

笔记本： 程序设计综合实验

创建时间： 2018-11-8 22:52

更新时间： 2018-11-9 11:32

URL： <https://blog.csdn.net/u011555996/article/details/77996921>

转 C# GDI+ 绘图

2017年09月15日 23:50:50 luckyone906 阅读数：1708

1 坐标系

(1) 坐标原点：在窗体或控件的左上角，坐标为(0,0)

(2) 正方向：X轴正方向为水平向右，Y轴正方向为竖直向下

(3) 单位：在设置时，一般以像素为单位，像素(Pixel)是由图像(Picture)和元素(Element)组成，是用来计算数码影像的一种单位。

把影像放大数倍，会发现这些连续的色调其实是有许多色彩相近的小方点组成，这些小方点是构成影像的最小单位—像素。

图形的质量是有像素决定，像素越大，分辨率也越大。

2 命名空间 --- System.Drawing

(1) System.Drawing 提供了对GDI+基本图形功能的访问

(2) System.Drawing 常用基本类及结构

类	说明
Bitmap	用于处理有像素数据定义的图像的对象。
Brush	定义用于填充图形形状的内部对象。
Font	定义特定的文本格式。
Graphics	封装一个GDI+绘图图画，无法继承此类。
Pen	用于绘制直线和曲线的对象，无法继承此类。
Region	指示由矩形和路径构成的图形形状的内部，无法继承此类。
Color	表示RGB颜色。

Point	定义二维平面中定义的点。
Rectangle	存储一组整数，共4个，表示一个矩形的位置和大小。
Size	存储一个有序整数对，通常为矩形的宽和高。

3 Graphics类

Graphics类封装了一个GDI+绘制界面，提供将对象绘制到显示界面的方法。使用GDI+创建图形图像时，需要先创建Graphics对象，即在哪里画图。

共有3种类型的绘图界面：

- （1） 窗体和控件
- （2） 打印机
- （3） 内存的位图

创建图形对象的3中方法：

- （1） 控件类的OnPaint()方法参数PaintEventArgs获取Graphics对象
- （2） 窗体类或控件类中的CreateGraphics()方法获得Graphics对象
- （3） 从位图对象(Bitmap)产生一个Graphics对象

Graphics类的常用方法

名称	说明
Dispose	释放Graphics使用的所有资源。
DrawEllipse	绘制椭圆，有高度，宽度，一对坐标。
DrawArc	绘制弧形。
DrawLine	绘制一条直线，由2个点指定。
DrawPolygon	绘制由一组Point结构定义的多边形。
DrawRectangle	绘制矩形。
DrawPie	绘制一个扇形。
DrawCurve	绘制曲线，由参数Point数组指定。

FillEllipse	填充边框所定义的椭圆的内部。
FillRegion	填充Region的内部。
ScaleTransform	将制定的缩放操作应用于次Graphics。
TanslateTransform	平移更改坐标系统的原点。

