

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра інформаційних систем та технологій

**Лабораторна робота № 5**

з дисципліни «Технології та засоби розробки комп'ютерної графіки та  
мультимедіа»

Тема: «Побудова та переміщення графічних об'єктів на площині»

Виконав:

студент групи ІО-25

Льоскін Іван Вадимович

Дата здачі 25.12.2024

Захищено з балом \_\_\_\_\_

Перевірила:

ст. вик. кафедри ІСТ

Хмелюк Марина Сергіївна

Київ 2024

**Завдання:**

- Побудувати геометричну фігуру та матрицю перетворень будь-якої геометричної фігури, здійснити її переміщення, обертання, відображення, розтягування на екрані.
- Вивести на екран будь-який текст і здійснити його переміщення.

**Хід роботи:**

Розміщуємо кнопки на полі та приступаємо до написання коду:

```
public partial class Form1 : Form
{
    private PointF[] trianglePoints;
    private string displayedText = "Slim Shady";
    private float textX = 100;
    private float textY = 100;
    private Font textFont = new Font("Arial", 16);

    public Form1()
    {
        InitializeComponent();
        InitializeTriangle();
        this.KeyDown += HandleKeyPress;
        this.KeyPreview = true;
    }

    private void Form1_Load(object sender, EventArgs e)
    {

    }

    private void InitializeTriangle()
    {
        trianglePoints = new PointF[]
        {
            new PointF(200, 50), // top point
            new PointF(150, 200), // bottom left point
            new PointF(250, 200), // bottom right point
        };
    }
}
```

```
}
```

```
protected override void OnPaint(PaintEventArgs e)
{
    Graphics g = e.Graphics;
    g.FillPolygon(Brushes.Yellow, trianglePoints);
    RectangleF textArea = new RectangleF(textX, textY,
this.ClientSize.Width, this.ClientSize.Height);
    PointF textPosition = GetCenteredTextPosition(displayedText, textArea,
g);
    g.DrawString(displayedText, textFont, Brushes.Black, textPosition);
}
```

```
private PointF GetCenteredTextPosition(string text, RectangleF area,
Graphics g)
{
    {
        SizeF textSize = g.MeasureString(text, textFont);
        float centerX = area.X + (area.Width - textSize.Width) / 2;
        float centerY = area.Y + (area.Height - textSize.Height) / 2;
        return new PointF(centerX, centerY);
    }
}
```

```
private PointF CalculateCenter()
{
    float x = 0, y = 0;
    foreach (var point in trianglePoints)
    {
        x += point.X;
        y += point.Y;
    }
    return new PointF(x / trianglePoints.Length, y / trianglePoints.Length);
}
```

```
private void TranslateShape(int dx, int dy)
```

```

{
    for (int i = 0; i < trianglePoints.Length; i++)
    {
        trianglePoints[i].X += dx;
        trianglePoints[i].Y += dy;
    }
}

```

```

private void RotateShape(float angleDegrees)
{
    Matrix rotationMatrix = new Matrix();
    PointF center = CalculateCenter();
    rotationMatrix.RotateAt(angleDegrees, center);
    rotationMatrix.TransformPoints(trianglePoints);
}

```

```

private void TransformShape(float scaleX, float scaleY)
{
    Matrix transformMatrix = new Matrix();
    PointF center = CalculateCenter();
    transformMatrix.Translate(-center.X, -center.Y, MatrixOrder.Append);
    transformMatrix.Scale(scaleX, scaleY, MatrixOrder.Append);
    transformMatrix.Translate(center.X, center.Y, MatrixOrder.Append);
    transformMatrix.TransformPoints(trianglePoints);
}

```

```

private void ResizeShape(float scaleFactor)
{
    TransformShape(scaleFactor, scaleFactor);
}

```

```

private void HandleKeyPress(object sender, KeyEventArgs e)
{
    const int moveStep = 5;

    switch (e.KeyCode)

