Tutorial – Drawing

Spirit Du

Ver. 1.0, 18th September, 2007

Ver. 2.0, 3rd November, 2008

Contents

About This Document

In this tutorial, two families of methods are introduced: Graphics.FillXXX() family and Graphics.DrawXXX() family. These two families of methods are the base of GUI. Moreover, no Windows Form designer is used that means everything is coded by the programmer's hand. A funny picture like Figure 1 is created.

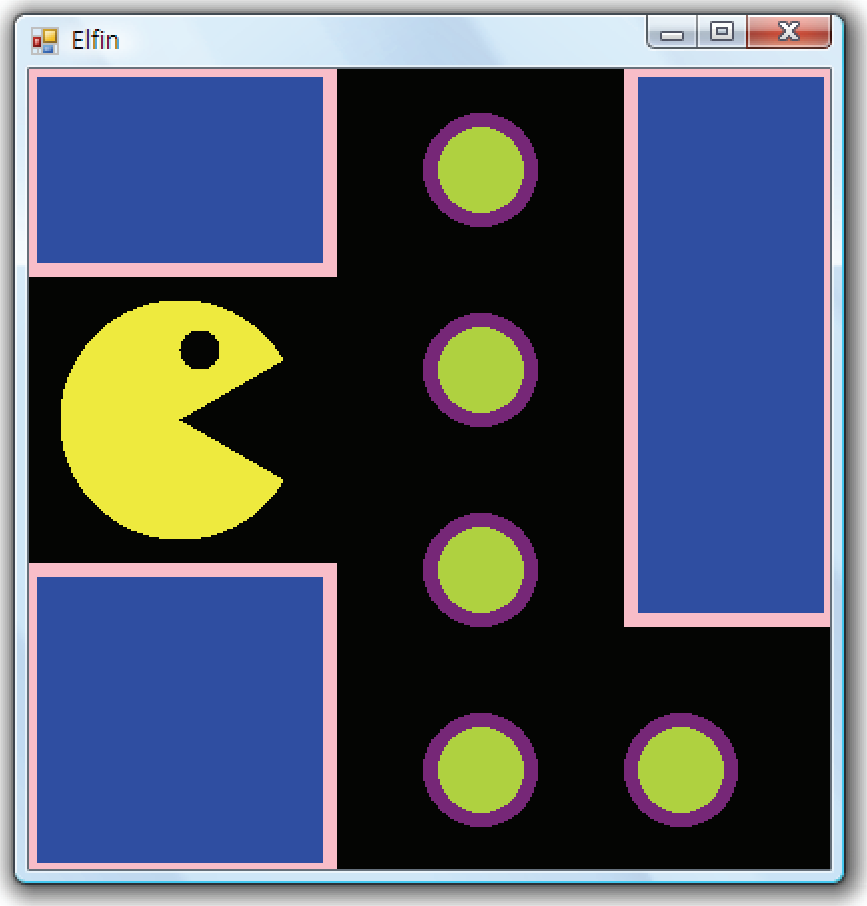

Figure 1 Elfin Game Map

Tutorial – Elfin

Step 1 Setup Environment

Create a C# Console Application project. Add the references System.Windows.Forms and System.Drawing into the project, and then add a new class: ElfinForm that inherits the Form class into the project with the following code.

```
using System.Windows.Forms;
using System.Drawing;

namespace Tutorial {
    class ElfinForm : Form {

        protected int WINDOW_WIDTH = 400;
        protected int WINDOW_HEIGHT = 400;
        protected int BALL_SIZE = 50;

        public ElfinForm() {
            SetClientSizeCore(WINDOW_WIDTH, WINDOW_HEIGHT);
            BackColor = Color.Black;
            Text = "Elfin";
        }
    }
}
```

Step 2 Fill Shapes with Brush

The outline of a Form is able to modify by overriding the OnPaint() method. When you get Graphics instance from PaintEventArgs object, you can do many things though its member methods, for examples, to fill shapes. You may find that when you want to fill shapes, you need a brush. However, you do not need to create any brush; you can find many brushes predefined in Brushes class.

```
Graphics.FillRectangle(Brush brush, int x, int y, int width, int height)
Graphics.FillEllipse(Brush brush, int x, int y, int width, int height)
Graphics.FillPie(Brush brush, int x, int y, int width, int height, int startAngle, int sweepAngle)
```

Figure 2 The FillXXX() family

Step 3 Make the Picture

Now, add a member method which override the OnPaint() method, and then we use those methods described in Step 2 to draw three blocks to identify the road, five circles as gems on the road. Then, we use FillPie() and FillEllipse() methods to draw the classic eflin.

```
protected override void OnPaint(PaintEventArgs e) {
    base.OnPaint(e);
    Graphics g = e.Graphics;

    // Three blue blocks
    g.FillRectangle(Brushes.Blue, 0, 0, 150, 100);
    g.FillRectangle(Brushes.Blue, 0, 250, 150, 150);
    g.FillRectangle(Brushes.Blue, 300, 0, 100, 275);

    // The elfin
    g.FillPie(Brushes.Yellow, 15, 115, 120, 120, 30.0f, 300.0f);
    g.FillEllipse(Brushes.Black, 75, 130, 20, 20);

    // Five balls
    g.FillEllipse(Brushes.GreenYellow, 200, 25, BALL_SIZE, BALL_SIZE);
    g.FillEllipse(Brushes.GreenYellow, 200, 125, BALL_SIZE, BALL_SIZE);
    g.FillEllipse(Brushes.GreenYellow, 200, 225, BALL_SIZE, BALL_SIZE);
    g.FillEllipse(Brushes.GreenYellow, 200, 325, BALL_SIZE, BALL_SIZE);
    g.FillEllipse(Brushes.GreenYellow, 300, 325, BALL_SIZE, BALL_SIZE);
}
```

