

ใบงานที่ 17

การเขียนโปรแกรมกราฟฟิกส์ด้วย GDI+ (1)

1. กล่าวนำ

ใบงานนี้ มีวัตถุประสงค์ เพื่อให้นักศึกษา ได้รู้จักกับ GDI+ ซึ่งจะช่วยให้ นักศึกษาสามารถ

- อธิบายระบบพิกัดใน GDI+ ได้
- สามารถใช้คำสั่งพื้นฐานของ GDI+ ในการวาดกราฟฟิกส์อย่างง่ายได้

2. เนื้อหา

2.1 เกี่ยวกับ GDI+

GDI+ (Graphics Device Interface Plus) เป็นกราฟฟิกส์ไลบรารี ซึ่ง Microsoft จัดเตรียมไว้ให้นักพัฒนาโปรแกรมได้ใช้ทำงานด้านกราฟฟิกส์ โดยที่ GDI+ จะมีลักษณะเป็น OOP เต็มตัว

ก่อนที่จะมี GDI+ ไมโครซอฟท์มีไลบรารีชื่อ GDI ซึ่งอยู่ในไฟล์ gdi32.dll ติดตั้งมาบน Windows ทุกตัว จนกระทั่งมาถึงยุค Windows XP ทางไมโครซอฟท์ก็ได้ปรับปรุง GDI ให้มีความสามารถมากขึ้น โดยให้ชื่อว่า GDI+

2.2 ความสามารถของ GDI+

ความสามารถของ GDI+ จะมีอยู่ 3 ส่วนหลักๆ คือ

- **Vector graphics** เป็นการสร้างกราฟฟิกส์ 2 มิติ ด้วยรูปทรงเรขาคณิตพื้นฐาน เช่น เส้นตรง(Line) เส้นโค้ง (Curve) สี่เหลี่ยม(Rectangle) วงรี (Ellipse) เป็นต้น
- **Imaging** นอกจากความสามารถในการวาดภาพด้วยรูปทรงเรขาคณิตแล้ว GDI+ ยังมีความสามารถในด้านการแสดงผล, Load/Save ไฟล์ภาพชนิดต่างๆ ในรูปของ bitmap (bitmap ในที่นี้หมายถึงไฟล์ภาพทุกชนิด ที่แสดงเป็น pixel และไม่สามารถอธิบายด้วยรูปทรงเรขาคณิตอย่างง่าย ๆ ได้ เช่น ภาพถ่าย ภาพสแกน เป็นต้น)
- **Typology** เป็นการจัดการเกี่ยวกับฟอนต์ ซึ่งมีความสามารถในการแสดงผลแบบ anti aliasing ทำให้ขอบของตัวอักษรดูเรียบขึ้นมากกว่าการเขียนโปรแกรมด้วย GDI

2.3 ระบบพิกัดใน GDI+

ใน GDI+ จะมีระบบพิกัดที่ใช้งานอยู่เป็นจำนวน 3 ระบบ คือ World coordinate, Page coordinate และ Device coordinate

World coordinate จะทำงานกับระบบหน่วยที่ใช้ในชีวิตประจำวัน เช่น หน่วยมิลลิเมตร หรือ หน่วยนิ้ว เป็นต้น

Page coordinate จะทำงานกับระบบ coordinate บนหน้ากระดาษ และจะสามารถพิมพ์ออกทางเครื่องพิมพ์ตามขนาดที่กำหนดบน pagecoordinate

