Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №7

за 3 семестр

По дисциплине: «ТРИПС»

Тема: «Анимация в WPF-приложениях»

Выполнила:

Студент 2 курса

Группы ПО-6(1)

Мартынович Д. М.

Проверил:

Войцехович О.Ю.

2021

|  |
| --- |
| Задание 1  Добавьте в WPF-приложение MultiEdit, разработанное в лабораторной работе №5, эффекты анимации. |

<Window x:Class="Lab\_6\_TRIPS.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:Lab\_6\_TRIPS"

mc:Ignorable="d"

Title="MultiEdit" Height="450" Width="800">

<Window.Resources>

<Style TargetType="Button" x:Key="property">

<Style.Triggers>

<Trigger Property="IsMouseOver" Value="True">

<Trigger.Setters>

<Setter Property="Foreground" Value="Red" />

<Setter Property="FontSize" Value="14" />

</Trigger.Setters>

</Trigger>

</Style.Triggers>

</Style>

<Style TargetType="TextBox" x:Key="MainTextBlock">

<Setter Property="Width" Value="350"/>

<Setter Property="Height" Value="300"/>

<Setter Property="Control.Background">

<Setter.Value>

<LinearGradientBrush>

<LinearGradientBrush.GradientStops>

<GradientStop Color="White" Offset="1" />

<GradientStop Color="Red" Offset="0.5" />

<GradientStop Color="White" Offset="0" />

</LinearGradientBrush.GradientStops>

</LinearGradientBrush>

</Setter.Value>

</Setter>

<Style.Triggers>

<DataTrigger Binding="{Binding ElementName=ComoBox1, Path=SelectedIndex}" Value="0">

<DataTrigger.Setters>

<Setter Property="Foreground" Value="DarkBlue" />

<Setter Property="FontSize" Value="15"/>

</DataTrigger.Setters>

</DataTrigger>

<DataTrigger Binding="{Binding ElementName=ComoBox1, Path=SelectedIndex}" Value="1">

<DataTrigger.Setters>

<Setter Property="Foreground" Value="Yellow" />

<Setter Property="FontSize" Value="25"/>

</DataTrigger.Setters>

</DataTrigger>

<DataTrigger Binding="{Binding ElementName=ComoBox1, Path=SelectedIndex}" Value="2">

<DataTrigger.Setters>

<Setter Property="Foreground" Value="Orange" />

<Setter Property="FontSize" Value="46"/>

</DataTrigger.Setters>

</DataTrigger>

</Style.Triggers>

</Style>

<Style BasedOn="{StaticResource property}" TargetType="Button" x:Key="CloseButton">

<Style.Triggers>

<MultiDataTrigger>

<MultiDataTrigger.Conditions>

<Condition Binding="{Binding ElementName=textbox1, Path=Text}" Value=""/>

<Condition Binding="{Binding ElementName=textbox2, Path=Text}" Value=""/>

</MultiDataTrigger.Conditions>

<MultiDataTrigger.Setters>

<Setter Property="IsEnabled" Value="True"/>

</MultiDataTrigger.Setters>

</MultiDataTrigger>

</Style.Triggers>

<Style.Setters>

<Setter Property="IsEnabled" Value="False"/>

</Style.Setters>

</Style>

</Window.Resources>

<Grid Background="WhiteSmoke">

<TextBox x:Name="textbox1" Style="{StaticResource MainTextBlock}" HorizontalAlignment="Left" Margin="37,94,0,0" TextWrapping="Wrap" Width="271" Height="255" VerticalAlignment="Top"/>

<TextBox x:Name="textbox2" Style="{StaticResource MainTextBlock}" HorizontalAlignment="Right" Margin="0,94,98,0" TextWrapping="Wrap" Width="271" Height="255" VerticalAlignment="Top"/>

<Button x:Name="OpenB" Style="{StaticResource property}" Content="Открыть" HorizontalAlignment="Left" Margin="37,34,0,0" VerticalAlignment="Top" Height="27" Width="79"/>

<Button x:Name="ClearB" Style="{StaticResource property}" Content="Очистить" HorizontalAlignment="Left" Margin="140,34,0,0" VerticalAlignment="Top" Height="27" Width="78" Click="Button\_Click\_1"/>

<Button x:Name="CloseB" Style="{StaticResource CloseButton}" Content="Закрыть" HorizontalAlignment="Left" Margin="249,34,0,0" VerticalAlignment="Top" Height="27" Width="79" Click="Button\_Click"/>

<ComboBox x:Name="ComoBox1" HorizontalAlignment="Left" Margin="357,39,0,0" VerticalAlignment="Top" Width="120" SelectedIndex="0">

<ComboBoxItem Content="1"/>

<ComboBoxItem Content="2"/>

<ComboBoxItem Content="3"/>

</ComboBox>

</Grid>

</Window>

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;

using System.Windows.Media.Animation;

namespace Lab\_6\_TRIPS

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

DoubleAnimation buttonOpen = new DoubleAnimation();

buttonOpen.From = OpenB.ActualWidth;

buttonOpen.To = 79;

buttonOpen.Duration = TimeSpan.FromSeconds(3);

