

繪圖與多媒體

C# 2008 在螢幕上顯示的表單、控制項或文字，都是以繪圖的方式，繪出的圖形。

當在表單或控制項建立 **Graphics** 物件後，就可以在表單或控制項中繪圖，包括：繪製文字、直線、矩形、多邊形、圓形、橢圓形、圓弧，也可將圖形填滿。

播放多媒體音效的最簡單方法，就是利用 **Windows Media Player** 來播放，但 **Windows Media Player** 預設並未出現在工具箱中，必須自行加入。

C# 2008 提供的 **System.Media.SoundPlayer** 類別可以直接播放 **.wav** 音效檔和系統音效。

學習重點

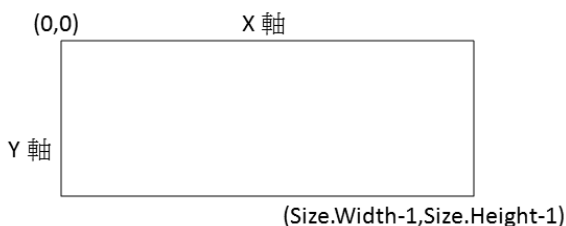
- 繪圖座標
- 建立畫布
- 建立畫筆和筆刷
- 繪製文字和直線
- 繪製矩形和多邊形
- 繪製曲線
- 繪製填滿圖形
- 圖形平移、縮放和旋轉
- 圖檔的讀取和儲存
- 多媒體播放程式
- **System.Media.SoundPlayer** 播放 **.wav** 音效檔和系統音效

14.1 繪圖工具

C# 2008 在螢幕上顯示的表單、控制項或文字，都是以繪圖方式所繪出的圖形。

14.1.1 繪圖座標

繪圖座標系統的基本單位是像素 (Pixels)，可以在表單或控制項建立畫布，畫布繪圖區域是一個矩形，它是以左上角為原點，X 軸座標由左至右，Y 軸座標由上至下。



當在控制項建立畫布後，整個畫布的寬度和高度如下：

物件屬性	說明
Size.Width	整個控制項的寬度。
Size.Height	整個控制項的高度。

但表單控制項因為含有標題列、功能表列、狀態列和框線的寬度，因此其實際可顯示的寬度和高度如下：

表單物件屬性	說明
ClientSize.Width	表單實際可顯示的寬度。
ClientSize.Height	表單實際可顯示的高度。

14.1.2 建立畫布

C# 2008 可以在表單或控制項中繪圖，而繪圖前最先的宣告就是建立畫布，當在表單或控制項建立 Graphics 物件後，就可以在表單或控制項繪圖，宣告的語法如下：

```
Graphics 畫布物件變數;
畫布物件變數=控制項名稱.CreateGraphics;
```

例如：在表單上建立畫布 g：

```
Graphics g ;
g=this.CreateGraphics;
```

又如：在圖片方塊 pictureBox1 上建立畫布 g：

```
Graphics g;
g=pictureBox1.CreateGraphics;
```

也可以將兩行合併成一行：

```
Graphics g =pictureBox1.CreateGraphics;
```

Paint 事件繪圖

除了上述的方法外，在 Paint 事件也可以使用其 PaintEventArgs 類別的參數 e 取得繪圖物件，以繪製圖形。

Paint 事件繪圖的語法：

```
Private void 物件名稱_Paint(object sender, PaintEventArgs e)
```

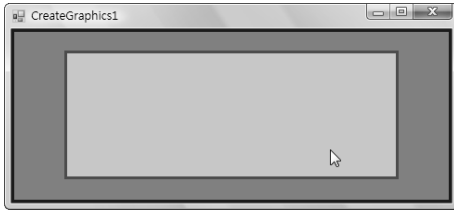
例如：Form 表單的 Paint 事件繪製矩形。

```
private void Form1_Paint(object sender, PaintEventArgs e)
{
    Pen pen = new Pen(Color.Blue, 5);
    e.Graphics.DrawRectangle(pen, 0, 0, 100,100); //繪矩形
}
```

範例：在表單和圖片方塊上建立畫布

當表單視窗縮小或放大時，都會觸發 Paint 事件，在 Paint 事件中，將表單建立成灰色畫布，並以藍色畫筆在表單的實際繪圖區繪製一個藍色矩形。如果在圖片方塊按一下滑鼠左鍵，則會將圖片方塊建立成淺綠色畫布並繪製一個紅色矩形。(CreateGraphics1.sln)

執行結果



範例說明

開啟已經建立完成的 `CreateGraphics1.sln` 專案，本範例示範如何在表單和圖片方塊上建立畫布繪製矩形。

程式碼

```
19 private void pictureBox1_Click(object sender, EventArgs e)
20 {
21     Graphics g = pictureBox1.CreateGraphics(); //在 PictureBox1
        建立繪圖物件
22     Pen pen =new Pen(Color.Red, 5); //建立紅色畫筆，畫筆粗細=5 個
        Pixels
23     int width, height ;
24     width = pictureBox1.Size.Width; //取得控制項大小
25     height = pictureBox1.Size.Height;
26     g.Clear(Color.LightGreen); //PictureBox1 清除為淡綠色
27     g.DrawRectangle(pen, 0, 0, width - 1, height - 1); //繪矩形
28 }
29
30 private void Form1_Paint(object sender, PaintEventArgs e)
31 {
32     Pen pen = new Pen(Color.Blue, 5); //建立藍色畫筆，畫筆粗細=5 個
        Pixels
33     int width, height ;
34     width = this.ClientSize.Width ; //取得表單實際繪圖區的大小
35     height = this.ClientSize.Height;
36     e.Graphics.Clear(Color.Gray); //表單清除為灰色
37     e.Graphics.DrawRectangle(pen, 0, 0, width - 1, height - 1); //
        繪藍色矩形
```

```
38 }
```

程式說明

```
21 在 pictureBox1 建立繪圖物件。
22 建立紅色畫筆，畫筆粗細為 5 個 Pixels。
23-25 取得 pictureBox1 控制項大小。
26 pictureBox1 清除為淡綠色。
27 pictureBox1 上繪紅色矩形外框。
30-38 只要表單大小調整縮小或放大，都會自動觸發 Paint 事件。
33-35 取得表單實際繪圖區的大小。
36 在 Paint 事件中，可以使用 e.Graphics 來繪圖，
    e.Graphics.Clear(Color.Gray) 將表單清除為灰色。
37 以藍色畫筆在表單上繪製矩形。
19-28 在圖片方塊 pictureBox1 按一下滑鼠左鍵，會將圖片方塊建立成淺綠色畫布
    並繪製一個紅色矩形。
```

14.1.3 建立畫筆和筆刷

畫筆 Pen 物件

建立畫布後要繪製圖形前，必須先建立畫筆，建立畫筆物件的語法如下：

```
Pen 畫筆=new Pen(畫筆顏色, 畫筆粗細);
```

例如：建立藍色，畫筆粗細為 5 的畫筆 pen：

```
Pen pen =new Pen(Color.Blue,5);
```

畫筆物件建立後，也可以使用 Color 和 Width 屬性更改畫筆顏色和畫筆粗細，例如：

```
pen.Color=Color.Red;
pen.Width=2;
```

筆刷 Brush 物件

如果要填滿圖形或顯示文字，則可以使用筆刷物件，也可以使用筆刷來建立畫筆。常用的筆刷類別有：

類別	說明
SolidBrush	<p>建立單一顏色的筆刷。</p> <p>例如：建立紅色筆刷，並以紅色筆刷建立畫筆 pen：</p> <pre>SolidBrush sb =new SolidBrush(Color.Red); Pen pen =new Pen(sb,2);</pre>
TextureBrush	<p>建立以圖形物件當作圖案的筆刷。</p> <p>例如：以 <bmp1.bmp> 檔的圖案建立圖案的筆刷，並以此圖案筆刷建立畫筆：</p> <pre>TextureBrush tb = new TextureBrush ("bmp1.bmp"); Pen pen = new Pen(tb,2);</pre>
HatchBrush	<p>建立花紋筆刷。</p> <p>建立花紋筆刷的語法：</p> <pre>HatchBrush 花紋筆刷變數 = new (花紋樣式, 前景顏色, 背景顏色);</pre> <p>例如：建立藍、紅色波浪花紋筆刷，並以此花紋圖案筆刷建立畫筆。</p> <pre>HatchBrush hb = new HatchBrush(HatchStyle.Wave, Color.Blue, Color.Red); Pen pen = new Pen(hb, 10); // 建立花紋畫筆</pre> <p>註：花紋筆刷類別包含在 Drawing2D 的命名空間，因此，必須在程式的第一行引用，亦即要加入下列程式碼：</p> <pre>using System.Drawing.Drawing2D ;</pre>
LinearGradienBrush	<p>建立漸層筆刷。</p> <p>建立漸層筆刷的語法：</p> <pre>LinearGradienBrush 漸層筆刷變數 = new LinearGradienBrush (漸層矩形區域, 前景顏色, 背景顏色, 漸層傾斜角度);</pre> <p>漸層矩形區域是指漸層區塊的大小，通常會定義的比畫筆大，如果定義的比畫筆小，顯示時將會呈現磚塊的效果。漸層矩形區域可以宣告一個矩形區域，例如以整個 pictureBox1 為漸層區域的語法如下：</p> <pre>Rectangle rect1=new Rectangle(0, 0, pictureBox1.Size.Width,</pre>

```
pictureBox1.Size.Height)
```

例如：建立藍、紅色垂直漸層筆刷，並以此漸層筆刷建立畫筆。

```
Rectangle rect1 = new Rectangle(0, 0, pictureBox1.Size.Width,
pictureBox1.Size.Height);
```

```
LinearGradientBrush lgb = new LinearGradientBrush(rect1,
Color.Blue, Color.Red, 90);
```

```
Pen pen = new Pen(lgb, 10); // 建立漸層畫筆
```

註：漸層筆刷類別包含在 `Drawing2D` 的命名空間，因此，必須在程式的第一行引用，亦即要加入下列程式碼：

```
using System.Drawing.Drawing2D;
```

範例：在表單和圖片方塊上建立畫布

以 `pictureBox1` 為藍色畫布，使用單色、圖案、花紋和漸層不同的筆刷當作畫筆，繪製寬度不同的直線。(Pen1.sln)

