МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Український державний університет науки і технологій**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №2**

**з дисципліни «Комп’ютерна графіка»**

**на тему:** **«Системи координат та афінні перетворення.»**

Виконав: студент гр. ПЗ2011

Кулик С. В.

Прийняла: Шаповал І. В.

.

Дніпро, 2022

**Тема:** Системи координат та афінні перетворення.

**Мета:** Отримати навички використання афінних перетворень. Вивчити графічні можливості середи, в якій створювалась програма.

**Завдання**

У середовищі розробки Builder З++ розробити об'єктно-орієнтовану програму для виконання операцій перетворення над графічними об'єктами на площині.

Вимоги до програми:

- графічний об'єкт є ієрархією графічних об'єктів;

- користувач може обирати яка фігура буде змінюватися;

- перетворення повинні виконуватися над початковим об'єктом і його складовими, які задає користувач;

- перетворення повинні виконуватися відносно точки, розташування якої задає користувач;

- перетворення, які може робити користувач:

а) пересувати об’єкт вверх, вниз, вліво та вправо;

б) повертати об’єкт за часовою стрілкою або проти відносно обраної точки;

в) змінювати масштаб об’єкта по осі Х або осі Y відносно обраної точки;

**Індивідуальне завдання**

Фігура «Будинок», яка складається з:

* Даху;
* Стін
* Вікон
* Дверей

**Структура графічного об’єкту (у вигляді дерева)**

****

**Операції перетворення графічних об’єктів на площині у матричному вигляді**

Матриця переміщення:



Матриця масштабування:



Матриця повороту:



**Текст програми**

*Файл Ioperations.cs*

using System.Drawing;

namespace ComputerGraphicLB2

{

internal interface IOperations

{

void Move(int x, int y);

void Scale(int x, int y, double k);

void Rotate(int x, int y, int angle);

void Draw(Graphics g);

}

}

*Файл Ioperations.cs*

using System;

using System.Collections.Generic;

using System.Drawing;

namespace ComputerGraphicLB2

{

public class GraphicsElement : IOperations

{

public string name { get; set; }

protected PointF[] points;

public void Draw(Graphics g)

{

g.DrawLines(Pens.Black, points);

}

public void Move(int x, int y)

{

for (int i = 0; i < points.Length; i++)

{

PointF currentPoint = points[i];

currentPoint.X += x;

currentPoint.Y += y;

points[i] = currentPoint;

}

}

public void Rotate(int x, int y, int angle)

{

for (int i = 0; i < points.Length; i++)

{

PointF currentPoint = points[i];

double angleRadian = angle \* Math.PI / 180;

float ox = currentPoint.X;

float oy = currentPoint.Y;

currentPoint.X = (float)((ox - x) \* Math.Cos(angleRadian) - (oy - y) \* Math.Sin(angleRadian) + x);

currentPoint.Y = (float)((ox - x) \* Math.Sin(angleRadian) + (oy - y) \* Math.Cos(angleRadian) + y);

points[i] = currentPoint;

}

}

public void Scale(int x, int y, double k)

{

for (int i = 0; i < points.Length; i++)

{

PointF currentPoint = points[i];

currentPoint.X -= x;

currentPoint.Y -= y;

currentPoint.X = (float)(currentPoint.X \* k);

currentPoint.Y = (float)(currentPoint.Y \* k);

currentPoint.X += x;

currentPoint.Y += y;

points[i] = currentPoint;

}

}

}

public class Circle : GraphicsElement

{

public Circle(string name, int x, int y, int radius)

{

//уравнение окружности

this.name = name;

points = new PointF[101];

for (int i = 0; i < points.Length; i++)

{

points[i] = new PointF((float)(x + radius \* Math.Cos(i \* 2 \* Math.PI / 100)), (float)(y + radius \* Math.Sin(i \* 2 \* Math.PI / 100)));

}

}

}

public class Rectangle : GraphicsElement

{

public Rectangle(string name, int x, int y, int width, int height)

{

this.name = name;

points = new PointF[5];

points[0] = new PointF(x, y);

points[1] = new PointF(x + width, y);

points[2] = new PointF(x + width, y + height);

points[3] = new PointF(x, y + height);

points[4] = new PointF(x, y);

}

}

public class Triangle : GraphicsElement

{

public Triangle(string name, int x1, int y1, int x2, int y2, int x3, int y3)

{

this.name = name;

points = new PointF[4];

points[0] = new PointF(x1, y1);

points[1] = new PointF(x2, y2);

points[2] = new PointF(x3, y3);

points[3] = new PointF(x1, y1);

