**Міністерство освіти і науки України**

**Національний університет «Запорізька Політехніка»**

Кафедра програмних засобів

**ЗВІТ**

з лабораторної роботи №5

з дисципліни «Основи програмної інженерії» на тему:

«Обробка подій миші»

**Виконав:**

Студент групи КНТ-122 О. А. Онищенко

**Прийняли:**

Викладач: О. І. Качан

Викладач: Т. І. Каплієнко

2023

Обробка подій миші

Мета роботи

Навчитися основним принципам обробки подій маніпулятора типу миша усередовищі Visual Studio C#.

Завдання до роботи

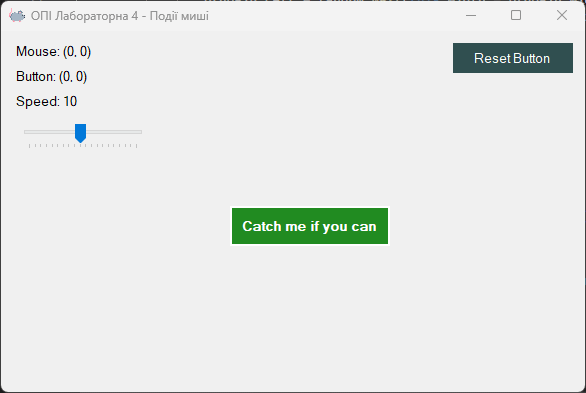
Виконати наступні завдання:

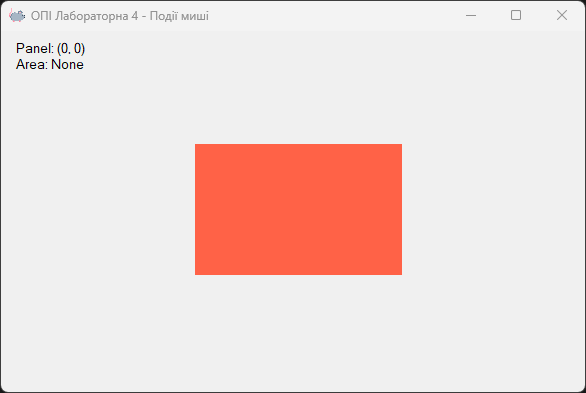
- реалізувати кнопку, чуттєву до руху миші. При її наведенні вона повинна рухатися у бік протилежний, або будь який інший бік, від курсору. Передбачити демонстрування координат курсору та кнопки, також додати можливість регулювання швидкості реакції кнопки. Кожний раз при натисканні кнопки виводити повідомлення про перемогу, та можливість спробувати ще;

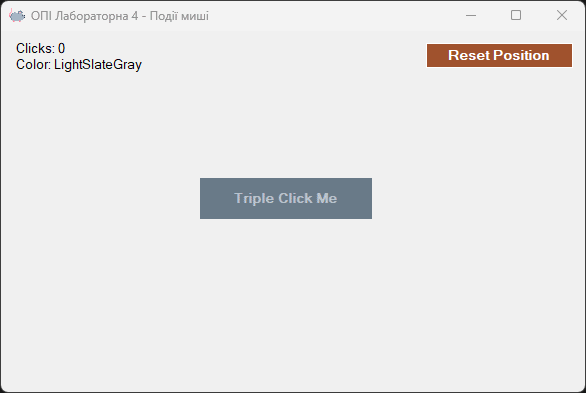
- реалізувати програму, де певний елемент управління можна переміщувати за допомогою миші у певні 4 області форми, де цей елемент притягується до різних сторін форми. Кожна область до певної сторони, без повторів. Стан об’єкту виводити на екран - координати, та сутність області де він знаходиться;

- реалізувати обробку подій потрійного натискання кнопки миші. Обрати елементи керування програмою (свій вибір обґрунтувати) та за допомогою потрійного натискання миші реалізувати зміну положення елементуізміну його стилю.