執行結果



範例說明

開啟已經建立完成的 `Pen1.sln` 專案，本範例示範如何使用筆刷當作畫筆。

加入 `HatchBrush` 和 `LinearGradientBrush` 命名空間。

```
9 using System.Drawing.Drawing2D;
```

程式碼

```
20 Graphics g; //在表單建立繪圖物件
```

```
21 Pen pen;
22
23 private void Form1_Load(object sender, EventArgs e)
24 {
25     g = pictureBox1.CreateGraphics(); // 在 pictureBox1 建立
        Graphics 物件
26     pen = new Pen(Color.Red, 10); // 預設畫筆為紅色畫筆，寬度 10
27 }
28
29 private void radioButton1_CheckedChanged(object sender,
        EventArgs e)
30 {
31     SolidBrush sb = new SolidBrush(Color.Gold);
32     pen = new Pen(sb, 10); // 金色畫筆
33 }
34
35 private void radioButton2_CheckedChanged(object sender,
        EventArgs e)
36 {
37     TextureBrush tb = new TextureBrush(new
        Bitmap(@"C:\VC2008\Ch14\icons\icon1.bmp"));
38     pen = new Pen(tb, 10); // 圖案畫筆
39 }
40
41 private void radioButton3_CheckedChanged(object sender,
        EventArgs e)
42 {
43     HatchBrush hb = new HatchBrush(HatchStyle.Wave, Color.Blue,
        Color.Red);
44     pen = new Pen(hb, 10); // 花紋畫筆
45 }
46
47 private void radioButton4_CheckedChanged(object sender,
        EventArgs e)
48 {
49     Rectangle rect1 = new Rectangle(0, 0, pictureBox1.Size.Width,
        pictureBox1.Size.Height);
```



```

50     LinearGradientBrush lgb = new LinearGradientBrush(rect1,
           Color.Blue, Color.Red, 90);
51     pen = new Pen(lgb, 10); // 漸層畫筆
52 }
53
54 private void button1_Click(object sender, EventArgs e)
55 {
56     g.Clear(Color.LightGreen) ; //畫布顏色為淺綠色
57     pen.Width = 10; // 畫筆寬度
58     for (int i = 0; i <= pictureBox1.Width; i = i + 36)
59     {
60         g.DrawLine(pen, i, 0, i, pictureBox1.Height);
61         pen.Width += 2; // 畫筆寬度
62     }
63 }

```

程式說明

- 9 定義花紋和漸層的筆刷必須引用「using System.Drawing.Drawing2D;」命名空間。
- 20-21 將畫布 g 和畫筆 pen 都定義為全域變數。
- 25 在 pictureBox1 建立畫布。
- 26 預設畫筆為紅色畫筆，寬度為 10。
- 54-63 按繪製圖形按鈕 (button1)，會先將畫布顏色設為淺綠色，並在畫布上依指定的畫筆繪製直線。
- 60 「g.DrawLine(pen, i, 0, i, pictureBox1.Height)」繪製直線，會在下面的章節說明。
- 61 漸漸增加畫筆寬度，每次加 2。
- 29-52 依不同的 RadioButton 選擇不同樣式的畫筆。

14.2 繪製文字和圖形

Graphics 物件提供許多的方法可以繪製文字、直線、矩形、多邊形、圓形、橢圓形、圓弧，也可將圖形填滿。

14.2.1 繪製文字和直線

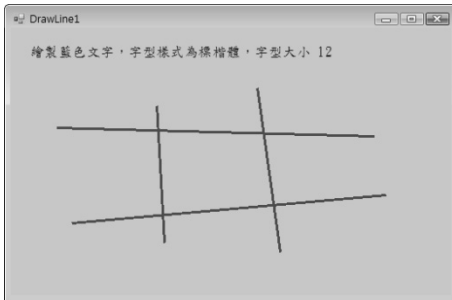
繪製文字和直線的方法：

方法	說明
DrawingString(String,Font,Brush,x,y)	<p>在 (x,y) 座標位置，以 Font 的字型和 Brush 的筆刷，繪製 String 的文字。</p> <p>例如：在座標 (20,20)，以藍色筆刷，繪製「繪製藍色文字，字型樣式為標楷體，字型大小 12」文字：</p> <pre>SolidBrush brush = new SolidBrush(Color.Blue); Font font = new Font("標楷體",12); g.DrawString("繪製藍色文字，字型樣式為標楷體，字型大小 12",font,brush,20,20);</pre>
DrawLine(Pen,x1,y1,x2,y2)	<p>使用 Pen 畫筆，從 (x1,y1) 到 (x2,y2) 座標繪製一條直線。</p> <p>例如：以藍色畫筆，自 (10,20) 到 (50,60) 繪製一條直線：</p> <pre>Graphics g =this.CreateGraphics // 建立 Graphics 物件 Pen pen As =new Pen(Color.Blue,3); g.DrawLine(pen,10,20,50,60);</pre>

範例：繪製文字和直線

以表單為淺綠色畫布，繪製藍色文字，並可使用滑鼠左鍵，繪製紅色直線。(DrawLine1.sln)

執行結果



範例說明

請開啟已經建立完成的 `DrawLine1.sln` 專案，本範例示範如何繪製文字和直線。

程式碼

```

19 Graphics g;           //在表單建立繪圖物件
20 Pen pen=new Pen(Color.Red, 3);
21 int PointCounter=0; // 記錄共已儲存多少的點座標
22 Point[] aryPoint = new Point[99]; // 儲存所有繪製的直線(直線起點,直
    線終點)的座標
23
24 private void Form1_Load(object sender, EventArgs e)
25 {
26     g = this.CreateGraphics();    // 建立 Graphics 物件
27 }
28
29 private void Form1_MouseDown(object sender, MouseEventArgs e)
30 {
31     aryPoint[PointCounter] = new Point(e.X, e.Y); // 將直線起點存
    入 aryPoint()陣列
32 }
33
34 private void Form1_MouseUp(object sender, MouseEventArgs e)
35 {
36     if (e.Button == MouseButton.Left) // 按滑鼠左鍵
37     {
38         PointCounter += 1;
39         aryPoint[PointCounter] = new Point(e.X, e.Y); //將直線終
    點存入 aryPoint()陣列

```

```
40     PointCounter += 1;
41     Form1_Paint(sender, null); // 以 Paint() 事件重繪，這樣即
    使表單縮放，仍會重繪圖形
42 }
43 }
44
45 private void Form1_Paint(object sender, PaintEventArgs e)
46 {
47     int x1, y1, x2, y2;
48     g.Clear(Color.LightGreen);
49     SolidBrush brush = new SolidBrush(Color.Blue);
50     Font font = new Font("標楷體", 12);
51     // 繪製文字
52     g.DrawString("繪製藍色文字，字型樣式為標楷體，字型大小 12", font,
    brush, 20, 20);
53
54     //以兩個點為一組，繪製全部紅色直線
55     for (int i = 0; i <= PointCounter; i = i + 2)
56     {
57         x1 = aryPoint[i].X;    //直線起點座標
58         y1 = aryPoint[i].Y;
59         x2 = aryPoint[i + 1].X; //直線終點座標
60         y2 = aryPoint[i + 1].Y;
61         g.DrawLine(pen, x1, y1, x2, y2); //繪製紅色直線
62     }
63 }
```

程式說明

- 19-21 建立全域變數。
- 22 Point[] aryPoint 用以儲存所有繪製的直線(直線起點,直線終點)的座標。
- 26 在表單上建立繪圖物件。
- 45-63 只要表單大小調整縮小或放大，都會自動觸發 Paint 事件。
- 49-52 繪製藍色文字,文字內容為「繪製藍色文字，字型樣式為標楷體，字型大小 12」。
- 55-62 因為要在 Paint 事件重繪直線，必須記憶所有建立的座標點，然後以兩個點為一組取出重繪，繪製全部已繪製過的紅色直線。
- 61 g.DrawLine(pen, x1, y1, x2, y2) 以 pen 繪筆自 (x1,y1) 至 (x2,y2) 繪製一條直線。

31	當按下滑鼠，將直線起點存入 aryPoint [] 陣列。
39	當放開滑鼠左鍵，將直線終點存入 aryPoint [] 陣列。
38,40	PointCounter 記錄共儲存多少的點。
41	當放開滑鼠左鍵，執行 Form1_Paint(sender, null) 重繪圖形。

14.2.2 繪製矩形和多邊形

繪製矩形和多邊形的方法：

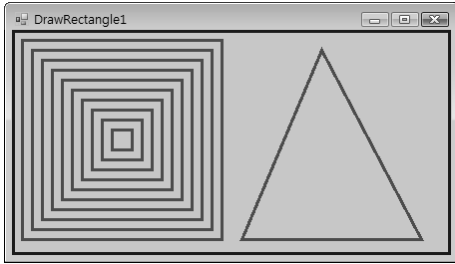
方法	說明
DrawRectangle(Pen,x,y,width,height)	<p>使用 Pen 畫筆 p，以 (x,y) 座標為矩形的左上角，繪製寬 width，高 height 的矩形。</p> <p>例如：以藍色畫筆，自 (10,20) 繪製寬 50，高 50 的矩形：</p> <pre>Graphics g = this.CreateGraphics; Pen p = new Pen(Color.Blue, 5); g.DrawRectangle(p, 10, 20, 50, 50);</pre>
DrawPolygon(Pen,Point())	<p>使用 Pen 畫筆 p，以 Point[] 陣列中的各個點為多邊形的端點，繪製多邊形。</p> <p>例如：以藍色畫筆，繪製三角形。</p> <pre>Graphics g =this.CreateGraphics; Pen p =new Pen(Color.Blue,3); Point[] p_Array = {new Point(310, 20), new Point(230, 210), new Point(410, 210)}; //三角形頂點 g.DrawPolygon(p, p_Array); //繪多邊形 (三角形)</pre>

範例：繪製矩形和多邊形

以表單為淺綠色畫布，繪製藍色矩形外框，紅色由小漸大的矩形外框和紅色三角形。

(DrawRectangle1.sln)

