UDC 004.42

MODELING AND CREATING ANIMATION USING STORYBOARD

V.I. Pylypenko, Senior Lecturer, Department of ICT
Kyiv National University of Technologies and Design
B.V. Kalensky, student
Kyiv National University of Technologies and Design

Keywords: WPF, XAML, animation, StoryBoard.

Modeling and creating animations using Storyboard in WPF allows you to create functional visual applications, as it can animate various properties of user interface (UI) elements over a specified time. This provides the ability to bring dynamics and interactivity to applications. Animations can include changes in the position, color, size, and other characteristics of elements. Storyboard is implemented through XAML markup, which makes it easy for developers to use. You can combine multiple animations, adjust the duration, and use functions to create smooth and effective transitions [1-3]. This allows you to achieve unique visual effects that enhance the overall experience of the user interacting with the application. The general appearance of the application window with object jumping animation is shown in Fig. 1.

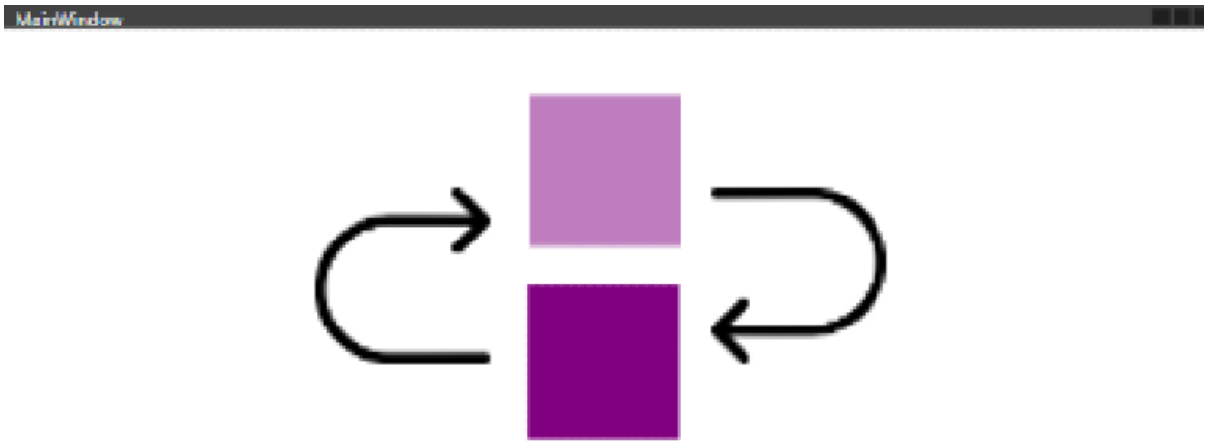

Figure 1 – Window with cube jumping animation from Storyboard

After creating the project, interface elements are added to the main file MainWindow.xaml, where the main window is defined. In particular, our cube object for animation. Adding a Grid container to XAML and creating an object inside it that will be animated is shown in Fig. 2.

```
<Rectangle Name="IceCube" Width="100" Height="100" Fill="Purple">
    <Rectangle.RenderTransform>
        <TranslateTransform />
    </Rectangle.RenderTransform>
</Rectangle>
```

Figure 2 - Using TranslateTransform to move an object along the X and Y axes

Creating an animation that will allow the rectangle to move up and down, simulating jumping, is shown in Figure 3. Here, a DoubleAnimation is added, which changes the Y property of the TranslateTransform.

```
<Storyboard RepeatBehavior="Forever" AutoReverse="True">
    <!-- Анімація переміщення по осі Y (стрибки) -->
    <DoubleAnimation Storyboard.TargetName="IceCube"
                     Storyboard.TargetProperty="RenderTransform.(TranslateTransform.Y)"
                     From="0" To="-150" Duration="0:0:1" />
</Storyboard>
```

Figure 3 – Creating an animation of raising a rectangle 150 pixels up and rotating it back

Adding an EventTrigger so that the animation starts automatically when the window is loaded in the Window.Loaded event is shown in Fig. 4. To do this, a trigger is created inside the Grid container.

```
<Grid.Triggers>
    <EventTrigger RoutedEvent="Window.Loaded">
        <BeginStoryboard>
            <Storyboard RepeatBehavior="Forever" AutoReverse="True">
                <!-- Анімація переміщення по осі Y (стрибки) -->
                <DoubleAnimation Storyboard.TargetName="IceCube"
                                 Storyboard.TargetProperty="RenderTransform.(TranslateTransform.Y)"
                                 From="0" To="-150" Duration="0:0:1" />
            </Storyboard>
        </BeginStoryboard>
    </EventTrigger>
</Grid.Triggers>
```

Figure 4 – Implementing the Window.Loaded event to start an animation

Using Storyboards in WPF makes it easy to implement complex animations for various interface elements. In this case, Storyboards are used to create a looping animation of an object jumping — a rectangle that moves along the Y axis, rising and falling back thanks to the RepeatBehavior and AutoReverse properties.

References

1. Yuen, S. (2020). Mastering Windows Presentation Foundation: Build responsive UIs for desktop applications with WPF. Packt Publishing Ltd.

2. Troelsen, A., & Japikse, P. (2022). WPF Controls, Layouts, Events, and Data Binding. In Pro C# 10 with. NET 6: Foundational Principles and Practices in Programming (pp. 1137-1195). Berkeley, CA: Apress.

3. Xie, W., Wang, Q., & Peng, D. (2023, September). Construction of Simulation Environment and Design of Path Planning for Maze Robot. In 2023 IEEE 6th International Conference on Information Systems and Computer Aided Education (ICISCAE) (pp. 270-274). IEEE.