Результати виконання роботи







Код

*// Завдання 1*

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace dev

{

    public partial class FormMain : Form

    {

        int speed = 10;

        Random random = new Random();

        public FormMain()

        {

            InitializeComponent();

            lblSpeed.Text = $"Speed: {speed}";

            trackBarSpeed.Minimum = 1;

            trackBarSpeed.Maximum = 20;

            trackBarSpeed.Value = 10;

            speed = trackBarSpeed.Value;

        }

        private void FormMain\_Load(object *sender*, EventArgs *e*)

        {

            btnMain.Text = "Catch me if you can";

            lblMousePos.Text = "Mouse: (0, 0)";

            lblBtnPos.Text = "Button: (0, 0)";

        }

        private void FormMain\_MouseMove(object *sender*, MouseEventArgs *e*)

        {

            int mouseX = *e*.X;

            int mouseY = *e*.Y;

            int buttonX = btnMain.Left + btnMain.Width / 2;

            int buttonY = btnMain.Top + btnMain.Height / 2;

            int dx = buttonX - mouseX;

            int dy = buttonY - mouseY;

            double angle = Math.Atan2(dy, dx);

            int distance = (int)Math.Sqrt(dx \* dx + dy \* dy);

            if (distance < 100)

            {

                int newX = buttonX + (int)(speed \* Math.Cos(angle));

                int newY = buttonY + (int)(speed \* Math.Sin(angle));

                if (newX < 0 || newX > this.Width)

                {

                    newX = buttonY - (int)(speed \* Math.Sin(angle));

                }

                btnMain.Left = newX - btnMain.Width / 2;

                btnMain.Top = newY - btnMain.Height / 2;

            }

            lblBtnPos.Text = $"Button: ({buttonX}, {buttonY})";

            lblMousePos.Text = $"Mouse: ({mouseX}, {mouseY})";

        }

        private void btnMain\_Click(object *sender*, EventArgs *e*)

        {

            DialogResult result = MessageBox.Show("You caught me! Well done!\nDo you want to try again?", "Congratulations! You won!", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

            if (result == DialogResult.Yes)

            {

                btnMain.Left = random.Next(this.Width - btnMain.Width);

                btnMain.Top = random.Next(this.Height - btnMain.Height);

            }

            else

            {

                this.Close();

            }

        }

        private void btnReset\_Click(object *sender*, EventArgs *e*)

        {

            btnMain.Left = random.Next(this.Width - btnMain.Width);

            btnMain.Top = random.Next(this.Height - btnMain.Height);

        }

        private void trackBarSpeed\_ValueChanged(object *sender*, EventArgs *e*)

        {

        }

    }

}

*// Завдання 2*

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace dev

{

    public partial class FormMain : Form

    {

        bool isDragging = false;

        Point offset;

        public FormMain()

        {

            InitializeComponent();

        }

        private void FormMain\_Load(object *sender*, EventArgs *e*)

        {

            pnlMain.BackColor = Color.Tomato;

            lblPos.Text = "Panel: (0, 0)\nArea: None";

        }

        private void pnlMain\_MouseDown(object *sender*, MouseEventArgs *e*)

        {

            if (*e*.Button == MouseButtons.Left)

            {

                isDragging = true;

                offset = new Point(*e*.X, *e*.Y);

            }

        }

        private void pnlMain\_MouseMove(object *sender*, MouseEventArgs *e*)

        {

            if (isDragging)

            {

                int panelX = pnlMain.Left;

                int panelY = pnlMain.Top;

                int newX = panelX + *e*.X - offset.X;

                int newY = panelY + *e*.Y - offset.Y;

                pnlMain.Left = newX;

                pnlMain.Top = newY;

                string area = "None";

                if (newX < this.Width / 2 && newY < this.Height / 2)

                    area = "Top-Left";

                else if (newX >= this.Width / 2 && newY < this.Height / 2)

                    area = "Top-Right";

                else if (newX < this.Width / 2 && newY >= this.Height / 2)

                    area = "Bottom-Left";

                else if (newX >= this.Width / 2 && newY >= this.Height / 2)

                    area = "Bottom-Right";

                lblPos.Text = $"Panel: ({newX}, {newY})\nArea: {area}";

            }

        }

        private void pnlMain\_MouseUp(object *sender*, MouseEventArgs *e*)

        {

            if (*e*.Button == MouseButtons.Left)

            {

                isDragging = false;

                int clientHeight = this.Height - SystemInformation.CaptionHeight - 2 \* SystemInformation.BorderSize.Height;

*// Snap the panel to the appropriate side*

                if (pnlMain.Left < this.Width / 2 && pnlMain.Top < this.Height / 2)

                    pnlMain.Location = new Point(0, 0);

                else if (pnlMain.Left >= this.Width / 2 && pnlMain.Top < this.Height / 2)

                    pnlMain.Location = new Point(this.Width - pnlMain.Width, 0);

                else if (pnlMain.Left < this.Width / 2 && pnlMain.Top >= this.Height / 2)

                    pnlMain.Location = new Point(0, clientHeight - pnlMain.Height);

                else if (pnlMain.Left >= this.Width / 2 && pnlMain.Top >= this.Height / 2)

                    pnlMain.Location = new Point(this.Width - pnlMain.Width, clientHeight - pnlMain.Height);