執行結果



範例說明

開啟已經建立完成的 `DrawRectangle1.sln` 專案，本範例示範如何繪製矩形和多邊形。

程式碼

```
19 private void Form1_Paint(object sender, PaintEventArgs e)
20 {
21     Graphics g = this.CreateGraphics();
22     g.Clear(Color.LightGreen); // 淺綠色畫布
23     Pen p = new Pen(Color.Blue, 5);
24     g.DrawRectangle(p, 0, 0, this.ClientSize.Width - 1,
25         this.ClientSize.Height - 1); //外框矩形
26     p.Color = Color.Red;
27     p.Width = 3;
28     for (int i = 1; i <= 10; i++) // 繪製 10 個由小變大的矩形
29         g.DrawRectangle(p, 10 + (i - 1) * 10, 10 + (i - 1) * 10,
30             200 - (i - 1) * 20, 200 - (i - 1) * 20);
31     Point[] p_Array = { new Point(310, 20), new Point(230, 210),
32         new Point(410, 210) };
33     g.DrawPolygon(p, p_Array); //繪多邊形(三角形)
34 }
```

程式說明

- 24 繪藍色矩形外框。
- 27-28 繪製 10 個由小漸漸變大的矩形。
- 29-30 建立 `p_Array[]` 陣列設定多邊形的端點，繪多邊形（本例是三角形）。

14.2.3 繪製曲線

繪圖物件可繪製曲線如橢圓、圓弧、扇形、貝茲曲線和曲線。

繪製曲線的方法如下：

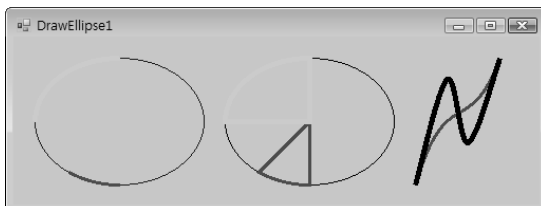
方法	說明
<p>DrawEllipse(Pen,x,y,width,height)</p> <p>或</p> <p>DrawEllipse(Pen,Rectangle)</p>	<p>使用 Pen 畫筆，以 (x,y) 座標為左上角，繪製寬 width，高 height 的橢圓。</p> <p>例如：以藍色畫筆，自 (10,20) 繪製寬 50，高 50 的橢圓。</p> <pre>Graphics g = this.CreateGraphics; Pen p As = new Pen(Color.Blue, 5); g.DrawEllipse(p, 10, 20, 50, 50);</pre> <p>或</p> <pre>Rectangle rect1 = new Rectangle(0, 20, 50, 50); Graphics g = this.CreateGraphics; Pen p = new Pen(Color.Blue, 5); g.DrawEllipse(p, rect1);</pre>
<p>DrawArc(Pen,x,y,width,height,startAngle,sweepAngle)</p> <p>或</p> <p>DrawArc(Pen,Rectangle,startAngle,sweepAngle)</p>	<p>使用 Pen 畫筆，在以 (x,y) 座標為左上角，寬 width，高 height 的橢圓路徑上，自起始角 startAngle 開始繪製圓弧，sweepAngle 為繪製圓弧的長度 (角度)，角度為正代表順時針，角度為負代表逆時針。</p> <p>例如：以藍色畫筆，在以 (20, 20, 160, 120) 的橢圓路徑上繪製橢圓，圓弧起始角為 270 度，逆時針共繪 90 度的圓弧。</p> <pre>Graphics g = this.CreateGraphics; Pen p = new Pen(Color.Blue, 1); g.DrawArc(p, 20, 20, 160, 120, 270, -90);</pre>
<p>DrawPie(Pen,x,y,width,height,startAngle,sweepAngle)</p> <p>或</p> <p>DrawPie(Pen,Rectangle,startAngle,</p>	<p>使用 Pen 畫筆，在以 (x,y) 座標為左上角，寬 width，高 height 的橢圓路徑上，自起始角 startAngle 開始繪製扇形，sweepAngle 為繪製扇形的長度 (角度)，角度為正代表順時針，角度為負代表逆時針。</p>

sweepAngle)	<p>例如：以藍色畫筆，在以 (20, 20, 160, 120) 的橢圓路徑上繪製扇形，扇形起始角為 270 度，逆時針共繪 90 度的扇形。</p> <pre>Graphics g = this.CreateGraphics; Pen p = new Pen(Color.Blue, 1); g.DrawPie(p, 20, 20, 160, 120, 270, -90);</pre>
DrawBezier(Pen,x1,y1,x2,y2,x3,y3,x4,y4)	<p>使用 Pen 畫筆，自 (x1,y1) 至 (x4,y4) 畫貝茲曲線，(x2,y2) 和 (x3,y3) 用以決定貝茲曲線的形狀，貝茲曲線一定要指定四個點。</p> <p>例如：</p> <pre>g.DrawBezier(p, new Point(380, 140), new Point(410, 40), new Point(430, 100), new Point(460, 20));</pre>
DrawCurve(Pen,Point())	<p>使用 Pen 畫筆，以 Point[] 陣列中的各個點為曲線的端點，繪製曲線。</p> <p>例如：</p> <pre>Point[] p_Array[] = {new Point(380, 140), new Point(410, 40), new Point(430, 100), new Point(460, 20)}; g.DrawCurve(p, p_Array);</pre>

範例：繪製橢圓、圓弧、扇形、貝茲曲線和曲線

以表單為淺綠色畫布，繪製兩個橢圓，然後在左邊的橢圓上繪兩個圓弧，在右邊的橢圓上繪兩個扇形，並在最右邊繪製一個紅色貝茲曲線和一個黑色曲線。(DrawEllipse1.sln)

執行結果



範例說明

開啟已經建立完成的 DrawEllipse1.sln 專案，本範例示範如何繪製曲線。

程式碼

```

19 private void Form1_Paint(object sender, PaintEventArgs e)
20 {
21     Graphics g = this.CreateGraphics();
22     g.Clear(Color.LightGreen); //淺綠色畫布
23     Pen p = new Pen(Color.Blue, 1);
24     g.DrawEllipse(p, 20, 20, 160, 120); //橢圓
25
26     p = new Pen(Color.Red, 3); //重設畫筆為紅色
27     g.DrawArc(p, 20, 20, 160, 120, 90, 45); //在左邊的橢圓上以紅色
    繪圓弧(圓弧起始 90 度，順時針繪 45 度的圓弧)
28
29     p = new Pen(Color.Gold, 5); //重設畫筆為金色
30     g.DrawArc(p, 20, 20, 160, 120, 270, -90); //在左邊的橢圓上以金
    色繪圓弧(圓弧起始 270 度，逆時針繪 90 度的圓弧)
31
32     p = new Pen(Color.Black, 1); //重設畫筆
33     Rectangle rect = new Rectangle(200, 20, 160, 120);
34     g.DrawEllipse(p, rect); //畫右邊的橢圓
35
36     p = new Pen(Color.Red, 3); //重設畫筆
37     g.DrawPie(p, rect, 90, 45); //在右邊的橢圓上以紅色繪扇形框(圓弧
    起始 90 度，順時針共繪 45 圓弧)
38
39     p = new Pen(Color.Gold, 5); //重設畫筆
40     g.DrawPie(p, rect, 270, -90); //在右邊的橢圓上以金色繪扇形框(圓
    弧起始 270 度，逆時針共繪 90 圓弧)
41
42     //繪紅色貝茲曲線
43     p = new Pen(Color.Red, 3); //重設畫筆
44     g.DrawBezier(p, new Point(380, 140), new Point(410, 40), new
    Point(430, 100), new Point(460, 20));
45

```

```

46     //繪黑色曲線
47     p = new Pen(Color.Black, 5); //重設畫筆
48     Point[] p_Array = { new Point(380, 140), new Point(410, 40),
                           new Point(430, 100), new Point(460, 20) };
49     g.DrawCurve(p, p_Array);
50 }

```

程式說明

23-24 繪左邊的橢圓。

26-27 在左邊的橢圓上以紅色繪圓弧（圓弧起始角 90 度，順時針繪 45 度的圓弧）。

29-30 在左邊的橢圓上以金色繪圓弧（圓弧起始角 270 度，逆時針繪 90 度的圓弧）。

32-34 畫右邊的橢圓。

36-40 在右邊的橢圓畫兩個扇形。

39-40 在右邊的橢圓上以金色繪扇形框（圓弧起始角 270 度，逆時針繪 90 度圓弧）。

43-44 以四個點繪紅色貝茲曲線。

47-49 以 p_Array[] 陣列為端點，繪黑色曲線。

14.2.4 繪製填滿圖形

繪圖物件也可以使用筆刷物件繪製填滿的圖形，包括實心矩形、實心多邊形、實心橢圓、實心扇形。

繪製填滿圖形的方法如下：

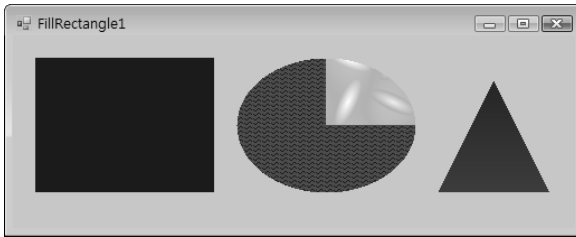
方法	說明
FillRectangle(Brush,x,y,width,h,height)	<p>使用 Brush 筆刷，以 (x,y) 座標為矩形的左上角，繪製寬 width，高 height 填滿的矩形。</p> <p>例：以藍色筆刷，自 (20,20) 繪製寬 160，高 120 的填滿矩形。</p> <pre> Graphics g = this.CreateGraphics; Brush Brush As = new SolidBrush(Color.Blue); g.FillRectangle(Brush, 20, 20, 160, 120); </pre>
FillPolygon(Brush,Point())	<p>使用 Brush 筆刷，以 Point[] 陣列中的各個點為多邊形的端點，繪製填滿的多邊形。</p> <p>例如：以藍紅色漸層色筆刷，繪製填滿的三角形。</p> <pre> Rectangle rect1 = new Rectangle(0, 0, this.Size.Width, </pre>

