Робота з гумовим контуром у 2D-графіці на мові C# з використанням Windows Forms

У 2D-графіці часто доводиться виділяти прямокутну область для виділення показників, частини рисунку. Зазвичай це виконується "гумовим" контуром, тобто при переміщенні мишки і натиснутій лівої кнопки будується новий прямокутник та видаляється старий.

Ефективно використовувати в *OnMouseMove* методику *малювання за принципом "виключаюче АБО"* (*exclusive-OR* або *XOR*), *XOR*-малювання не просто малює кольорові пікселі на екрані, а змінює кольори існуючих пікселів. Лінія, намальована в такий спосіб на чорному фоні, виглядає білою, а на голубому фоні – червоною. Перевага цієї методики у тому, що друга *XOR*-лінія, задана тими ж координатами, стирає повністю першу. Тобто повністю відновлює колір пікселів, які були до малювання першої лінії. У старих версіях C# *GDI+* не підтримується *XOR*-малювання і для стирання попереднього прямокутника використовується колір фону. Очевидно, що це не кращий спосіб. З'являються сліди фонових ліній, від яких треба позбутися. Для усунення цього обробка *OnMouseMove* закінчується викликом *Invalidate,* генеруючим подію *Paint.* У даному випадку повністю перемальовується малюнок. У випадку *XOR*-малювання необхідність у виклику *Invalidate* просто відпала б. Нижче наведені три приклади роботи з гумовим контуром. Перший і другий приклад виконані без використання *XOR*-малювання. У третьому прикладі програма виконана з використанням *XOR*-малювання.

Приклад 1.

У лістингу 1 та на рис. 1 відповідно наведені текст програми і результат її рішення без використання *XOR*-малювання. Детальний опис програми недоцільно, оскільки студенти мають досвід програмування резинового контуру при вивченні *Visual C++*.

Лістинг 1. (exam1)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace exam28
{
    public partial class Form1 : Form
    {
        Point ptbeg,ptend;
        Graphics grfx;
```

```
        public Form1()
        {
         InitializeComponent();
         Text = "Blockout flectangle with House";
         BackColor = SystemColors.Window;
         ForeColor = SystemColors.WindowText;
         ptbeg = ptend = Point.Empty;
         grfx = CreateGraphics();
        }

    private void Form1_MouseDown(object sender,MouseEventArgs e)
    {
      ptbeg = ptend = e.Location;
    }

   private void Form1_MouseMove(object sender, MouseEventArgs e)
   {
     if (e.Button != MouseButtons.Left) return;
     for (int i = 0; i < 2; i++)
      {
        grfx.DrawRectangle(new Pen((i == 0) ?
        BackColor : ForeColor),
        ptbeg.X, ptbeg.Y, ptend.X - ptbeg.X, ptend.Y - ptbeg.Y);
        ptend = e.Location;
       }
   }

    private void Form1_MouseUp(object sender, MouseEventArgs e)
      {
        if (e.Button == MouseButtons.Left) Invalidate();
      }

