МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Казанский национальный исследовательский технический университет

им. А.Н. Туполева – КАИ»

Институт компьютерных технологий и защиты информации

Отделение СПО ИКТЗИ (Колледж информационных технологий)

ЛАБОРАТОРНАЯ РАБОТА №7

по дисциплине

Разработка программных модулей

Тема: «Анимация в WPF-приложениях»

Работу выполнил

Студент гр.4335

Желваков А. С.

Принял

Преподаватель Лоповок Е.Е.

Казань 2024

**Задание 1**

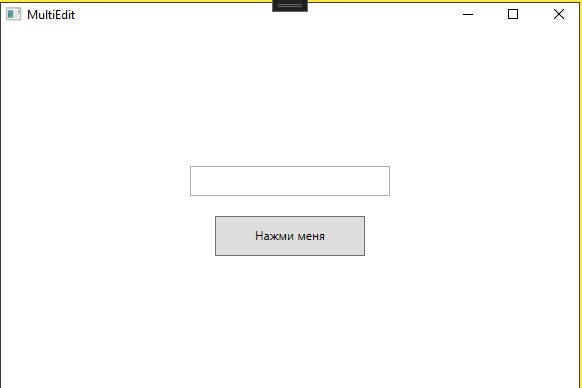


Рис. 1.1 - Начальное состояние экрана.

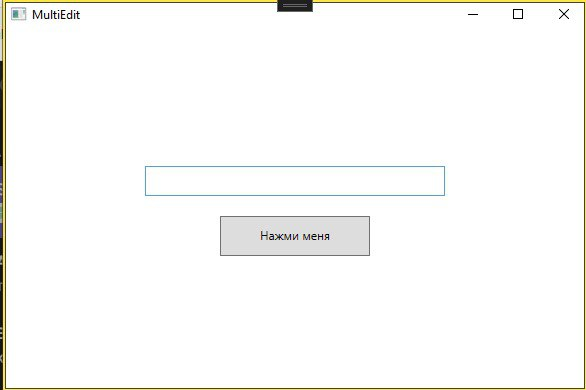


Рис. 1.2 - Состояние текстового поля после клика.

При клике на текстовое поле оно плавно расширяется.

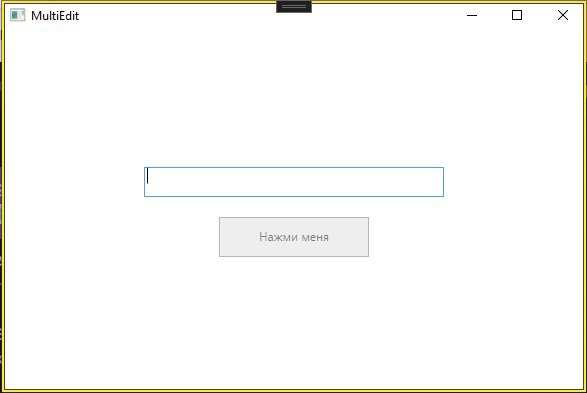


Рис. 1.3 - Состояние кнопки после наведения курсора.

Кнопка плавно становится прозрачнее после наведения курсора.

**Задание 2**

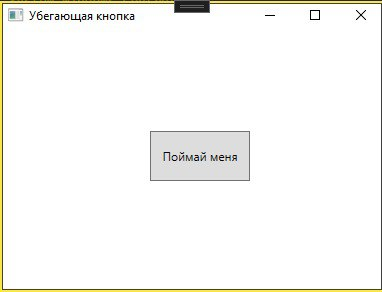


Рис. 2.1 - Начальное состояние экрана.

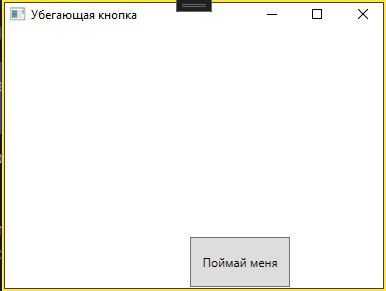


Рис. 2.2 - Кнопка после наведения.

