Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №1-2

По дисциплине: «ОСиСП»

Выполнил:

Студент 3 курса

Группы ПО-4(1)

Елисеев С. Г.

Проверила:

Дряпко А.В.

Брест 2021

Лабораторная работа № 1-2

Вариант 8:

Цель: приобрести практические навыки проектирования и разработки приложений с графическим пользовательским интерфейсом

Задание: Простейший графический редактор с функциями рисования карандашом, стёркой. Дополнительные требования к инструменту: возможность изменения цвета и толщины карандаша и стёрки. Предусмотреть возможность сохранения полученного изображения.

Код:

**Mainwindow.xaml.cs**

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using System.Globalization;

using Microsoft.Win32;

namespace TRIPS\_2

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void DrawingModeSelection(object sender, RoutedEventArgs e)

{

Ink.EditingMode = InkCanvasEditingMode.Ink;

}

private void DeletingModeSelection(object sender, RoutedEventArgs e)

{

Ink.EditingMode = InkCanvasEditingMode.EraseByStroke;

}

private void FormatingModeSelection(object sender, RoutedEventArgs e)

{

Ink.EditingMode = InkCanvasEditingMode.EraseByPoint;

}

private void ColorsList\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

ComboBoxItem selectedItem = (ComboBoxItem)(ColorsList.SelectedValue);

string value = (string)(selectedItem.Content);

Ink.DefaultDrawingAttributes.Color = (Color)ColorConverter.ConvertFromString(value);

}

private void BrushWidthChange(object sender, RoutedPropertyChangedEventArgs<double> e)

{

Ink.DefaultDrawingAttributes.Width = ChooseWidth.Value;

Ink.DefaultDrawingAttributes.Height = ChooseWidth.Value;

}

private void CloseWindow\_Click(object sender, RoutedEventArgs e)

{

Close();

}

private void SaveControlImage(

Visual baseElement, int imageWidth, int imageHeight, string pathToOutputFile)

{

// 1) get current dpi

var pSource = PresentationSource.FromVisual(Application.Current.MainWindow);

Matrix m = pSource.CompositionTarget.TransformToDevice;

double dpiX = m.M11 \* 96;

double dpiY = m.M22 \* 96;

// 2) create RenderTargetBitmap

var elementBitmap = new RenderTargetBitmap(imageWidth, imageHeight, dpiX, dpiY, PixelFormats.Default);

// 3) undo element transformation

var drawingVisual = new DrawingVisual();

using (DrawingContext drawingContext = drawingVisual.RenderOpen())

{

var visualBrush = new VisualBrush(baseElement);

drawingContext.DrawRectangle(

visualBrush,

null,

new Rect(new Point(0, 0), new Size(imageWidth / m.M11, imageHeight / m.M22)));

}

// 4) draw element

elementBitmap.Render(drawingVisual);

// 5) create PNG image

var encoder = new PngBitmapEncoder();

encoder.Frames.Add(BitmapFrame.Create(elementBitmap));

// 6) save image to file

using (var imageFile = new FileStream(pathToOutputFile, FileMode.Create, FileAccess.Write))

{

encoder.Save(imageFile);

imageFile.Flush();

imageFile.Close();

}

}

private string GetOutputFileName()

{

var dlgSave = new SaveFileDialog

{

DefaultExt = "png",

Title = "Choose output file",

Filter = "PNG Files|\*.png"

};

if (dlgSave.ShowDialog() == true)

return dlgSave.FileName;

return string.Empty;

}

private void Btn\_Save\_Click(object sender, RoutedEventArgs e)

{

// get output file name

string fileName = this.GetOutputFileName();

if (fileName == string.Empty)

return;

// select visual element

Visual element = null;

var elementWidth = 0;

var elementHeight = 0;

element = this.Ink;

elementWidth = (int)this.Ink.ActualWidth;

elementHeight = (int)this.Ink.ActualHeight;

// save selected element to file

this.SaveControlImage(element, elementWidth, elementHeight, fileName);

}

}

}

**MainWindow.xaml**

<Grid Name="baseGrid">

<InkCanvas Name="Ink" >

<InkCanvas.DefaultDrawingAttributes>

<DrawingAttributes/>

</InkCanvas.DefaultDrawingAttributes>

</InkCanvas>

<Grid Name="TopPanel" VerticalAlignment="Top" HorizontalAlignment="Left" Width="800" Height="99" Background="Gainsboro" Margin="0,0,-6.4,0">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="423\*"/>

<ColumnDefinition Width="377\*"/>

</Grid.ColumnDefinitions>

<TextBlock FontFamily="Arial" FontSize="18" Margin="20, 10, 0, 0" Grid.ColumnSpan="2"><Run Text="Choose color:"/></TextBlock>