```

```
{
  case Keys.W:
    TranslateShape(0, -moveStep);
    break;

  case Keys.S:
    TranslateShape(0, moveStep);
    break;

  case Keys.A:
    TranslateShape(-moveStep, 0);
    break;

  case Keys.D:
    TranslateShape(moveStep, 0);
    break;

  case Keys.I:
    textY -= moveStep;
    break;

  case Keys.K:
    textY += moveStep;
    break;

  case Keys.J:
    textX -= moveStep;
    break;

  case Keys.L:
    textX += moveStep;
    break;

  default:
    return;
}
```

```

        Invalidate();
    }

private void ScaleUp_Click(object sender, EventArgs e)
{
    ResizeShape(1.1f);
    Invalidate();
}

private void ScaleDown_Click(object sender, EventArgs e)
{
    ResizeShape(0.9f);
    Invalidate();
}

private void ReflectionX_Click(object sender, EventArgs e)
{
    TransformShape(1, -1);
    Invalidate();
}

private void ReflectionY_Click(object sender, EventArgs e)
{
    TransformShape(-1, 1);
    Invalidate();
}

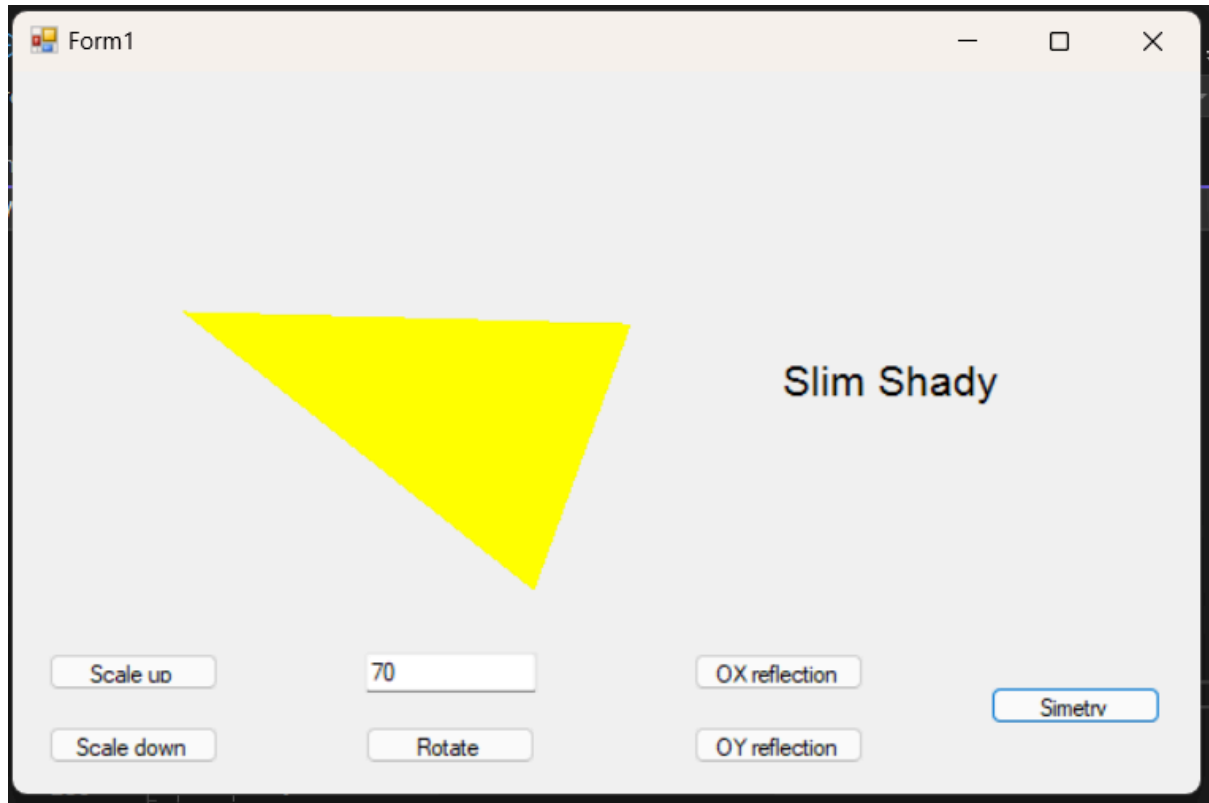
private void ReflectionSimetry_Click(object sender, EventArgs e)
{
    TransformShape(-1, -1);
    Invalidate();
}

private void Rotate_Click(object sender, EventArgs e)
{
    float degrees = float.Parse(DegreeBox.Text);

```

```
        RotateShape(degrees);  
        Invalidate();  
    }  
}
```

### Результат:



### Висновок:

Виконуючи цю лабораторну роботу, я вивчив, як відображати об'єкти на площині за допомогою двомірної матриці, та навчився маніпулювати графічними об'єктами всередині Windows Forms Application