

[](https://www.tutorialr.com/tutorials/)

Windows App SDK

Acrylic Material





# Acrylic Material

**Acrylic Material** shows how you can use **AcrylicBrush** with the **Windows App SDK** which can colour an

area of an application with a semi-transparent material that uses multiple effects including blur and noise.

## Step 1

Follow **Setup and Start** on how to get **Setup** and **Install** what you need for **Visual Studio 2022** and **Windows App SDK**.

|  |  |
| --- | --- |
| In **Windows 11** choose **Start** and then find or search for **Visual Studio 2022** and then select it. | Text  Description automatically generated |
| Once **Visual Studio 2022** has started select **Create a new project**. | **Graphical user interface, text  Description automatically generated** |
| Then choose the **Blank App, Packages (WinUI in Desktop)** and then select **Next**. | **Graphical user interface, text  Description automatically generated** |
| After that in **Configure your new project** type in the **Project name** as *AcrylicMaterial*, then select a Location and then select **Create** to start a new **Solution**. | **Graphical user interface, text, application, email  Description automatically generated** |

## Step 2

|  |  |
| --- | --- |
| Within **Solution** **Explorer** for the **Solution** and double-click on **MainWindow.xaml** to see the **XAML** for the **Main Window**. |  |

## Step 3

In the **XAML** for **MainWindow.xaml** there be some **XAML** for a **StackPanel**, this should be **Removed** by removing the following:

<StackPanel Orientation="Horizontal"

HorizontalAlignment="Center" VerticalAlignment="Center">

<Button x:Name="myButton" Click="myButton\_Click">Click Me</Button>

</StackPanel>

## Step 4

While still in the **XAML** for **MainWindow.xaml** above **</Window>**, type in the following **XAML**:

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<ComboBox Grid.Row="0" Margin="25" Name="Options" HorizontalAlignment="Stretch"

SelectionChanged="Options\_SelectionChanged">

<ComboBoxItem IsSelected="True">None</ComboBoxItem>

<ComboBoxItem>SystemControlAcrylicElementBrush</ComboBoxItem>

<ComboBoxItem>SystemControlAcrylicElementMediumHighBrush</ComboBoxItem>

<ComboBoxItem>SystemControlBaseHighAcrylicElementMediumBrush</ComboBoxItem>

</ComboBox>

<Viewbox Grid.Row="1">

<Grid>

<StackPanel Spacing="5" Orientation="Horizontal"

HorizontalAlignment="Center">

<Rectangle Width="50" Height="50" Fill="Black"/>

<Rectangle Width="50" Height="50" Fill="Gray"/>

<Rectangle Width="50" Height="50" Fill="Red"/>

<Rectangle Width="50" Height="50" Fill="Orange"/>

<Rectangle Width="50" Height="50" Fill="Yellow"/>

<Rectangle Width="50" Height="50" Fill="Green"/>

<Rectangle Width="50" Height="50" Fill="Cyan"/>

<Rectangle Width="50" Height="50" Fill="Blue"/>

<Rectangle Width="50" Height="50" Fill="Magenta"/>

<Rectangle Width="50" Height="50" Fill="Purple"/>

</StackPanel>

<Rectangle x:Name="Overlay"/>

</Grid>

</Viewbox>

</Grid>

This **XAML** features a **Grid** containing a ComboBox which has some of the **Acrylic** **Material** colours that can be used with **AcrylicBrush**, it also contains another **Grid** with a **StackPanel** with **Rectangle** elements that will be used to demonstrate the **AcrylicBrush** and a **Rectangle** of **Overlay** which will be coloured with the **AcrylicBrush**.

## Step 5

|  |  |
| --- | --- |
| Then, within **Solution** **Explorer** for the **Solution** select the arrow next to **MainWindow.xaml** then double-click on **MainWindow.xaml.cs** to see the **Code** for the **Main Window**. |  |

## Step 6

In the **Code** for **MainWindow.xaml.cs** there be a **Method** of **myButton\_Click(...)** this should be **Removed** by removing the following:

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

{

myButton.Content = "Clicked";

}

## Step 7

Once **myButton\_Click(...)** has been removed, below the end of **public MainWindow() { ... }** type in the following **Code**:

private void Options\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

if (Overlay != null && Windows.Foundation.Metadata.ApiInformation.IsTypePresent(

"Windows.UI.Xaml.Media.XamlCompositionBrushBase"))

{

string value = (Options.SelectedItem as ComboBoxItem).Content as string;

Overlay.Fill = value != "None" ?