<ComboBox FontFamily="Arial" FontSize="16" Name="ColorsList" Width="200" Height="30" HorizontalAlignment="Left" Margin="20,32,0,18" SelectionChanged="ColorsList\_SelectionChanged">

<ComboBoxItem Content="Black" FontFamily="Arial" FontSize="16"/>

<ComboBoxItem Content="Red" FontFamily="Arial" FontSize="16"/>

<ComboBoxItem Content="Orange" FontFamily="Arial" FontSize="16"/>

<ComboBoxItem Content="Yellow" FontFamily="Arial" FontSize="16"/>

<ComboBoxItem Content="Green" FontFamily="Arial" FontSize="16"/>

<ComboBoxItem Content="Cyan" FontFamily="Arial" FontSize="16"/>

<ComboBoxItem Content="Blue" FontFamily="Arial" FontSize="16"/>

<ComboBoxItem Content="Purple" FontFamily="Arial" FontSize="16"/>

</ComboBox>

<TextBlock Text="Choose mode:" FontFamily="Arial" FontSize="18" Margin="280,10,0,0" Grid.ColumnSpan="2"/>

<RadioButton Name="DrawingMode" Checked="DrawingModeSelection" Margin="280, 30, 0, 0" FontFamily="Arial" FontSize="12" Content="кисть&#xD;&#xA;" Grid.ColumnSpan="2">

</RadioButton>

<RadioButton Name="DeletingMode" Checked="DeletingModeSelection" Margin="280, 45, 0, 0" FontFamily="Arial" FontSize="12" Content="удалить объект&#xD;&#xA;" Grid.ColumnSpan="2">

</RadioButton>

<RadioButton Name="FormatingMode" Checked="FormatingModeSelection" Margin="280, 60, 0, 0" FontFamily="Arial" FontSize="12" Content="ластик" Grid.ColumnSpan="2">

</RadioButton>

<TextBlock FontFamily="Arial" FontSize="18" Margin="34.6,10,0,0" Grid.Column="1"><Run Text="Choose width:"/><InlineUIContainer>

</InlineUIContainer></TextBlock>

<Button x:Name="Btn\_Save" Content="Сохранить" Click="Btn\_Save\_Click" HorizontalAlignment="Right" VerticalAlignment="Bottom" Width="71" Height="35" Margin="0,0,10,0" Grid.Column="1"/>

<Slider Name="ChooseWidth" Maximum="100" Value="1" TickPlacement="Both" TickFrequency="10" Margin="76.6,35,100,0" Width="200" Foreground="Black" ValueChanged="BrushWidthChange" Grid.Column="1"/>

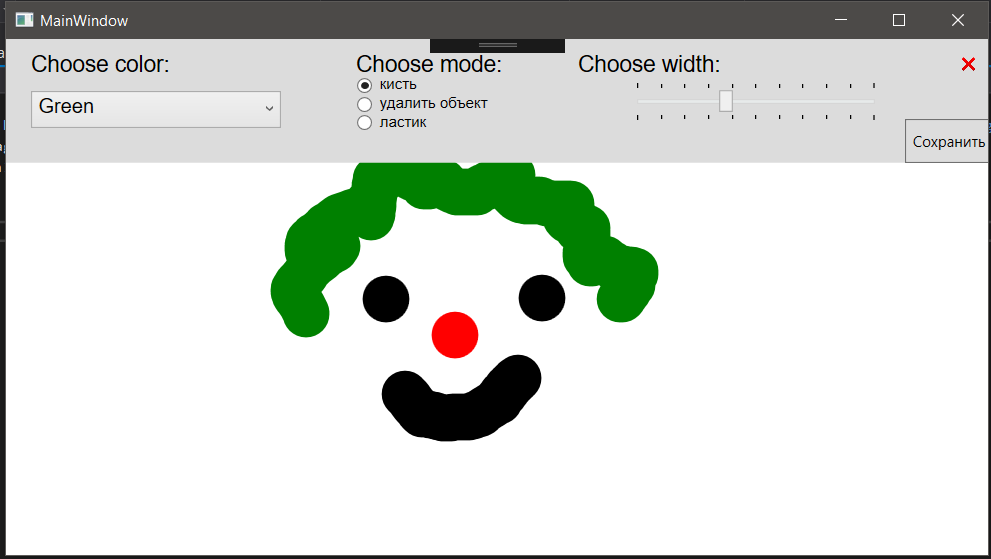
<Button Name="CloseWindow" Width="40" Height="40" BorderThickness="0" VerticalAlignment="Top" HorizontalAlignment="Right" Background="Gainsboro" Click="CloseWindow\_Click" Grid.Column="1" Margin="0,0,10,0">

<Image Source="https://emojio.ru/images/apple-b/274c.png" Height="11" Width="20"></Image>

</Button>

</Grid>

</Grid>

Результат:  


Сохранение:

