**Міністерство освіти і науки України**

**Національний університет «Львівська політехніка»**

**Інститут прикладної математики та фундаментальних наук**

**Кафедра прикладної математики**

**Звіт**

до лабораторної роботи №7

Варіант - 7

Виконала:

студентка групи

ПМ-32

Яворська Олена

Прийняв:

Строчик М. М.

Львів 2020

### Завдання:

### Створити проект для формування XML–документу з елементами SVG–графіки, функціонал якого надає можливості

### 1)додавання елементів та їх атрибутів у XML–файл з форми та збереження сформованого XML-файлу.

### 2)попередній перегляд SVG–документу на формі.

### 3)інтерактивність SVG–зображення на формі [та у браузері]∗.

### 4)перегляд згенерованого графічного зображення у браузері.

### Оскільки мій варіант у підгрупі сьомий, а усього варіантів завдань до лабораторної роботи 7- 6, я взяла завдання 1.

### Хід роботи:

### Створено форму з відповідними об’єктами.

### 

### 

### 2)Передбачено додавання елементів та їх атрибутів у XML–файл з форми та збереження сформованого XML-файлу:

### Оновлення прямокутника при зміні коду:

//при зміні xml на формі будується новий прямокутник

private void richTextBox1\_TextChanged(object sender, EventArgs e)

{

using (Graphics gr = pictureBox1.CreateGraphics())

{

var svgDoc = SvgDocument.FromSvg<SvgDocument>(richTextBox1.Text);

svg.RenderRectangle(gr, svgDoc);

}

### }

### Власне вивід xml в елемент форми richTextBox:

var xmlDoc = new XmlDocument

{

XmlResolver = null

};

xmlDoc.Load(filepath);

//виводимо xml код для перегляду/редагування

### richTextBox1.Text = xmlDoc.InnerXml;

### Збереження файлу:

//збереження xml з форми

private void button1\_Click(object sender, EventArgs e)

{

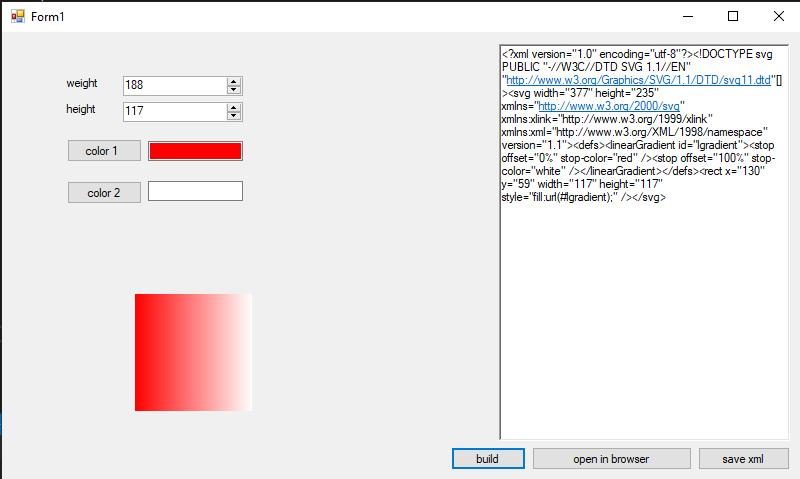
var svgDoc = SvgDocument.FromSvg<SvgDocument>(richTextBox1.Text);

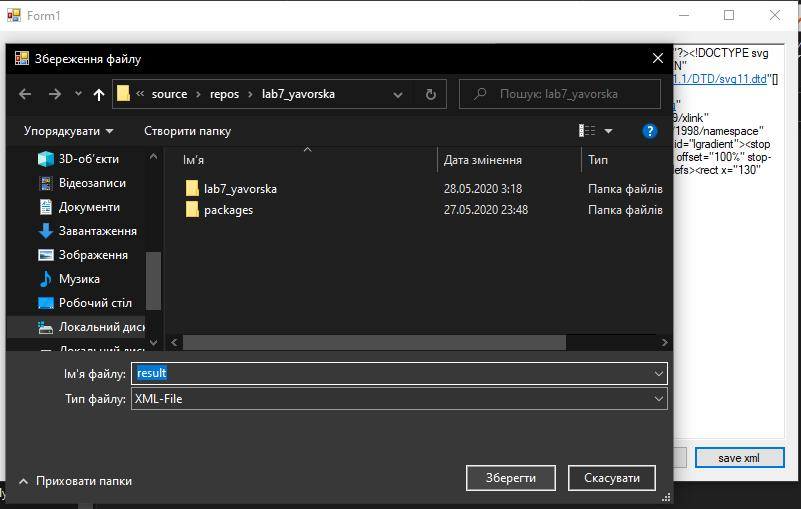
//задання формату для збереження, без цього у файлу формату не буде, просто назва

saveFileDialog1.Filter = "XML-File | \*.xml";