OpenB.BeginAnimation(Button.WidthProperty, buttonOpen);

DoubleAnimation buttonClear = new DoubleAnimation();

buttonClear.From = ClearB.ActualWidth;

buttonClear.To = 79;

buttonClear.Duration = TimeSpan.FromSeconds(3);

ClearB.BeginAnimation(Button.WidthProperty, buttonClear);

DoubleAnimation buttonClose = new DoubleAnimation();

buttonClose.From = CloseB.ActualWidth;

buttonClose.To = 79;

buttonClose.Duration = TimeSpan.FromSeconds(3);

CloseB.BeginAnimation(Button.WidthProperty, buttonClose);

}

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

{

Close();

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

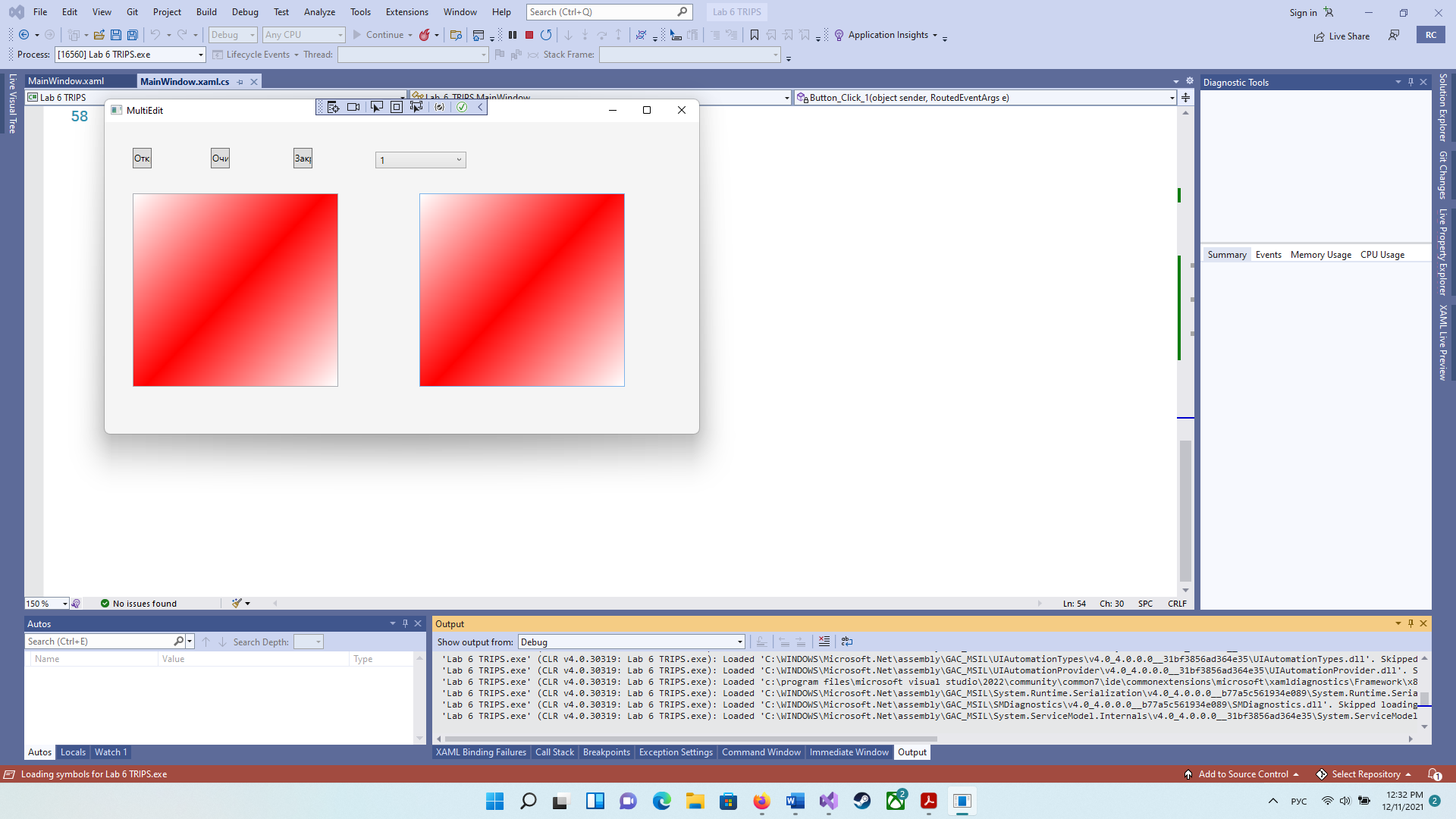
textbox1.Clear();