	<pre> this.Size.Height); //以整個表單為漸層區域 LinearGradientBrush lgb = new LinearGradientBrush(rect1, Color.Blue, Color.Red, 90); Point[] p_Array = {new Point(380, 140), new Point(430, 40), new Point(480, 140)}; g.FillPolygon(lgb, p_Array); </pre>
<pre> FillEllipse(Brush,x,y,width,h eight) 或 FillEllipse(Brush,Rectangle) </pre>	<p>使用 Brush 筆刷，以 (x,y) 座標為左上角，繪製寬 width，高 height 的填滿橢圓。</p> <p>例：以藍色筆刷，自 (20,20) 繪製寬 160，高 120 的填滿橢圓。</p> <pre> Graphics g = this.CreateGraphics; Brush Brush = new SolidBrush(Color.Blue); g.FillEllipse(Brush, 20, 20, 160, 120); </pre>
<pre> FillPie(Brush,x,y,width,height, startAngle,sweepAngle) 或 FillPie(Brush,Rectangle,star tAngle,sweepAngle) </pre>	<p>使用 Brush 筆刷，在以 (x,y) 座標為左上角，寬 width，高 height 的橢圓路徑上，自起始角 startAngle 開始繪製填滿扇形，sweepAngle 為繪製扇形的長度 (角度)，角度為正代表順時針，角度為負代表逆時針。</p> <p>例：以藍色筆刷，在以 (20, 20, 160, 120) 的橢圓路徑上繪製填滿扇形，扇形起始 270 度，逆時針繪 90 度圓弧。</p> <pre> Graphics g = this.CreateGraphics; Brush Brush = new SolidBrush(Color.Blue); g.FillPie(Brush , 20, 20, 160, 120, 270, -90); </pre>

範例：繪製填滿的矩形、橢圓、扇形和多邊形

以表單為淺綠色畫布，繪製填滿的藍色矩形、圖案橢圓、花紋扇形和藍紅漸層色多邊形 (三角形)。(FillRectangle1.sln)

執行結果



範例說明

請開啟已經建立完成的 `FillRectangle1.sln` 專案，本範例示範如何繪製填滿圖形。

命名空間程式碼

```
9 using System.Drawing.Drawing2D;
```

程式碼

```
20 private void Form1_Paint(object sender, PaintEventArgs e)
21 {
22     Graphics g = this.CreateGraphics();
23     g.Clear(Color.LightGreen); // 淺綠色畫布
24
25     Brush Brush = new SolidBrush(Color.Blue); // 藍色筆刷
26     g.FillRectangle(Brush, 20, 20, 160, 120); // 繪填滿的藍色矩形
27
28     TextureBrush tb = new TextureBrush(new
29         Bitmap(@"C:\VC2008\Ch14\icons\icon1.bmp")); // 圖案筆刷
30     Rectangle rect = new Rectangle(200, 20, 160, 120);
31     g.FillEllipse(tb, rect); // 畫右邊填滿圖案的橢圓
32
33     HatchBrush hb = new HatchBrush(HatchStyle.Wave, Color.Blue,
34         Color.Red); // 花紋筆刷
35     g.FillPie(hb, rect, 270, -270); // 在右邊的橢圓上以花紋筆刷繪填滿
36         的扇形(圓弧起始角 270 度，逆時針共繪 270 圓弧)
37
38     // 繪填滿的多邊形
39     Rectangle rect1 = new Rectangle(0, 0, this.Size.Width,
40         this.Size.Height); // 以整個表單為漸層區域
41     LinearGradientBrush lgb = new LinearGradientBrush(rect1,
42         Color.Blue, Color.Red, 90); // 漸層筆刷
```

```

38     Point[] p_Array = { new Point(380, 140), new Point(430, 40),
                             new Point(480, 140) };
39     g.FillPolygon(lgb, p_Array);
40 }

```

程式說明

- 9 定義花紋和漸層的筆刷必須引用「using System.Drawing.Drawing2D;」命名空間。
- 25-26 繪填滿的藍色矩形。
- 28-30 畫填滿圖案的橢圓。
- 32-33 在橢圓上以花紋筆刷繪填滿的扇形（圓弧起始角 270 度，逆時針繪 270 度圓弧）。
- 36-39 以漸層筆刷繪填滿的多邊形（本例是三角形）。

14.2.5 圖形平移、縮放和旋轉

繪圖物件提供平移、縮放和旋轉的方法，其方法如下：

方法	說明
TranslateTransform(x,y)	將繪圖物件右移平移 x 點，下移 y 點。
ScaleTransform(sx,sy)	將繪圖物件 X 軸縮放 sx 倍，Y 軸縮放 sy 倍，比例大於 1 代表放大，小於 1 代表縮小。
RotateTransform(angle)	將繪圖物件旋轉 angle 角度，角度為正代表順時針，角度為負代表逆時針。
ResetTransform()	將繪圖物件還原。

範例：圖形平移、縮放和旋轉

以表單為淺綠色畫布，繪製小精靈，按下**右移**、**縮放**和**旋轉**按鈕，即可操作小精靈。

(TranslateTransform1.sln)

執行結果



範例說明

開啟已經建立完成的 `TranslateTransform1.sln` 專案，本範例示範圖形平移、縮放和旋轉。

程式碼

```

19 Graphics g;
20 Single x=0, y=0, angle=0; // 平移(x,y) 旋轉 angle 角度
21 Single mode = 1;          // 模式
22 Single radiox=1, radioy=1; //縮放比例
23
24 private void Form1_Paint(object sender, PaintEventArgs e)
25 {
26     g = e.Graphics;
27     g.Clear(Color.LightGreen); //淺綠色畫布
28     Brush Brush = new SolidBrush(Color.Blue); //藍色筆刷
29     Rectangle rect = new Rectangle(70, 20, 100, 100); //小精靈繪
        圖區域
30     g.TranslateTransform(x, y); // 平移
31     g.ScaleTransform(radiox, radioy); // 縮放
32     g.RotateTransform(angle); // 旋轉
33     if (mode == 4) //還原
34     {
35         radiox = 1; radioy = 1;
36         angle = 0; x = 0; y = 0;
37         g.ResetTransform();
38     }
39     g.FillPie(Brush, rect, 270, -270); //在右邊的橢圓上以藍色筆刷繪
        填滿的扇形(圓弧起始角 270 度，逆時針共繪 270 圓弧)
40 }
41
42 private void button1_Click(object sender, EventArgs e)

```

```

43 {
44     mode = 1;
45     x += 10;
46     y += 0;
47     this.Refresh(); // 執行 Form1_Paint()
48 }
49
50 private void button2_Click(object sender, EventArgs e)
51 {
52     mode = 2;
53     radiox *= 1.1F;
54     radioy *= 1.1F;
55     this.Refresh(); // 執行 Form1_Paint()
56 }
57
58 private void button3_Click(object sender, EventArgs e)
59 {
60     mode = 3;
61     angle += 10;
62     this.Refresh() ; //執行 Form1_Paint()
63 }
64
65 private void button4_Click(object sender, EventArgs e)
66 {
67     mode = 4;
68     this.Refresh() ; //執行 Form1_Paint()
69 }

```

程式說明

- 19-22 宣告全域變數，預設平移(x,y) 為 (0,0)，縮放比例 (radiox, radioy) 為 (1,1)，旋轉角度 angle 為 0 度。
- 26 在 Paint 事件中也可以以「g=e.Graphics」取得 Graphics 繪圖物件。
- 42-48 按下 **右移按鈕** 每次會向右平移 10 個點。
- 50-56 按下 **縮放按鈕** 每次會縮放 (1.1,1.1)。
- 58-63 按下 **旋轉按鈕** 每次會順時針旋轉 10 度。
- 47 this.Refresh() 用以執行 Form1_Paint()，如此才能將小精靈，按下 **右移、縮放** 和 **旋轉** 按鍵後移到新的位置。

-
- 30-32 不管小精靈移到那個位置，均可以在新位置上連續縮放和旋轉，因此要將右移、縮放和旋轉功能合併一起執行，否則小精靈無法連續動作。
- 33-38 還原。
-

14.2.6 圖檔的讀取和儲存

C# 2008 可以將圖檔載入記憶體，並設定為畫布，然後在畫布上繪製圖形，亦可將畫布上的圖形存檔。

Bitmap 類別

Bitmap 可以說是一張點陣圖的記憶體圖像，它是儲存在記憶體中，表單上並不會顯示，如果要顯示此記憶體圖像，通常必須再透過其他控制項的屬性設定，將 Bitmap 顯示在指定的控制項上。

例如：以 <S04.bmp> 的圖檔當作記憶體畫圖像 bmp，並將記憶體畫圖像 bmp 顯示在 PictureBox1 圖片控制項中。

```
Bitmap bmp = new Bitmap(@"C:\VC2008\ch14\icons\S04.bmp");
pictureBox1.SizeMode = pictureBoxSizeMode.AutoSize; //自動調整大小
pictureBox1.Image = bmp; //顯示在 pictureBox1 圖片控制項中
```

也可以將記憶體圖像宣告為指定寬度和高度的空白圖像。

例如：宣告記憶體圖像 bmp 為 (800,600) 的空白圖像：

```
Bitmap bmp = new Bitmap(800, 600);
```

Bitmap 的重要方法：

方法	說明
FromImage	<p>宣告一個記憶體畫布，畫布內容為 Bitmap 記憶體圖像。</p> <p>例如：宣告 g 為淺藍色的記憶體畫布，畫布內容為 bmp 記憶體圖像，畫布大小和表單相同，畫布顏色為淺綠色：</p> <pre>Bitmap bmp = new Bitmap(this.Width, this.Height); Graphics g = Graphics.FromImage(bmp);</pre>

	<code>g.Clear(Color.LightGreen);</code>
Clone	取得 Bitmap 上指定區域的圖像。
Dispose	釋放 Bitmap。
GetPixel	取得 Bitmap 上指定像素 (Pixel) 的顏色。
SetPixel	設定 Bitmap 上指定像素 (Pixel) 的顏色。
MakeTransparent	去除 Bitmap 的背景顏色 (底色透明)。
Save	將 Bitmap 的圖像存成檔案。

範例：Bitmap 圖檔的讀取和儲存

以 <S04.bmp> 的圖檔當作記憶體圖像 bmp1，並將 bmp1 顯示在 pictureBox1 圖片控制項中，利用 **圖片擷取**、**圖片複製**、**灰階處理** 和 **存檔** 按鈕對 bmp1 作圖檔處理。(BitMap1.sln)

執行結果



範例說明

開啟已經建立完成的 BitMap1.sln 專案，本範例示範記憶體圖像宣告、圖片擷取、圖片複製、灰階處理和圖片存檔。

程式碼

```

19 Bitmap bmp1;
20 private void Form1_Load(object sender, EventArgs e)
21 {
22     bmp1 = new Bitmap(@"C:\VC2008\ch14\icons\S04.bmp"); //以
        "S04.bmp" 的圖檔當作記憶體圖像 bmp1