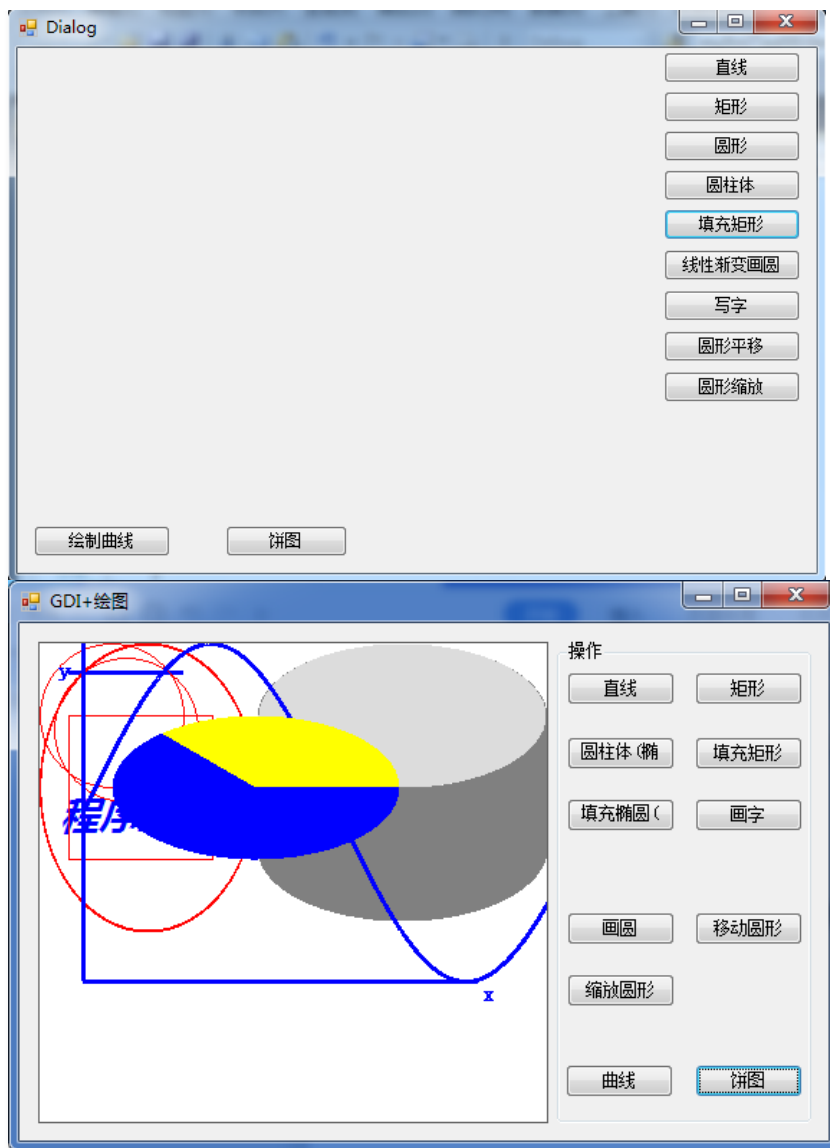
4 绘图工具类

类名	说明
Pen	设置画笔的颜色，线条粗细和线条样式(实线和虚线)。
Brush	用于填充图形，设置笔刷的样式，颜色及线条的粗细。

5 Brush类的派生类

名称	说明
ImageBrush	图像绘制区域。
LinearGradientBrush	线性渐变绘制区域。
RadialGradientBrush	径向渐变绘制区域，焦点定义渐变的开始，椭圆定义渐变的终点。
SolidColorBrush	单色绘制区域。
VideoBrush	视频内容绘制区域。

6 案例 免费下载地址 <http://download.csdn.net/detail/taoerit/8350869>



7 代码

[csharp]

```

1. using System;
2. using System.Collections.Generic;
3. using System.ComponentModel;
4. using System.Data;
5. using System.Drawing;
6. using System.Linq;
7. using System.Text;
8. using System.Windows.Forms;
9. using System.Drawing.Drawing2D;
10.
11.
12. namespace GDI绘图
13. {
14.     public partial class MainDialog : Form
15.     {
16.         public MainDialog()
17.         {

