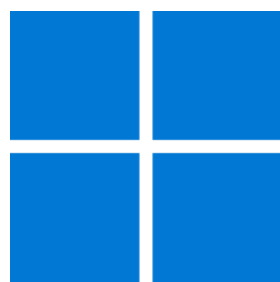




Windows App SDK



High or Low

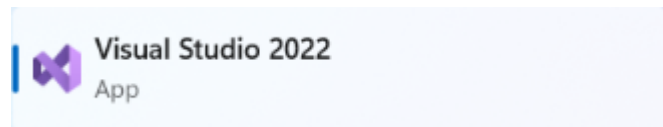
High or Low

High or Low shows how you can create simple **Card** based game where you guess if the next card is higher or lower than the last, ignoring suits so *Eight of Diamonds* is higher than *Six of Hearts*, using a toolkit from **NuGet** using the **Windows App SDK**.

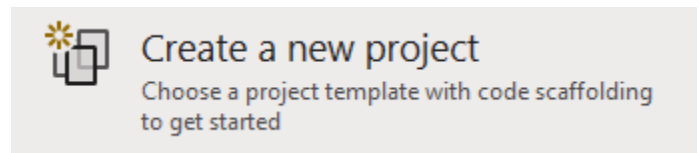
Step 1

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

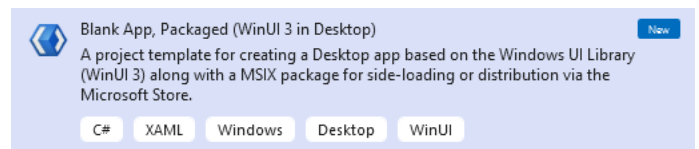
In **Windows 11** choose **Start** and then find or search for **Visual Studio 2022** and then select it.



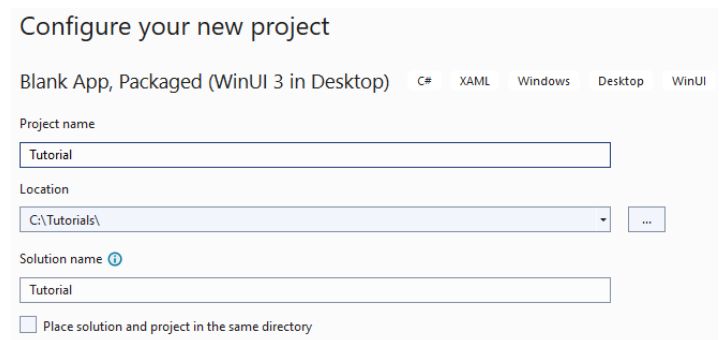
Once **Visual Studio 2022** has started select **Create a new project**.



Then choose the **Blank App, Packages (WinUI in Desktop)** and then select **Next**.

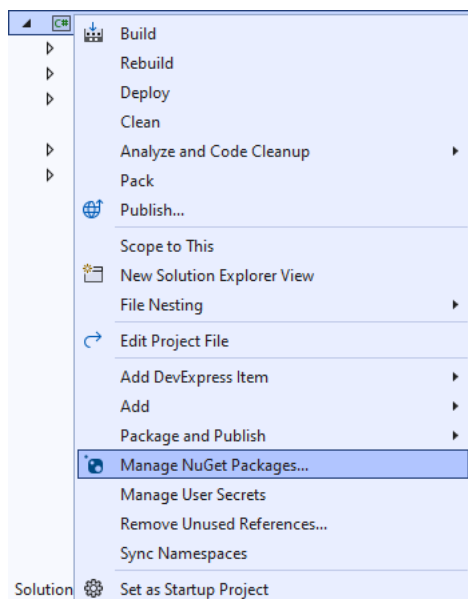


After that in **Configure your new project** type in the **Project name** as *HighOrLow*, then select a Location and then select **Create** to start a new **Solution**.