```

```
23     pictureBox1.SizeMode = PictureBoxSizeMode.AutoSize; //大小自  
      動調整  
24     pictureBox1.Image = bmp1; // 顯示在 PictureBox1 圖片控制項中  
25 }  
26  
27 private void button1_Click(object sender, EventArgs e)  
28 {  
29     Bitmap bmp2;  
30     Rectangle rect = new Rectangle(0, 0, 100, 100);  
31     bmp2 = bmp1.Clone(rect, bmp1.PixelFormat); //擷取 bmp1 上(0,  
      0, 100, 100)的圖像  
32     pictureBox2.Image = bmp2;//透過 PictureBox2 顯示之  
33 }  
34  
35 private void Button2_Click(object sender, EventArgs e)  
36 {  
37     Bitmap bmp3 = new Bitmap(bmp1.Width, bmp1.Height);  
38     for (int i = 0; i <= bmp1.Width - 1; i++)  
39     {  
40         for (int j = 0; j <= bmp1.Height - 1; j++)  
41         {  
42             Color pixelColor = bmp1.GetPixel(i, j); //取得 bmp1 上  
              指定像素 (Pixel) 的顏色  
43             bmp3.SetPixel(i, j, pixelColor); //設定 bmp3 上  
              指定像素 (Pixel) 的顏色  
44         }  
45     }  
46     pictureBox3.Image = bmp3; // 透過 PictureBox3 顯示之  
47 }  
48  
49 private void button3_Click(object sender, EventArgs e)  
50 {  
51     Bitmap bmp4 = new Bitmap(bmp1.Width, bmp1.Height);  
52     for (int i = 0; i <= bmp1.Width - 1; i++)  
53     {  
54         for (int j = 0; j <= bmp1.Height - 1; j++)  
55         {
```

```

56         Color pixelColor = bmp1.GetPixel(i, j); //取得 bmp1 上
           指定像素 (Pixel) 的顏色
57         int a = pixelColor.A; // 分離像素的屬性
58         int r = pixelColor.R;
59         int b = pixelColor.B;
60         int g = pixelColor.G;
61
62         a = Convert.ToInt32(a * 0.9); //透明度 90%
63         r = 255 - r; // 灰階處理
64         g = 255 - g;
65         b = 255 - b;
66         Color newColor = Color.FromArgb(a, r, g, b);
67         bmp4.SetPixel(i, j, newColor); // 設定 bmp4 顏色為灰階
           處理, 透明度 90%
68     }
69 }
70     pictureBox4.Image = bmp4; // 透過 PictureBox4 顯示之
71 }
72
73 private void button4_Click(object sender, EventArgs e)
74 {
75     bmp1.Save("newFile.bmp");
76 }

```

程式說明

22 以 <S04.bmp> 的圖檔當作記憶體圖像 bmp1。

23-24 將記憶體圖像 bmp1 顯示在 pictureBox1 圖片控制項中。

27-33 按下圖片擷取按鈕, 將擷取 bmp1 上(0, 0, 100, 100)的圖像, 並放入 bmp2 中, 然後透過 pictureBox2 顯示出來。

35-47 按下圖片複製按鈕會以 GetPixel() 方法和 SetPixel() 方法將 bmp1 上所有像素 (Pixel) 複製到 bmp3, 然後透過 pictureBox3 顯示出來。所以 pictureBox3 顯示的是 pictureBox1 複製的圖像。

49-71 按下灰階處理按鈕會以 GetPixel() 將所有的像素取出, 並將像素的 ARGB 屬性分離, 然後以灰階處理的公式「 $f=255-f$ 」完成灰階運算。

57-60 將像素的 ARGB 屬性分離。

62-65 透明度 90%, 灰階處理公式 $f=255-f$ 。

66 「Color newColor = Color.FromArgb(a, r, g, b)」將新的 ARGB 屬性組合成 newColor。

67 設定 bmp4 顏色為灰階處理, 透明度 90%。

73-76 按下存檔按鈕會將 bmp1 存成 <newFile.bmp> 檔。

使用 Bitmap 物件建立 Graphics 物件

Bitmap 僅是一張記憶體的圖像，無法繪製圖形，如果要用在 Bitmap 繪製圖形，必須使用 Graphics 物件的 FromImage 方法，將 Bitmap 物件建立成一個 Graphics 物件，也就是以 Bitmap 圖像建立一個記憶體畫布。

例如：以 pictureBox1 圖片，建立記憶體圖像 bmp，再以此記憶體圖像 bmp，建立一個可以繪製圖形的 Graphics 物件 myDraw，然後在 myDraw 記憶體畫布上畫圖：

```
// bmp 的大小和 pictureBox1 相同
Bitmap bmp = new Bitmap(this.PictureBox1.Width,
    this.PictureBox1.Height);
// 以記憶體圖像 bmp 建立 myDraw 記憶體畫布
Graphics myDraw = Graphics.FromImage(bmp);
MyDraw.Clear(this.pictureBox1.BackColor); //畫布背景色
MyDraw.DrawLine(new pen(Color.Red,2),x,y,e.X,e.Y); //可以繪圖了
```

在畫布上顯示 Bitmap 物件

Bitmap 可以說是一張記憶體的圖像，表單上並不會顯示，但可以透過 Graphics 物件的 DrawImage() 方法，將 Bitmap 圖像顯示在另外的畫布（例如表單或圖片）上來顯示。

DrawImage() 的方法如下：

方法	說明
DrawImage(Bitmap,x,y)	在 (x,y) 座標顯示 Bitmap 圖像。
DrawImage(Bitmap,x,y, width,height)	在 (x,y) 座標以 (width,height) 的大小顯示 Bitmap 圖像。

例如：將以記憶體圖像 bmp 建立的 myDraw 記憶體畫布的圖像，以 pictureBox1 將圖片顯示出來：

```
//以 pictureBox1 建立畫布 g
Graphics g = pictureBox1.CreateGraphics;
//設記憶體圖像 bmp 的大小和 pictureBox1 相同
```

```

Bitmap bmp = new Bitmap(pictureBox1.Width,pictureBox1.Height)
// 以記憶體圖像 bmp 建立 MyDraw 畫布
Graphics myDraw = Graphics.FromImage(bmp);
MyDraw.Clear(pictureBox1.BackColor); //畫布背景色
// 直線是繪在 MyDraw 畫布(也是記憶體圖像 bmp 中),其實是用 MyDraw 繪圖物件繪
// 在記憶體圖像 bmp 中
MyDraw.DrawLine(pen,x,y,e.X,e.Y); //可以繪圖了
// 模擬繪圖中... 但這些圖像都只繪在記憶體圖像 bmp 中,並不會顯示出來
// 以 DrawImage 將記憶體圖像 bmp 顯示在 pictureBox1,亦即表單 g 中
g.DrawImage(bmp, 0, 0);
// 以下是常犯的語法錯誤,因為使用的 MyDraw 是不對的。
// g.DrawImage(MyDraw, 0, 0);

```

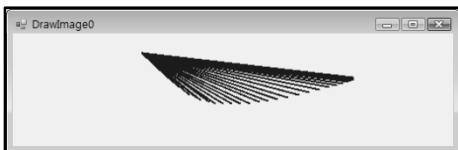
讀到這裡,是否會覺得模糊呢?甚麼是記憶體圖像、記憶體畫布,實在有點抽象。接下來我們要用這個觀念解決繪圖中的殘影問題,或許會讓您更有體會一些。

當我們繪製一條直線時,正確動作是按下滑鼠左鍵 (p1 點),開始以 p1 為起點,滑鼠移動到新的位置 (p2 點),會出現一條 p1 到 p2 暫時性的直線,用以提示要繪製直線的輪廓,直到放開滑鼠 (p3 點),才真正由 p1 到 p3 繪製一條直線。

範例：殘影的讀取和儲存

本例中可以滑鼠左鍵繪製兩點間的一條直線。因未使用記憶體圖像、記憶體畫布繪製,繪製一條直線時,將會有如下殘影問題。(DrawImage0.sln)

執行結果



程式碼

```

19 Graphics g;
20 Pen pen = new Pen(Color.Blue, 2);
21 Point point;

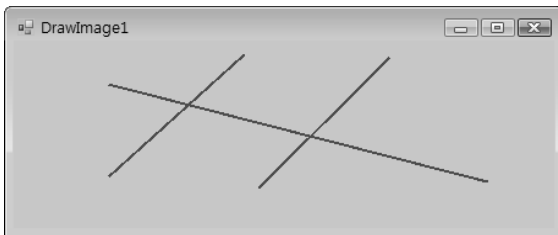
```

```
22 private void Form1_Load(object sender, EventArgs e)
23 {
24     g = this.CreateGraphics(); // 建立 Graphics 物件
25 }
26
27 private void Form1_MouseDown(object sender, MouseEventArgs e)
28 {
29     point = new Point(e.X, e.Y);
30 }
31
32 private void Form1_MouseMove(object sender, MouseEventArgs e)
33 {
34     if (e.Button == MouseButtons.Left) // 按滑鼠左鍵
35         g.DrawLine(pen, point.X, point.Y, e.X, e.Y);
36 }
37
38 private void Form1_MouseUp(object sender, MouseEventArgs e)
39 {
40     g.DrawLine(pen, point.X, point.Y, e.X, e.Y);
41 }
```

範例：繪製紅色直線 (解決殘影問題)

當滑鼠按下左鍵，即可繪製紅色直線，放開即完成紅色直線的繪製。(DrawImage1.sln)

執行結果



範例說明

開啟已經建立完成的 DrawImage1.sln 專案，本範例示範解決殘影問題。

程式碼

```
19 Graphics g;
20 Pen pen = new Pen(Color.Red, 2);
21 int x, y;
22 Bitmap bmp;
23 Graphics Draw;
24 private void Form1_Load(object sender, EventArgs e)
25 {
26     g = this.CreateGraphics(); //建立 Graphics 物件(整個表單)
27     bmp = new Bitmap(this.Width, this.Height); //整個表單的大小
28     Draw = Graphics.FromImage(bmp); //以 bmp 建立 記憶體畫布 Draw
29     Draw.Clear(Color.LightGreen); // 記憶體畫布 Draw 的顏色為淺綠色
30 }
31
32 private void Form1_MouseDown(object sender, MouseEventArgs e)
33 {
34     x = e.X; y = e.Y; //記憶直線開始的位置
35 }
36
37 private void Form1_MouseMove(object sender, MouseEventArgs e)
38 {
39     if (e.Button == MouseButtons.Left) // 按滑鼠左鍵
40     {
41         g.DrawImage(bmp, 0, 0); // 在表單上顯示 bmp 記憶體圖像
42         g.DrawLine(pen, x, y, e.X, e.Y); //在表單上繪暫態直線
43     }
44 }
45
46 private void Form1_MouseUp(object sender, MouseEventArgs e)
47 {
48     Draw.DrawLine(pen, x, y, e.X, e.Y); //在記憶體畫布 Draw 繪直線
49 }
50
51 private void Form1_Paint(object sender, PaintEventArgs e)
52 {
53     g.DrawImage(bmp, 0, 0); // 在表單上顯示 bmp 記憶體圖像
```

