

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

Технології та засоби розробки комп'ютерної графіки та мультимедіа

Лабораторна робота №4

Виконав:
студент групи
ІО-25
Льоскін І. В.
Перевірила:
Хмелюк М. С.

Київ – 2024

Тема: Переміщення зображення по довільній траєкторії. Переміщення зображення за допомогою клавіатури. Переміщення зображення за допомогою маніпулятора миші

Завдання:

- намалювати або підключити об'єкт (зображення)
- змінити координати зображення і вивести його на нове місце і так далі
- відновити екран в місці, де було зображення
- реалізувати безперервну траєкторію руху зображення
- розташувати на екрані об'єкти-перешкоди, і забезпечити відбивання від них зображення, яке рухається
- додати зображення та забезпечити управління переміщенням і перемиканням між ними за допомогою клавіатури
- додати зображення та забезпечити управління переміщенням і перемиканням між зображеннями за допомогою маніпулятора миші

Хід роботи:

Аналогічно попереднім лабораторним роботам створюємо новий проект. Також створюємо нову іконку, встановлюємо її, додаємо задній фон формі, змінюємо ім'я форми та її текст. Також додаємо з Toolbox компонент PictureBox, загрузаємо у папку Resources довільне зображення, та у властивостях PictureBox, вказуємо полю SizeMode значення Zoom, щоб було видно всю картинку і вона не була обрізаною. Також додаємо чотири кнопки для того щоб “Зробити крок”, “Повернутись у стартове положення”, “Запустити нескінченне переміщення”, “Відкрити нову форму” Після чого приступаємо до написання коду. Для цього перейдемо в файл UserInteractionForm.cs:

```
{
    public UserInteractionForm()
    {
        InitializeComponent();
        this.ClientSize = new Size(800, 600);

        // create obstacle
        obstacle1 = new Panel
        {
            BackColor = Color.Red,
            Size = new Size(70, 70),
            Location = new Point(300, 150)
        };
        // add obstacle on the form
    }
}
```

```

        this.Controls.Add(obstacle1);
        initialImagePosition = new Point(100, 100);
    }

    private Panel obstacle1;
    private Point initialImagePosition;
    Timer moveTimer = new Timer();
    int xSpeed = 20;
    int ySpeed = 20;

    // move PictureBox on press the button
    private void moveButton_Click(object sender, EventArgs e)
    {
        if (pictureBox1.Location.X >= ClientSize.Width - pictureBox1.Width
|| pictureBox1.Location.X <= 0)
        {
            xSpeed = -xSpeed;
        }
        if (pictureBox1.Location.Y >= ClientSize.Height - pictureBox1.Height
|| pictureBox1.Location.Y <= 50)
        {
            ySpeed = -ySpeed;
        }

        pictureBox1.Location = new Point(pictureBox1.Location.X + xSpeed,
pictureBox1.Location.Y + ySpeed);
    }

    // set PictureBox on start position
    private void resetButton_Click(object sender, EventArgs e)
    {
        moveTimer.Stop();
        pictureBox1.Location = initialImagePosition;
        xSpeed = Math.Abs(xSpeed);
        ySpeed = Math.Abs(ySpeed);
    }

    // start infinity move of PictureBox with x and y speed
    private void InfMoveButton_Click(object sender, EventArgs e)
    {
        moveTimer.Tick += (s, EventArgs) =>
        {
            if (IsColliding(pictureBox1, obstacle1))
            {
                if (pictureBox1.Location.X < obstacle1.Location.X)
                {
                    xSpeed = -Math.Abs(xSpeed); // change vector of
movement if met right side of obstacle

```

```

        }
        else
        {
            xSpeed = Math.Abs(xSpeed); // change vector of
movement if met left side of obstacle
        }

        if (pictureBox1.Location.Y < obstacle1.Location.Y)
        {
            ySpeed = -Math.Abs(ySpeed); // change vector of
movement if met top side of obstacle
        }
        else
        {
            ySpeed = Math.Abs(ySpeed); // change vector of
movement if met bottom side of obstacle
        }
    }

    // change vector of movement if met one of the sides
    if (pictureBox1.Location.X >= ClientSize.Width -
pictureBox1.Width || pictureBox1.Location.X <= 0)
    {
        xSpeed = -xSpeed;
    }
    if (pictureBox1.Location.Y >= ClientSize.Height -
pictureBox1.Height || pictureBox1.Location.Y <= 50)
    {
        ySpeed = -ySpeed;
    }

    // update position
    pictureBox1.Location = new Point(pictureBox1.Location.X +
xSpeed, pictureBox1.Location.Y + ySpeed);
};
    moveTimer.Start();
}

// method that check the colliding with obstacle
private bool IsColliding(Control pictureBox, Panel obstacle)
{
    Rectangle pictureBoxRect = pictureBox.Bounds;
    Rectangle obstacleRect = obstacle.Bounds;

    return pictureBoxRect.Intersects(obstacleRect);
}

private void pictureBox1_Click(object sender, EventArgs e)
{
}

private void Form1_Load(object sender, EventArgs e)
{

```

```

    }
    // open second form
    private void openButton_Click(object sender, EventArgs e)
    {
        Form1 secondForm = new Form1();
        secondForm.Show();
    }
}

```

Пояснення коду для форми UserInteractionForm:

Ініціалізація форми, задаємо розмір, створюємо перешкоду якій визначаємо колір, розмір та позицію. Розміщуємо перешкоду на формі та ініціалізуємо стартову позицію зображення

```

public UserInteractionForm()
{
    InitializeComponent();
    this.ClientSize = new Size(800, 600);

    // create obstacle
    obstacle1 = new Panel
    {
        BackColor = Color.Red,
        Size = new Size(70, 70),
        Location = new Point(300, 150)
    };
    // add obstacle on the form
    this.Controls.Add(obstacle1);
    initialImagePosition = new Point(100, 100);
}

```

*Визначаємо змінні що вказували при створенні форми, а також створюємо значення, які будуть відповідати за відстань на яку переміщуватиметься **PictureBox***

```

private Panel obstacle1;
private Point initialImagePosition;
Timer moveTimer = new Timer();
int xSpeed = 20;
int ySpeed = 20;

```

Переміщуємо зображення на задану “швидкість”, але перед цим перевіряємо, чи зіштовхнулось зображення з однією зі сторін форми. Якщо так тоді змінюємо напрямок імітуючи відбивання

```
// move PictureBox on press the button
1 reference
private void moveButton_Click(object sender, EventArgs e)
{
    if (pictureBox1.Location.X >= ClientSize.Width - pictureBox1.Width || pictureBox1.Location.X <= 0)
    {
        xSpeed = -xSpeed;
    }
    if (pictureBox1.Location.Y >= ClientSize.Height - pictureBox1.Height || pictureBox1.Location.Y <= 50)
    {
        ySpeed = -ySpeed;
    }

    pictureBox1.Location = new Point(pictureBox1.Location.X + xSpeed, pictureBox1.Location.Y + ySpeed);
}
}
```

Зупиняємо таймер, встановлюємо зображенню початкову позицію та встановлюємо значення швидкості по модулю, тобто додатні

```
// set PictureBox on start position
1 reference
private void resetButton_Click(object sender, EventArgs e)
{
    moveTimer.Stop();
    pictureBox1.Location = initialImagePosition;
    xSpeed = Math.Abs(xSpeed);
    ySpeed = Math.Abs(ySpeed);
}
}
```

*Для того щоб запустити нескінченний рух по траєкторії необхідно спершу зробити перевірку чи не було зіткнень зі сторонами переешкоди, якщо вона була, тоді визначаємо з якою саме і відповідно змінюємо напрямок руху. Якщо зіткнення було з лівою або правою стороною, тоді змінюємо **xSpeed**, якщо з верхньою чи нижньою - **ySpeed**. Після чого оновлюємо позицію для **PictureBox***

```

moveTimer.Tick += (s, EventArgs) =>
{
    if (IsColliding(pictureBox1, obstacle1))
    {
        if (pictureBox1.Location.X < obstacle1.Location.X)
        {
            xSpeed = -Math.Abs(xSpeed); // change vector of movement if met right side of obstacle
        }
        else
        {
            xSpeed = Math.Abs(xSpeed); // change vector of movement if met left side of obstacle
        }

        if (pictureBox1.Location.Y < obstacle1.Location.Y)
        {
            ySpeed = -Math.Abs(ySpeed); // change vector of movement if met top side of obstacle
        }
        else
        {
            ySpeed = Math.Abs(ySpeed); // change vector of movement if met bottom side of obstacle
        }
    }

    // change vector of movement if met one of the sides
    if (pictureBox1.Location.X >= ClientSize.Width - pictureBox1.Width || pictureBox1.Location.X <= 0)
    {
        xSpeed = -xSpeed;
    }
    if (pictureBox1.Location.Y >= ClientSize.Height - pictureBox1.Height || pictureBox1.Location.Y <= 50)
    {
        ySpeed = -ySpeed;
    }

    // update position
    pictureBox1.Location = new Point(pictureBox1.Location.X + xSpeed, pictureBox1.Location.Y + ySpeed);
};
moveTimer.Start();

```

Функція яка перевіряє чи було зіткнення з перешкодою

```

private bool IsColliding(Control pictureBox, Panel obstacle)
{
    Rectangle pictureBoxRect = pictureBox.Bounds;
    Rectangle obstacleRect = obstacle.Bounds;

    return pictureBoxRect.Intersects(obstacleRect);
}

```

Кнопка що відкриває другу форму

```

// open second form
1 reference
private void openButton_Click(object sender, EventArgs e)
{
    Form1 secondForm = new Form1();
    secondForm.Show();
}

```

Тепер приступаємо до роботи зі взаємодією користувача та рухом зображень. Для цього створюємо нову форму, а в ній дві компоненти **PictureBox**. Після чого приступаємо до написання коду для Form1.cs:

```
{
    public Form1()
    {
        InitializeComponent();
        this.KeyDown += new KeyEventHandler(Form1_KeyDown);

        // setup current active PictureBox as pictureBox1 by default
        activePictureBox = pictureBox1;

        pictureBox1.MouseDown += new MouseEventHandler(PictureBox_MouseDown);
        pictureBox1.MouseMove += new MouseEventHandler(PictureBox_MouseMove);
        pictureBox1.MouseUp += new MouseEventHandler(PictureBox_MouseUp);

        pictureBox2.MouseDown += new MouseEventHandler(PictureBox_MouseDown);
        pictureBox2.MouseMove += new MouseEventHandler(PictureBox_MouseMove);
        pictureBox2.MouseUp += new MouseEventHandler(PictureBox_MouseUp);
    }

    private PictureBox activePictureBox; // current PictureBox
    private bool isDragging = false; // flag for dragging
    private Point mouseOffset; // cursor shift related of PictureBox

    private void Form1_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode == Keys.Tab)
        {
            activePictureBox = activePictureBox == pictureBox1 ? pictureBox2
: pictureBox1;
        }
        int moveDistance = 10; // distance for moving the PictureBox

        // change coordinates of pictureBox1 depending of pressed button
        if (e.KeyCode == Keys.Up)
        {
            if (activePictureBox.Location.Y > 20)
            {
                activePictureBox.Location = new
Point(activePictureBox.Location.X, activePictureBox.Location.Y - 20);
            }
        }
        else if (e.KeyCode == Keys.Down)
        {
            if (activePictureBox.Bottom < ClientSize.Height - 20)
            {
                activePictureBox.Location = new
Point(activePictureBox.Location.X, activePictureBox.Location.Y + 20);
            }
        }
        else if (e.KeyCode == Keys.Left)
```



```

    {
        if (activePictureBox.Left > 20)
        {
            activePictureBox.Location = new
Point(activePictureBox.Location.X - 20, activePictureBox.Location.Y);
        }
    }
    else if (e.KeyCode == Keys.Right)
    {
        if (activePictureBox.Right < ClientSize.Width - 20)
        {
            activePictureBox.Location = new
Point(activePictureBox.Location.X + 20, activePictureBox.Location.Y);
        }
    }
    Invalidate(); // render
}

private void PictureBox_MouseDown(object sender, MouseEventArgs e)
{
    // set active PictureBox
    activePictureBox = sender as PictureBox;

    if (activePictureBox != null && e.Button == MouseButtons.Left)
    {
        // run the procces of drugging
        isDragging = true;

        // calculste cursor shift related to PictureBox
        mouseOffset = new Point(e.X, e.Y);
    }
}

private void PictureBox_MouseMove(object sender, MouseEventArgs e)
{
    if (isDragging && activePictureBox != null)
    {
        // calculate new position of PictureBox
        Point newLocation = activePictureBox.Location;
        newLocation.X += e.X - mouseOffset.X;
        newLocation.Y += e.Y - mouseOffset.Y;
        activePictureBox.Location = newLocation;
    }
}

private void PictureBox_MouseUp(object sender, MouseEventArgs e)
{
    // end dragging
    if (e.Button == MouseButtons.Left)
    {
        isDragging = false;
    }
}