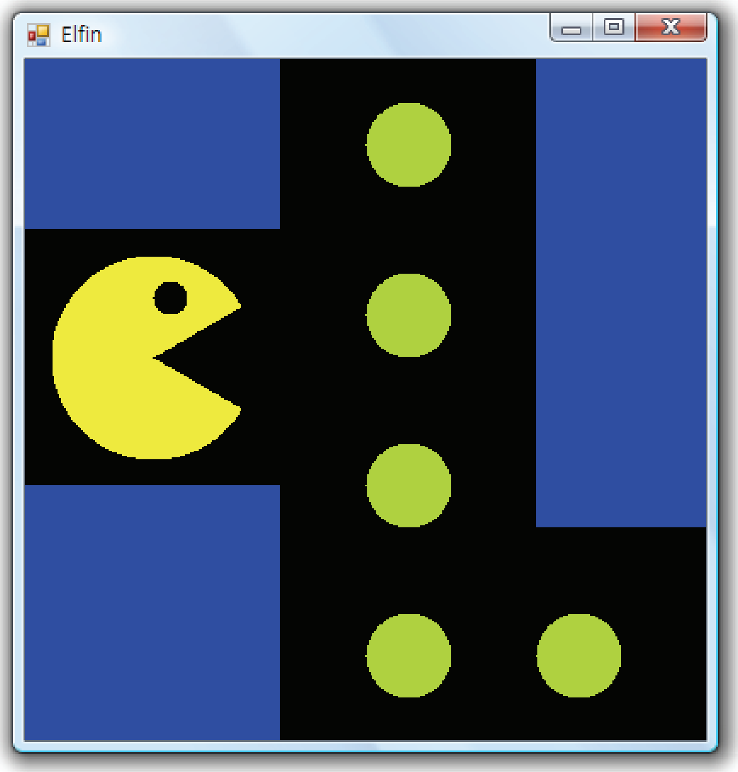

Figure 3 Semi-finished good

Step 4 Draw vs. Fill

Beside of FillXXX() methods which always draw shapes with background color, there are still many DrawXXX() methods which just sketch the outline of shape. There are different to FillXXX methods; they use a pen to sketch, not a brush. There is no FillLine() method because it is meaningless!

```
Graphics.DrawRectangle(Pen pen, int x, int y, int width, int height)
Graphics.DrawEllipse(Pen pen, int x, int y, int width, int height)
Graphics.DrawPie(Pen pen, int x, int y, int width, int height, int startAngle, int sweepAngle)
Graphics.DrawLine(Pen pen, int x1, int y1, int x2, int y2);
```

Figure 4 The DrawXXX() family

Step 5 A Thick Pen

In order to sketch the thick wall around the block, we create a thick pen instead of the system defined pen. Pen class has four constructors to create a pen depending on your needs. Here is a simple way to create a pen with specific color and thickness.

```
// Create a purple thick pen
Pen thickPen = new Pen(Color.Purple, 7.0f);
```

Step 6 Walls and Shells

At the last step, we use the thick pen as a parameter pass to DrawXXX method and draw walls and the shells of the gems. You should put the following codes after the codes of Step 3 in OnPaint() method.

```
// Give a border to each ball
g.DrawEllipse(thickPen, 200, 25, 50, 50);
g.DrawEllipse(thickPen, 200, 125, 50, 50);
g.DrawEllipse(thickPen, 200, 225, 50, 50);
g.DrawEllipse(thickPen, 200, 325, 50, 50);
g.DrawEllipse(thickPen, 300, 325, 50, 50);

// Change the color of pan to pink
thickPen.Color = Color.Pink;

// Create walls
g.DrawRectangle(thickPen, 0, 0, 150, 100);
g.DrawRectangle(thickPen, 0, 250, 150, 150);
g.DrawRectangle(thickPen, 300, 0, 100, 275);
```

Step 7 Just have Fun

Modify the entry class (or create an entry class if it is not existing in the created project) for the

project, and then create a form instance from the class we designed. If you still have time during the class, you can add more funny objects on the screen. You can modify the output type from Console Application to Windows Application in the project property setting page, so that no black console windows is shown anymore.

```
using System.Windows.Forms;


namespace Tutorial {
    class MainEntry {
        static void Main(string[] args) {
            Form form = new ElfinForm();
            Application.Run(form);
        }
    }
}
```

- The End -