54 }

程式說明

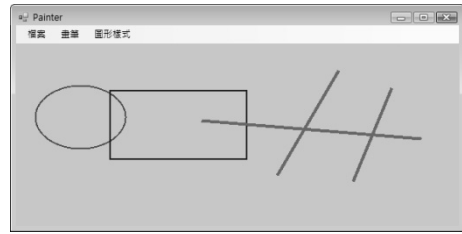
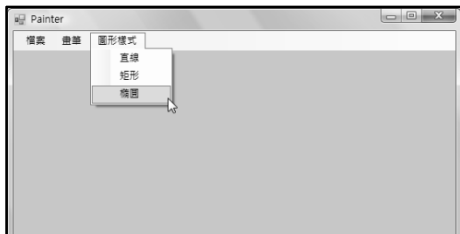
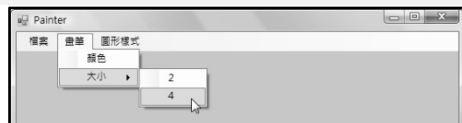
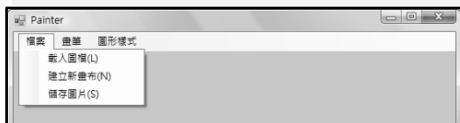
- 19-23 宣告全域變數。
- 26 以表單建立畫布 g。
- 27-28 以 bmp 建立記憶體畫布 Draw，bmp 的大小和表單相同。
- 29 記憶體畫布 Draw 的顏色為淺綠色。
- 41 在表單上顯示 bmp 記憶體圖像。實際圖像是透過記憶體畫布 Draw 畫在 bmp 記憶體圖像區，但因記憶體畫布 Draw 僅存在記憶體中，無法顯示在表單中，所以再透過畫布 g 將 bmp 記憶體圖像畫在表單中（顯示在表單中）。
- 42 在表單上繪製一直線，因為每次移動滑鼠都會先將 bmp 記憶體圖像繪製在表單上，然後再繪製此一直線，所以感覺上這條直線是一條提示直線。
- 48 在記憶體畫布 Draw 繪直線，真正要繪製的直線是畫在記憶體畫布 Draw 中，但螢幕上無法顯示，所以要透過「g.DrawImage(bmp, 0, 0)」再將之顯示在畫布 g 上。
- 53 當表單縮放時，會觸發 Paint() 事件，表單上會顯示 bmp 記憶體圖像。

以前當學生時，小畫家是經常繳交的作業，它看似不難，但其實要寫的完整卻相當不易，下面的例子，即是要以此為例，這個例子為避免太複雜，只有畫直線、矩形、橢圓的功能，但其實已把畫圖的架構表現出來。

範例：簡易小畫家

簡易小畫家可以設定畫筆的顏色和粗細，並選擇在新的畫布或以舊有圖檔當作畫布，然後在畫布上畫直線、矩形或橢圓，也可將繪製後的圖形存檔。(Painter.sln)

執行結果



範例說明

開啟已經建立完成的簡易小畫家 Painter.sln 專案。

程式碼

```

19 Graphics g; // 全域變數的表單畫布
20 Pen pen = new Pen(Color.Red, 2);
21 int x, y; // 記憶繪圖開始的位置
22 Bitmap bmp; // 全域變數的記憶體圖像
23 Graphics Draw; // 全域變數的記憶體畫布
24 String File; // 圖檔名稱
25 int style; // 圖形樣式
26 private void Form1_Load(object sender, EventArgs e)
27 {
28     g = this.CreateGraphics(); // 建立 Graphics 物件 (整個表單)
29     bmp = new Bitmap(this.Width, this.Height); // 整個表單的大小
30     Draw = Graphics.FromImage(bmp); // 以 bmp 建立 記憶體畫布 Draw
31     Draw.Clear(Color.LightGreen); // 設記憶體畫布 Draw 的顏色為淺綠色
32 }
33
34 private void Form1_MouseDown(object sender, MouseEventArgs e)
35 {
36     x = e.X; y = e.Y; // 記憶繪圖開始的位置
37 }
38
39 private void Form1_MouseMove(object sender, MouseEventArgs e)
40 {
41     if (e.Button == MouseButton.Left) // 按滑鼠左鍵
42     {
43         g.DrawImage(bmp, 0, 0); // 在表單上顯示 bmp 記憶體圖像
44         // 計算繪圖的矩形區域
45         Rectangle rect = new Rectangle(Math.Min(x, e.X),
46             Math.Min(y, e.Y), Math.Abs(e.X - x), Math.Abs(e.Y - y));
47         switch (style)
48         {
49             case 1:
50                 g.DrawLine(pen, x, y, e.X, e.Y); // 在表單上繪動態直線

```

```
50             break;
51         case 2:
52             g.DrawRectangle(pen, rect); //在表單上繪暫態矩形
53             break;
54         case 3:
55             g.DrawEllipse(pen, rect);    //在表單上繪暫態橢圓
56             break;
57     }
58 }
59 }
60
61 private void Form1_MouseUp(object sender, MouseEventArgs e)
62 {
63     //計算繪圖的矩形區域
64     Rectangle rect = new Rectangle(Math.Min(x, e.X), Math.Min(y,
65         e.Y), Math.Abs(e.X - x), Math.Abs(e.Y - y));
66     switch (style)
67     {
68         case 1:
69             Draw.DrawLine(pen, x, y, e.X, e.Y); // 在記憶體畫布
70             Draw 繪直線
71             break;
72         case 2:
73             Draw.DrawRectangle(pen, rect);    // 在記憶體畫布
74             Draw 繪矩形
75             break;
76         case 3:
77             Draw.DrawEllipse(pen, rect);    // 在記憶體畫布
78             Draw 繪橢圓
79             break;
80     }
81 }
82
83 private void Form1_Paint(object sender, PaintEventArgs e)
84 { g.DrawImage bmp, 0, 0; } // 在表單上顯示 bmp 記憶體圖像
85 }
```

```
82 private void 儲存圖片 ToolStripMenuItem_Click(object sender,
    EventArgs e)
83 {
84     saveFileDialog1.Filter =
        "bmp (*.bmp) | *.bmp | jpg (*.jpg) | *.jpg";
85     if (saveFileDialog1.ShowDialog() == DialogResult.OK)
86     {
87         File = saveFileDialog1.FileName;
88         bmp.Save(File);
89     }
90 }
91
92 private void 載入圖檔 ToolStripMenuItem_Click(object sender,
    EventArgs e)
93 {
94     openFileDialog1.Filter =
        "bmp (*.bmp) | *.bmp | jpg (*.jpg) | *.jpg";
95     openFileDialog1.FileName = "b1.bmp";
96     if (openFileDialog1.ShowDialog() == DialogResult.OK)
97     {
98         File = openFileDialog1.FileName;
99         bmp = new Bitmap(File); // 整個表單的大小
100        Draw = Graphics.FromImage(bmp); // 以 bmp 建立 記憶
            體畫布 Draw
101        this.Size = bmp.Size; //重設表單大小
102        g = this.CreateGraphics(); //建立 Graphics 物件(整個表單)
103        g.DrawImage(bmp, 0, 0); //在表單上顯示 bmp 記憶體圖像
104    }
105 }
106
107 private void 建立新畫布 ToolStripMenuItem_Click(object sender,
    EventArgs e)
108 {
109     bmp = new Bitmap(600, 300); //表單大小 (600, 300)
110     Draw = Graphics.FromImage(bmp); // 以 bmp 建立 記憶體畫
        布 Draw
111     Draw.Clear(Color.LightGreen); //記憶體畫布 Draw 的顏
        色為淺綠色
```

```
112     this.Size = bmp.Size;        // 重設表單大小 (600, 300)
113     g = this.CreateGraphics(); //建立 Graphics 物件 (整個表單)
114     g.DrawImage(bmp, 0, 0);     // 在表單上顯示 bmp 記憶體圖像
115 }
116
117 private void 顏色 ToolStripMenuItem_Click(object sender,
        EventArgs e)
118 {
119     if (colorDialog1.ShowDialog() == DialogResult.OK)
120         pen.Color = colorDialog1.Color; //設定繪圖的顏色
121 }
122
123 private void ToolStripMenuItem2_Click(object sender, EventArgs
        e)
124 { pen.Width = Convert.ToInt32(ToolStripMenuItem2.Text); } //
        細筆
125
126 private void ToolStripMenuItem3_Click(object sender, EventArgs
        e)
127 { pen.Width = Convert.ToInt32(ToolStripMenuItem3.Text); } //
        粗筆
128
129 private void 直線 ToolStripMenuItem_Click(object sender,
        EventArgs e)
130 { style = 1; } //直線
131
132 private void 矩形 ToolStripMenuItem_Click(object sender,
        EventArgs e)
133 { style = 2; } // 矩形
134
135 private void 橢形 ToolStripMenuItem_Click(object sender,
        EventArgs e)
136 { style = 3; } // 橢圓
```

程式說明

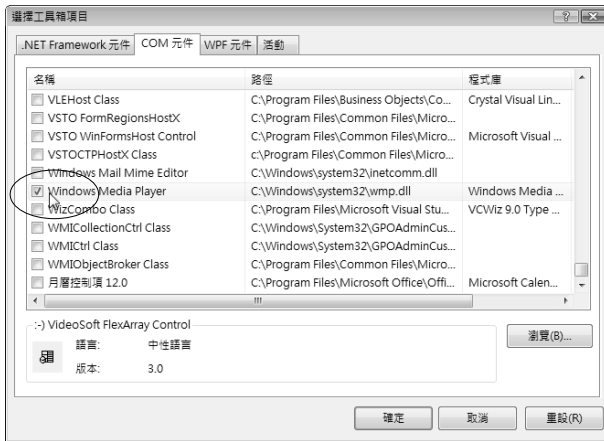
- 19-25 設定全域變數。
- 28-31 預設以表單建立畫布，並以整個表單的大小建立記憶體圖像 bmp，再以 bmp 建立淺綠色的記憶體畫布 Draw。