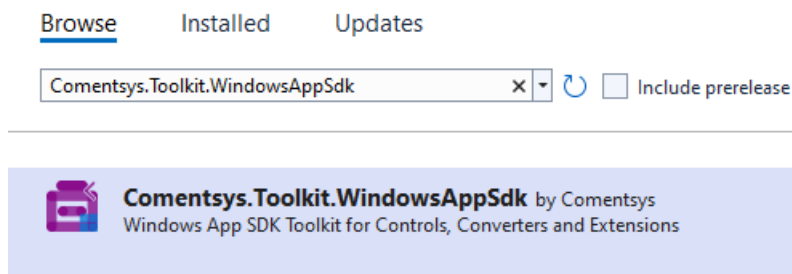
Step 2

Then in **Visual Studio** within **Solution Explorer** for the **Solution**, right click on the **Project** shown below the **Solution** and then select **Manage NuGet Packages...**



Step 3

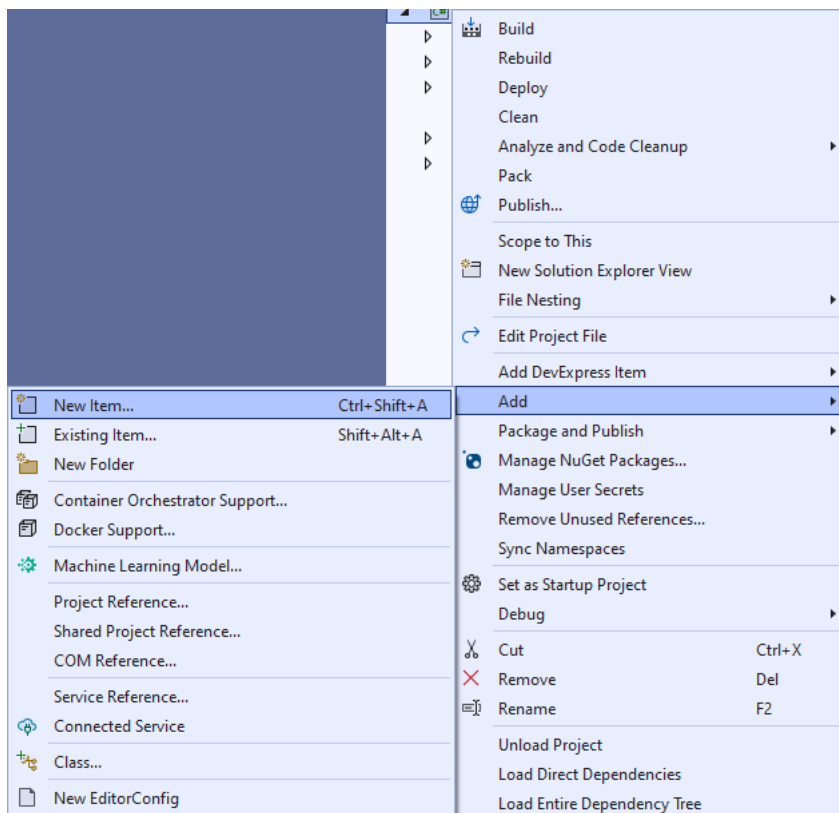
Then in the **NuGet Package Manager** from the **Browse** tab search for **Comentsys.Toolkit.WindowsAppSdk** and then select **Comentsys.Toolkit.WindowsAppSdk** by **Comentsys** as indicated and select **Install**



This will add the package for **Comentsys.Toolkit.WindowsAppSdk** to your **Project**. If you get the **Preview Changes** screen saying **Visual Studio is about to make changes to this solution. Click OK to proceed with the changes listed below.** You can read the message and then select **OK** to **Install** the package,, then you can close the **tab** for **Nuget: HighOrLow** by selecting the **x** next to it.