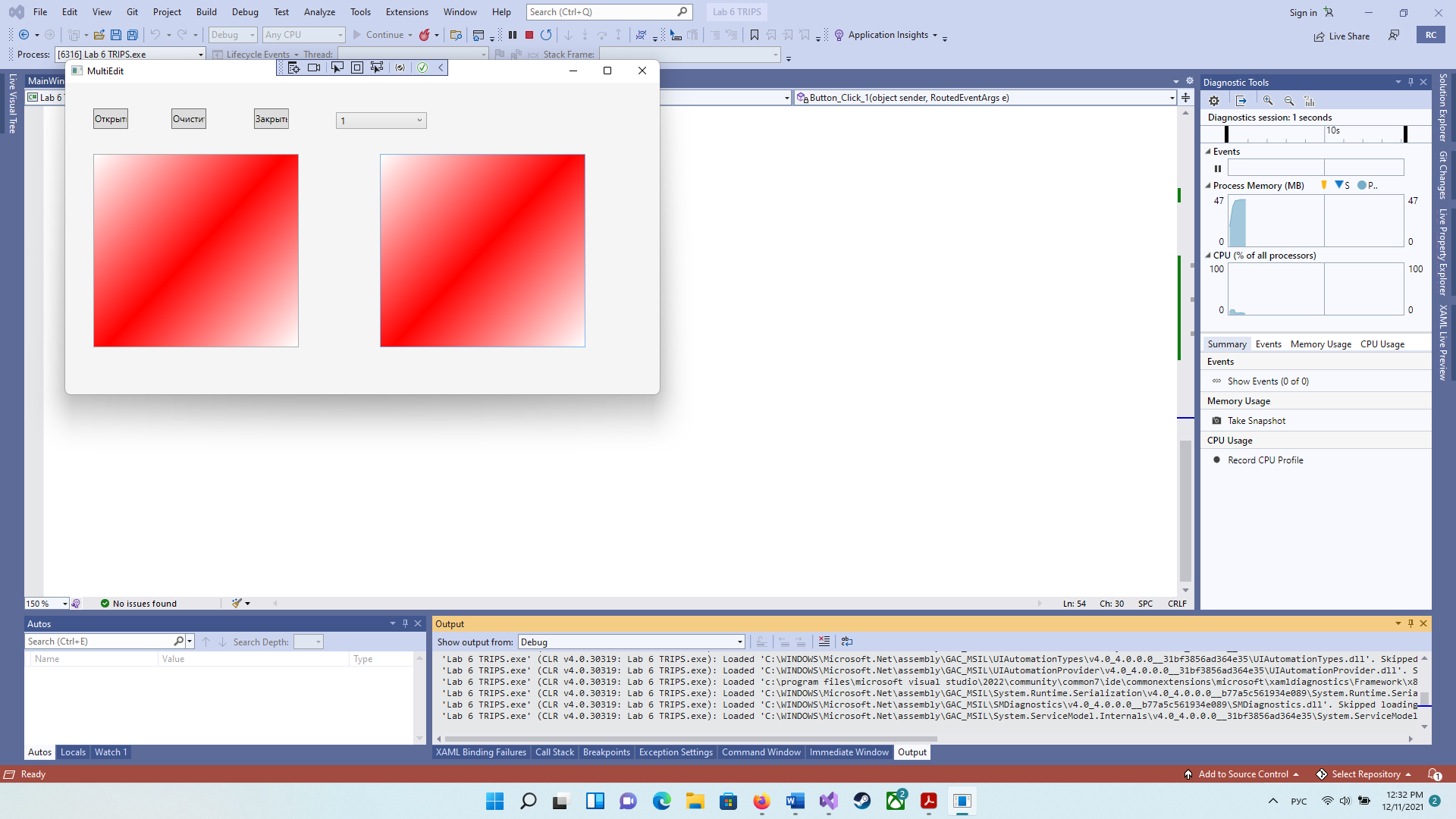
textbox2.Clear();

