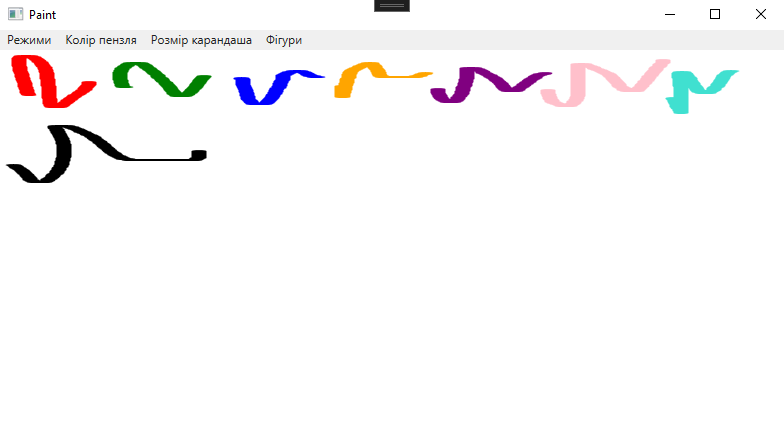
ЛАБОРАТОРНА РОБОТА №6

ТЕМА: ТРАНСФОРМАЦІЇ ТА ЕФЕКТИ У WPF



| MainWindow.xaml |
| --- |
| <Window x:Class="WpfApp5.MainWindow"  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"  xmlns:local="clr-namespace:WpfApp5"  mc:Ignorable="d"  Title="Paint" Height="450" Width="800">  <Grid>  <Grid.RowDefinitions>  <RowDefinition Height="20"/>  <RowDefinition Height="\*"/>  </Grid.RowDefinitions>  <Menu Name="mainMenu" Grid.Row="0">  <MenuItem Header="Режими">  <MenuItem Name="DrawModeMenuItem" Header="Малювання" Click="MenuItem\_Checked"/>  <MenuItem Name="EditModeMenuItem" Header="Редагування" Click="MenuItem\_Checked"/>  <MenuItem Name="DeleteModeMenuItem" Header="Видалення" Click="MenuItem\_Checked"/>  </MenuItem>  <MenuItem Header="Колір пензля">  <MenuItem Header="Червоний" Click="MenuItemColor\_Click"/>  <MenuItem Header="Зелений" Click="MenuItemColor\_Click"/>  <MenuItem Header="Синій" Click="MenuItemColor\_Click"/>  <MenuItem Header="Помаранчевий" Click="MenuItemColor\_Click"/>  <MenuItem Header="Фіолетовий" Click="MenuItemColor\_Click"/>  <MenuItem Header="Рожевий" Click="MenuItemColor\_Click"/>  <MenuItem Header="Бірюзовий" Click="MenuItemColor\_Click"/>  <MenuItem Header="Чорний" Click="MenuItemColor\_Click"/>  </MenuItem>  <MenuItem Header="Розмір карандаша">  <MenuItem Header="Normal" Click="MenuItemPenSize\_Click"/>  <MenuItem Header="5" Click="MenuItemPenSize\_Click"/>  <MenuItem Header="6" Click="MenuItemPenSize\_Click"/>  <MenuItem Header="7" Click="MenuItemPenSize\_Click"/>  <MenuItem Header="8" Click="MenuItemPenSize\_Click"/>  <MenuItem Header="9" Click="MenuItemPenSize\_Click"/>  <MenuItem Header="10" Click="MenuItemPenSize\_Click"/>  <MenuItem Header="11" Click="MenuItemPenSize\_Click"/>  <MenuItem Header="12" Click="MenuItemPenSize\_Click"/>  <MenuItem Header="13" Click="MenuItemPenSize\_Click"/>  <MenuItem Header="14" Click="MenuItemPenSize\_Click"/>  <MenuItem Header="15" Click="MenuItemPenSize\_Click"/>  </MenuItem>  <MenuItem Header="Фігури">  <MenuItem Name="LineMenuItem" Header="Лінія" IsCheckable="True" Click="MenuItemShape\_Click"/>  <MenuItem Name="CircleMenuItem" Header="Коло" IsCheckable="True" Click="MenuItemShape\_Click"/>  <MenuItem Name="RectangleMenuItem" Header="Прямокутник" IsCheckable="True" Click="MenuItemShape\_Click"/>  </MenuItem>  </Menu>  <InkCanvas Name="inkCanvas" Grid.Row="1" Background="White" MouseLeftButtonDown="InkCanvas\_MouseLeftButtonDown"/>  </Grid>  </Window> |