}

}

public class Picture : IOperations

{

public List<GraphicsElement> elements = new List<GraphicsElement>();

public List<bool> selected = new List<bool>();

public Picture()

{

elements.Add(new Rectangle("Стіна", 200, 300, 500, 250));

elements.Add(new Triangle("Дах", 200, 300, 450, 150, 700, 300));

elements.Add(new Circle("Вікно 1", 300, 400, 50));

elements.Add(new Circle("Вікно 2", 600, 400, 50));

elements.Add(new Rectangle("Двері", 400, 400, 100, 150));

for (int i = 0; i < elements.Count; i++)

{

selected.Add(false);

}

}

public void Draw(Graphics g)

{

foreach (var item in elements)

{

item.Draw(g);

}

}

public void Move(int x, int y)

{

for (int i = 0; i < elements.Count; i++)

{

if (selected[i])

{

elements[i].Move(x, y);

}

}

}

public void Rotate(int x, int y, int angle)

{

for (int i = 0; i < elements.Count; i++)

{

if(selected[i])

elements[i].Rotate(x, y, angle);

}

}

public void Scale(int x, int y, double k)

{

for (int i = 0; i < elements.Count; i++)

{

if (selected[i])

elements[i].Scale(x, y, k);

}

}

public void checkIt(int index, bool value)

{

selected[index] = value;

}

}

}

*Файл Form1.cs*

using System;

using System.Drawing;

using System.Windows.Forms;

namespace ComputerGraphicLB2

{

public partial class Form1 : Form

{

Picture picture;

int cX, cY;

public Form1()

{

InitializeComponent();

picture = new Picture();

foreach (var item in picture.elements)

{

checkedListBox1.Items.Add(item.name);

}

}

private void pictureBox1\_Paint(object sender, PaintEventArgs e)

{

picture.Draw(e.Graphics);

e.Graphics.DrawEllipse(new Pen(Color.Red, 2), cX, cY, 5, 5);

}

private void buttonMoveUp\_Click(object sender, EventArgs e)

{

picture.Move(0, -5);

pictureBox1.Refresh();

}

private void checkedListBox1\_ItemCheck(object sender, ItemCheckEventArgs e)

{

picture.checkIt(e.Index, e.NewValue == CheckState.Checked);

}

private void buttonMoveRight\_Click(object sender, EventArgs e)

{

picture.Move(5, 0);

pictureBox1.Refresh();

}

private void buttonMoveLeft\_Click(object sender, EventArgs e)

{

picture.Move(-5, 0);

pictureBox1.Refresh();

}

private void buttonMoveDown\_Click(object sender, EventArgs e)

{

picture.Move(0, 5);

pictureBox1.Refresh();

}

private void pictureBox1\_MouseClick(object sender, MouseEventArgs e)

{

cX = e.X;

cY = e.Y;

pictureBox1.Refresh();

}

private void buttonRotateLeft\_Click(object sender, EventArgs e)

{

picture.Rotate(cX,cY,-5);

pictureBox1.Refresh();

}

private void buttonScaleUp\_Click(object sender, EventArgs e)

{

picture.Scale(cX, cY, 1.1);

pictureBox1.Refresh();

}

private void buttonScaleDown\_MouseClick(object sender, MouseEventArgs e)

{

picture.Scale(cX, cY, 0.9);

pictureBox1.Refresh();

}

private void pictureBox1\_Resize(object sender, EventArgs e)

{

pictureBox1.Refresh();

}

private void buttonCheckAll\_Click(object sender, EventArgs e)

{

for (int i = 0; i < checkedListBox1.Items.Count; i++)

{

checkedListBox1.SetItemChecked(i, true);

}

}

private void checkBoxCheckAll\_CheckedChanged(object sender, EventArgs e)

{

for (int i = 0; i < checkedListBox1.Items.Count; i++)

{

checkedListBox1.SetItemChecked(i, checkBoxCheckAll.Checked);

}

}

private void buttonRotateRight\_Click(object sender, EventArgs e)

{

picture.Rotate(cX, cY, 5);