    protected override void  OnPaint(PaintEventArgs e)
    {
        //return;
    if (ptbeg.Equals(Point.Empty)) return;
    if (Math.Abs(ptbeg.X - ptend.X) < 2 ||
             Math.Abs(ptbeg.Y - ptend.Y) < 2) return;
    e.Graphics.DrawRectangle(new Pen(ForeColor),ptbeg.X,ptbeg.Y,
                        ptend.X - ptbeg.X, ptend.Y - ptbeg.Y);
    }
  }
}
```

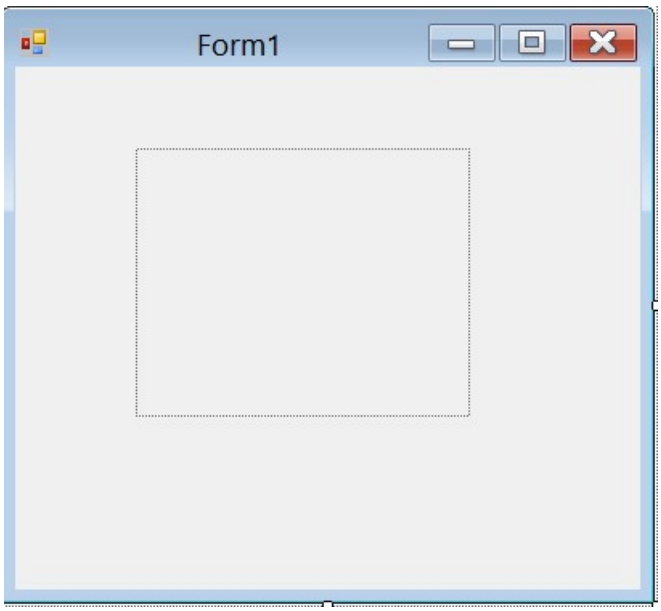

Рис. 1. До формування гумового контуру без використання *XOR*

Приклад 2.

У прикладі використаний інтерфейс, який використовується для двох класів відповідно в *Form*1 та *Form*2 (лістинги 2а і 2б). У першій формі виводиться гумовий прямокутник. У другій формі виводиться еліпс у вигляді гумового контуру. Інтерфейс схожий на визначення класу. Він містить методи, властивості і індексатор. Але інтерфейс містить лише сигнатури (*signatures*) цих членів, але не їх тіла. Як відомо, клас може бути спадкоємцем іншого класу, а клас, який не виглядає нічиїм спадкоємцем, насправді є спадкоємцем *Object.*

Лістинг 2а. (exam2)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace exam29
{
    interface ICapture
    {
        void OnLostCapture();
        void Down(Point a);
        void Move(Point a);
    }
    public partial class Form1 : Form, ICapture
    {
        public Point ptbeg, ptend;
```

```
     Form2 f = null;

     public Form1()
     {
         InitializeComponent();
         Text = "Blockout";
         BackColor = SystemColors.Window;
         ForeColor = SystemColors.WindowText;
         ptbeg = ptend = Point.Empty;
         CaptureLoss win = new CaptureLoss();
         win.control = this;
         win.AssignHandle(Handle);
     }

    private void Form1_Paint(object sender, PaintEventArgs e)
    {
        if (Math.Abs(ptbeg.X - ptend.X) < 5 ||
             Math.Abs(ptbeg.Y - ptend.Y) < 5)
        ptbeg.X = ptend.X = ptbeg.Y = ptend.Y;
        e.Graphics.DrawRectangle(new Pen(ForeColor), ptbeg.X,
           ptbeg.Y, ptend.X - ptbeg.X, ptend.Y - ptbeg.Y);
     }

public void OnLostCapture()
     {
        // Text ="OnLostCapture";
         vr++;
         Text = vr.ToString() + "C  ";

         Invalidate();
     }
     public void Down(Point a)
     {
         ptbeg = ptend = a;
     }

     int vr = 0;

    public void Move(Point a)
    {
      vr++;
      Text = vr.ToString() + "M  ";
     Graphics grfx = CreateGraphics();
     grfx.DrawRectangle(new Pen(BackColor), ptbeg.X, ptbeg.Y,
         ptend.X - ptbeg.X, ptend.Y - ptbeg.Y);
     ptend = a;
     grfx.DrawRectangle(new Pen(ForeColor), ptbeg.X, ptbeg.Y,
         ptend.X - ptbeg.X, ptend.Y - ptbeg.Y);
     grfx.Dispose();
    }

  private void form2ToolStripMenuItem_Click(object sender,
                                            EventArgs e)
```

```
        {
            if (f == null)
            {
                f = new Form2();
                f.Show();
            }
            else
            {
                f.Close();
                f = null;
            }
        }
    }

class CaptureLoss:NativeWindow
{
    public ICapture control;

    protected override void  WndProc(ref Message m)
{
    // WM_LBUTTONDOWN - 513
     if (m.Msg == 513)
    {
        short a = (short)m.LParam;
        short b = (short)(((int)m.LParam)/65536);
        Point pp = new Point(a, b);
        control.Down(pp);
    }
      // WM_MOUSEMOVE - 512
     if (m.Msg == 512&&((int)m.WParam)==1)
     {
         short a = (short)m.LParam;
         short b = (short)(((int)m.LParam) / 65536);
         Point pp = new Point(a, b);
         control.Move(pp);
     }

     // WM_CAPTURECHANGED - 533
     if (m.Msg == 533)
     control.OnLostCapture();
     base.WndProc(ref m);
      }
    }
 }
```

