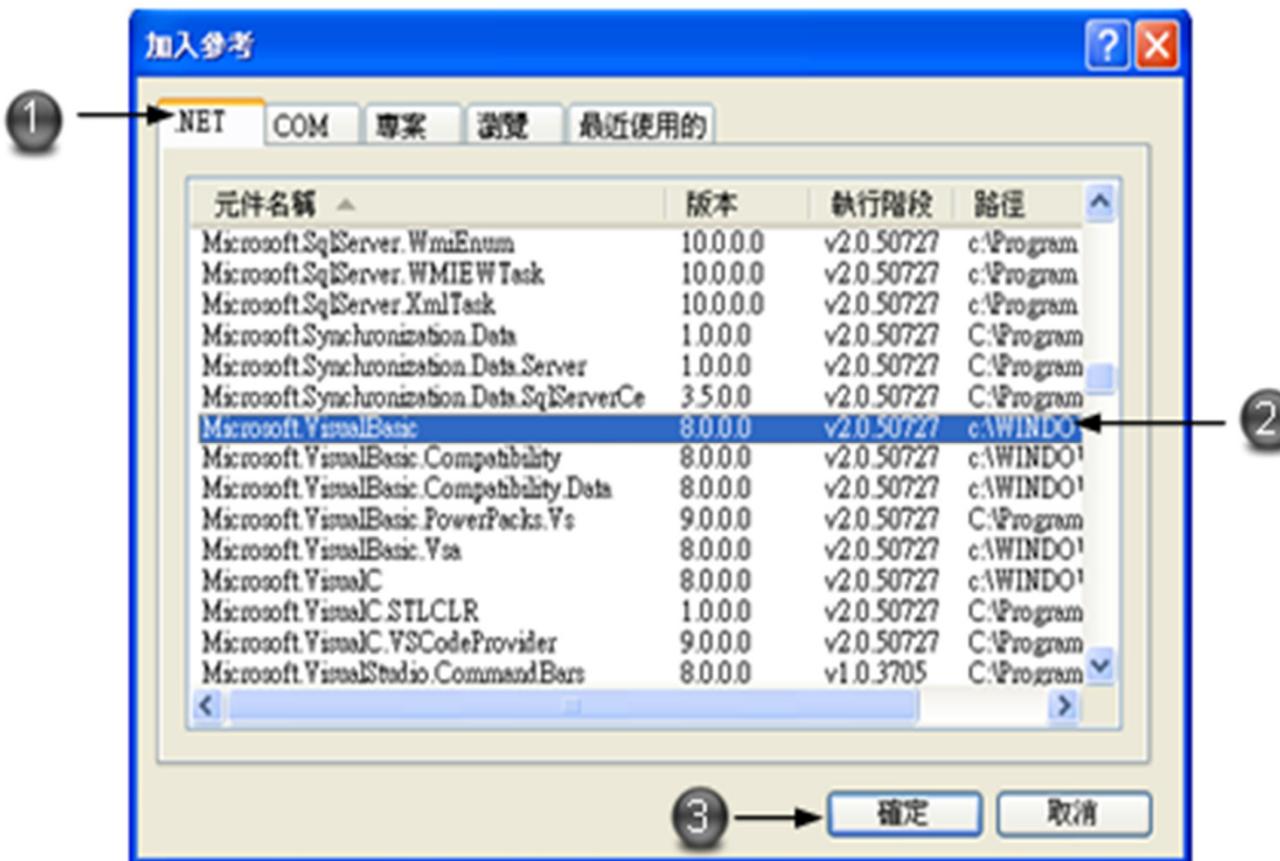


13.5 音效與多媒體播放

13.5.1 播放聲音檔

1. Computer 物件

- 在 VB 2008 提供 `My.Computer.Audio` 物件，可在程式中很容易播放指定的 `*.wav` 聲音檔。
- 由於 .NET 中所有類別與物件都可互相呼叫，C# 若要用 VB 2008 的 `My.Computer.Audio` 物件來播放聲音檔
- 先執行功能表【專案(P)/加入參考(R)】指令，選取「`Microsoft.VisualBasic`」元件的參考：





