УО «Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №3

По дисциплине: “Технологии разработки интерфейса программных средств„

Тема: Основные элементы управления WPF.

Вариант 6

Выполнил: Комиссаров А. Е.

Брест 2021 Группа ПО-7, 2 курс.

Проверила: Войцехович О.Ю.

Цель: Ознакомиться с основными элементами управления WPF. Получить знания о обработчиках событий языка WPF.

Ход работы:

**Задание 1.**  Разработать WPF-приложение с меню, панелью инструментов и строкой состояния. С помощью пунктов меню пользователь может изменять цвет фона окна, получить информацию о разработчике, а также закрыть окно. Кнопки панели инструментов дублируют команды меню. При наведении на пункты меню или кнопки панели инструментов в строке состояния отображается информация об этих элементах управления.

**Код программы:**

**Содержание файла MainWindow.xaml:**

<Window x:Class="WpfApp2.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:WpfApp2"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="600" ResizeMode="NoResize">

<StackPanel Name="mainpanel">

<Menu Background="White" Width="600">

<MenuItem Header="\_Цвет фона окна" ToolTip="Выбор цвета заднего фона окна.">

<RadioButton GroupName="Color\_Bg" Content="\_Белый" Click="color\_wh" />

<RadioButton GroupName="Color\_Bg" Content="\_Красный" Click="color\_rd"/>

<RadioButton GroupName="Color\_Bg" Content="\_Желтый" Click="color\_yl"/>

<RadioButton GroupName="Color\_Bg" Content="\_Зеленый" Click="color\_gr"/>

</MenuItem>

<MenuItem Header="\_О разработчике" Click="about\_dev" ToolTip="Информация о разработчике"/>

<MenuItem Header="\_Закрыть" Click="close\_window" ToolTip="Закрытие окна"/>

</Menu>

<ToolBar Height="40" Width="600" Background="White" Margin="0 0 0 0">

<Button Click="color\_wh" ToolTip="Белый">

<Image Source="https://www.pinclipart.com/picdir/big/182-1823226\_change-the-png-icon-change-color-icon-png.png"></Image>

</Button>

<Separator/>

<Button Click="color\_rd" ToolTip="Красный">

<Image Source="https://www.pinclipart.com/picdir/big/182-1823226\_change-the-png-icon-change-color-icon-png.png"></Image>

</Button>

<Separator/>

<Button Click="color\_yl" ToolTip="Желтый">

<Image Source="https://www.pinclipart.com/picdir/big/182-1823226\_change-the-png-icon-change-color-icon-png.png"></Image>

</Button>

<Separator/>

<Button Click="color\_gr" ToolTip="Зеленый">

<Image Source="https://www.pinclipart.com/picdir/big/182-1823226\_change-the-png-icon-change-color-icon-png.png"></Image>

</Button>

<Separator/>

<Button Click="about\_dev" ToolTip="Информация о разработчике">

<Image Source="https://icon-library.com/images/about-icon/about-icon-23.jpg"></Image>

</Button>

<Separator/>

<Button Click="close\_window" ToolTip="Закрыть окно">

<Image Source="https://cdn.iconscout.com/icon/free/png-256/exit-93-458494.png"></Image>

</Button>

</ToolBar>

<StatusBar Margin="0 330 0 0" Height="25">

<TextBlock Text="Строка состояния ..." />

<StatusBarItem HorizontalAlignment="Right" >

<TextBlock Text="Строка состояния: ОК" />

</StatusBarItem>

</StatusBar>

</StackPanel>

</Window>

**Содержание файла MainWindow.xaml.cs:**

namespace WpfApp2

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void color\_wh(object sender, RoutedEventArgs e)

{

mainpanel.Background = Brushes.White;

}

private void color\_rd(object sender, RoutedEventArgs e)

{

mainpanel.Background = Brushes.Red;

}

private void color\_yl(object sender, RoutedEventArgs e)

{

mainpanel.Background = Brushes.Yellow;

}

private void color\_gr(object sender, RoutedEventArgs e)

{

mainpanel.Background = Brushes.Green;

}

private void about\_dev(object sender, RoutedEventArgs e)

{

MessageBox.Show("Комиссаров, ПО-7, 2 курс.");

}

private void close\_window(object sender, RoutedEventArgs e)

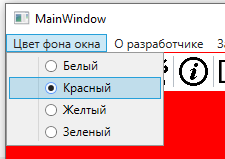
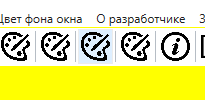
{

Close();

}}}

**Результат выполнения программы:**

Рис. 1 – Окно программы при запуске (Белый фон)



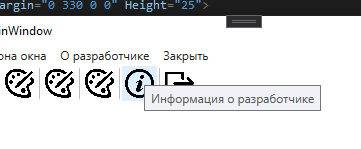
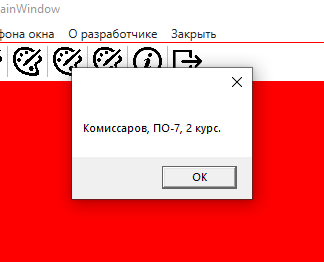
 Рис. 2, 3 – Нажатие на кнопки выбора цвета из меню и из панели инструментов.

Рис. 3, 4 – Наведение курсора мыши на кнопку, появление всплывающей подсказки, а также нажатие на кнопку «О разработчике».

**Задание 2.**  Разработать WPF-приложение «Графический редактор» с выпадающим списком для выбора цвета кисти, ползунком для выбора размеров кисти и зависимыми переключателями для выбора режима работы: «рисование», «редактирование», «удаление».