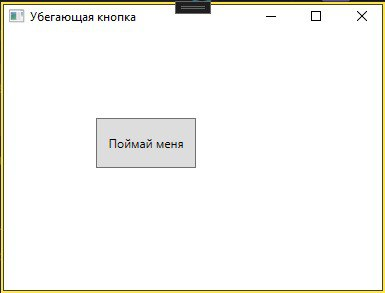


Рис. 2.3 - Кнопка после последующего наведения.

Листинг.

Задание 1

<Window x:Class="MultiEdit.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="MultiEdit" Height="400" Width="600">

<Window.Resources>

<Storyboard x:Key="ButtonMouseEnterAnimation">

<DoubleAnimation Storyboard.TargetProperty="Opacity"

To="0.5"

Duration="0:0:0.3"

AutoReverse="True"/>

</Storyboard>

<Storyboard x:Key="TextBoxFocusAnimation">

<DoubleAnimation Storyboard.TargetProperty="Width"

To="300"

Duration="0:0:0.2"/>

</Storyboard>

<Storyboard x:Key="TextBoxLostFocusAnimation">

<DoubleAnimation Storyboard.TargetProperty="Width"

To="200"

Duration="0:0:0.2"/>

</Storyboard>

</Window.Resources>

<StackPanel Orientation="Vertical" HorizontalAlignment="Center" VerticalAlignment="Center">

<TextBox Width="200" Height="30" Margin="10"

GotFocus="TextBox\_GotFocus"

LostFocus="TextBox\_LostFocus"/>

<Button Content="Нажми меня" Width="150" Height="40" Margin="10">

<Button.Triggers>

<EventTrigger RoutedEvent="Button.MouseEnter">

<BeginStoryboard Storyboard="{StaticResource ButtonMouseEnterAnimation}"/>

</EventTrigger>

</Button.Triggers>

</Button>

</StackPanel>

</Window>

using System.Windows;

using System.Windows.Controls;

using System.Windows.Media.Animation;

namespace MultiEdit

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void TextBox\_GotFocus(object sender, RoutedEventArgs e)

{

Storyboard focusAnimation = (Storyboard)FindResource("TextBoxFocusAnimation");

Storyboard.SetTarget(focusAnimation, (TextBox)sender);

focusAnimation.Begin();

}

private void TextBox\_LostFocus(object sender, RoutedEventArgs e)

{

Storyboard lostFocusAnimation = (Storyboard)FindResource("TextBoxLostFocusAnimation");

Storyboard.SetTarget(lostFocusAnimation, (TextBox)sender);

lostFocusAnimation.Begin();

}

}

}

Задание 2

<Window x:Class="RunawayButtonApp.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="Убегающая кнопка" Height="300" Width="400">

<Canvas x:Name="MainCanvas">

<Button x:Name="RunawayButton" Content="Поймай меня" Width="100" Height="50"

Canvas.Left="150" Canvas.Top="100"

MouseEnter="RunawayButton\_MouseEnter"/>

</Canvas>

</Window>

using System;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Input;

namespace RunawayButtonApp

{

public partial class MainWindow : Window

{

private Random random = new Random();

public MainWindow()

{

InitializeComponent();

}

private void RunawayButton\_MouseEnter(object sender, MouseEventArgs e)

{

double windowWidth = MainCanvas.ActualWidth;

double windowHeight = MainCanvas.ActualHeight;

double buttonWidth = RunawayButton.ActualWidth;

double buttonHeight = RunawayButton.ActualHeight;

double newLeft = random.Next(0, (int)(windowWidth - buttonWidth));

double newTop = random.Next(0, (int)(windowHeight - buttonHeight));

Canvas.SetLeft(RunawayButton, newLeft);

Canvas.SetTop(RunawayButton, newTop);

}

}

}