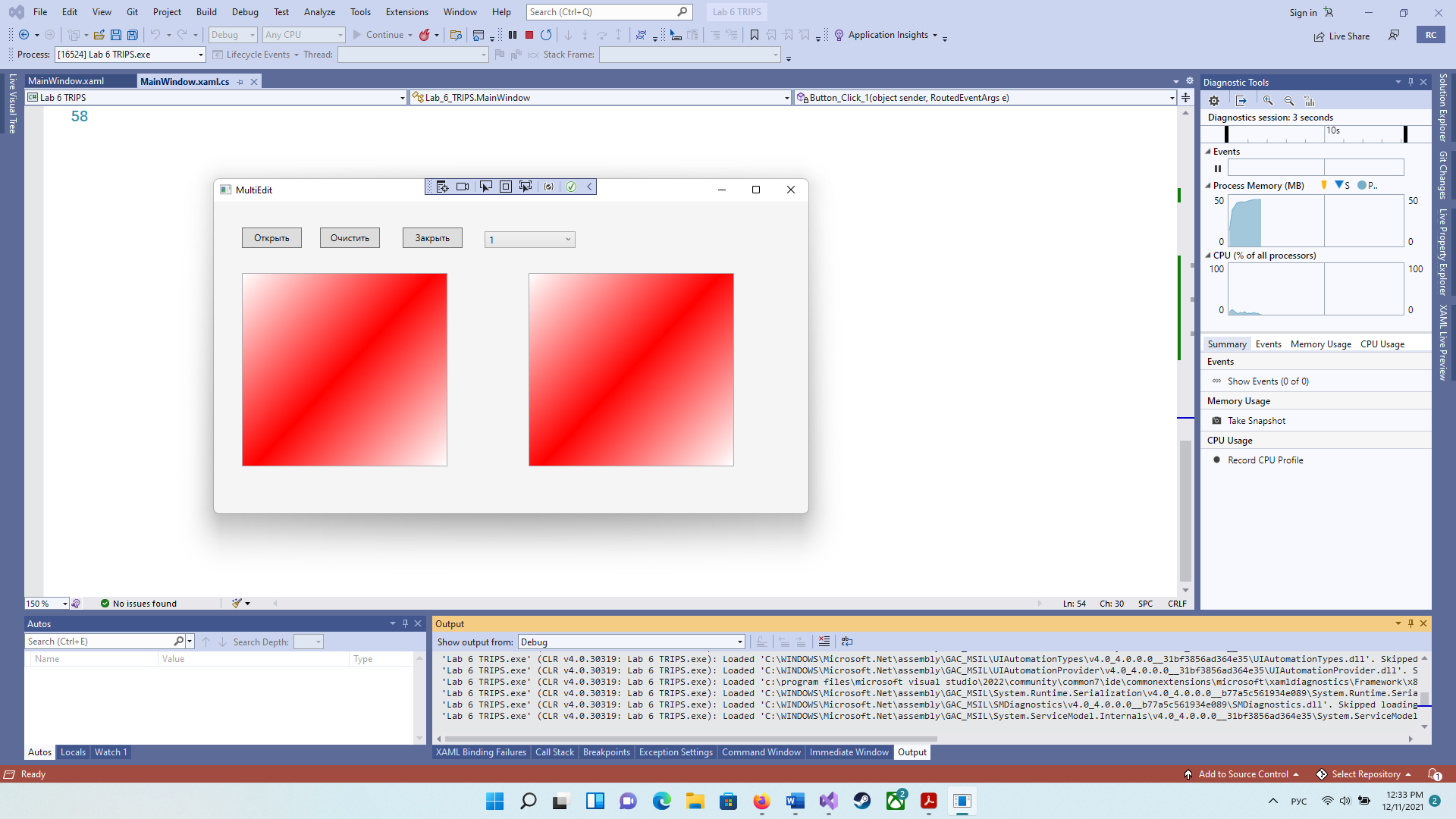
}

}

}







|  |
| --- |
| Задание 2  Разработайте WPF-приложение «Убегающая кнопка»: при наведении курсора мыши на кнопку она смещается на некоторое расстояние от курсора. Событие наведения курсора мыши – MouseEnter. |

<Window x:Class="Lab\_7\_TRIPS.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:Lab\_7\_TRIPS"

mc:Ignorable="d"

Title="Убегающая кнопка" Height="450" Width="800">

<Grid >

<Button x:Name="Click" MouseEnter="Click\_MouseEnter" Content="Нажмите" Margin="0,342,634,0" Click="Click\_Click" Width="54" HorizontalAlignment="Right" Height="26" VerticalAlignment="Top" />

<Button Content="Выйти" HorizontalAlignment="Left" Margin="685,350,0,0" VerticalAlignment="Top" Click="Button\_Click" RenderTransformOrigin="-1.792,-0.756"/>

</Grid>

</Window>

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;

using System.Windows.Media.Animation;

namespace Lab\_7\_TRIPS

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

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

{

MessageBox.Show("Nice reflex");

}

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

{

Close();

}

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

{

Random r = new Random();

int t = r.Next(15, 300);

int rig = r.Next(15, 700);

ThicknessAnimation animation = new ThicknessAnimation()

{

From = new Thickness(Click.Margin.Left, Click.Margin.Top, Click.Margin.Right, Click.Margin.Bottom),

