**ЛАБОРАТОРНА РОБОТА №8, 9**

**Тема: ВИКОРИСТАННЯ STORYBOARD ДЛЯ СТВОРЕННЯ АНІМАЦІЇ**

**Мета:** ознайомитись із основами створення анімацій на базі Storyboard. Навчитися створювати анімації та анімувати різні властивості елементів управління.

**Завдання 13 (частина 1 та частина 2):** Анімація рамки: Зміна кольору рамки елемента при наведенні. Голографічний ефект: Анімуйте елемент, щоб він виглядав так, ніби "плаває" над поверхнею з різними кольорами.

**Хід роботи**

**Завдання 1:** Анімація рамки: Зміна кольору рамки елемента при наведенні.

**Мета:** Створити ефект зміни кольору рамки елемента при наведенні курсора миші.

**Лістинг код:**

<!-- Елемент для анімації рамки -->

<Border x:Name="AnimatedBorder"

Width="200"

Height="100"

BorderThickness="5"

CornerRadius="10"

HorizontalAlignment="Center"

VerticalAlignment="Top"

Margin="0,50,0,0">

<Border.BorderBrush>

<SolidColorBrush x:Name="BorderBrushRef" Color="DarkBlue"/>

</Border.BorderBrush>

<TextBlock Text="Наведи на мене!"

HorizontalAlignment="Center"

VerticalAlignment="Center"

FontSize="16"/>

<!-- Тригер, який реагує на наведення миші -->

<Border.Triggers>

<EventTrigger RoutedEvent="Border.MouseEnter">

<BeginStoryboard>

<Storyboard>

<!-- Анімація кольору рамки до червоного -->

<ColorAnimation

Storyboard.TargetName="BorderBrushRef"

Storyboard.TargetProperty="Color"

To="Red"

Duration="0:0:0.3"/>

</Storyboard>

</BeginStoryboard>

</EventTrigger>

<EventTrigger RoutedEvent="Border.MouseLeave">

<BeginStoryboard>

<Storyboard>

<!-- Анімація повернення до початкового кольору -->

<ColorAnimation

Storyboard.TargetName="BorderBrushRef"

Storyboard.TargetProperty="Color"

To="DarkBlue"

Duration="0:0:0.5"/>

</Storyboard>

</BeginStoryboard>

</EventTrigger>

</Border.Triggers>

</Border>

**Принцип роботи:**

1. Використовуються EventTrigger для обробки подій MouseEnter та MouseLeave
2. ColorAnimation безпосередньо змінює властивість Color кисті SolidColorBrush
3. Різна тривалість анімації (0.3с для входу, 0.5с для виходу) створює природніший ефект

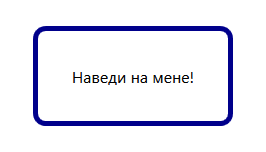


Рис.1. Завдання 1. Без наведення курсора миші

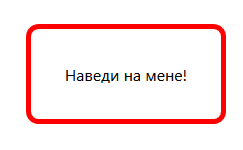


Рис.2. Завдання 1. При наведенні курсора миші

**Завдання 2:** Голографічний ефект: Анімуйте елемент, щоб він виглядав так, ніби "плаває" над поверхнею з різними кольорами.

**Мета:** Створити ефект елемента, що "плаває" над поверхнею з плавною зміною кольорів.

**Лістинг код:**

<Border x:Name="HolographicElement"

Width="150"

Height="150"

Background="{DynamicResource HoloBrush}"

CornerRadius="75"

HorizontalAlignment="Center"

VerticalAlignment="Top"

Margin="0,228,0,0"

RenderTransformOrigin="0.5, 0.5">

<Border.Resources>

<!-- Оголошуємо кисть як ресурс -->

<SolidColorBrush x:Key="HoloBrush" Color="Purple"/>

<Storyboard x:Key="HolographicAnimation" RepeatBehavior="Forever">

<!-- Анімація руху -->

<DoubleAnimationUsingKeyFrames

Storyboard.TargetName="HoloTranslate"

Storyboard.TargetProperty="Y"

Duration="0:0:2" AutoReverse="True">

<LinearDoubleKeyFrame KeyTime="0:0:0" Value="0"/>

<LinearDoubleKeyFrame KeyTime="0:0:1" Value="-15"/>

</DoubleAnimationUsingKeyFrames>

<!-- Анімація кольору - звертаємось до ресурсу -->

<ColorAnimationUsingKeyFrames

Storyboard.TargetName="HolographicElement"

Storyboard.TargetProperty="(Background).(SolidColorBrush.Color)"

Duration="0:0:4" AutoReverse="True">

<LinearColorKeyFrame KeyTime="0:0:0" Value="Purple"/>

<LinearColorKeyFrame KeyTime="0:0:1" Value="Blue"/>

<LinearColorKeyFrame KeyTime="0:0:2" Value="Cyan"/>

<LinearColorKeyFrame KeyTime="0:0:3" Value="BlueViolet"/>

</ColorAnimationUsingKeyFrames>

<!-- Анімація непрозорості -->

<DoubleAnimationUsingKeyFrames

Storyboard.TargetName="HolographicElement"

Storyboard.TargetProperty="Opacity"

Duration="0:0:1.5" AutoReverse="True">

<LinearDoubleKeyFrame KeyTime="0:0:0" Value="0.8"/>

<LinearDoubleKeyFrame KeyTime="0:0:0.75" Value="1"/>

</DoubleAnimationUsingKeyFrames>

</Storyboard>

</Border.Resources>

<!-- Transform для анімації руху -->

<Border.RenderTransform>

<TransformGroup>

<TranslateTransform x:Name="HoloTranslate" X="0" Y="0"/>

</TransformGroup>

</Border.RenderTransform>

<TextBlock Text="Голограмма"

HorizontalAlignment="Center"

VerticalAlignment="Center"

Foreground="White"

FontWeight="Bold"/>

<Border.Triggers>

<EventTrigger RoutedEvent="Border.Loaded">

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

</EventTrigger>

</Border.Triggers>

</Border>

**Принцип роботи:**

1. Анімація руху (TranslateTransform.Y) - створює ефект "плавучості"
2. Анімація кольору (Background.Color) - створює переливання кольорів
3. Анімація прозорості (Opacity) - додає ефект миготіння
4. RepeatBehavior="Forever" робить анімацію безкінечною
5. AutoReverse="True" забезпечує плавний зворотний рух



Рис.3. Завдання 2. Елемент у первісному стані голограма



Рис.4. Завдання 2. Елемент у русі та зміна кольору з миготінням

**Висновок:** У ході виконання лабораторної роботи було успішно освоєно технологію створення анімацій за допомогою Storyboard у WPF. Практично реалізовано два типи анімаційних ефектів: зміну кольору рамки при наведенні миші та складний голографічний ефект "плавання" з переливом кольорів. Робота демонструє, що Storyboard є потужним інструментом для створення інтерактивних інтерфейсів, дозволяючи керувати різними властивостями елементів (колір, позиція, прозорість) через різноманітні типи анімацій (ColorAnimation, DoubleAnimationUsingKeyFrames). Використання EventTrigger дозволило прив'язати анімації до подій без написання програмного коду, що підкреслює ефективність декларативного підходу WPF. Отримані навички можуть бути використані для покращення користувацького досвіду шляхом додавання плавних переходів, візуальних підказок та естетично привабливих ефектів до інтерфейсів програм.

**Виконано:** Куліш Владою Іванівною, група БІП-1-22.