```

```

private void Form1_Load(object sender, EventArgs e)
{
    }
}

```

Пояснення коду:

*Ініціалізуємо форму, підписуємось на подію яка відповідатиме за взаємодію з клавіатурою **Form1_KeyDown**, створюємо змінну, яка слідує за тим, який **PictureBox** зараз вибраний, для обох зображень підписуємось на події що відповідають за натискання клавіші миші, перетягування миші та відпускання клавіші: **PictureBox_MouseDown**, **PictureBox_MouseMove** та **PictureBox_MouseUp** - відповідно*

```

public Form1()
{
    InitializeComponent();
    this.KeyDown += new KeyEventHandler(Form1_KeyDown);

    // setup current active PictureBox as pictureBox1 by default
    activePictureBox = pictureBox1;

    pictureBox1.MouseDown += new MouseEventHandler(PictureBox_MouseDown);
    pictureBox1.MouseMove += new MouseEventHandler(PictureBox_MouseMove);
    pictureBox1.MouseUp += new MouseEventHandler(PictureBox_MouseUp);

    pictureBox2.MouseDown += new MouseEventHandler(PictureBox_MouseDown);
    pictureBox2.MouseMove += new MouseEventHandler(PictureBox_MouseMove);
    pictureBox2.MouseUp += new MouseEventHandler(PictureBox_MouseUp);
}

```

Ініціалізуємо змінні, що відповідають за поточне зображення, показник чи клавіша натиснута, та значення на яке зсунуто зображення

```

private PictureBox activePictureBox; // current PictureBox
private bool isDragging = false; // lag for dragging
private Point mouseOffset; // cursore shift related of PictureBox

```

*Коли натискаємо на клавішу **Tab** то змінюється курування з поточного зображення на інше, якщо були натиснуті **стрілки**, тоді іде перевірка чи можна робити зсув у відповідну сторону - якщо так тоді зсуваємо зображення*

```

private void Form1_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Tab)
    {
        activePictureBox = activePictureBox == pictureBox1 ? pictureBox2 : pictureBox1;
    }
    int moveDistance = 10; // distance for moving the PictureBox

    // change coordinates of pictureBox1 depending of pressed button
    if (e.KeyCode == Keys.Up)
    {
        if (activePictureBox.Location.Y > 20)
        {
            activePictureBox.Location = new Point(activePictureBox.Location.X, activePictureBox.Location.Y - 20);
        }
    }
    else if (e.KeyCode == Keys.Down)
    {
        if (activePictureBox.Bottom < ClientSize.Height - 20)
        {
            activePictureBox.Location = new Point(activePictureBox.Location.X, activePictureBox.Location.Y + 20);
        }
    }
    else if (e.KeyCode == Keys.Left)
    {
        if (activePictureBox.Left > 20)
        {
            activePictureBox.Location = new Point(activePictureBox.Location.X - 20, activePictureBox.Location.Y);
        }
    }
    else if (e.KeyCode == Keys.Right)
    {
        if (activePictureBox.Right < ClientSize.Width - 20)
        {
            activePictureBox.Location = new Point(activePictureBox.Location.X + 20, activePictureBox.Location.Y);
        }
    }
    Invalidate(); // render
}

```

Визначаємо яке зображення буде перетягнуто. Якщо таке зображення існує і натиснута ліва кнопка миші, тоді сетаємо значення **isDragging** в **true**, щоб знати, що іде процес перетягування. Розраховується зсув курсора миші відносно верхнього лівого кута **PictureBox**

```

private void PictureBox_MouseDown(object sender, MouseEventArgs e)
{
    // set active PictureBox
    activePictureBox = sender as PictureBox;

    if (activePictureBox != null && e.Button == MouseButtons.Left)
    {
        // run the procces of drugging
        isDragging = true;

        // calculste cursor shift related to PictureBox
        mouseOffset = new Point(e.X, e.Y);
    }
}