Лістинг 2б (exam2)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
```

```
using System.Text;
using System.Windows.Forms;

namespace exam29
{
    public partial class Form2 : Form, ICapture
    {
        public Point ptbeg, ptend;

        public Form2()
        {
            InitializeComponent();
            Text = "Blockout 2";
            BackColor = SystemColors.Window;
            ForeColor = SystemColors.WindowText;
            ptbeg = ptend = Point.Empty;
            CaptureLoss win = new CaptureLoss();
            win.control = this;
            win.AssignHandle(Handle);
        }

       private void Form2_Paint(object sender, PaintEventArgs e)
       {
           if (Math.Abs(ptbeg.X - ptend.X) < 5 ||
                Math.Abs(ptbeg.Y - ptend.Y) < 5)
           ptbeg.X = ptend.X = ptbeg.Y = ptend.Y;
           e.Graphics.DrawEllipse(new Pen(ForeColor), ptbeg.X,
             ptbeg.Y, ptend.X - ptbeg.X, ptend.Y - ptbeg.Y);
        }

        public void OnLostCapture()
        {
            Invalidate();
        }

        public void Down(Point a)
        {
            ptbeg = ptend = a;
        }

     public void Move(Point a)
      {
         Graphics grfx = CreateGraphics();
         grfx.DrawEllipse(new Pen(BackColor), ptbeg.X, ptbeg.Y,
             ptend.X - ptbeg.X, ptend.Y - ptbeg.Y);
         ptend = a;
         grfx.DrawEllipse(new Pen(ForeColor), ptbeg.X, ptbeg.Y,
             ptend.X - ptbeg.X, ptend.Y - ptbeg.Y);
```