Device coordinate จะทำงานกับ pixel บนจอหรือบนเครื่องพิมพ์

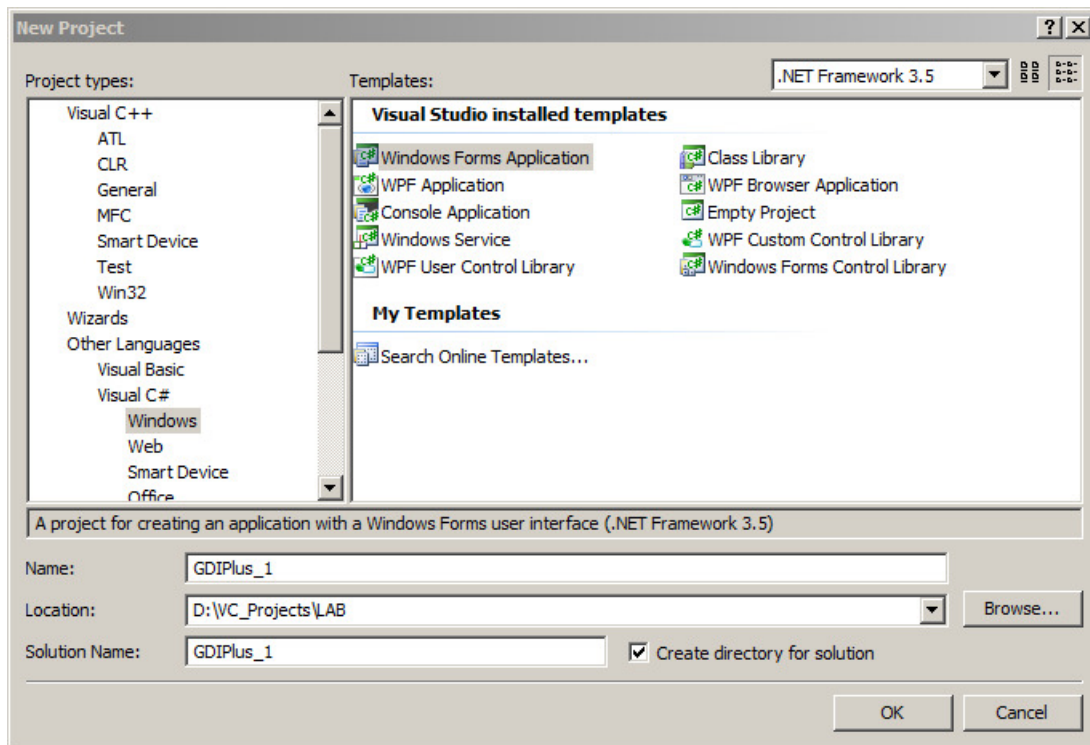
2.4 C-sharp.NET

เพื่อให้สามารถเขียนโปรแกรมได้ง่ายยิ่งขึ้น ใบบงานที่เกี่ยวกับ GDI+ ทั้งหมดจะเขียนด้วยภาษา C# เป็นหลัก อย่างไรก็ตาม นักศึกษาสามารถนำ GDI+ ไปใช้กับโปรแกรมได้ทุกภาษา (รวมถึง C++ ด้วย แต่ต้องมีการเตรียมสภาพแวดล้อมที่ยุ่งยากกว่า)

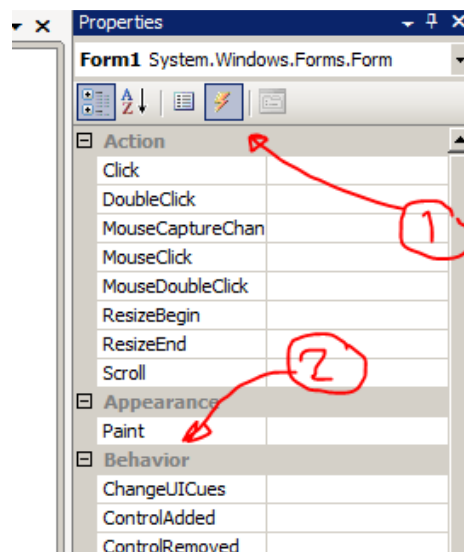
3. การทดลองย่อย 1 เริ่มต้นกับ C# และ GDI+

3.1. เรียกโปรแกรม Visual Studio 2008

3.2. สร้าง Project ใหม่ เป็นชนิด C#



3.3. เมื่อ Wizard สร้าง Project เสร็จแล้ว จะนำเรามาที่หน้าต่าง Form1.cs[Design] ให้คลิกที่ปุ่ม Events ของ Properties pane ตาม (1) และ Double click ที่ Paint ตาม (2)



3.4. เพื่อให้โปรแกรมของเราสามารถใช้งาน GDI+ ในการวาดภาพ 2D ได้ ให้ทำการเพิ่ม

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

```
7 | using System.Text;
8 | using System.Windows.Forms;
9 | using System.Drawing.Drawing2D;
10 |
11 | namespace GDIPlus_1
12 | {
13 |     public partial class Form1 : Form
14 |     {
```

3.5. ให้เพิ่มบรรทัดต่อไปนี้ลงในฟังก์ชัน `private void Form1_Paint(object sender, PaintEventArgs e)` (ในที่นี้จะอธิบายโปรแกรมไปด้วย การเพิ่มจริงๆ ให้ทำตามรูปข้างล่าง)

3.5.1. สร้าง Object ของกราฟิกส์ โดยคำสั่ง `Graphics g = e.Graphics;` ซึ่ง Object ชื่อ `e` ถูกส่งผ่านมาจาก `argument` ของฟังก์ชัน

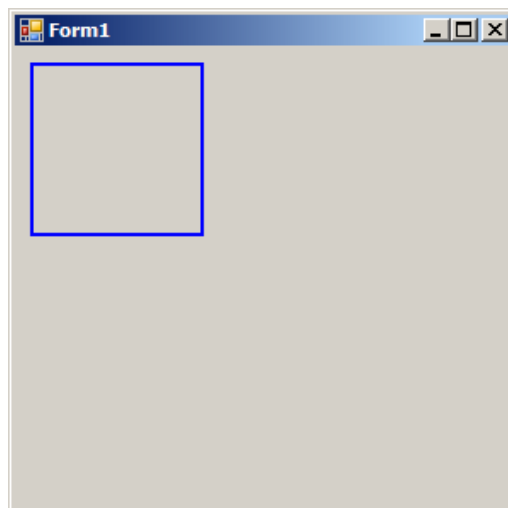
3.5.2. เพิ่มออบเจกต์ของปากกา สีน้ำเงินขนาด 2 พิกเซล ด้วยคำสั่ง `Pen = new Pen(Color.Blue, 2);`

