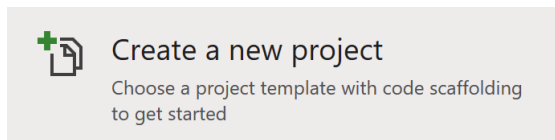


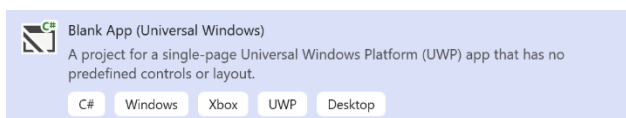
# Universal Windows Platform – Order Game

**Order Game** shows how to create a simple game where the objective is to re-arrange numbered squares in the correct sequence in the quickest time possible

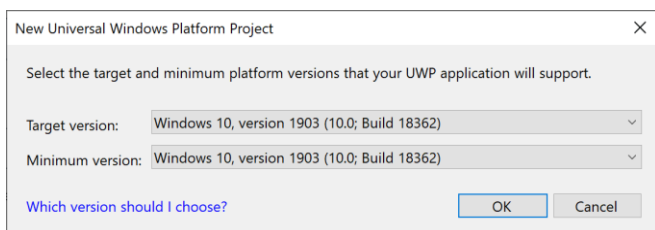
## Step 1



Follow **Setup and Start** on how to Install and/or Get Started with **Visual Studio 2019** if not already or in **Windows 10** choose **Start**, find and select **Visual Studio 2019** then from the **Get started** screen select **Create a new project**



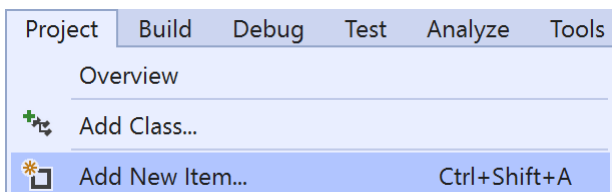
Then choose **Blank App (Universal Windows)** and select **Next** and then in **Configure your new project** enter the **Project name** as **OrderGame** and select **Create**



Finally, in **New Universal Windows Platform Project** pick the **Target version** and **Minimum version** to be at least **Windows 10, version 1903 (10.0; Build 18362)** and then select **OK**

Target Version will control the most recent features of Windows 10 your application can use. To make sure you always have the most recent version, check for any Notifications or Updates in Visual Studio 2019

## Step 2



Choose **Project** then **Add New Item...** from the **Menu** in **Visual Studio 2019**

## Step 3



Then choose **Code File** from **Add New Item** in **Visual Studio 2019**, enter the **Name** as **Library.cs** and select **Add**

# Universal Windows Platform – Order Game

## Step 4

In the **Code** View of **Library.cs** will be displayed and in this the following should be entered:

```
using System;
using System.Collections.Generic;
using System.Collections.ObjectModel;
using System.Linq;
using Windows.UI.Popups;
using Windows.UI.Xaml.Controls;

public class Library
{
    private const string title = "Order Game";
    private const int size = 6;

    private DateTime _timer;
    private ObservableCollection<int> _list =
        new ObservableCollection<int>();
    private Random _random = new Random((int)DateTime.Now.Ticks);
}
```

There are **using** statements to include necessary functionality. **\_list** is a **ObservableCollection<int>** represents the elements of the game and **Random** is used to create the numbers for the order of the elements

Then below the **private Random \_random = new Random((int)DateTime.Now.Ticks);** line the following **methods** should be entered:

```
private void Show(string content, string title)
{
    _ = new MessageDialog(content, title).ShowAsync();
}

private List<int> Choose(int start, int total)
{
    return Enumerable.Range(start, total)
        .OrderBy(r => _random.Next(start, total)).ToList();
}
```

**Show** method is used to display a **MessageDialog** and **Choose** is used to pick randomised numbers to order the elements in the game