```
            grfx.Dispose();
        }
    }
}
```

Клас також може бути спадкоємцем одного або декількох інтерфейсів. Якщо клас спадкує інтерфейс, у ньому повинні бути реалізовані всі методи і властивості, визначені у інтерфейсі. Інтерфейси допомагають підвищувати ступінь абстрактності класів, так як для визначення змінної замість імені класу або структури може бути використано ім'я відповідного інтерфейсу, після чого клас може викликати методи і властивості, визначені в цьому інтерфейсі. У нашому випадку, використовуючи інтерфейс, для кожної з двох форм підключаються різні функції:

void OnLostCapture();
void Down(Point a);
void Move(Point a);

Функція *void OnLostCapture*() підключається при генерації відгуку WM_CAPTURECHANGED (код = 533). Це відбувається коли вікно втрачає захоплення мишки штатним (при відпусканні кнопки мишки) або нештатним чином. Для цього знадобиться клас *NativeWindow*. Дві інші функції підключаються на відгук натискання і переміщення мишки.

Ці функції підключаються по відгукам у функції *WndProc* класу *NativeWindow*.

Прив'язка двох форм до єдиної функції *WndProc* виконується у кожній формі з використанням операторів

CaptureLoss win = new CaptureLoss();
win.control = this;
win.AssignHandle(Handle);

Тут виконується створення об'єкту класу *CaptureLoss*(), похідного від класу *NativeWindow*. Потім інтерфейсу (як базовому класу) привласнюється об'єкт класу *Form*1 або *Form*2. У функції *AssignHandle* здійснюється прив'язка через дескриптор вікна об'єкту класу *CaptureLoss*() до *Form*1 або *Form*2. На рис. 2. наведено результат рішення задачі.

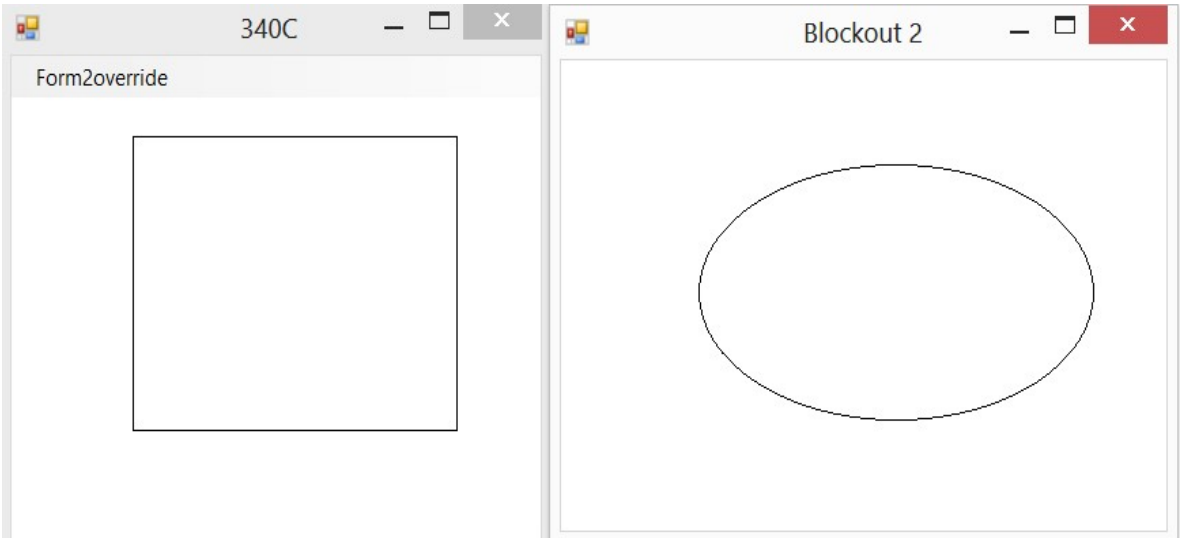

Рис. 2. Використання інтерфейсу при малюванні гумових контурів

Приклад 3. Цей приклад зроблено на основі *прикладу 2*. Однак є дві суттєві відмінності. По-перше, при виході мишки за межі клієнтського вікна будується прямокутник, утворений перетином клієнтського вікна і прямокутником, утворений мишкою. При цьому припиняється захоплення мишки. По-друге, у програмі малювання виконується з використанням *XOR*-малювання. Як вже зазначалося перевага цієї методики у тому, що друга *XOR*-лінія, задана тими ж координатами що і перша, стирає повністю першу лінію. У прикладі використаний інтерфейс, який використовується для двох класів відповідно до *Form*1 та *Form*2 (лістинги 3а і 3б). Результат рішення наведено на рис. 3.

Лістинг 3а (exam3)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace exam30
{
  interface ICapture
    {
        void OnLostCapture();
        void Down(Point a);
        void Move(Point a);
    }

    public partial class Form1 : Form, ICapture
    {
```

```
 public Point ptbeg, ptend;
 Form2 f = null;
 public Form1()
 {
     InitializeComponent();
     Text = "Blockout";
     BackColor = SystemColors.Window;
     ForeColor = SystemColors.WindowText;
     ptbeg = ptend = Point.Empty;
     CaptureLoss win = new CaptureLoss();
     win.control = this;
     win.AssignHandle(this.Handle);
 }

private void Form1_Paint(object sender, PaintEventArgs e)
{
  Graphics gr = CreateGraphics();
  gr.DrawRectangle(new Pen(ForeColor), ptbeg.X, ptbeg.Y,
  ptend.X - ptbeg.X, ptend.Y - ptbeg.Y);
  gr.Dispose();
}

 public void OnLostCapture()
 {
    Invalidate();
 }

 public void Down(Point a)
 {
     ptbeg = ptend = a;
 }

 public void Move(Point a)
 {
     Rectangle rect;
     Point pt1, pt2;
     pt1 = PointToScreen(ptbeg);
     Peres(ref pt1);
     pt2 = PointToScreen(ptend);
     rect = new Rectangle(pt1.X, pt1.Y, pt2.X - pt1.X,
                          pt2.Y - pt1.Y);
     ControlPaint.DrawReversibleFrame(rect,
         Color.FromArgb(255, 255, 0), FrameStyle.Dashed);
     ptend = a;
     Peres(ref ptend);
     pt2 = PointToScreen(ptend);
     rect = new Rectangle(pt1.X, pt1.Y, pt2.X - pt1.X,
                          pt2.Y - pt1.Y);
     ControlPaint.DrawReversibleFrame(rect,
         Color.FromArgb(255, 255, 0), FrameStyle.Dashed);
 }
```

```
     private void Peres(ref Point pt)
     {
      pt.X = (pt.X < ClientRectangle.Left) ?
              ClientRectangle.Left : pt.X;
      pt.X = (pt.X > ClientRectangle.Right) ?
              ClientRectangle.Right : pt.X;
      pt.Y = (pt.Y < ClientRectangle.Top + menuStrip1.Height) ?
              ClientRectangle.Top + menuStrip1.Height : pt.Y;
      pt.Y = (pt.Y > ClientRectangle.Bottom) ?
              ClientRectangle.Bottom : pt.Y;
        }

     private void form2ToolStripMenuItem_Click(object sender,
                                               EventArgs e)
        {
            if (f == null)
            {
                f = new Form2();
                f.Show();
            }
            else
            {
                f.Close();
                f = null;
            }
        }
    }

class CaptureLoss : NativeWindow
{
  public ICapture control;

  protected override void  WndProc(ref Message m)
   {
    // WM_LBUTTONDOWN - 513
     if (m.Msg == 513)
    {
        short a = (short)m.LParam;
        short b = (short)(((int)m.LParam)/65536);
        Point pp = new Point(a, b);
        control.Down(pp);
    }

     // WM_MOUSEMOVE - 512
     if (m.Msg == 512&&((int)m.WParam)==1)
     {
         short a = (short)m.LParam;
         short b = (short)(((int)m.LParam) / 65536);
         Point pp = new Point(a, b);
         control.Move(pp);
     }
```

```
          // WM_CAPTURECHANGED - 533

          if (m.Msg == 533)
          control.OnLostCapture();
          base.WndProc(ref m);
    }
   }
}
```