-
- 36 當按下滑鼠，以 x 、 y 分別記錄滑鼠的初始位置。
- 46-57 利用全域變數 `Style` 判別繪圖模式，因為繪矩形和橢圓時繪製方向並不一定是由左上往右下，第 45 行即是在計算此繪製矩形和橢圓的矩形區域。
- 41-58 這部份程式應該是本範例最重要的關鍵，也就是在處理上一個範例討論的殘影問題。先將記憶體圖像 `bmp` 畫在表單畫布 `g` 上，然後在表單畫布 `g` 繪製圖形，因為每次移動滑鼠時都會先將記憶體圖像 `bmp` 繪製在表單上，然後再繪製圖形，所以感覺上這個圖形是一個提示圖形。
- 61-77 放開滑鼠則實際繪製圖形，圖形是畫在記憶體畫布 `Draw` 中（而不是表單畫布 `g` 中），其實是用 `Draw` 繪圖物件繪在記憶體圖像 `bmp` 中。
- 79-80 當表單畫布縮放時更新顯示。
- 82-90 儲存繪製的圖形。
- 92-105 以圖檔建立新的表單畫布。
- 107-115 建立新的表單畫布。
- 117-121 以 `colorDialog` 設定繪圖的顏色。
- 123-136 選擇劃筆的粗細和顏色。
-

14.3 多媒體

14.3.1 多媒體播放程式

播放多媒體音放的最簡單方法，就是利用 Windows Media Player 來播放，但 Windows Media Player 預設並未出現在工具箱中，必須自行加入。請選擇功能表 **工具 \ 選擇工具箱項目 \ COM 元件**，勾選 **Windows Media Player** 後按 **確定** 鈕。設定完成後，工具箱即會出現 Windows Media Player 控制項。



Windows Media Player 元件的屬性：

屬性	說明
URL	播放的檔名或 URL 網址。
stretchToFit	是否自動調整成視窗的大小，預設值為 False ，表示不自動調整。
fullScreen	是否全螢幕顯示，預設值為 False ，表示不以全螢幕顯示。
playState	多媒體播放狀態。
settings	以 Settings 物件，設定 Windows Media Player 元件。
Ctlcontrols	以 Ctlcontrols 物件，設定 Windows Media Player 的播放、暫停或停止。

Settings 物件的屬性：

屬性	說明
autoStart	是否自動播放，預設值為 False ，表示不自動播放。
mute	是否靜音，預設值為 False ，表示不是靜音。
PlayCount	設定播放的次數。
volume	設定音量，其值是 1~100。

Ctlcontrols 物件的方法：

方法	說明
play	繼續播放。
pause	暫停播放。
stop	停止播放。

範例：多媒體播放

以 **Windows Media Player** 播放多媒體檔案，並設定每次自動播放一次。如果多媒體已在播放中，則可以將表單放大為全螢幕或縮小為原來大小，在表單中自定三個按鈕，分別用來手動控制多媒體的播放、暫停和停止。(MediaPlayer1.sln)

執行結果：



(可以開啟 <C:\VC2008\ch14\完成參考\MediaPlayer1\MediaPlayer1.sln> 專案觀看結果)

動手實作

1. 新增專案：新增 **Windows Form 應用程式** 專案，專案名稱為「MediaPlayer1」。
2. 從工具箱中加入一個 WindowsMediaPlayer、一個 MenuStrip、三個 Button、一個 OpenFileDialog，並設定如下屬性：

物件類別	Name 屬性	屬性設定	說明
WindowsMedia Player	axWindowsMedi aPlayer1	Dock=Fill	將表單填滿。
MenuStrip	menuStrip1	加入三個 ToolStripMenuItem 項目， 檔案 ToolStripMenuItem、 全螢幕 ToolStripMenuItem、調整 成視窗大小 ToolStripMenuItem。	加入載入多媒體檔案、全螢幕、調整成視窗大小三個 ToolStripMenuItem 項目。
Button	button1 到 button3 按鈕	無	自定的按鈕：可以控制多媒體的播放、暫停和停止。
OpenFileDialog	openFileDialog1	無	載入多媒體檔案對話方塊。

設計完成畫面如下：



- 1 axWindowsMediaPlayer1。
- 2 menuStrip1。
- 3 button1 到 button3 按鈕。
- 4 openFileDialog1。

3. 加入程式碼。


```
19 private void Form1_Load(object sender, EventArgs e)
20 {
21     openFileDialog1.FileName =
        @"C:\VC2008\ch14\music\songbird.mp3";
22     axWindowsMediaPlayer1.Dock = DockStyle.Fill; //將表單填滿
23     axWindowsMediaPlayer1.stretchToFit = true; //自動調整成視窗的
        大小
24     axWindowsMediaPlayer1.URL = openFileDialog1.FileName;
25     axWindowsMediaPlayer1.settings.playCount = 1; //播放一次
26     axWindowsMediaPlayer1.settings.autoStart = true; //自動播放
27 }
28
29 private void button1_Click(object sender, EventArgs e)
30 {
31     axWindowsMediaPlayer1.Ctlcontrols.play(); //播放
32 }
33
34 private void button2_Click(object sender, EventArgs e)
35 {
36     axWindowsMediaPlayer1.Ctlcontrols.pause(); //暫停
37 }
38
39 private void button3_Click(object sender, EventArgs e)
40 {
41     axWindowsMediaPlayer1.Ctlcontrols.stop(); //停止
42 }
43
44 private void 檔案 FToolStripMenuItem_Click(object sender,
        EventArgs e)
45 {
46     if (openFileDialog1.ShowDialog() == DialogResult.OK) // 開
        啟多媒體檔案
47     {
48         this.Text = "多媒體檔名：" + openFileDialog1.FileName;
49         axWindowsMediaPlayer1.URL = openFileDialog1.FileName;
50     }
51 }
```

```
52
53 private void 全螢幕 FToolStripMenuItem_Click(object sender,
    EventArgs e)
54 {
55     // 在播放模式才允許調整成全螢幕
56     if (axWindowsMediaPlayer1.playState ==
        WMPLib.WMPPlayState.wmppsPlaying)
57         axWindowsMediaPlayer1.fullScreen = true;
58 }
59
60 private void 調整成視窗大小 SToolStripMenuItem_Click(object sender,
    EventArgs e)
61 {
62     // 在播放模式才允許調整成視窗大小
63     if (axWindowsMediaPlayer1.playState ==
        WMPLib.WMPPlayState.wmppsPlaying)
64         axWindowsMediaPlayer1.fullScreen = false;
65 }
```

程式說明

21-26 預設開啟 <C:\VC2008\ch14\music\songbird.mp3>，設定自動播放一次。
46-51 以開啟舊檔對話方塊開啟多媒體檔案後。
53-58 只有在播放模式才允許將視窗調整成全螢幕。
60-65 只有在播放模式才允許將視窗復原。
41 按手動結束按鈕執行「axWindowsMediaPlayer1.Ctlcontrols.stop()」來停止播放。

14.3.2 System.Media 播放 .wav 音效檔和系統音效

System.Media 命名空間包含用來播放 .wav 音效檔和系統音效。

在這個命名空間中包含下列類別，C# 2008 提供的 **System.Media.SoundPlayer** 類別可以直接播放 .wav 音效檔，**System.Media.SystemSounds** 類別可以播放系統音效。

SoundPlayer

SoundPlayer 類別可以直接播放 .wav 音效檔，**SoundPlayer** 類別常用的方法如下：

播放模式	說明
Play	使用新執行緒播放 .wav 檔，如果 .wav 檔還未載入則先將其載入。
PlayLooping	使用新執行緒重複循環播放 .wav 檔。
PlaySync	方法使用目前的執行緒播放 .wav 檔，可在載入完成前防止執行緒處理其他訊息。
Stop	停止播放。

例如：重複循環播放 <chimes.wav> 音效檔。

```
SoundPlayer simpleSound = new
    System.Media.SoundPlayer(@"c:\Windows\Media\chimes.wav");
simpleSound.PlayLooping(); //重複循環播放
```

SystemSounds

SystemSounds 類別，可以播放系統音效，SystemSounds 提供的音效有：

音效	說明
Asterisk	檔案下載完畢音效。
Beep	提示訊息視窗音效。
Exclamation	警告音效。
Hand	指令錯誤音效。
Question	無法確認音效。

例如：播放提示訊息視窗音效。

```
System.Media.SystemSounds.Beep.Play(); //提示訊息視窗音效
```

範例：以 System.Media.SoundPlayer、System.Media.SystemSounds 播放 .wav 的音效檔和系統音效

以 System.Media.SoundPlayer 播放 .wav 的音效檔，以 System.Media.SystemSounds 播放系統音效，按下 **載入 wav** 按鈕，可以開啟音效檔，按 **播放 wav 檔** 按鈕會播放 .wav 檔。

wav 檔的播放模式可以選擇 Play 或 PlayLooping，按下 **播放系統音效** 可以選擇播放 Asterisk 或 Beep 的系統音效。(AudioPlayer1.sln)

執行結果：



範例說明

開啟已經建立完成的 AudioPlayer1.sln 專案，可以播放 .wav 的音效檔和系統音效。

加入命名空間程式碼

```
9 using System.Media; // 新加入的命名空間
10 using System.IO; // 新加入的命名空間
```