## Universal Windows Platform – Order Game

Then below the **private List<int> Choose(...)** { ... } **method** the following **method** should be entered:

```
private void Layout(GridView grid)
{
    int index = 0;
    _list.Clear();
    grid.IsEnabled = true;
    grid.ItemsSource = null;
    _timer = DateTime.UtcNow;
    List<int> numbers = Choose(1, size * size);
    while (index < numbers.Count)
    {
        _list.Add(numbers[index]);
        index++;
    }
    grid.ItemsSource = _list;
}
```

Layout method is used to create the look-and-feel of the game and the Grid and call Add method of \_list

Finally after the **private void Layout(...)** { ... } **method** the following **public methods** should be entered:

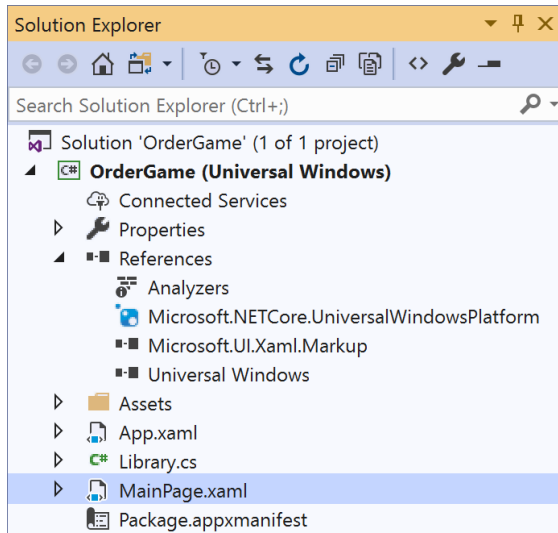
```
public void New(GridView grid)
{
    Layout(grid);
}

public void Order(GridView grid)
{
    if (_list.OrderBy(o => o).SequenceEqual(_list))
    {
        TimeSpan duration = (DateTime.UtcNow - _timer).Duration();
        Show($"Completed in {duration:hh\\:mm\\:ss}", title);
        grid.IsEnabled = false;
    }
}
```

New method will setup and start playing the game by calling Layout method and Order method will check to see if the elements are in order then display a message to say how long the game took to complete

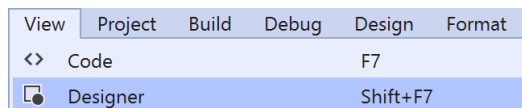
# Universal Windows Platform – Order Game

## Step 5



In the **Solution Explorer** of **Visual Studio 2019** select **MainPage.xaml**

## Step 6



Choose **View** then **Designer** from the **Menu** in **Visual Studio 2019**

# Universal Windows Platform – Order Game

## Step 7

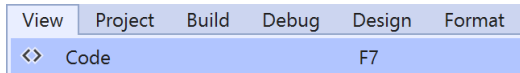
In the **Design** View and **XAML** View of **Visual Studio 2019** will be displayed, and in this between the **Grid** and **/Grid** elements enter the following **XAML**:

```
<Viewbox>
  <GridView Margin="50" Name="Display"
  HorizontalAlignment="Center" VerticalAlignment="Center"
  SelectionMode="Single" CanDrag="True" CanDragItems="True"
  AllowDrop="True" CanReorderItems="True"
  DragItemsCompleted="Display_DragItemsCompleted">
    <GridView.ItemTemplate>
      <DataTemplate>
        <Grid Width="50" Height="50"
        Background="{ThemeResource SystemAccentColor}">
          <TextBlock Text="{Binding}" FontSize="24"
          HorizontalAlignment="Center"
          VerticalAlignment="Center"
          Foreground="White"/>
        </Grid>
      </DataTemplate>
    </GridView.ItemTemplate>
    <GridView.ItemsPanel>
      <ItemsPanelTemplate>
        <ItemsWrapGrid Orientation="Horizontal"
        MaximumRowsOrColumns="6"/>
      </ItemsPanelTemplate>
    </GridView.ItemsPanel>
  </GridView>
</Viewbox>
<CommandBar VerticalAlignment="Bottom">
  <AppBarButton Icon="Page2" Label="New" Click="New_Click"/>
</CommandBar>
```