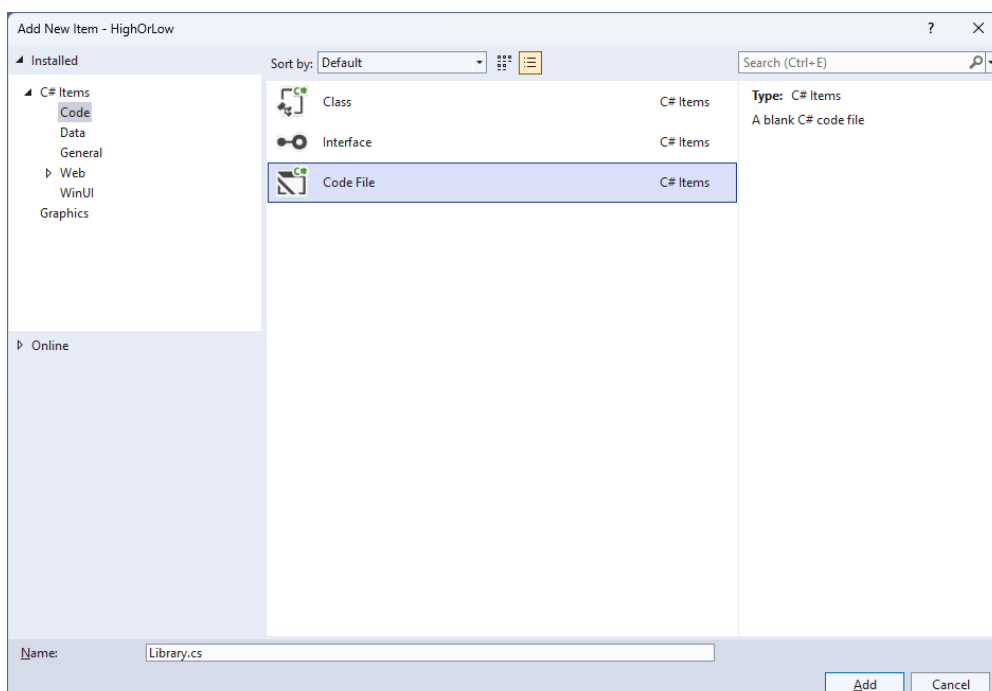
Step 4

Then in **Visual Studio** within **Solution Explorer** for the **Solution**, right click on the **Project** shown below the **Solution** and then select **Add** then **New Item...**



Step 5

Then in **Add New Item** from the **C# Items** list, select **Code** and then select **Code File** from the list next to this, then type in the name of *Library.cs* and then **Click** on **Add**.



Step 6

You will now be in the **View** for the **Code** of *Library.cs*, within this first type the following **Code**:

```
using Comentsys.Toolkit.WindowsAppSdk;
using Microsoft.UI;
using Microsoft.UI.Xaml;
using Microsoft.UI.Xaml.Controls;
using Microsoft.UI.Xaml.Input;
using Microsoft.UI.Xaml.Media;
using System;
using System.Collections.Generic;
using System.Linq;

public class Library
{
    private const string title = "High or Low";
    private const string higher = "Higher";
    private const string lower = "Lower";
    private const int minimum = 1;
    private const int maximum = 52;
    private const int suit = 13;
    private const int total = 4;

    private readonly Random _random = new((int)DateTime.UtcNow.Ticks);

    private StackPanel _panel = new();
    private Dialog _dialog;

    private List<int> _values = new();
    private bool _over = false;
    private int _card = 1;

    // Choose, Get Card, Set Card & Get Confirm

    // Play

    // Add Card, Layout & New
}
```

The **Class** that has been defined in so far *Library.cs* has **using** for the package of **Comentsys.Toolkit.WindowsAppSdk** amongst others needed. There are also some **const** and **readonly** values for parts of the game including **Random** which will be used to select randomised numbers as well as other values and elements to be used in the game.

Step 7

Still in the **Class** for *Library.cs* after the **Comment** of `// Choose, Get Card, Set Card & Get Confirm` type the following **Methods**:

```
private List<int> Choose(int minimum, int maximum, int total) =>
    Enumerable.Range(minimum, maximum)
        .OrderBy(r => _random.Next(minimum, maximum))
        .Take(total).ToList();

public Card GetCard(int value, string name = "") => new()
{
    Back = new SolidColorBrush(Colors.Red),
    Margin = new Thickness(10),
    Value = value,
    Name = name
};

public void SetCard(int value, string name) =>
    ((Card)_panel.FindName(name)).Value = value;

private async Task<bool> GetConfirmAsync(int card)
{
    var confirm = new StackPanel()
    {
        Orientation = Orientation.Vertical
    };
    confirm.Children.Add(new TextBlock()
    {
        HorizontalTextAlignment = TextAlignment.Center,
        Text = "Is Next Card?"
    });
    confirm.Children.Add(new Viewbox()
    {
        Height = 150,
        Child = GetCard(_values[card])
    });
    return await _dialog.ConfirmAsync(confirm, higher, lower);
}
```

Choose is used to select a list of randomised numbers, **GetCard** is used to get a **Card** which is a **Control** used to display a playing card, **SetCard** will update the **Value** of a **Card** and **GetConfirmAsync** will be used to get a confirmation dialog for the choice of whether the next playing card is *Higher* or *Lower* than the previous playing card.