**Код программы:**

**Содержимое файла MainWindow.xaml:**

<Window x:Class="WpfApp2.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:WpfApp2"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="600" ResizeMode="NoResize">

<StackPanel>

<Menu Background="White" Width="600">

<MenuItem Header="\_Цвет кисти" ToolTip="Выбор цвета.">

<RadioButton GroupName="Color\_Bg" Content="\_Белый" Click="Color\_wh" />

<Separator/>

<RadioButton GroupName="Color\_Bg" Content="\_Красный" Click="Color\_rd"/>

<RadioButton GroupName="Color\_Bg" Content="\_Желтый" Click="Color\_yl"/>

<RadioButton GroupName="Color\_Bg" Content="\_Зеленый" Click="Color\_gr"/>

</MenuItem>

<MenuItem Header="\_Толщина кисти">

<Slider Name="Slider1" Height="25" Width="100" Minimum="1" Maximum="15" Value="2" MouseLeave="Thickness"/>

</MenuItem>

<MenuItem Header="\_Режим работы">

<RadioButton GroupName="mode" Content="\_Редактирование" Click="Mode\_edit"/>

<RadioButton GroupName="mode" Content="\_Рисование" Click="Mode\_draw"/>

<RadioButton GroupName="mode" Content="\_Удаление" Click="Mode\_erase"/>

</MenuItem>

</Menu>

<InkCanvas Height="420" Name="drawboard">

<InkCanvas.DefaultDrawingAttributes>

<DrawingAttributes Color="Red" Height="2" Width="2"/>

</InkCanvas.DefaultDrawingAttributes>

</InkCanvas>

</StackPanel>

</Window>

**Содержимое файла MainWindow.xaml.cs:**

namespace WpfApp2

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void Color\_wh(object sender, RoutedEventArgs e)

{

drawboard.DefaultDrawingAttributes.Color = Colors.White;

}

private void Color\_rd(object sender, RoutedEventArgs e)

{

drawboard.DefaultDrawingAttributes.Color = Colors.Red;

}

private void Color\_yl(object sender, RoutedEventArgs e)

{

drawboard.DefaultDrawingAttributes.Color = Colors.Yellow;

}

private void Color\_gr(object sender, RoutedEventArgs e)

{

drawboard.DefaultDrawingAttributes.Color = Colors.Green;

}

private void Mode\_erase(object sender, RoutedEventArgs e)

{

drawboard.EditingMode = InkCanvasEditingMode.EraseByPoint;

}

private void Mode\_draw(object sender, RoutedEventArgs e)

{

drawboard.EditingMode = InkCanvasEditingMode.Ink;

}

private void Mode\_edit(object sender, RoutedEventArgs e)

{

drawboard.EditingMode = InkCanvasEditingMode.Select;

}

private void Thickness(object sender, RoutedEventArgs e)

{

drawboard.DefaultDrawingAttributes.Width = Slider1.Value;

drawboard.DefaultDrawingAttributes.Height = Slider1.Value;

}

}

}

**Результат выполнения программы:**

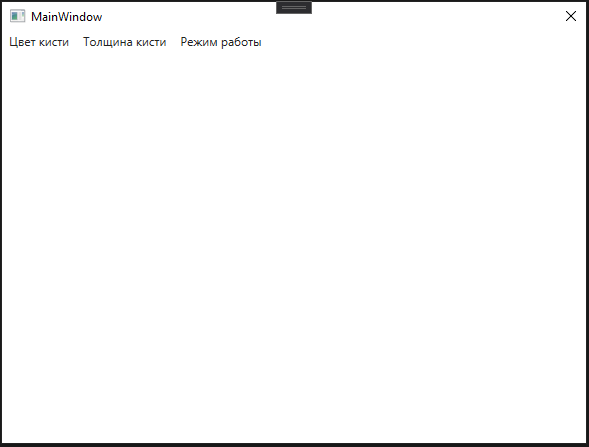


Рис. 5 – Запуск приложения «Графический редактор»

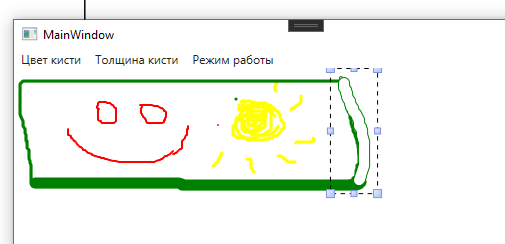
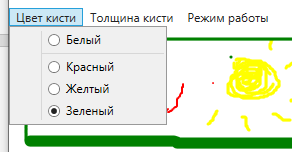
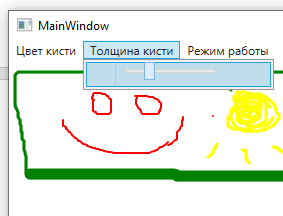
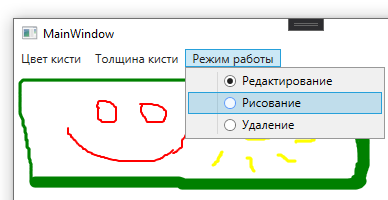


Рис. 6 – В пределах окна можно рисовать, выбирать цвет кисти, толщину кисти, а также менять режимы работы с приложением (Редактирование, Удаление, Рисование).

 а) б)

в)

Рис. 7 – а) Меню выбора цвета кисти; б) меню выбора толщины кисти; в) меню выбора режима работы с приложением.

Вывод: Я ознакомился с основными элементами управления WPF. Получил знания о обработчиках событий языка WPF.