■ 在程式開頭引用 Microsoft.VisualBasic 命名空間：

using Microsoft.VisualBasic;

using Microsoft.VisualBasic.Devices;

■ 接著可用下面語法建立 Computer 物件。透過此物件可播放音效、處理鍵盤、滑鼠...等資訊。

Computer Computer 變數=new Computer();



2. PlaySystemSound 方法

- 使用 Computer 物件的 PlaySystemSound 方法，可用在背景播放一次系統音效。語法：

`Computer變數.Audio.PlaySystemSound(系統音效);`

`Computer變. Audio.PlaySystemSound(System.Media.SystemSounds.Beep);`

- SystemSound 提供 Asterisk、Beep、Exclamation、Hand、Question 等5個系統音效。
因 PlaySystemSound 方法是在背景播放系統音效，
程式仍會繼續執行。



3. Play方法

- 使用 Computer 物件的 Play 方法，除可播放指定的 WAV 語音檔外，還可設定播放的模式。語法：

Computer變數.Audio.Play("音效檔", AudioPlayMode);

- [例]

Computer變數.Audio.Play ("C:\\good.wav", AudioPlayMode.Background);

- AudioPlayMode 播放模式三種形式，視程式需要指定適當播放模式：

成員	說明
WaitToComplete	音效播放完畢後才繼續執行程式碼。
Background	在背景播放音效程式繼續執行。
BackgroundLoop	在背景反覆播放音效，直到執行 Stop 方法才停止。



4. Stop方法

■ Computer 物件的 Stop 方法，可停止 Play 方法在背景反覆播放的 WAV 語音檔。語法：

Computer變數.Audio.Stop()