```

Якщо іде перетягування і таке зображення існує, тоді обчислюємо поточне значення позиції поточного зображення

2 references

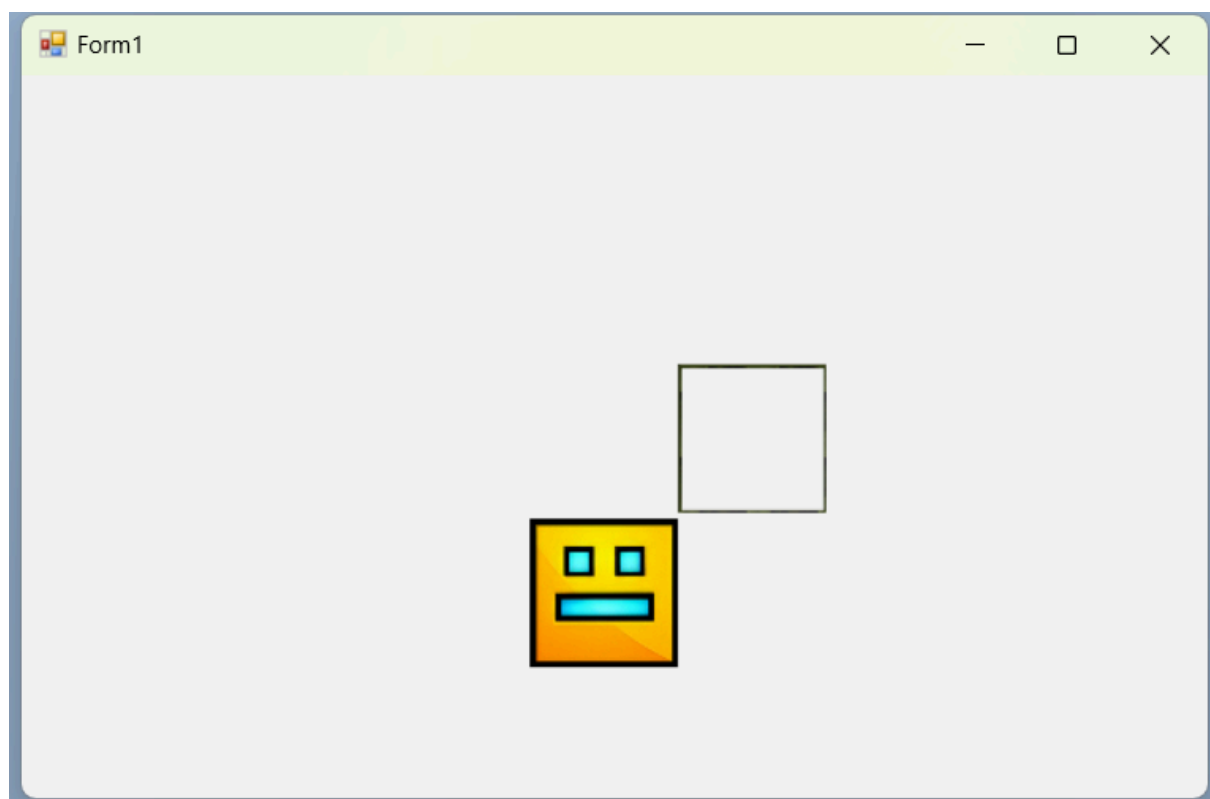
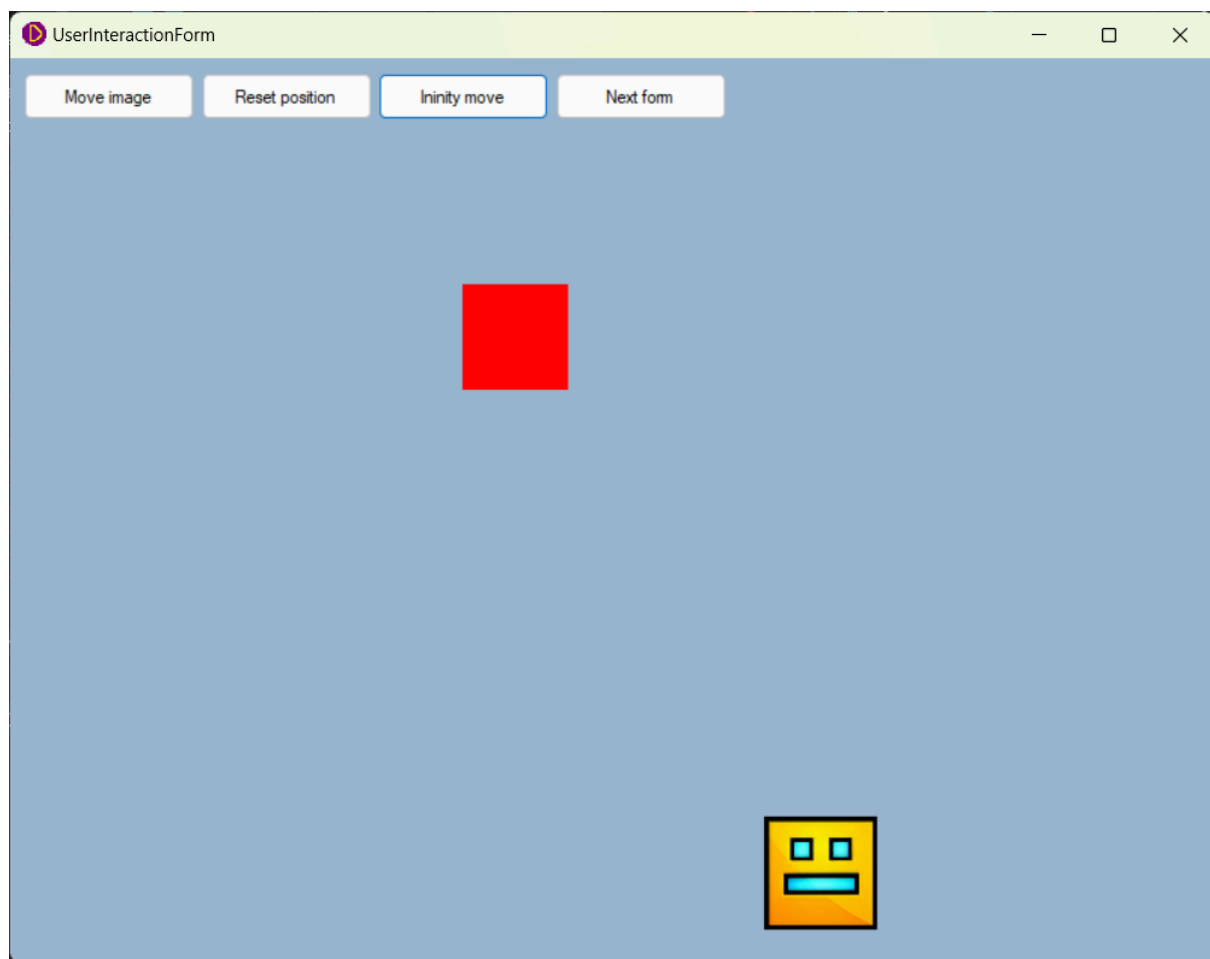
```
private void PictureBox_MouseMove(object sender, MouseEventArgs e)
{
    if (isDragging && activePictureBox != null)
    {
        // calculate new position of PictureBox
        Point newLocation = activePictureBox.Location;
        newLocation.X += e.X - mouseOffset.X;
        newLocation.Y += e.Y - mouseOffset.Y;
        activePictureBox.Location = newLocation;
    }
}
```

Накінець, якщо ми відпускаємо ліву кнопку миші, тоді процес перетягування завершено

2 references

```
private void PictureBox_MouseUp(object sender, MouseEventArgs e)
{
    // end dragging
    if (e.Button == MouseButtons.Left)
    {
        isDragging = false;
    }
}
```

Результат:



Висновок:

Виконуючи цю лабораторну роботу я зіштовхнувся з проблемою того, як відстежити відбивання від перешкод і це було дуже цікаво. Також ознайомився зі взаємодією графічних компонентів з клавіатурою та методом drag and drop в Windows Form Application