程式碼

```
22 String file=@"c:\Windows\Media\chimes.wav"; //音效檔檔名預設
23 int PlayStyle = 1; //預設音效播放模式
24 int PlaySystemStyle = 1; //預設系統音效播放模式
25 SoundPlayer simpleSound = new
    System.Media.SoundPlayer(@"c:\Windows\Media\chimes.wav");
26
27 private void Form1_Load(object sender, EventArgs e)
28 {
29     openFileDialog1.Filter = "wav 音效檔 (*.wav) | *.wav";
30     openFileDialog1.InitialDirectory = @"C:\Windows\Media"; //預
    設的目錄
31 }
32
33 private void btnLoad_Click(object sender, EventArgs e)
34 {
35     if (openFileDialog1.ShowDialog() == DialogResult.OK) // 開
    啟音效檔檔案
```

```
36     {
37         this.Text = "音效檔檔名：" + openFileDialog1.FileName; //
           顯示檔名
38         file = openFileDialog1.FileName;
39     }
40 }
41
42 private void btnPlay_Click(object sender, EventArgs e)
43 {
44     FileInfo f = new System.IO.FileInfo(file);
45     if (f.Exists) // 確認檔案存在，以免誤動作
46     {
47         simpleSound = new System.Media.SoundPlayer(file);
48         if (PlayStyle == 1) // 播放音效檔
49             simpleSound.Play();
50         else if (PlayStyle == 2) // 重複循環播放
51             simpleSound.PlayLooping();
52     }
53 }
54
55 private void radioButton3_CheckedChanged(object sender,
           EventArgs e)
56 { PlayStyle = 1; } // 播放音效檔
57
58 private void radioButton4_CheckedChanged(object sender,
           EventArgs e)
59 { PlayStyle = 2; } // 重複循環播放
60
61 private void btnStop_Click(object sender, EventArgs e)
62 { simpleSound.Stop(); }
63
64 private void BtnPlaySystem_Click(object sender, EventArgs e)
65 {
66     if (PlaySystemStyle == 1) // 播放系統音效檔
67         System.Media.SystemSounds.Asterisk.Play(); // 檔案下載完
           畢音效
68     else if (PlaySystemStyle == 2)
```

```
69         System.Media.SystemSounds.Beep.Play();           //提示訊息視  
           窗音效  
70     }  
71  
72     private void radioButton1_CheckedChanged(object sender,  
           EventArgs e)  
73     { PlaySystemStyle = 1; } // 播放 Asterisk 檔案下載完畢音效  
74  
75     private void radioButton2_CheckedChanged(object sender,  
           EventArgs e)  
76     { PlaySystemStyle = 2; } // 播放 Beep 提示訊息視窗音效
```

程式說明

- 9-10 加入命名空間，「System.Media」用以播放音效，「System.IO」則用以檢查檔案。
- 22-25 載入預設的音效檔，預設播放的模式。
- 29-30 設定開啟舊檔對話方塊的預設路徑和副檔名。
- 42-53 依不同的模式播放選取的音效檔。
- 64-70 依不同的模式播放系統音效。
- 55-59 選取不同的音效播放模式。
- 72-76 選取不同的系統音效播放模式。
- 62 按停止播放按鈕執行「System.Media.SoundPlayer.Stop()」來停止播放。

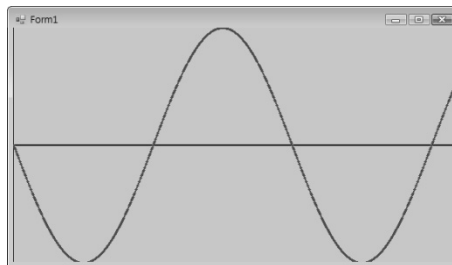
習作園地

一、選擇題

- () C# 2008 繪製直線時，必須使用下列那一個物件來指定線條的色彩和粗細？
(A)Point (B)Pen (C)Brush (D)Pixel
- () C# 2008 繪製文字內容時，必須使用下列那一個物件來當作筆刷？
(A)Point (B)Pen (C)Brush (D)Pixel
- () 繪製圓弧「DrawArc(Pen,x,y,width,height,startAngle,sweepAngle)」中，若 sweepAngle 為負代表意義為何？
(A)順時針 (B)逆時針 (C)停止繪製圓弧 (D)不正確
- () 將 Bitmap 物件建立成一個 Graphics 物件，必須使用下列何種方法？
(A)Clone (B)Save (C)DrawImage (D)FromImage
- () C# 2008 提供的 System.Media.SystemSounds 類別可以直接播放系統音效和甚麼類型的音效檔？
(A).wav (B).mid (C).mp3 (D).wmv

二、實作題

- 以 DrawEllipse 繪製小橢圓，在表單上繪出紅色正弦波圖形。

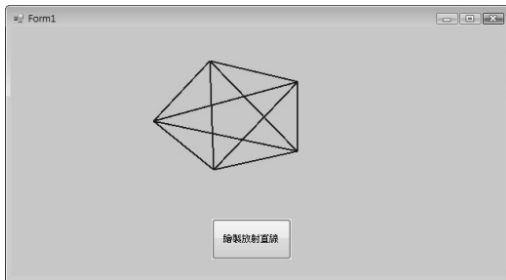


【提示】

正弦波的公式是 $y=\sin(x)$ ， x 必須使用徑度，角度和徑度的轉換公式為 $\text{angle} = x / 180 * \text{Math.PI}$ ，本例中要注意型別轉換的問題。

```
// 以紅色繪正弦波
p = new Pen(Color.Red, 2);
int h = y1;
double angle, y;
float tmpy, tmpx;
for (double x = 0; x <= x2; x++)
{
    angle = x / 180 * Math.PI;
    y = Convert.ToDouble(y1) + Math.Sin(angle) * h;
    tmpx = Convert.ToSingle(x);
    tmpy = Convert.ToSingle(y);
    g.DrawEllipse(p, tmpx, tmpy, 1, 1); //繪製紅色圓點
}
}
```

2. 以按滑鼠左鍵可以設定各直線的頂點，按 **繪製放射直線** 就可以繪製各頂點互相連接的放射直線。



以按滑鼠左鍵設定五個頂點，按 **繪製放射直線** 執行的結果。

【提示】

- 一、建立陣列 `Point[] aryPoint=new Point [99]`，儲存所有繪製的直線的頂點座標。
- 二、當按下滑鼠左鍵時將直線頂點座標記錄到 `aryPoint[]` 陣列，並繪製一個小點。

```
if (e.Button == MouseButton.Left) // 按滑鼠左鍵
{
    aryPoint[PointCounter] = new Point(e.X, e.Y); //將直線頂點存入
    aryPoint[]陣列
    PointCounter += 1;
    g.DrawEllipse(pen, e.X, e.Y, 1, 1); //繪小點
}
}
```

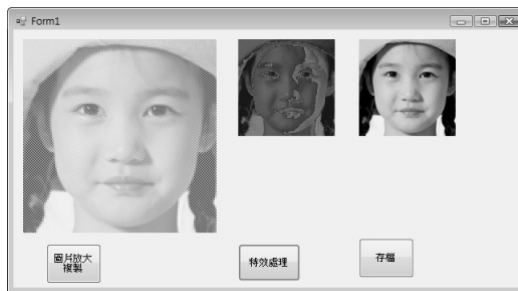

三、繪出放射直線圖形

```

for (int i = 0; i <= PointCounter - 1; i++)
{
    for (int j = 0; j <= PointCounter - 1; j++)
        g.DrawLine(pen, aryPoint[i].X, aryPoint[i].Y, aryPoint[j].X,
            aryPoint[j].Y);
}

```

3. 載入 <C:\VC2008\ch14\icons\S04.bmp> 的圖檔後，按下 **圖片放大複製** 按鈕會以 GetPixel() 方法和 SetPixel() 方法將圖片放大兩倍，按下 **特效處理** 按鈕會以 GetPixel() 將所有的像素取出，並將像素的 ARGB 屬性分離，然後以特效處理的公式 $f=f-128$ (如果 f 大於 128) 完成特效處理運算，按下 **存檔** 按鈕會將圖像存成 <newFile.bmp> 檔。



【提示】

圖片放大兩倍處理：

```

Bitmap bmpLarge = new Bitmap(bmpSource.Width*2, bmpSource.Height* 2);
for (int i = 0; i <= bmpSource.Width - 1; i++)
{
    for (int j = 0; j <= bmpSource.Height - 1; j++)
    {
        Color pixelColor = bmpSource.GetPixel(i, j); //取得 bmpSource
            上指定像素 (Pixel) 的顏色
        bmpLarge.SetPixel(i * 2, j * 2, pixelColor); //設定 bmpLarge
            上指定像素 (Pixel) 的顏色
        bmpLarge.SetPixel(i * 2 + 1, j * 2 + 1, pixelColor); //
            設定 bmpLarge 上指定像素 (Pixel) 的顏色
    }
}

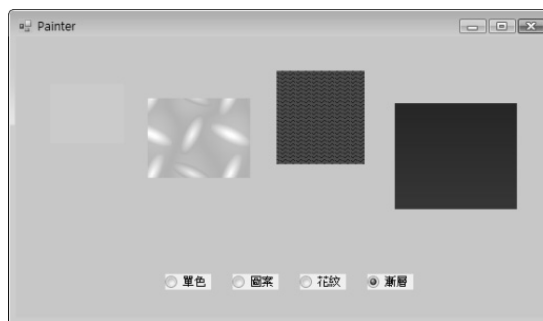
```

```
}  
picLarge.Image = bmpLarge; // 透過 picLarge 顯示之
```

特效處理：

```
Color pixelColor = bmpSource.GetPixel(i, j); //取得 bmpSource 上指定  
    像素 (Pixel) 的顏色  
double a = pixelColor.A; //分離像素的屬性  
int r = pixelColor.R;  
int b = pixelColor.B;  
int g = pixelColor.G;  
  
a = a * 0.9; //透明度 90%  
if (r > 128) //特效處理  
    r = r - 128;  
if (g > 128)  
    g = g - 128;  
if (b > 128)  
    b = b - 128;  
Color newColor = Color.FromArgb((int)a, r, g, b);  
bmpARGB.SetPixel(i, j, newColor);
```

4. 先選擇單色、圖案、花紋和漸層等不同形式的筆刷後，用滑鼠左鍵在表單上繪製不同圖案的填滿矩形。



【提示】

以花紋筆刷繪製填滿矩形：

```

using System.Drawing.Drawing2D;
...
Brush MyBrush; // 筆刷樣式
MyBrush = new HatchBrush(HatchStyle.Wave, Color.Blue, Color.Red);
g.DrawImage bmp, 0, 0); // 在表單上顯示 bmp 記憶體圖像
//計算繪圖的矩形區域
Rectangle rect = new Rectangle(Math.Min(x, e.X), Math.Min(y, e.Y),
    Math.Abs(e.X - x), Math.Abs(e.Y - y));
g.FillRectangle(MyBrush, rect); // 在表單上繪暫態填滿矩形

```

5. 以 `System.Media.SoundPlayer` 播放 `.wav` 的音效檔，按下 **載入 wav** 按鈕，可以開啟音效檔供 **播放 wav 檔** 按鈕播放，`wav` 檔的播放模式可以選擇 `Play` 和 `PlayLooping`，按 **停止播放** 鈕將停止播放。預設檔案目錄是 `<C:\Windows\Media>`，播放模式為 `PlayLooping` 反覆播放。



Handwriting practice lines consisting of 15 horizontal dotted lines on a white background.

