Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №4

за 1 семестр

По дисциплине: «ТРИПС»

Тема: «Анимация»

Выполнил:

Студент 2 курса

Группы ПО-4(1)

Иваненко И. Л.

Проверила:

Войцехович О.Ю.

2020

Цель работы: изучить анимацию в графической системе WPF.

**Задание 1**. Доработать MultiEdit с использованием анимации.

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

mc:Ignorable="d"

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

<Window.Resources>

<Style x:Key="Gradient">

<Setter Property="Control.Background">

<Setter.Value>

<LinearGradientBrush StartPoint="0,0" EndPoint="0,0.5">

<LinearGradientBrush.GradientStops>

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

<GradientStop Color="Red" Offset="2"/>

</LinearGradientBrush.GradientStops>

</LinearGradientBrush>

</Setter.Value>

</Setter>

</Style>

<Style TargetType="TextBox">

<Setter Property="Opacity" Value="0.5"/>

<Setter Property="Margin" Value="0,40,0,0"/>

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

<Style.Triggers>

<EventTrigger RoutedEvent = "MouseEnter">

<BeginStoryboard>

<Storyboard>

<DoubleAnimation From="25" To="85" Storyboard.TargetProperty="Height"/>

<ThicknessAnimation From="0,40,0,0" To="0" Storyboard.TargetProperty="Margin"/>

<DoubleAnimation From="12" To="30" Storyboard.TargetProperty="FontSize"/>

</Storyboard>

</BeginStoryboard>

</EventTrigger>

<EventTrigger RoutedEvent = "MouseLeave">

<BeginStoryboard>

<Storyboard>

<DoubleAnimation From="85" To="25" Storyboard.TargetProperty="Height"/>

<ThicknessAnimation From="0" To="0,40,0,0" Storyboard.TargetProperty="Margin"/>

<DoubleAnimation From="30" To="12" Storyboard.TargetProperty="FontSize"/>

</Storyboard>

</BeginStoryboard>

</EventTrigger>

</Style.Triggers>

</Style>

</Window.Resources>

<Grid>

<Grid>

<Grid.RowDefinitions>

<RowDefinition/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<StackPanel Grid.Row="0" Grid.Column="0" Style="{StaticResource Gradient}">

<TextBox></TextBox>

<TextBox></TextBox>

<TextBox></TextBox>

<TextBox></TextBox>

</StackPanel>

<StackPanel Grid.Row="0" Grid.Column="1" Style="{StaticResource Gradient}">

<TextBox></TextBox>

<TextBox></TextBox>

<TextBox></TextBox>

<TextBox></TextBox>

</StackPanel>

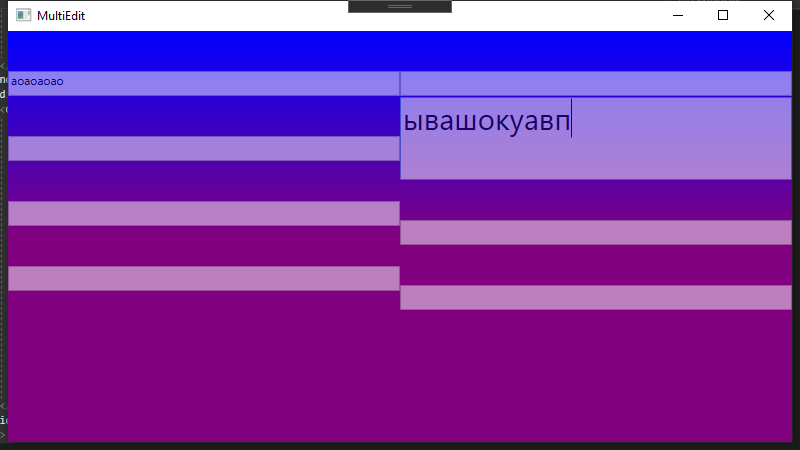
</Grid>

</Grid>

</Window>

Результат:

Теперь Изменение размеров окна и шрифта TextBox Происходит при использовании анимации



**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="RunningButton.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:RunningButton"

mc:Ignorable="d"

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

<Grid>

<Button Content="Кнiпка" Width="60" Height="20">

<Button.Triggers>

<EventTrigger RoutedEvent="Button.MouseEnter">

<BeginStoryboard>

<Storyboard>

<ThicknessAnimation

From="0"

To="300, 0, 0, 300"

Duration="0:0:0.2"