Application.Current.Resources[value] as AcrylicBrush : null;

}

}

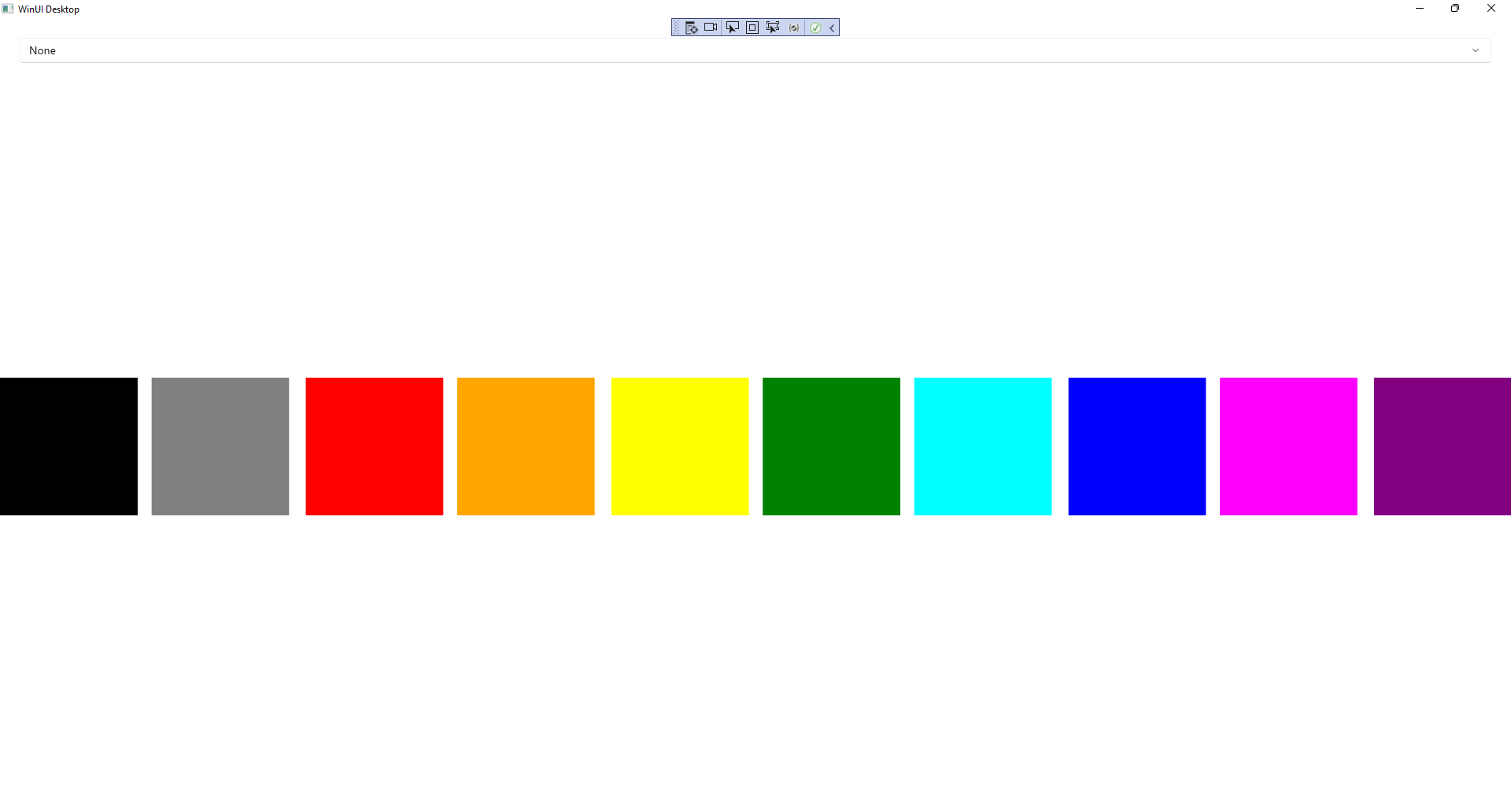
The **Method** of **Options\_SelectionChanged** will be triggered by the **Event** of selecting an item from the **ComboBox** then the code will check if the **Rectangle** of **Overlay** has a value along with checking to see if the **Acrylic Material** is supported, if so it will use the **SelectedItem** from the **ComboBox** and will set the **Property** for **Fill** of the **Rectangle** of **Overlay** to the **AcrylicBrush** that was selected in the **ComboBox** this uses the **Conditional Operator** of **?** and **:** where if the value before the **?** is **true**, then it will set the **Fill** to one of the **Acrylic Material** resources, if it is **false** it will set it to **null** or have no **Fill**.