            }

        }

    }

}

*// Завдання 3*

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace dev

{

    public partial class FormMain : Form

    {

        int clicks = 0;

        Timer timer;

        Color[] colors = new Color[]

        {

            Color.Orange,

            Color.YellowGreen,

            Color.ForestGreen,

            Color.DimGray,

            Color.IndianRed,

            Color.Brown,

            Color.Firebrick,

            Color.Maroon,

            Color.DarkRed,

            Color.Chocolate,

            Color.LightSalmon,

            Color.LightSeaGreen,

            Color.LightSkyBlue,

            Color.LightSlateGray,

            Color.Lime,

            Color.LimeGreen

        };

        public FormMain()

        {

            InitializeComponent();

        }

        private void FormMain\_Load(object *sender*, EventArgs *e*)

        {

            timer = new Timer();

            timer.Interval = SystemInformation.DoubleClickTime;

            timer.Tick += Timer\_Tick;

            btnMain.BackColor = Color.YellowGreen;

            btnMain.ForeColor = ControlPaint.LightLight(btnMain.BackColor);

            lblInfo.Text = $"Clicks: {clicks}\nColor: {btnMain.BackColor.Name}";

        }

        private void Timer\_Tick(object *sender*, EventArgs *e*)

        {

            timer.Stop();

            clicks = 0;

        }

        private void btnMain\_Click(object *sender*, EventArgs *e*)

        {

            clicks++;

            lblInfo.Text = $"Clicks: {clicks}\nColor: {btnMain.BackColor.Name}";

            if (clicks == 3)

            {

                timer.Stop();

                clicks = 0;

                TripleClickAction();

                lblInfo.Text = $"Clicks: {clicks}\nColor: {btnMain.BackColor.Name}";

            }

            else

            {

                timer.Start();

            }

        }

        private void TripleClickAction()

        {

            Random random = new Random();

            int newX = random.Next(20, this.Width - btnMain.Width - 20);

            int newY = random.Next(20, this.Height - btnMain.Height - 20);

            btnMain.Left = newX;

            btnMain.Top = newY;

            int index = random.Next(0, colors.Length);

            btnMain.BackColor = colors[index];

            btnMain.ForeColor = ControlPaint.LightLight(btnMain.BackColor);

        }

        private void btnReset\_Click(object *sender*, EventArgs *e*)

        {

            btnMain.Location = new Point(199, 147);

        }

    }

}

Висновки

Таким чином, ми навчилися основним принципам обробки подій маніпулятора типу миша усередовищі Visual Studio C#.

Контрольні питання

**Які події миші оброблюються в C#?**

Події миші обробляються в C# за допомогою делегатів та обробників подій. Делегат - це тип, який визначає сигнатуру методу, а обробник події - це метод, який відповідає сигнатурі делегату і виконує певні дії при виникненні події. Події миші генеруються елементом управління або формою, яка отримує ввід миші, і їх можна обробити, приєднавши до події обробник події.

**Які параметри подій миші?**

Параметрами подій миші зазвичай є об'єкт, який представляє відправника події, та об'єкт MouseEventArgs, який містить інформацію про стан миші. Об'єкт MouseEventArgs має такі властивості, як Button, Clicks, Delta, Location, а також координати X та Y. Залежно від типу події миші, деякі з цих властивостей можуть мати різні значення. Наприклад, властивість Button вказує, яка кнопка миші була натиснута або відпущена, властивість Clicks вказує кількість натискань і відпускань кнопки миші, властивість Delta вказує величину переміщення коліщатка, а властивість Location вказує положення вказівника миші відносно верхнього лівого кута елемента управління.

**Як отримати поточні координати курсору?**

Для отримання поточних координат курсору можна скористатися статичною властивістю Cursor.Position класу Cursor. Ця властивість повертає структуру типу Point, яка представляє позицію курсору в екранних координатах. Для перетворення екранних координат у клієнтські координати елемента управління або форми можна використати метод PointToClient класу Control. Наприклад, щоб отримати поточні координати курсору відносно форми, можна написати