```

```
18.         InitializeComponent();
19.     }
20.
21.     private void MainDialog_Load(object sender, EventArgs e)
22.     {
23.
24.     }
25.
26.     private void lineButton_Click(object sender, EventArgs e)
27.     {
28.         // 画直线
29.         Graphics gra = this.CreateGraphics();
30.         Pen pen = new Pen(Color.Red);
31.         pen.Width = 2;
32.         Point startPoint = new Point(20,20);
33.         Point endPoint = new Point(70,20);
34.         gra.DrawLine(pen,startPoint,endPoint);
35.
36.         pen.Dispose();
37.         gra.Dispose();
38.     }
39.
40.     private void rectangleButton_Click(object sender, EventArgs e)
41.     {
42.         //画矩形
43.         Graphics gra = this.CreateGraphics();
44.         Pen pen = new Pen(Color.Red);
45.         gra.DrawRectangle(pen, 20,50, 100,100);
46.         pen.Dispose();
47.         gra.Dispose();
48.     }
49.     private void cylinderButton_Click(object sender, EventArgs e)
50.     {
51.         //圆柱体,有许多个椭圆有底部逐渐叠起来的,最后填充颜色
52.
53.         int height = this.ClientSize.Height - 150;
54.         int width = this.ClientSize.Width - 50;
55.         int vHeight = 200;
56.         int vWidth = 100;
57.         Graphics gra = this.CreateGraphics();
58.         gra.Clear(Color.White);
59.         Pen pen = new Pen(Color.Gray,2);
60.         SolidBrush brush = new SolidBrush(Color.Gainsboro);
61.
62.         for (int i = height / 2; i > 0;i-- )
63.         {
64.             gra.DrawEllipse(pen,width/2,i,vHeight,vWidth);
65.         }
66.
67.         gra.FillEllipse(brush,width/2,0,vHeight,vWidth);
68.     }
69.
70.     private void fillRectangleButton_Click(object sender, EventArgs e)
71.     {
```

```
72.         //画矩形
73.         Graphics gra = this.CreateGraphics();
74.         Pen pen = new Pen(Color.Red,3);
75.         Brush brush = pen.Brush;
76.         Rectangle rect = new Rectangle(20,50,100,100);
77.         gra.FillRectangle(brush,rect);
78.         gra.Dispose();
79.     }
80.
81.     private void drawEllipseButton_Click(object sender, EventArgs e)
82.     {
83.         Graphics gra = this.CreateGraphics();
84.         Rectangle rect = new Rectangle(0,0,200,100);
85.         LinearGradientBrush brush = new LinearGradientBrush(rect,Color.Orange,Color.Purple,90);
86.
87.         gra.FillEllipse(brush,rect);
88.         gra.Dispose();
89.     }
90.
91.     private void fontButton_Click(object sender, EventArgs e)
92.     {
93.         Graphics gra = this.CreateGraphics();
94.         Font font = new Font("隶书",24,FontStyle.Italic);
95.         Pen pen = new Pen(Color.Blue,3);
96.         gra.DrawString("Windows应用程序设计",font,pen.Brush,10,100);
97.     }
98.
99.     private void ellipseButton_Click(object sender, EventArgs e)
100.    {
101.        // 画圆形
102.        Graphics gra = this.CreateGraphics();
103.        Pen pen = new Pen(Color.Red);
104.        gra.DrawEllipse(pen, 0, 0, 200,100);
105.        pen.Dispose();
106.        gra.Dispose();
107.    }
108.
109.    private void moveEllipseButton_Click(object sender, EventArgs e)
110.    {
111.        // 移动圆形
112.        Graphics gra = this.CreateGraphics();
113.        Pen pen = new Pen(Color.Red);
114.        gra.TranslateTransform(10,10); // 改变起坐标(10,10)
115.        gra.DrawEllipse(pen, 0, 0, 200, 100);
116.
117.        gra.Dispose();
118.    }
119.
120.    private void scaleEllipseButton_Click(object sender, EventArgs e)
121.    {
122.        // 缩放圆形
123.        float xScale = 1.5F;
124.        float yScale = 2F;
125.        Graphics gra = this.CreateGraphics();
```

```

125.         Pen pen = new Pen(Color.Red);
126.         gra.ScaleTransform(xScale, yScale); // X轴放大1.5倍, Y轴放大2倍
127.         gra.DrawEllipse(pen, 0, 0, 200, 100);
128.         gra.Dispose();
129.     }
130.
131.     private void curveButton_Click(object sender, EventArgs e)
132.     {
133.         //绘制曲线
134.         Graphics gra = this.CreateGraphics();
135.         Pen pen = new Pen(Color.Blue, 3);
136.         Point oo1 = new Point(30, this.ClientSize.Height - 100);
137.         Point oo2 = new Point(this.ClientSize.Width - 50, this.ClientSize.Height - 100);
138.         gra.DrawLine(pen, oo1, oo2);
139.         Point oo3 = new Point(30, 30);
140.         gra.DrawLine(pen, oo1, oo3);
141.         Font font = new System.Drawing.Font("宋体", 12, FontStyle.Bold);
142.         gra.DrawString("X", font, pen.Brush, oo2);
143.         gra.DrawString("Y", font, pen.Brush, oo3);
144.
145.         int x1 = 0, x2 = 0;
146.         double a = 0;
147.         double y1 = 0, y2 = this.ClientSize.Height - 100;
148.         for (x2 = 0; x2 < this.ClientSize.Width; x2++)
149.         {
150.             a = 2 * Math.PI * x2 / (this.ClientSize.Width);
151.             y2 = Math.Sin(a);
152.             y2 = (1 - y2) * (this.ClientSize.Height - 100) / 2;
153.             gra.DrawLine(pen, x1 + 30, (float)y1, x2 + 30, (float)y2);
154.             x1 = x2;
155.             y1 = y2;
156.         }
157.         gra.Dispose();
158.     }
159.
160.     private void piechartButton_Click(object sender, EventArgs e)
161.     {
162.         //饼图
163.         Graphics gra = this.CreateGraphics();
164.         Pen pen = new Pen(Color.Blue, 3);
165.         Rectangle rect = new Rectangle(50, 50, 200, 100);
166.         Brush brush = new SolidBrush(Color.Blue);
167.         gra.FillPie(pen.Brush, rect, 0, 60);
168.         gra.FillPie(brush, rect, 60, 150);
169.         brush = new SolidBrush(Color.Yellow);
170.         gra.FillPie(brush, rect, 210, 150);
171.
172.     }
173.
174. }
175. }

```