To = new Thickness(Click.Margin.Left, t, rig, Click.Margin.Bottom),

Duration = TimeSpan.FromSeconds(1),

SpeedRatio = 2

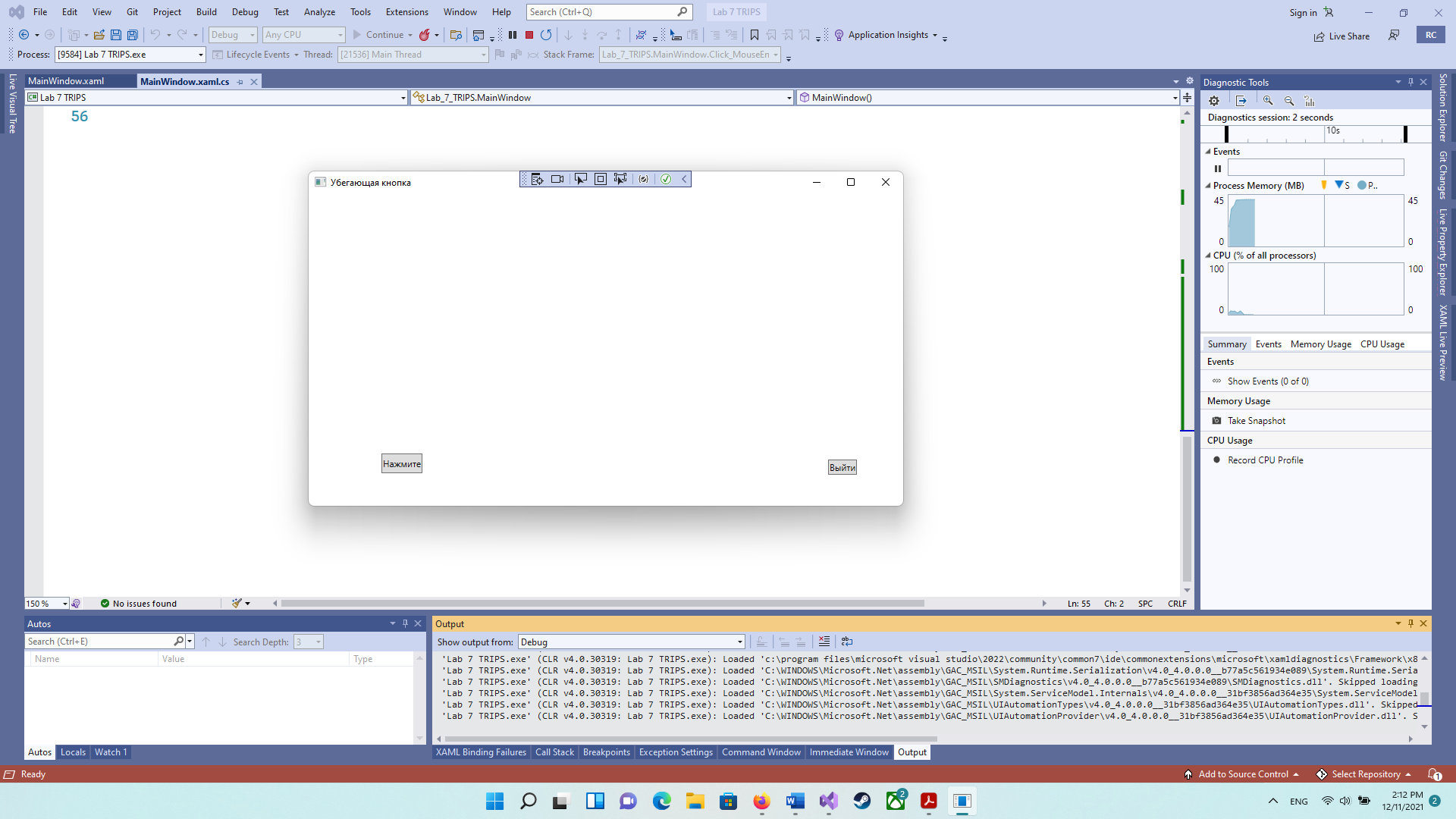
};