//з допомогою saveFileDialog зберігаємо,куди треба

if (saveFileDialog1.ShowDialog() == DialogResult.OK)

{



//svgDoc.Write(saveFileDialog1.FileName);

svgDoc.Write(saveFileDialog1.FileName);

}

### }

### 3) Передбачено попередній перегляд SVG–документу на формі:

var xmlDoc = new XmlDocument

{

XmlResolver = null

};

xmlDoc.Load(filepath);

//виводимо xml код для перегляду/редагування

### richTextBox1.Text = xmlDoc.InnerXml;

### 4) SVG–зображення на формі інтерактивне, ми можемо міняти розмір/колір:

### 

### 5) Перегляд згенерованого графічного зображення у браузері:

### 

//перегляд в браузері

private void in\_browser\_button\_Click(object sender, EventArgs e)

{

if (File.Exists(filepath))

{

System.Diagnostics.Process.Start(filepath);

}

### }

### Результуючий файл( код у додатку):

### 

**Висновок:** на цій лабораторній роботі я працювала з обробкою зображень у форматі SVG, реалізувала редагування і збереження xml-коду створеного прямокутника, перегляд зображення у браузері та на формі, роботу зі зміною розміру/кольору прямокутника, тобто інтерактивність зображення.

**Додаток:**

Program.cs:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace lab3\_yavO

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

Form1.cs:

using System;

using System.IO;

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;

using System.Xml;

using Svg;

using System.Xml.Linq;

namespace lab7\_yavorska

{

public partial class Form1 : Form

{

private Settings settings;

private readonly svg svg = new svg();

//де зберігаємо

string filepath = System.IO.Path.GetFileNameWithoutExtension(Application.ExecutablePath) + ".svg";

public Form1()

{

InitializeComponent();

settings = Settings.Empty(pictureBox1.Bounds.Size);

//максимальна/мінімальна висота/ширина, яку ми можемо задати

//якщо задати більші/менші значення, ширина/висота буде максимом/мінімумом відповідно

//тобто помилки не буде, буде значення в межах допустимого , найближче до введеного

weight.Maximum = pictureBox1.Width;

weight.Value = settings.Dimensions.Width;

height.Maximum = pictureBox1.Height;

height.Value = settings.Dimensions.Height;

//візуалізація кольорів градієнту

color1.BackColor = settings.ColorFrom;

color2.BackColor = settings.ColorTo;

}

private void label2\_Click(object sender, EventArgs e)

{

}

//ввід ширини нашого прямокутника

private void weight\_ValueChanged(object sender, EventArgs e)

{

settings.Dimensions.Width = Convert.ToInt32(weight.Value);

}

//ввід висоти нашого прямокутника

private void height\_ValueChanged(object sender, EventArgs e)

{

settings.Dimensions.Width = Convert.ToInt32(height.Value);

}

//зміна першого кольору градієнту

private void color\_1\_Click(object sender, EventArgs e)

{

if (colorDialog1.ShowDialog() != System.Windows.Forms.DialogResult.Cancel)

{

color1.BackColor = colorDialog1.Color;

settings.ColorFrom = colorDialog1.Color;

}

}

//зміна 2 кольору градієнту

private void color\_2\_Click(object sender, EventArgs e)

{

if (colorDialog2.ShowDialog() != System.Windows.Forms.DialogResult.Cancel)

{

color2.BackColor = colorDialog2.Color;

settings.ColorTo = colorDialog2.Color;

}

}

//власне побудова прямокутника і виведення xml

private void build\_button\_Click(object sender, EventArgs e)

{

//pictureBox1.BackColor

if (pictureBox1.Image != null)

{

//щоразу очищаємо пікчербокс перед побудовою прямок

pictureBox1.Image.Dispose();

pictureBox1.Image = null;

}

using (Graphics gr = pictureBox1.CreateGraphics())

{//функції з файлу/класу xml, власне будують на основі наших параметрів

var rectangleSvg = svg.RectangleToSVG(settings, pictureBox1.Width, pictureBox1.Height, filepath);

