

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

Windows App SDK

Order Game





# Order Game

**Order Game** shows how you can create a simple game where the objective is to arrange **Squares** in the

correct sequence in the quickest time possible 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. | 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 *OrderGame*, then select a Location and then select **Create** to start a new **Solution**. | **Graphical user interface, text, application, email  Description automatically generated** |

## 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…**

Graphical user interface, application

Description automatically generated

## 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**

Graphical user interface, text, application, email

Description automatically generated

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: OrderGame** by selecting the **x** next to it.

## Step 4

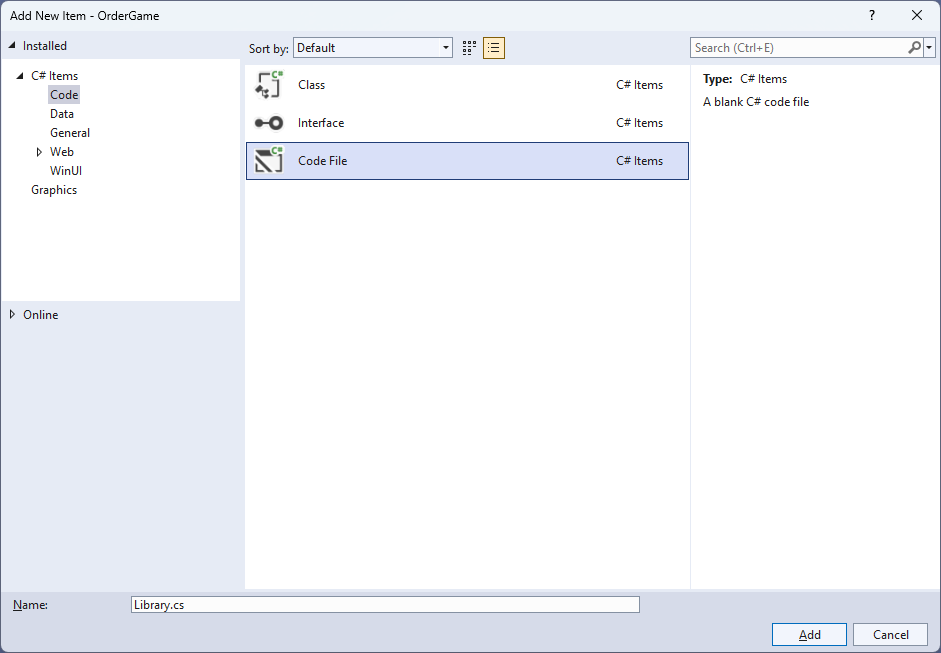
Then in **Visual Studio** within **Solution** **Explorer** for the **Solution**, right click on the **Project** shown below the **Solution** and then select **Add** then **New Item…**

Table

Description automatically generated with low confidence

## 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.Xaml;

using Microsoft.UI.Xaml.Controls;

using System;

using System.Collections.Generic;

using System.Collections.ObjectModel;

using System.Linq;

public class Library

{

private const string title = "Order Game";

private const int size = 6;

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

private readonly ObservableCollection<int> \_values = new();

private Dialog \_dialog;

private GridView \_view;

private DateTime \_start;

private List<int> Choose(int minimum, int maximum, int total) =>

Enumerable.Range(minimum, maximum)

.OrderBy(r => \_random.Next(minimum, maximum))

.Take(total).ToList();

private void Completed()

{

if (\_values.OrderBy(o => o).SequenceEqual(\_values))

{

TimeSpan duration = (DateTime.UtcNow - \_start).Duration();

\_dialog.Show($"Completed in {duration:hh\\:mm\\:ss}", title);

\_view.IsEnabled = false;

}

}

// Layout & New

}

**Class** defined so far *Library.cs* has **using** for packageof **Comentsys.Toolkit.WindowsAppSdk** and others. It also has **Constants** to represent things needed in the game and there are **Variables** to keep track of values used in the game along with **Methods** for **Choose** to pick the random sequence of numbers to be put in **Order** and **Completed** which will determine if the game has been finished and display a **Dialog** showing the time it took do to so.

## Step 7

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

private void Layout(Grid grid)

{

grid.Children.Clear();

\_view = new()

{

ItemsPanel = grid.Resources[nameof(ItemsPanelTemplate)]

as ItemsPanelTemplate,

ItemTemplate = grid.Resources[nameof(DataTemplate)]

as DataTemplate,

SelectionMode = ListViewSelectionMode.Single,

CanReorderItems = true,

ItemsSource = \_values,

CanDragItems = true,

AllowDrop = true,

IsEnabled = true,

CanDrag = true,

};

\_view.DragItemsCompleted += (ListViewBase sender,

DragItemsCompletedEventArgs args) =>

Completed();

grid.Children.Add(\_view);

}

public void New(Grid grid)

{

\_dialog = new Dialog(grid.XamlRoot, title);

\_values.Clear();

\_start = DateTime.UtcNow;

var values = Choose(1, size \* size, size \* size);

foreach (var value in values)

{

\_values.Add(value);

}

Layout(grid);

}

**Layout** will create the layout for the game which comprises of an **ItemsPanel** which will use **Templates** to create the look-and-feel for the elements of the game and will check if the game is finished using **Completed** when the elements are reordered and **New** will setup and start a new game.