The first block of XAML the main user interface features a ViewBox containing a GridView that will be the look-and-feel of the game which has the ability to reorder items and has a DataTemplate which will represent the items themselves and the ItemsPanel of the GridView is a ItemsWrapGrid to make the items display correctly and the second block of XAML is the CommandBar which contains New to setup and start the game

# Universal Windows Platform – Order Game

## Step 8



Choose **View** then **Code** from the **Menu** in **Visual Studio 2019**

## Step 9

Once in the **Code** View, below the end of `public MainPage() { ... }` the following Code should be entered:

```
Library library = new Library();

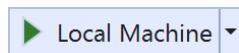
private void New_Click(object sender, RoutedEventArgs e)
{
    library.New(Display);
}

private void Display_DragItemsCompleted(ListViewBase sender,
    DragItemsCompletedEventArgs args)
{
    library.Order(Display);
}
```

Below the MainPage method an instance of the `Library` class is created. In the `New_Click(...)` Event handler will setup and start the game using the `New` method in the `Library` class and the `Display_DragItemsCompleted` event handler will call the `Order` method in the `Library` class

# Universal Windows Platform – Order Game

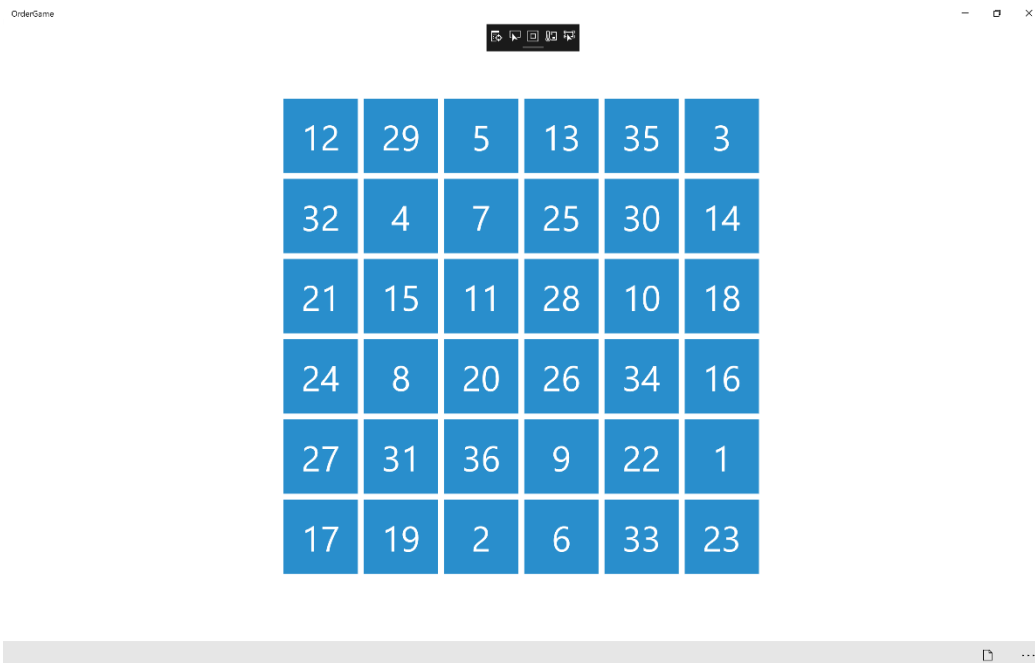
## Step 10



That completes the **Universal Windows Platform** Application, in **Visual Studio 2019** select **Local Machine** to run the Application

## Step 11

Once the Application is running use **New** to start playing, you can 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 positions.



## Step 12



To Exit the Application, select the **Close** button in the top right of the Application