Storyboard.TargetProperty

="Margin"

/>

</Storyboard>

</BeginStoryboard>

</EventTrigger>

<EventTrigger RoutedEvent="Button.MouseLeave">

<BeginStoryboard>

<Storyboard>

<ThicknessAnimation

From="300, 0, 0, 300"

To="0"

Duration="0:0:0.2"

Storyboard.TargetProperty

="Margin"

/>

</Storyboard>

</BeginStoryboard>

</EventTrigger>

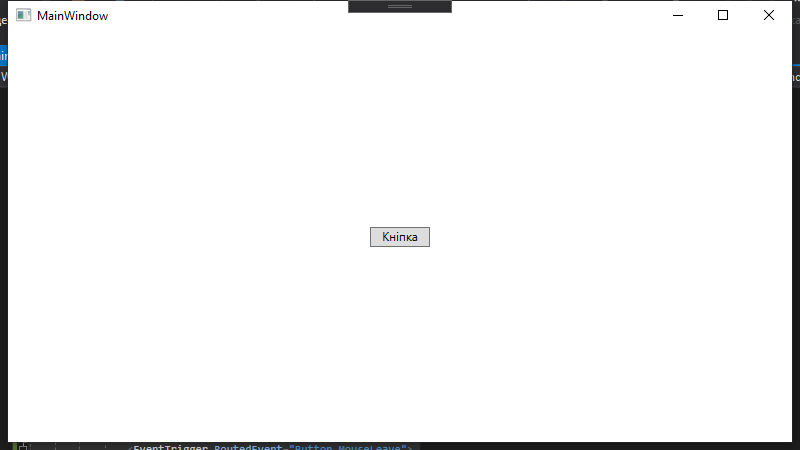
</Button.Triggers>

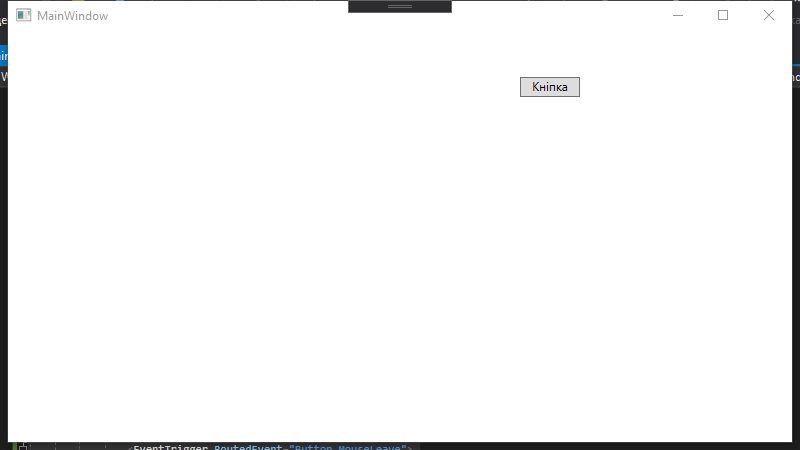
</Button>

</Grid>

</Window>

Результат:





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



XAML:

<Window x:Class="WpfApp4.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:WpfApp4"

mc:Ignorable="d"

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

<Grid>

<Ellipse Width="400" Height="400">

<Ellipse.Fill>

<RadialGradientBrush GradientOrigin="0.5,0.5" RadiusX="0.75" RadiusY="0.75">

<GradientStop Color="Yellow" Offset="0.4"/>

<GradientStop Color="Red" Offset="-0.2"/>

</RadialGradientBrush>

</Ellipse.Fill>

<Ellipse.Triggers>

<EventTrigger RoutedEvent="Window.Loaded">

<BeginStoryboard>

<Storyboard>

<DoubleAnimationUsingKeyFrames

Storyboard.TargetProperty="Fill.GradientStops[0].Offset"

Duration="0:0:4"

RepeatBehavior="Forever"