## Step 8

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

## Step 9

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 10

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

xmlns:ui="using:Comentsys.Toolkit.WindowsAppSdk"

The **XAML** for **<Window>** should then look as follows:

<Window

xmlns:ui="using:Comentsys.Toolkit.WindowsAppSdk"

x:Class="OrderGame.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:local="using:OrderGame"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

mc:Ignorable="d">

## Step 11

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

<Grid>

<Viewbox>

<Grid Margin="50" Name="Display"

HorizontalAlignment="Center"

VerticalAlignment="Center" Loaded="New">

<Grid.Resources>

<DataTemplate x:Name="DataTemplate">

<ui:Piece Value="{Binding}" IsSquare="True"

Fill="Black" Foreground="White" />

</DataTemplate>

<ItemsPanelTemplate x:Name="ItemsPanelTemplate">

<ItemsWrapGrid Orientation="Horizontal"

MaximumRowsOrColumns="6"/>

</ItemsPanelTemplate>

</Grid.Resources>

</Grid>

</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 **Grid**. It has a **Loaded** event handler for **New** which is also shared by the **AppBarButton** and the **Grid** also has **Resources** which defines the **Templates**.

## Step 12

|  |  |
| --- | --- |
| 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 13

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 14

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);

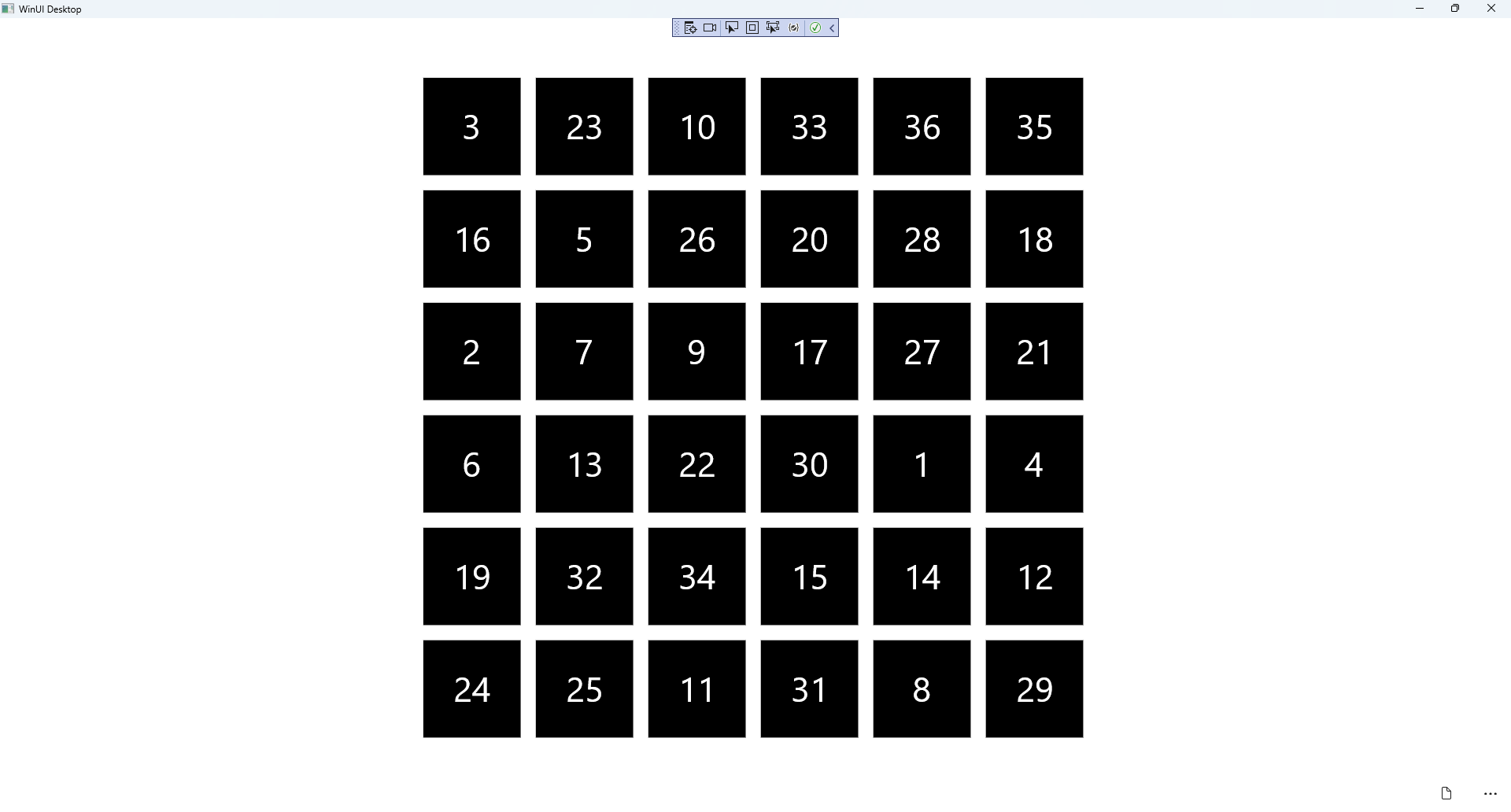
Here an **Instance** of the **Class** of **Library** is created then below this is the **Method** of **New** that will be used with **Event Handler** from the **XAML**, this **Method** uses Arrow Syntax with the **=>** for an Expression Bodywhich is useful when a **Method** only has one line.

## Step 15

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

## Step 16

Once running you win by putting all the numbers in **Order** from *1* to *36* from left to right as quickly as possible by dragging and moving them into the correct **Order**, or you can select *New*to start a new game.

****

## Step 17

|  |  |
| --- | --- |
| 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)! |  |