Step 8

While still in the **Class** for *Library.cs* after the **Comment** of `// Play` type in the following **Method**:

```
private async void Play(Card card)
{
    if (card.Name == $"{_values[_card]}")
    {
        int next = _card;
        int prev = _card - 1;
        var isHigher = await GetConfirmAsync(prev);
        var source = _values[prev] % suit;
        source = source == 0 ? suit : source;
        var target = _values[next] % suit;
        target = target == 0 ? suit : target;
        if ((isHigher == true && target > source) ||
            (isHigher == false && target < source))
        {
            SetCard(_values[next], $"{_values[next]}");
            _card++;
            if (_card == _values.Count)
            {
                _dialog.Show("Congratulations - You Win!");
                _over = true;
            }
        }
        else
        {
            var content = new StackPanel()
            {
                Orientation = Orientation.Vertical
            };
            content.Children.Add(new TextBlock()
            {
                HorizontalTextAlignment = TextAlignment.Center,
                Text = $"Incorrect - Next Card was {(isHigher ? higher : lower)}!"
            });
            content.Children.Add(new Viewbox()
            {
                Height = 150,
                Child = GetCard(_values[_card])
            });
            _dialog.Show(content);
            _over = true;
        }
    }
    else
        _dialog.Show("Select Next Card!");
}
```

Play is used to check if the card selected is either higher or lower than the previous playing card, should all the playing cards be guessed correctly then it goes onto the next one or if not then it will show what the correct playing card would have been and the game is over, if all are guessed correctly that wins the game.

Step 9

While still in the **Class** for *Library.cs* after the **Comment** of **// Add Card, Layout & New** type in the following **Methods**:

```
public void AddCard(StackPanel panel, string name, int value)
{
    var card = GetCard(value, name);
    card.Tapped += (object sender, TappedRoutedEventArgs e) =>
    {
        if (_over)
            _dialog.Show("Game Over!");
        else
            Play((Card)sender);
    };
    panel.Children.Add(card);
}

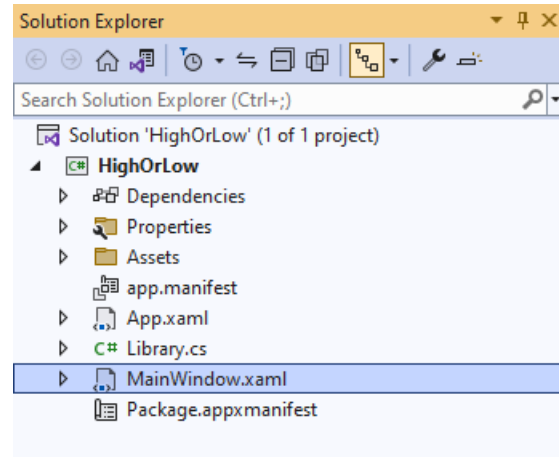
private void Layout(StackPanel panel)
{
    panel.Children.Clear();
    for (int index = 0; index < total; index++)
    {
        AddCard(panel, $"{_values[index]}", index == 0 ? _values[index] : 0);
    }
}

public void New(StackPanel panel)
{
    _card = 1;
    _over = false;
    _panel = panel;
    _dialog = new Dialog(panel.XamlRoot, title);
    _values = Choose(minimum, maximum, total);
    Layout(panel);
}
```

AddCard is used to add the playing cards for the game and will check if the game is over or use **Play** for the next playing card to be picked. **Layout** is used to create the layout for the game using **AddCard** and **New** is used to start a game.

Step 10

Then from **Solution Explorer** for the **Solution** double-click on **MainWindow.xaml** to see the **XAML** for the **Main Window**.