## Step 8

|  |  |
| --- | --- |
| That completes the **Windows App SDK** application. In **Visual Studio 2022** from the **Toolbar** select **AcrylicMaterial (Package)** to **Start** the application. |  |

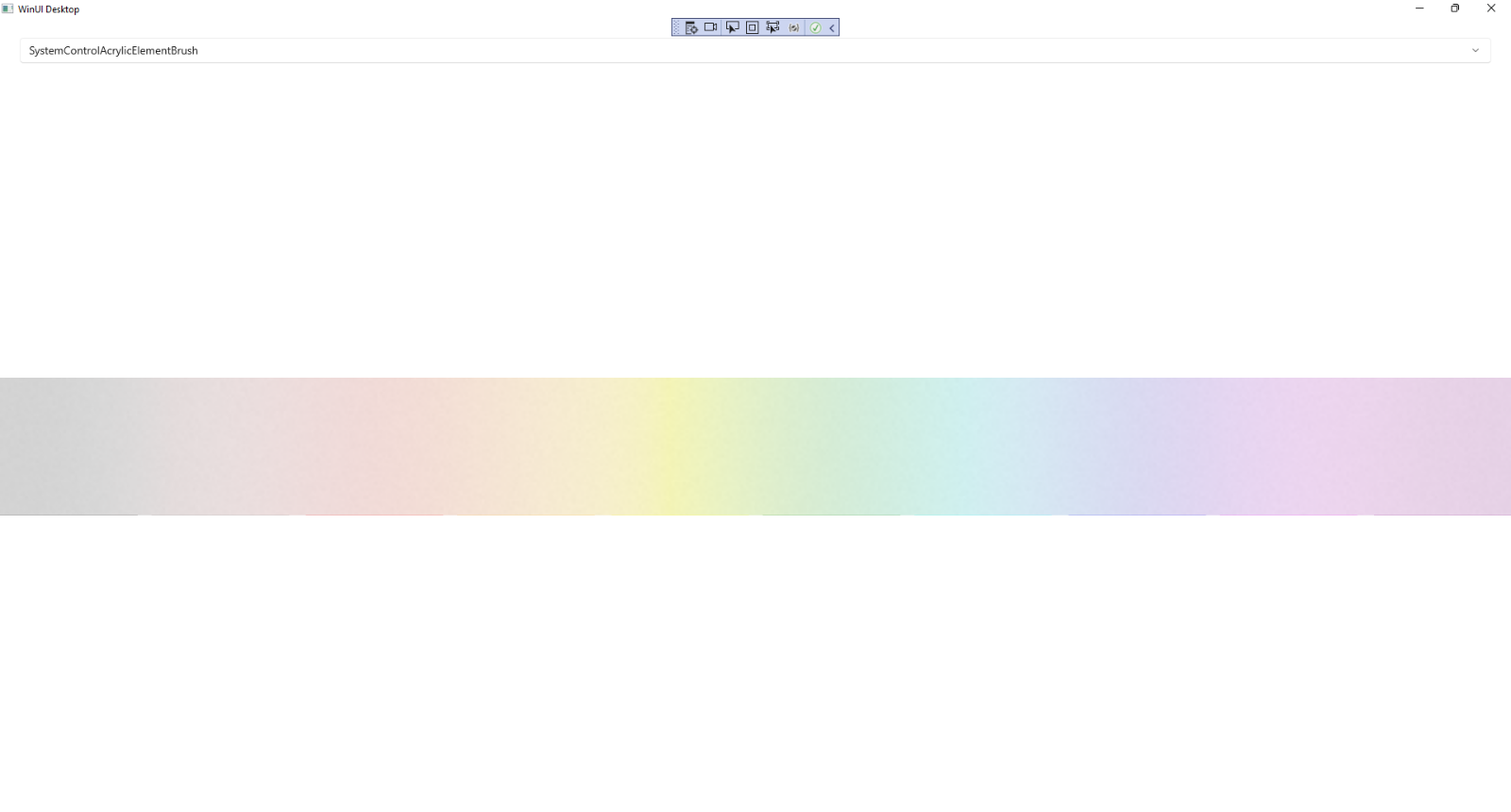
## Step 9

Once running you should see the **ComboBox** and the **Rectangle** Elements.

****

## Step 10

If you select an item from the **ComboBox** you will see the **Acrylic Material** of the **AcrylicBrush** being applied to the **Rectangle** of **Overlay**.



## Step 11

|  |  |
| --- | --- |
| To **Exit** the **Windows App SDK** application, select the **Close** button from the top right of the application as that concludes this **Tutorial** for **Windows App SDK** from [tutorialr.com](https://tutorialr.com)! |  |