| MainWindow.xaml.cs |
| --- |
| using System;  using System.Collections.Generic;  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;  namespace WpfApp5  {  /// <summary>  /// Interaction logic for MainWindow.xaml  /// </summary>  public partial class MainWindow : Window  {  private Color selectedColor = Colors.Black;  private enum ShapeType { Line, Circle, Rectangle }  private ShapeType currentShape = ShapeType.Line;  public MainWindow()  {  InitializeComponent();  inkCanvas.DefaultDrawingAttributes.Color = Colors.Black;  inkCanvas.DefaultDrawingAttributes.Width = 2;  }  public void MenuItem\_Checked(object sender, RoutedEventArgs e)  {  MenuItem menuItem = (MenuItem)sender;  string mode = menuItem.Header.ToString();  switch (mode)  {  case "Малювання":  inkCanvas.EditingMode = InkCanvasEditingMode.Ink;  break;  case "Редагування":  inkCanvas.EditingMode = InkCanvasEditingMode.Select;  break;  case "Видалення":  inkCanvas.EditingMode = InkCanvasEditingMode.EraseByPoint;  break;  }  }  public void MenuItemPenSize\_Click(object sender, RoutedEventArgs e)  {  MenuItem menuItem = (MenuItem)sender;  string size = menuItem.Header.ToString();  switch (size)  {  case "Normal":  inkCanvas.DefaultDrawingAttributes.Width = 2;  break;  case "5":  inkCanvas.DefaultDrawingAttributes.Width = 5;  break;  case "6":  inkCanvas.DefaultDrawingAttributes.Width = 6;  break;  case "7":  inkCanvas.DefaultDrawingAttributes.Width = 7;  break;  case "8":  inkCanvas.DefaultDrawingAttributes.Width = 8;  break;  case "9":  inkCanvas.DefaultDrawingAttributes.Width = 9;  break;  case "10":  inkCanvas.DefaultDrawingAttributes.Width = 10;  break;  case "11":  inkCanvas.DefaultDrawingAttributes.Width = 11;  break;  case "12":  inkCanvas.DefaultDrawingAttributes.Width = 12;  break;  case "13":  inkCanvas.DefaultDrawingAttributes.Width = 13;  break;  case "14":  inkCanvas.DefaultDrawingAttributes.Width = 14;  break;  case "15":  inkCanvas.DefaultDrawingAttributes.Width = 15;  break;  }  }  public void MenuItemColor\_Click(object sender, RoutedEventArgs e)  {  MenuItem menuItem = (MenuItem)sender;  string color = menuItem.Header.ToString();  switch (color)  {  case "Червоний":  inkCanvas.DefaultDrawingAttributes.Color = Colors.Red;  break;  case "Зелений":  inkCanvas.DefaultDrawingAttributes.Color = Colors.Green;  break;  case "Синій":  inkCanvas.DefaultDrawingAttributes.Color = Colors.Blue;  break;  case "Помаранчевий":  inkCanvas.DefaultDrawingAttributes.Color = Colors.Orange;  break;  case "Фіолетовий":  inkCanvas.DefaultDrawingAttributes.Color = Colors.Purple;  break;  case "Рожевий":  inkCanvas.DefaultDrawingAttributes.Color = Colors.Pink;  break;  case "Бірюзовий":  inkCanvas.DefaultDrawingAttributes.Color = Colors.Turquoise;  break;  case "Чорний":  inkCanvas.DefaultDrawingAttributes.Color = Colors.Black;  break;  }  }  private void MenuItemShape\_Click(object sender, RoutedEventArgs e)  {  MenuItem shapeMenuItem = (MenuItem)sender;  foreach (MenuItem item in mainMenu.Items.OfType<MenuItem>().Where(x => x.Name != "DrawModeMenuItem" && x.Name != "EditModeMenuItem" && x.Name != "DeleteModeMenuItem"))  {  item.IsEnabled = false;  }  shapeMenuItem.IsEnabled = true;  if (shapeMenuItem == LineMenuItem)  {  currentShape = ShapeType.Line;  }  else if (shapeMenuItem == CircleMenuItem)  {  currentShape = ShapeType.Circle;  }  else if (shapeMenuItem == RectangleMenuItem)  {  currentShape = ShapeType.Rectangle;  }  }  public void InkCanvas\_MouseLeftButtonDown(object sender, MouseButtonEventArgs e)  {  Point position = e.GetPosition(inkCanvas);  if (currentShape == ShapeType.Line)  {  Line line = new Line  {  X1 = position.X,  Y1 = position.Y,  X2 = position.X + 30,  Y2 = position.Y + 30,  Stroke = new SolidColorBrush(selectedColor),  StrokeThickness = 2  };  inkCanvas.Children.Add(line);  }  else if (currentShape == ShapeType.Circle)  {  Ellipse circle = new Ellipse  {  Width = 30,  Height = 30,  Fill = new SolidColorBrush(selectedColor)  };  InkCanvas.SetLeft(circle, position.X);  InkCanvas.SetTop(circle, position.Y);  inkCanvas.Children.Add(circle);  }  else if (currentShape == ShapeType.Rectangle)  {  Rectangle rectangle = new Rectangle  {  Width = 30,  Height = 30,  Fill = new SolidColorBrush(selectedColor)  };  InkCanvas.SetLeft(rectangle, position.X);  InkCanvas.SetTop(rectangle, position.Y);  inkCanvas.Children.Add(rectangle);  }  }  }  } |