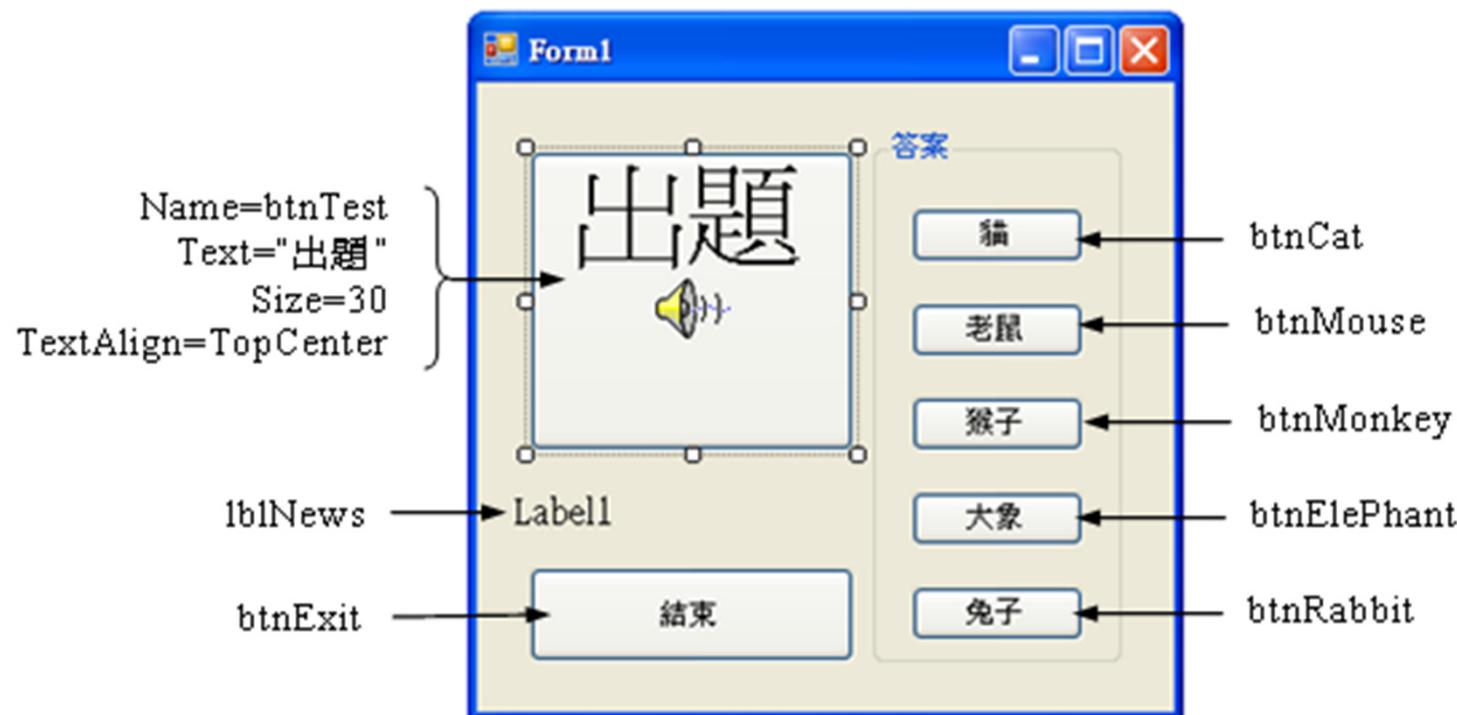
範例演練

在視窗左邊有一個 **出題** 鈕，按一下就唸一個英文單字。共有五題，聽完後請選擇答案。若聽不清楚，再按 **出題** 鈕會重讀一次。答對或答錯時，各會播出一段語音。





Step2 設計輸出入介面





Step3 撰寫程式碼

FileName : playsound.sln

```
01 using Microsoft.VisualBasic;
02 using Microsoft.VisualBasic.Devices;
03
04 namespace playsound
05 {
06     public partial class Form1 : Form
07     {
08         public Form1()
09         {
10             InitializeComponent();
11         }
12
13         int num = -1;
14         string[] test = { "elephant", "monkey", "cat", "rabbit", "mouse" };
15         bool ans = true;
```



```
16 private void Form1_Load(object sender, EventArgs e)
17 {
18     Computer myComputer = new Computer();
19     myComputer.Audio.PlaySystemSound(System.Media.SystemSounds.Beep);
20     lblNews.Text = "按 出題 鈕開始測驗！";
21     btnCat.Click += new EventHandler(myClick);
22     btnMouse.Click += new EventHandler(myClick);
23     btnMonkey.Click += new EventHandler(myClick);
24     btnElephant.Click += new EventHandler(myClick);
25     btnRabbit.Click += new EventHandler(myClick);
26 }
```



```
27 private void btnTest_Click(object sender, EventArgs e)
28 {
29     Computer myComputer = new Computer();
30     if (ans == true)
31     {
32         if (num != 4)
33         {
34             num += 1;
35         }
36         else
37         {
38             num = 0;
39         }
40         myComputer.Audio.Play(test[num]+".wav", AudioPlayMode.WaitToComplete);
41         ans = false;
42     }
43     else
44     {
45         myComputer.Audio.Play(test[num]+".wav", AudioPlayMode. WaitToComplete);
46     }
47     lblNews.Text = "第 " + (num + 1) + " 題";
48 }
49 }
```



```
50  private void myClick(object sender, EventArgs e)
51  {
52      Computer myComputer = new Computer();
53      Button btn = (Button)sender;
54      if (ans == false)
55      {
56          if (btn.Tag == test[num])
57          {
58              myComputer.Audio.Play("good.wav", AudioPlayMode.WaitToComplete);
59              lblNews.Text = "您真內行！！";
60          }
61          else
62          {
63              myComputer.Audio.Play("bad.wav", AudioPlayMode.WaitToComplete);
64              lblNews.Text = "請再好好加油！";
65          }
66          ans = true;
67      }
68      else
69      {
70          myComputer.Audio.PlaySystemSound(System.Media.SystemSounds.
71              Asterisk);
71          lblNews.Text = "請先按 出題 鈕！";
72      }
73  }
```