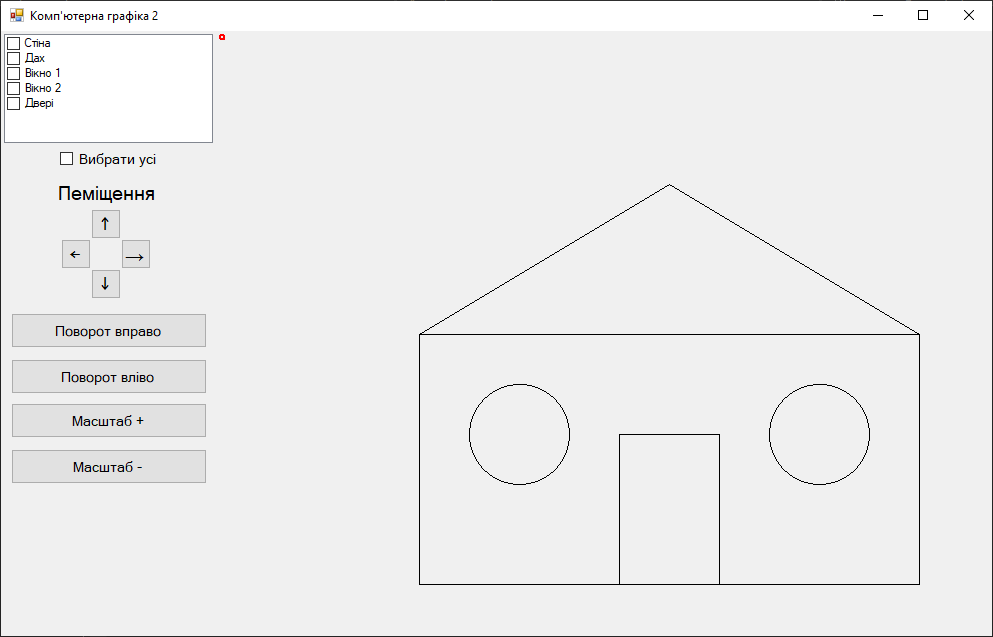
pictureBox1.Refresh();

}

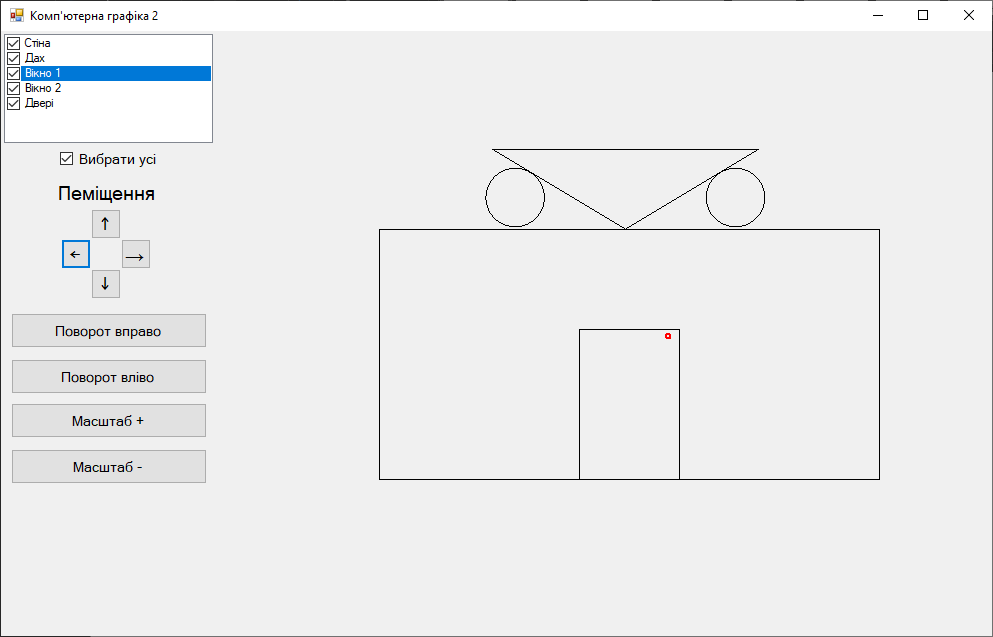
}

}

**Демонстрація роботи програми**

**

*Рис. 1 – Початкова форма*

**

*Рис. 2 – Форма після афінних пертворень*

**Аналіз результатів**

Всі створені графічні примітиви можна переміщати, масштабувати та повертати як в купі, так і окремо. Масшатабування та вовод відбуваються в залежності від розміщення точки опору.

**Висновок**

Під час виконання лабораторної роботи вивчав афінні перетворення. До них входять переміщення, повороту та масшабування. Кожен об’єкт можна редагувати як поодинці, так і окремо. Однак на мою думку, не зручно робити перетворення натискаючи по декілька раз на кнопки, краще реалізовувати контроль мишкою.