Introduction

Nova.Avalonia.UI is a control library built for Avalonia. It focuses on controls that are themeable, accessible, and ready to drop into desktop, web, and mobile experiences.

Available controls

- **Shimmer**: Skeleton loading effect for async data scenarios.
- **Avatar** and **AvatarGroup**: Identity visuals with initials, images, status badges, and grouping support.

How to use these docs

- Start with Getting Started to install the package and register the styles.
- Browse the individual control pages under **Controls** for API details and usage patterns.
- Refer to the API reference for full class members when you need to extend or customize behaviors.

Getting Started

Follow these steps to install Nova.Avalonia.UI, register its styles, and place your first control in a view.

Prerequisites

- Avalonia 11 or later
- .NET 9 (the library currently targets `net9.0`)

Install the NuGet package

From your application project, install the library:

```
dotnet add package Nova.Avalonia.UI
```

Register the control styles

Add the Nova styles to your `Application.Styles` so the controls pick up their templates. Keep your base theme (for example, `FluentTheme`) before the style include.

```
<Application xmlns="https://github.com/avaloniaui"
             xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
             x:Class="MyApp.App">
    <Application.Styles>
        <FluentTheme />
        <StyleInclude Source="avares://Nova.Avalonia.UI/Themes/Controls.axaml" />
    </Application.Styles>
</Application>
```

Use the controls in XAML

Declare the namespace for the controls and drop them into your layout. This example shows a shimmer placeholder wrapping content and a simple avatar.

```
<UserControl xmlns="https://github.com/avaloniaui"
             xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
             xmlns:nova="clr-
namespace:Nova.Avalonia.UI.Controls;assembly=Nova.Avalonia.UI">

    <StackPanel Spacing="16">
        <nova:Shimmer IsLoading="True" LoadingText="Loading profile">
            <StackPanel Spacing="8">
                <TextBlock FontSize="18" Text="Profile" />
```

```
                <Border Height="120" CornerRadius="12" Background="#1F1F1F" />
            </StackPanel>
        </nova:Shimmer>

        <nova:Avatar DisplayName="Avery Patel" Status="Online" />
    </StackPanel>
</UserControl>
```

Next, explore the individual control pages to see customization options and platform-specific notes.

Shimmer

The `Shimmer` control shows a lightweight skeleton while your content is loading. It inspects the visual tree beneath it to draw shapes that match text, images, and buttons, then animates a gradient sweep over the placeholders.

Add a Shimmer placeholder

Wrap the content that loads asynchronously in a `Shimmer`. Toggle `IsLoading` to switch between the placeholder and the real content.

```
<UserControl xmlns="https://github.com/avaloniaui"
             xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
             xmlns:nova="clr-namespace:Nova.Avalonia.UI.Controls;assembly=Nova.Avalonia.

    <nova:Shimmer IsLoading="True" LoadingText="Loading profile">
        <StackPanel Spacing="8">
            <TextBlock FontSize="20" Text="Profile" />
            <Border Height="160" CornerRadius="12" Background="#202020" />
            <Button Content="Refresh" Width="120" />
        </StackPanel>
    </nova:Shimmer>
</UserControl>
```

When `IsLoading` is `True`, Shimmer disables hit testing on the child content and announces the loading state to screen readers.

Customize the effect

Use the following properties to align the effect with your theme:

- `HighlightBrush` sets the moving gradient. Bind it to a `DynamicResource` for theme switching.
- `ShimmerOpacity` adjusts the overlay opacity. The default is `0.5`.
- `ShimmerAngle` sets the gradient angle in degrees.
- `LoadingText` defines the automation name announced while loading.

```
<nova:Shimmer IsLoading="True"
              HighlightBrush="{DynamicResource AccentGradient}"
              ShimmerOpacity="0.35"
              ShimmerAngle="12"
              LoadingText="Loading dashboard cards" />
```

Show loaded content

Set `IsLoading` to `False` when your data is ready. The child content becomes visible and interactive, and the automation name is cleared.

```
  // ViewModel
  public bool IsBusy { get; set; }

<nova:Shimmer IsLoading="{Binding IsBusy}">
    <ItemsControl Items="{Binding Orders}" />
</nova:Shimmer>
```

Avatar

The `Avatar` control presents a person's identity using an image, initials, icon, or custom content. It includes automatic background generation, size presets, and optional presence status indicators.