AutoReverse="True">

<LinearDoubleKeyFrame KeyTime="0%" Value="0.4"/>

<LinearDoubleKeyFrame KeyTime="10%" Value="0.5"/>

<LinearDoubleKeyFrame KeyTime="20%" Value="0.6"/>

<LinearDoubleKeyFrame KeyTime="30%" Value="0.7"/>

<LinearDoubleKeyFrame KeyTime="40%" Value="0.8"/>

<LinearDoubleKeyFrame KeyTime="50%" Value="0.9"/>

<LinearDoubleKeyFrame KeyTime="60%" Value="1.0"/>

<LinearDoubleKeyFrame KeyTime="70%" Value="1.1"/>

<LinearDoubleKeyFrame KeyTime="80%" Value="1.2"/>

<LinearDoubleKeyFrame KeyTime="90%" Value="1.3"/>

<LinearDoubleKeyFrame KeyTime="100%" Value="1.4"/>

</DoubleAnimationUsingKeyFrames>

</Storyboard>

</BeginStoryboard>

</EventTrigger>

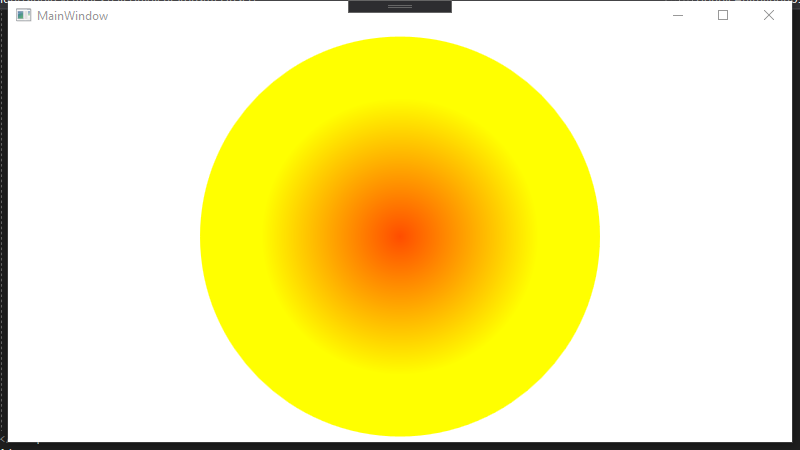
</Ellipse.Triggers>

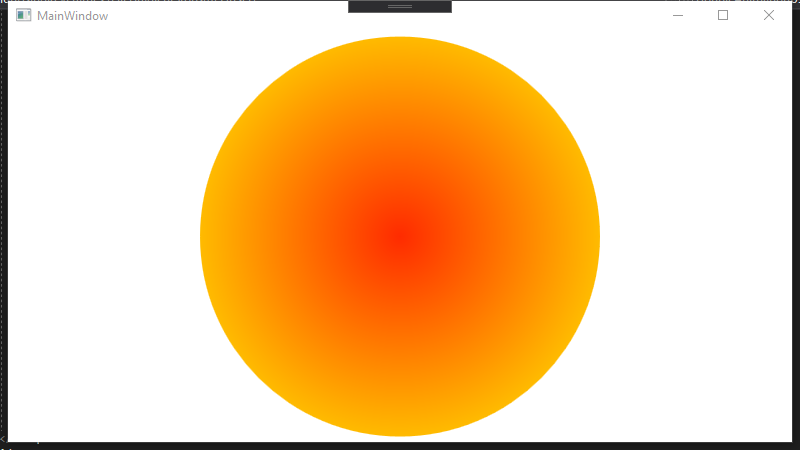
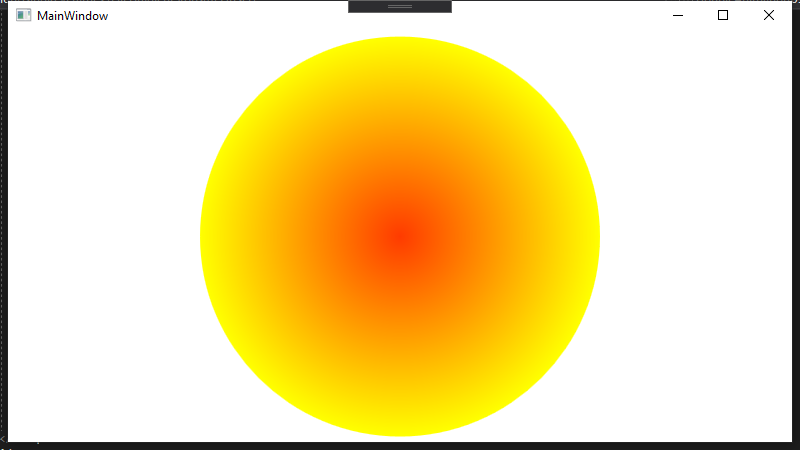
</Ellipse>

</Grid>

</Window>

Результат:





Вывод: изучил принципы работы с анимацией в WPF.