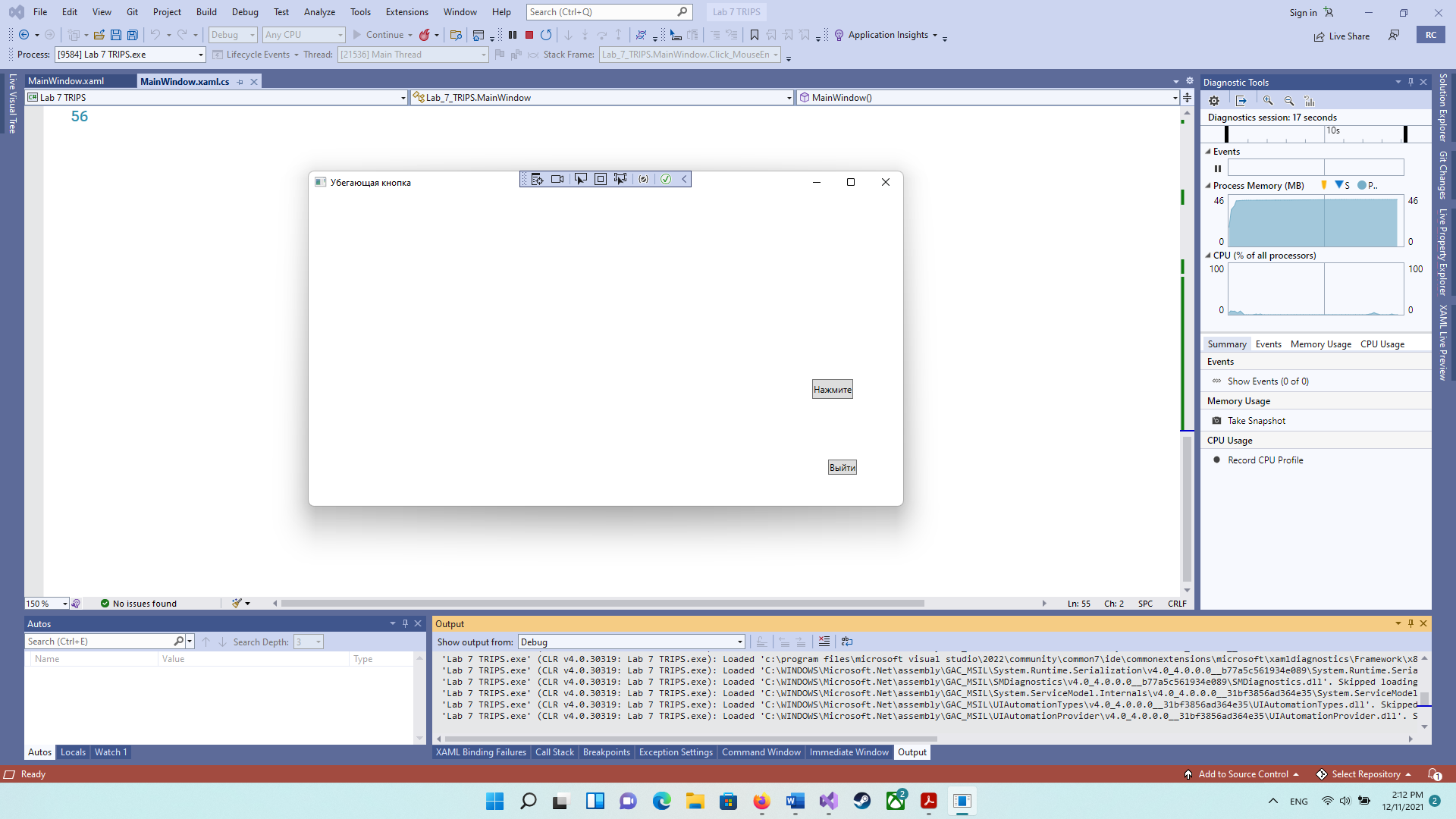
Click.BeginAnimation(ContentControl.MarginProperty, animation);