Create an avatar

Declare an `Avatar` and set `DisplayName`. With the default `DisplayMode` of `Auto`, the control will render initials when no image or icon is provided.

```
<UserControl xmlns="https://github.com/avaloniaui"
             xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
             xmlns:nova="clr-namespace:Nova.Avalonia.UI.Controls;assembly=Nova.Avalonia.

    <nova:Avatar DisplayName="Alex Martin" />
</UserControl>
```

Use images, icons, or custom content

Choose a display mode explicitly when you need to control the visual output:

- `DisplayMode="Image"` uses the `ImageSource` bitmap.
- `DisplayMode="Icon"` shows the provided `Icon` content.
- `DisplayMode="Content"` renders any custom `Content`.

```
<StackPanel Spacing="12">
    <nova:Avatar DisplayName="Jamie Fox" ImageSource="avares://Assets/jamie.png" Display
    <nova:Avatar DisplayName="Operations" DisplayMode="Icon">
        <nova:Avatar.Icon>
            <PathIcon Data="M18,13 L6,13 6,11 18,11z" />
        </nova:Avatar.Icon>
    </nova:Avatar>
    <nova:Avatar DisplayName="Admin" DisplayMode="Content">
        <nova:Avatar.Content>
            <Ellipse Fill="#F59E0B" Width="18" Height="18" />
        </nova:Avatar.Content>
    </nova:Avatar>
</StackPanel>
```

> **NOTE**
>
> If `DisplayMode` is left as `Auto`, the control picks an image when available, otherwise initials, then icon, then content.

Size, shape, and color

`Avatar` supports preset sizes via `Size` (`ExtraSmall` through `ExtraLarge`) and a `Custom` option controlled by `CustomSize`. Use `Shape` to switch between `Circle`, `Square`, and `Rectangle` corners.

The control can auto-generate a background color from the display name when `AutoGenerateBackground` is `True`, or you can set `BackgroundColor` and `ForegroundColor` directly.

```
<UniformGrid Columns="3" Rows="1" Margin="0,12,0,0">
    <nova:Avatar DisplayName="Kim Lee" Size="Small" />
    <nova:Avatar DisplayName="Drew Parker" Size="Large" Shape="Square" BackgroundColor="
    <nova:Avatar DisplayName="Avery Patel" Size="Custom" CustomSize="80" Shape="Rectangl
</UniformGrid>
```

Show presence status

Attach a status indicator with the `Status` property. You can override the default color per status with `StatusColor`.

```
<StackPanel Orientation="Horizontal" Spacing="10">
    <nova:Avatar DisplayName="Taylor Reed" Status="Online" />
    <nova:Avatar DisplayName="Morgan" Status="Away" />
    <nova:Avatar DisplayName="Jordan" Status="Busy" StatusColor="#C026D3" />
</StackPanel>
```

Tooltips automatically display the `DisplayName` when `ShowTooltip` is `True`, which helps identify users when only initials or icons are visible.

Arrange multiple avatars with AvatarGroup

Use `AvatarGroup` to stack or wrap multiple `Avatar` controls with configurable overlap and overflow handling. Combine `Spacing` and `MaxVisibleAvatars` to control layout, and place any remaining avatars in an overflow badge.

```
<StackPanel Spacing="12">
    <nova:AvatarGroup MaxVisibleAvatars="3" Spacing="-8">
        <nova:Avatar DisplayName="Taylor Reed" Status="Online" />
        <nova:Avatar DisplayName="Morgan Lee" Status="Away" />
        <nova:Avatar DisplayName="Jamie Fox" Status="Busy" />
        <nova:Avatar DisplayName="Avery Patel" />
    </nova:AvatarGroup>

    <nova:AvatarGroup Orientation="Vertical" Spacing="4">
        <nova:Avatar DisplayName="Ops" DisplayMode="Icon">
            <nova:Avatar.Icon>
                <PathIcon Data="M18,13 L6,13 6,11 18,11z" />
```

```
            </nova:Avatar.Icon>
        </nova:Avatar>
        <nova:Avatar DisplayName="Engineering" Status="Online" />
    </nova:AvatarGroup>
</StackPanel>
```

`AvatarGroup` also exposes `BorderBrush` and `BorderThickness` to add a ring around the stack when you need a stronger visual boundary against busy backgrounds.