//richTextBox1.Text = rectangleSvg

svg.RenderRectangle(gr, rectangleSvg);

try

{

var xmlDoc = new XmlDocument

{

XmlResolver = null

};

xmlDoc.Load(filepath);

//виводимо xml код для перегляду/редагування

richTextBox1.Text = xmlDoc.InnerXml;

}

finally

{

richTextBox1.TextChanged += richTextBox1\_TextChanged;

}

}

}

//перегляд в браузері

private void in\_browser\_button\_Click(object sender, EventArgs e)

{

if (File.Exists(filepath))

{

System.Diagnostics.Process.Start(filepath);

}

}

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

{

}

//при зміні xml на формі будується новий прямокутник

private void richTextBox1\_TextChanged(object sender, EventArgs e)

{

using (Graphics gr = pictureBox1.CreateGraphics())

{

var svgDoc = SvgDocument.FromSvg<SvgDocument>(richTextBox1.Text);

svg.RenderRectangle(gr, svgDoc);

}

}

//збереження xml з форми

private void button1\_Click(object sender, EventArgs e)

{

var svgDoc = SvgDocument.FromSvg<SvgDocument>(richTextBox1.Text);

//задання формату для збереження, без цього у файлу формату не буде, просто назва

saveFileDialog1.Filter = "XML-File | \*.xml";

//з допомогою saveFileDialog зберігаємо,куди треба

if (saveFileDialog1.ShowDialog() == DialogResult.OK)

{

//svgDoc.Write(saveFileDialog1.FileName);

svgDoc.Write(saveFileDialog1.FileName);

}

}

}

}

Settings.cs:

using System.Drawing;

namespace lab7\_yavorska

{

class Settings

{

public Size Dimensions;

public Color ColorFrom;

public Color ColorTo;

public static Settings Empty(Size pict)

{

return new Settings

{//дефолтні значення кольору і розміру картинки

Dimensions = new Size(pict.Width / 2, pict.Height / 2),

ColorFrom = Color.Red,

ColorTo = Color.White

};

}

}

}

Svg.cs:

using System.Drawing;

using System.IO;

using Svg;

namespace lab7\_yavorska

{

class svg

{

//починаємо рисувавати,як відомо, з лівого верхнього кута, тому нам треба його координати

private Point getTopLeft(Settings settings, int width, int height)

{

return new Point(

(width - settings.Dimensions.Width) / 2,

(height - settings.Dimensions.Height) / 2

);

}

public SvgDocument RectangleToSVG(Settings settings, int width, int height, string filepath)

{

SvgDocument svg = new SvgDocument();

svg.Width = width;

svg.Height = height;

SvgLinearGradientServer gradientFill = new SvgLinearGradientServer

{

ID = "lgradient"

};

SvgGradientStop from = new SvgGradientStop();

from.Offset = new SvgUnit(SvgUnitType.Percentage, 0.0f);

from.StopColor = new SvgColourServer(settings.ColorFrom);

SvgGradientStop to = new SvgGradientStop();

to.Offset = new SvgUnit(SvgUnitType.Percentage, 100.0f);

to.StopColor = new SvgColourServer(settings.ColorTo);

gradientFill.Children.Add(from);

gradientFill.Children.Add(to);

//побудова прямокутника(з верхнього лівого кута) і його заливка

SvgRectangle rectangle = new SvgRectangle();

rectangle.SetRectangle(new RectangleF(getTopLeft(settings, width, height), settings.Dimensions));

rectangle.Fill = gradientFill;

SvgDefinitionList defs = new SvgDefinitionList();

defs.Children.Add(gradientFill);

svg.Children.Add(defs);

svg.Children.Add(rectangle);

//збереження прямокутника

File.WriteAllText(filepath, svg.GetXML());

return svg;

}

public void RenderRectangle(Graphics gr, SvgDocument document)

{

using (ISvgRenderer renderer = SvgRenderer.FromGraphics(gr))

{

document.RenderElement(renderer);

}

}

}

}

Result.xml:

<?xml version="1.0" encoding="utf-8"?>

<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">

<svg width="377" height="235" xml="http://www.w3.org/XML/1998/namespace" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xml="http://www.w3.org/XML/1998/namespace" version="1.1">

<defs>

<linearGradient id="lgradient">

<stop offset="0%" stop-color="red" />

<stop offset="100%" stop-color="white" />

</linearGradient>

</defs>

<rect x="130" y="59" width="117" height="117" style="fill:url(#lgradient);" /></svg>