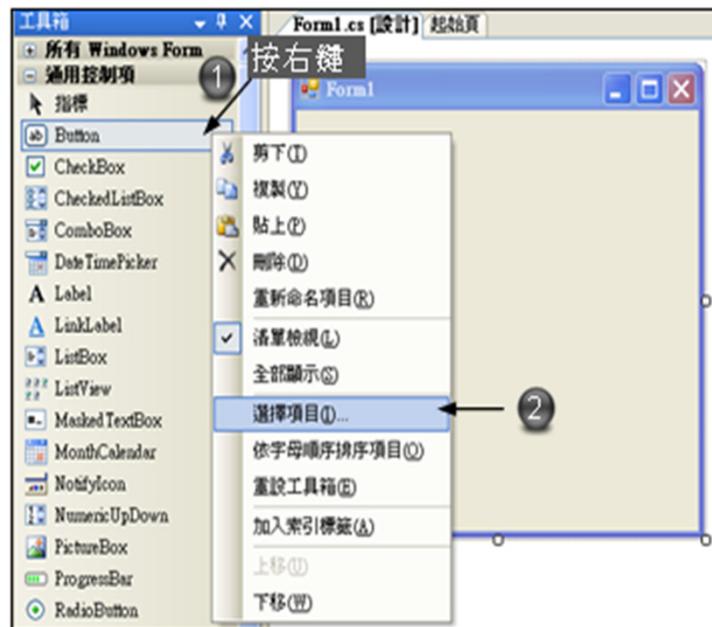


```
74  private void btnExit_Click(object sender, EventArgs e)
75  {
76      Application.Exit();
77  }
78 }
79 }
```



13.5.2 播放多媒體

- 在 C# 2008 可引用 COM 的 Windows Media Player 元件，透過此元件可以製作多媒體播放程式。
- 引用 Windows Media Player 元件的步驟：
- Step1 新增 Windows Media Player 元件到工具箱。