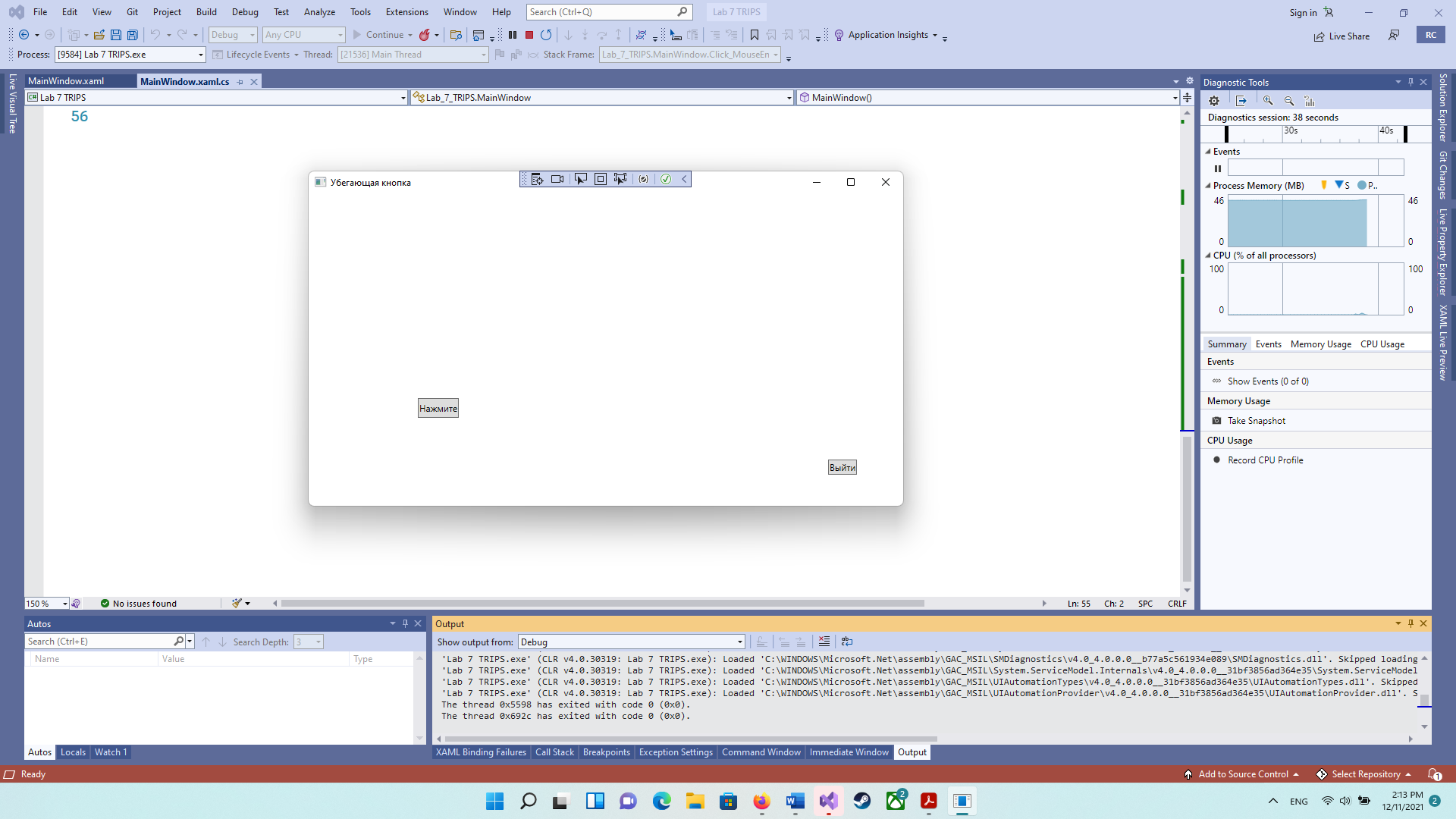
}

}

}







|  |
| --- |
| Задание 3  Разработайте WPF-приложение «Пульсар», изображающее круг, плавно меняющий свое состояние по следующей схеме: |

<Window x:Class="Lab\_7\_zad\_2.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:Lab\_7\_zad\_2"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800">

<Grid>

<Ellipse Width="100" Height="100">

<Ellipse.Fill>

<RadialGradientBrush>

<GradientStop Color="Yellow" Offset="1"/>

<GradientStop Color="Red" Offset="0"/>

</RadialGradientBrush>

</Ellipse.Fill>

<Ellipse.Triggers>

<EventTrigger RoutedEvent="Window.Loaded">

<BeginStoryboard>

<Storyboard>

<ColorAnimation Storyboard.TargetProperty="Fill.GradientStops[0].Color" AutoReverse="True" To="Red" Duration="0:0:3" RepeatBehavior="Forever" />