3.5.3. วาดสี่เหลี่ยมด้วยคำสั่ง `g.DrawRectangle(bluepen, 10, 10, 100, 100);`

3.5.4. คืนหน่วยความจำให้ระบบโดยการลบออบเจกต์ปากกาสีน้ำเงิน โดยคำสั่ง `bluepen.Dispose();`

```
19 |
20 | private void Form1_Paint(object sender, PaintEventArgs e)
21 | {
22 |     Graphics g = e.Graphics;
23 |     Pen bluepen = new Pen(Color.Blue, 2);
24 |     g.DrawRectangle(bluepen, 10, 10, 100, 100);
25 |     bluepen.Dispose();
26 |
27 | }
```

3.6. ทดลอง Build และ Run โปรแกรม



4. การทดลองย่อย 2 การใช้สี

4.1 โดยการผสมค่าสี

เพิ่ม Code ต่อไปนี้ลงในฟังก์ชัน `private void Form1_Paint(object sender, PaintEventArgs e)` แล้วทดลอง Run โปรแกรม

```
19
20 private void Form1_Paint(object sender, PaintEventArgs e)
21 {
22     Graphics g = e.Graphics;
23     Pen bluepen = new Pen(Color.Blue, 2);
24     g.DrawRectangle(bluepen, 10, 10, 100, 100);
25     bluepen.Dispose();
26
27     + { Pen somePen = new Pen(Color.FromArgb(255,120,200));
28         g.DrawEllipse(somePen, 20, 20, 200, 200);
29         somePen.Dispose();
30
31
32
33
34
```

4.2 โดยการใช้ methode FromName

เพิ่ม Code ต่อไปนี้ลงในฟังก์ชัน `private void Form1_Paint(object sender, PaintEventArgs e)` แล้วทดลอง Run โปรแกรม

```
20 private void Form1_Paint(object sender, PaintEventArgs e)
21 {
22     Graphics g = e.Graphics;
23     Pen bluepen = new Pen(Color.Blue, 2);
24     g.DrawRectangle(bluepen, 10, 10, 100, 100);
25     bluepen.Dispose();
26
27     Pen somePen = new Pen(Color.FromArgb(255,120,200));
28     g.DrawEllipse(somePen, 20, 20, 200, 200);
29     somePen.Dispose();
30
31     + { Color col = Color.FromName("LightGreen");
32         this.BackColor = col;
33
34
```

5. การทดลองย่อย 3 การใช้ปากกา

5.1 การทดลองเปลี่ยนขนาดและสีของปากกา โดยใช้ properties Width และ Color

```
20 private void Form1_Paint(object sender, PaintEventArgs e)
21 {
22     Graphics g = e.Graphics;
23     // create black pen, Width = 1
24     Pen mypen = new Pen(Color.Black, 1);
25     // Draw rectangle
26     g.DrawRectangle(mypen,10,10,200,200);
27     // change pen Width to 4, color to pink
28     mypen.Width = 4;
29     mypen.Color = Color.Pink;
30     g.DrawEllipse(mypen,10,10,200,200);
31
32 }
```

5.2 เปลี่ยนชนิดของปากกาเป็นเส้นประ

```
20 private void Form1_Paint(object sender, PaintEventArgs e)
21 {
22     Graphics g = e.Graphics;
23     // create black pen, Width = 1
24     Pen mypen = new Pen(Color.Black, 1);
25     mypen.DashStyle = DashStyle.Dash;
26     // Draw rectangle
27     g.DrawRectangle(mypen, 10, 10, 200, 200);
28     // change pen Width to 1, color to pink
29     mypen.Width = 4;
30     mypen.Color = Color.Pink;
31     g.DrawEllipse(mypen, 10, 10, 200, 200);
32 }
33
```

5.3 ใช้ Pen ร่วมกับ Brush

```
19 private void Form1_Paint(object sender, PaintEventArgs e)
20 {
21     Graphics g = e.Graphics;
22     // create Solid Brush
23     Brush myBrush = new SolidBrush(Color.DarkGreen);
24     // use brush with pen
25     Pen mypen = new Pen(myBrush, 5);
26     g.DrawEllipse(mypen, 10, 10, 200, 200);
27     mypen.Dispose();
28     myBrush.Dispose();
29 }
30
31
32
```

5.4 ใช้ Pen ร่วมกับ HatchBrush เพื่อสร้างลายเส้นแบบต่างๆ

```
20 private void Form1_Paint(object sender, PaintEventArgs e)
21 {
22     Graphics g = e.Graphics;
23     // create Solid Brush
24     HatchBrush myBrush = new HatchBrush(HatchStyle.DarkVertical,
25     Color.White, Color.Violet);
26     // use brush with pen
27     Pen mypen = new Pen(myBrush, 5);
28
29     g.DrawEllipse(mypen, 10, 10, 200, 200);
30     mypen.Dispose();
31     myBrush.Dispose();
32 }
33
```

ทดลองเปลี่ยน Color และ HatchStyle เป็นแบบต่างๆ เพื่อดูความเปลี่ยนแปลง