Step 11

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 12

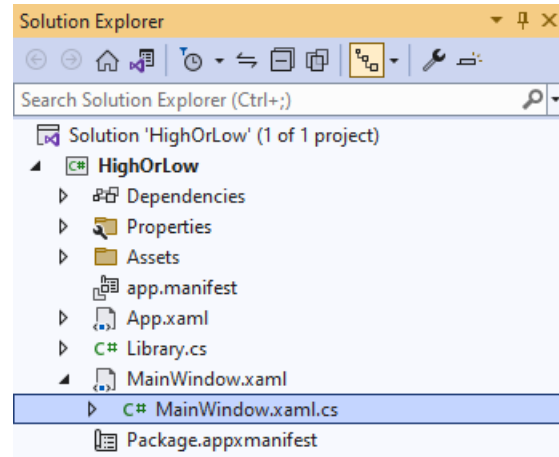
While still in the **XAML** for **MainWindow.xaml** above **</Window>**, type in the following **XAML**:

```
<Grid>
    <Viewbox>
        <StackPanel Margin="50" Name="Display" Orientation="Horizontal"
        HorizontalAlignment="Center" VerticalAlignment="Center" Loaded="New"/>
    </Viewbox>
    <CommandBar VerticalAlignment="Bottom">
        <AppBarButton Icon="Page2" Label="New" Click="New"/>
    </CommandBar>
</Grid>
```

This **XAML** contains a **Grid** with a **Viewbox** which will scale a **StackPanel**. It has a **Loaded** event handler for **New** which is also shared by the **AppBarButton**.

Step 13

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 14

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 15

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

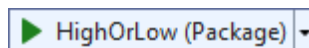
```
private readonly Library _library = new();

private void New(object sender, RoutedEventArgs e) =>
    _library.New(Display);
```

Here an **Instance** of the **Class** of **Library** is created then below this is the **Method** of **New** that will be used with **Event Handler** from the **XAML**, this **Method** uses Arrow Syntax with the **=>** for an Expression Body which is useful when a **Method** only has one line.

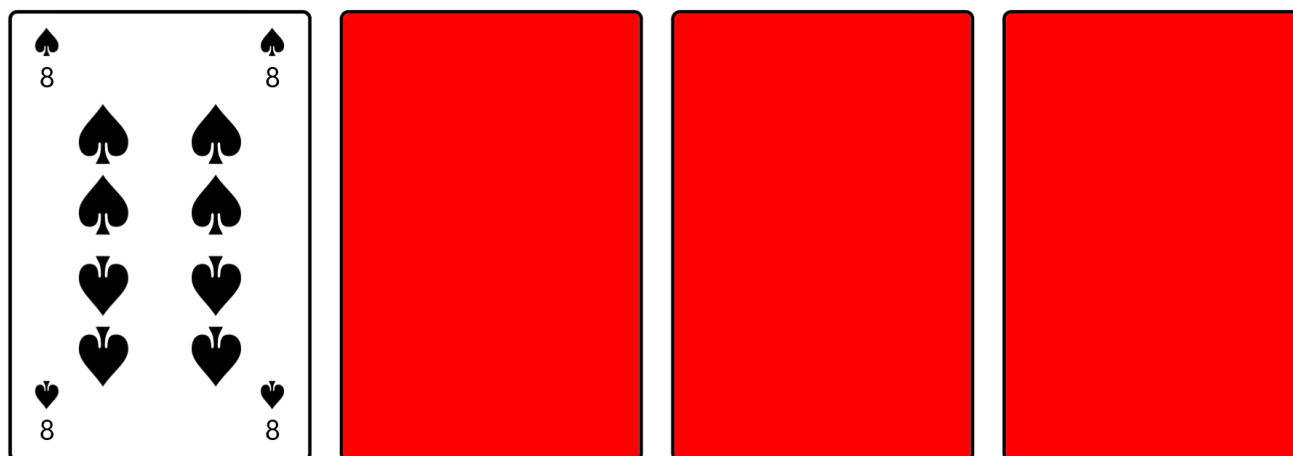
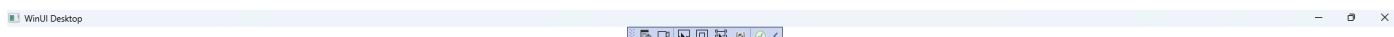
Step 16

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



Step 17

Once running you can then select the next playing card and then you will be asked if it is **Higher** or **Lower** if you guess correctly you can proceed to the next card until you guess them all, if you guess any incorrectly you will lose, or you can select *New* to start a new game.



Step 18

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!