</Storyboard>

</BeginStoryboard>

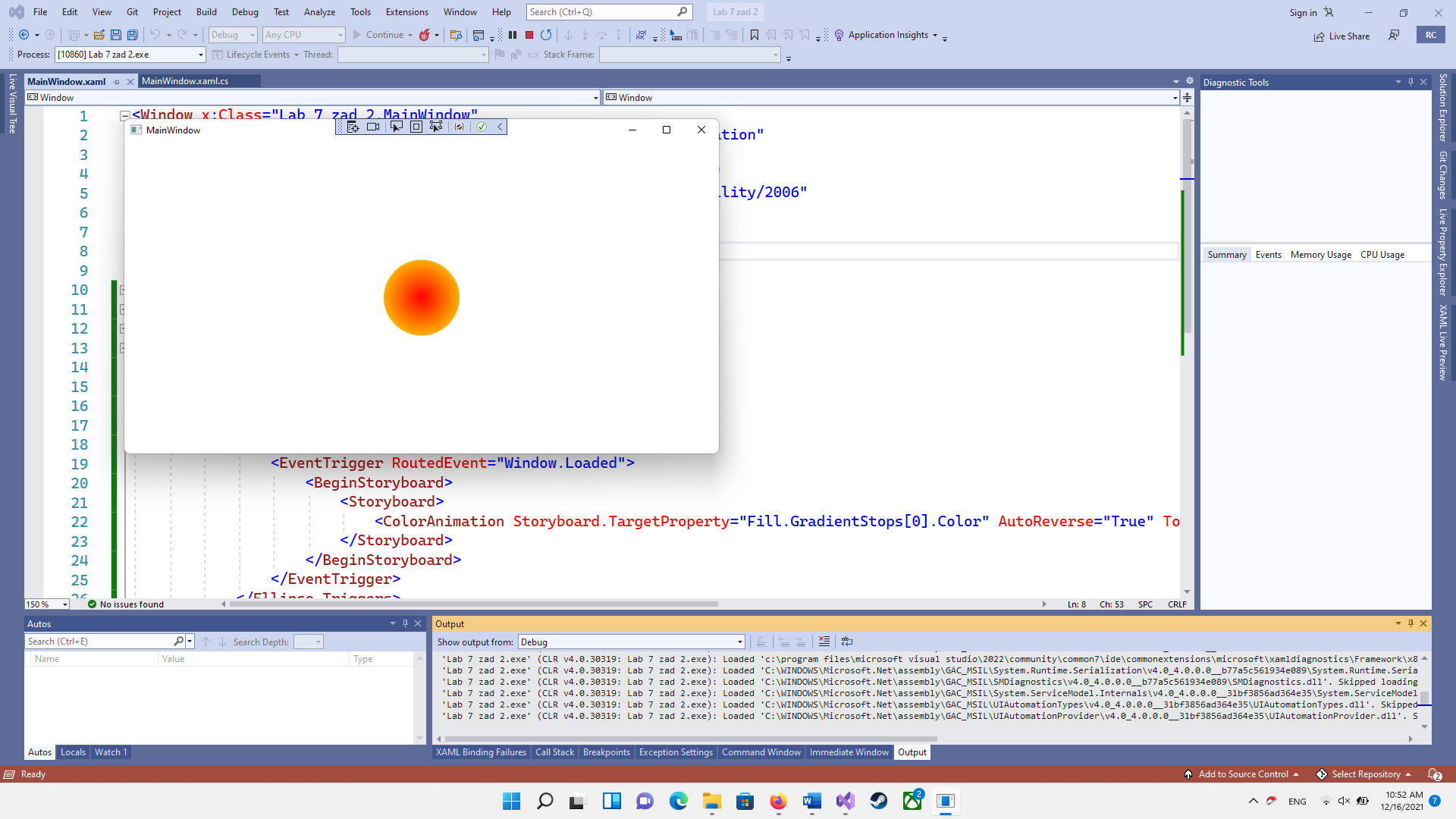
</EventTrigger>

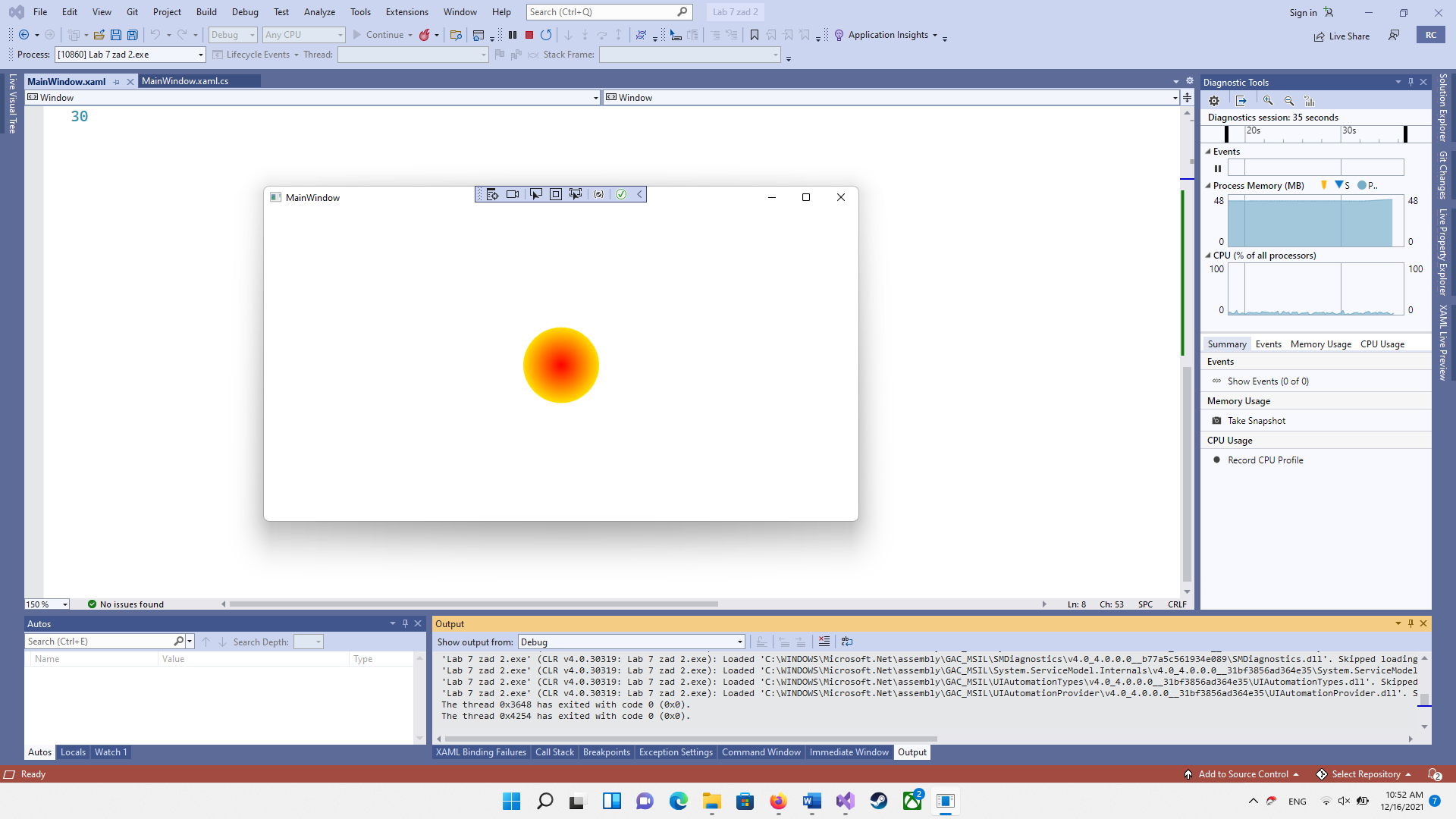
</Ellipse.Triggers>

</Ellipse>

</Grid>

</Window>





Вывод: научился использовать анимацию в WPF-приложениях.