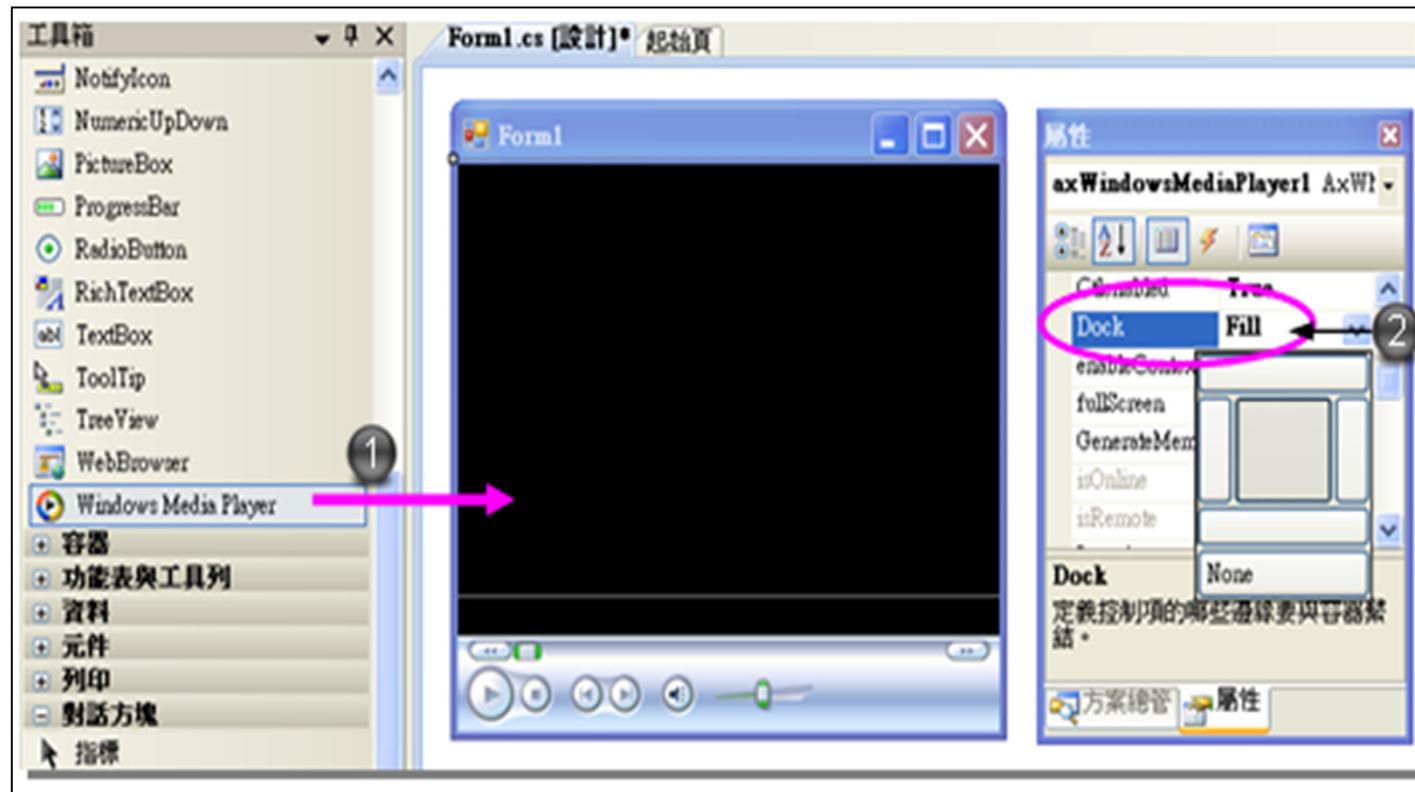


■ Step2 加入「Windows Media Player」元件





■ Step3 工具箱新增 Windows Media Player 將此元件拖曳到表單上填滿整個表單





■ Step4 用下面屬性設定 Windows Media Player 元件



確定

取消

套用(A)



畫面說明

1. 來源

可為檔案名稱或**URL(U)**，用來設定欲播放多媒體的檔案路徑。

2. 選取模式有四種模式：

模式	功能說明
Full 預設值	有狀態視窗、播放、停止、靜音、音量控制、上一段、下一段、快轉、倒帶等所有完整的多媒體操作控制列。 有狀態視窗、播放、停止、靜音、音量控制、上一段、下一段、快轉、倒帶等所有完整的多媒體操作控制列。
None	沒有任何可以操作多媒體播放的控制列，只顯示播放視訊的畫面。
Mini	只有狀態視窗、播放、停止、靜音及音量控制等控制項。
Invisible	多媒體的畫面全部隱藏。多媒體的畫面全部隱藏。



■ 3. 播放選項

- ① 自動啟動(O)：程式執行時自動播放影片。
- ② 適當延伸調整(S)：播放時自動調整長寬。
- ③ 全螢幕播放(F)：程式執行時程式執行時以全螢幕播放。
- ④ 播放次數(N)：設定播放次數。

■ 4. 音量設定

用來調整音量大小、左右聲道以及是否靜音。



範例演練

製作一個可播放多媒體的應用程式。執行功能表中[檔案/開啟] 指令，由開啟檔 \chap13\playMovie\bin\debug 資料夾選取動畫檔為**fishMove.avi**，可播放魚的覓食介紹。功能表中的選取模式功能有：**None**、**Mini**、**Full**、**Invisible**四種選項。





Step1 設計輸出入介面





Step2 撰寫程式碼

```
// FileName : playmovie.sln

01 private void 開啟ToolStripMenuItem_Click(object sender, EventArgs e)
02 {
03     if (openFileDialog1.ShowDialog() == DialogResult.OK)
04     {
05         axWindowsMediaPlayer1.URL = openFileDialog1.FileName;
06     }
07 }
08 private void noneToolStripMenuItem_Click(object sender, EventArgs e)
09 {
10     axWindowsMediaPlayer1.uiMode = "None";
11 }
```



```
12 private void miniToolStripMenuItem_Click(object sender, EventArgs e)
13 {
14     axWindowsMediaPlayer1.uiMode = "Mini";
15 }
16 private void fullToolStripMenuItem_Click(object sender, EventArgs e)
17 {
18     axWindowsMediaPlayer1.uiMode = "Full";
19 }
20 private void invisibleToolStripMenuItem_Click(object sender, EventArgs e)
21 {
22     axWindowsMediaPlayer1.uiMode = "Invisible";
23 }
24 private void 結束ToolStripMenuItem_Click(object sender, EventArgs e)
25 {
26     Application.Exit();
27 }
28 }
```