Лістинг 3б (exam3)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace exam30
{
    public partial class Form2 : Form, ICapture
    {
        public Point ptbeg, ptend;

        public Form2()
        {
            InitializeComponent();
            Text = "Blockout 2";
            BackColor = SystemColors.Window;
            ForeColor = SystemColors.WindowText;
            ptbeg = ptend = Point.Empty;
            CaptureLoss win = new CaptureLoss();
            win.control = this;
            win.AssignHandle(Handle);
        }

       private void Form2_Paint(object sender, PaintEventArgs e)
       {
           if (Math.Abs(ptbeg.X - ptend.X) < 5 ||
               Math.Abs(ptbeg.Y - ptend.Y) < 5)
           ptbeg.X = ptend.X = ptbeg.Y = ptend.Y;
           e.Graphics.DrawEllipse(new Pen(ForeColor), ptbeg.X,
               ptbeg.Y,ptend.X - ptbeg.X, ptend.Y - ptbeg.Y);
        }

        public void OnLostCapture()
        {
          Invalidate();
        }
```

```
        public void Down(Point a)
        {
            ptbeg = ptend = a;
        }

      public void Move(Point a)
      {
          Graphics grfx = CreateGraphics();
          grfx.DrawEllipse(new Pen(BackColor), ptbeg.X, ptbeg.Y,
          ptend.X - ptbeg.X, ptend.Y - ptbeg.Y);
          ptend = a;
          grfx.DrawEllipse(new Pen(ForeColor), ptbeg.X, ptbeg.Y,
          ptend.X - ptbeg.X, ptend.Y - ptbeg.Y);
          grfx.Dispose();
      }
  }
}
```

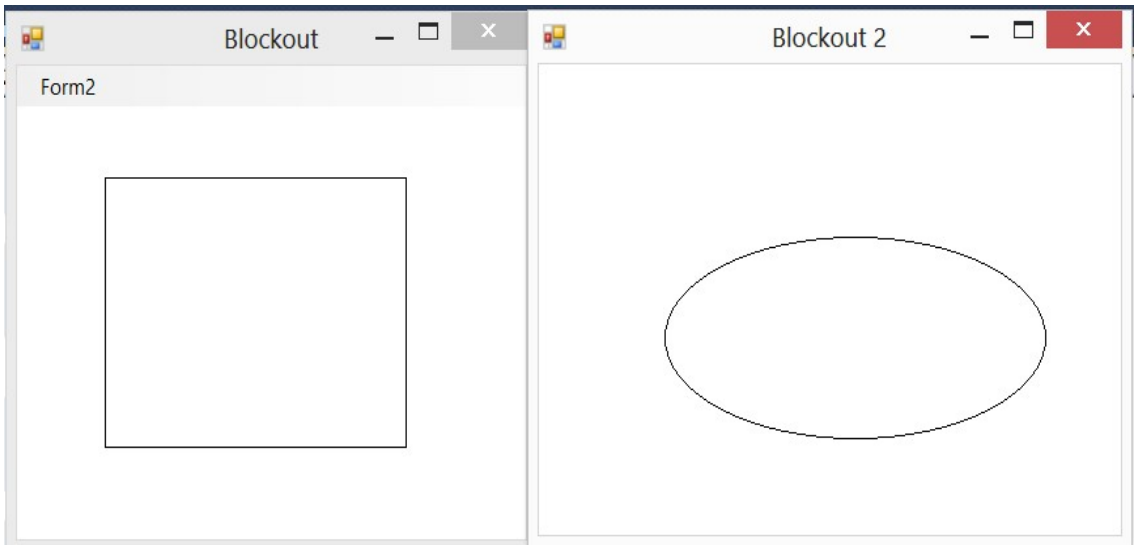

Рис. 3. До *XOR*-малювання гумових контурів

У програмі перетин прямокутників виконується функцією *Peres*(). *XOR*-малювання виконується при використанні статичного методу класу *ControlPaint*:

ControlPaint.DrawReversibleFrame().

Приклад використання цієї функції у програмі:

ControlPaint.DrawReversibleFrame(rect, Color.FromArgb(255, 255, 0), FrameStyle.Dashed).