**Fruit Game** shows how to create a simple game to match three randomly selected **fruit**

## 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** |
| A screenshot of a cell phone  Description automatically generated | Then choose **Blank App (Universal Windows)** and select **Next** and then in **Configure your new project** enter the **Project name** as **FruitGame** and select **Create** |
| A screenshot of a social media post  Description automatically generated | 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

|  |  |
| --- | --- |
| A screenshot of a cell phone  Description automatically generated | Choose **Project** then **Add New Item...** from the **Menu** in **Visual Studio 2019** |

## Step 3

|  |  |
| --- | --- |
| A close up of a logo  Description automatically generated | Then choose **Code File** from **Add New Item** in **Visual Studio 2019**, enter the **Name** as **Library.cs** and select **Add** |

## Step 4

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

|  |
| --- |
| using System;  using System.Linq;  using Windows.UI.Xaml;  using Windows.UI.Xaml.Controls;  using Windows.UI.Xaml.Media;  public class Library  {  private const string title = "Fruit Game";  private const int size = 3;  private readonly string[] values =  {  "\U0001F34E", // Apple  "\U0001F347", // Grapes  "\U0001F34B", // Lemon  "\U0001F352", // Cherry  "\U0001F34C", // Banana  "\U0001F349", // Melon  "\U0001F34A", // Orange  "\U0001F514", // Bell  };  private int \_spins = 0;  private int[] \_board = new int[size];  private Random \_random = new Random((int)DateTime.Now.Ticks);  } |

There are using statements to include necessary functionality. values is a string[] represents the different fruit that will be shown and Random is used to randomly pick these for the game

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

|  |
| --- |
| private Viewbox Fruit(int type)  {  TextBlock textblock = new TextBlock()  {  Text = values[type],  IsColorFontEnabled = true,  Margin = new Thickness(2),  TextLineBounds = TextLineBounds.Tight,  FontFamily = new FontFamily("Segoe UI Emoji"),  HorizontalTextAlignment = TextAlignment.Center  };  return new Viewbox()  {  Child = textblock  };  } |

Fruit **method** is used to display a the **fruit** using a TextBlock and **emoji** characters

Next below the **private Viewbox Fruit(...) { ... }** **method** the following **method** should be entered:

|  |
| --- |
| private async void Show(string content, int type)  {  ContentDialog dialog = new ContentDialog()  {  Title = $"{content} {title}",  PrimaryButtonText = "OK",  Content = new Viewbox()  {  Width = 150,  Height = 150,  Child = Fruit(type)  }  };  await dialog.ShowAsync();  } |

Show **method** is used to display a simple ContentDialog and uses the Fruit **method** as part of the Content

After the **private async void Show(...) { ... }** **method** the following **method** should be entered:

|  |
| --- |
| private void Add(Grid grid, int column, int type)  {  Viewbox viewbox = new Viewbox()  {  Width = 100,  Height = 100,  Child = Fruit(type),  Stretch = Stretch.Uniform  };  viewbox.SetValue(Grid.ColumnProperty, column);  grid.Children.Add(viewbox)  } |

Add **method** is used to create a Viewbox and uses the Fruit **method** to set the Child of this

Then after the **private void Add(...) { ... }** **method** the following **method** should be entered:

|  |
| --- |
| private void Layout(Grid grid)  {  grid.Children.Clear();  grid.ColumnDefinitions.Clear();  // Setup Grid  for (int column = 0; (column < size); column++)  {  grid.ColumnDefinitions.Add(new ColumnDefinition());  \_board[column] = \_random.Next(0, values.Count());  Add(grid, column, \_board[column]);  }  } |

Layout **method** is used to setup the look-and-feel of the game using a **Grid** and calling the Add **method**

Next after the **private void Add(...) { ... }** **method** the following **method** should be entered:

|  |
| --- |
| private void Winner()  {  \_spins++;  if (\_board.All(item => item == \_board.First()))  {  Show($"Spin {\_spins} matched", \_board.First());  \_spins = 0;  }  } |

Winner **method** is used to check if all the **fruit** match and if they do call Show to display a message

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

|  |
| --- |
| public void New(Grid grid)  {  Layout(grid);  Winner();  } |

New **method** will setup and start playing the game by calling Layout and Winner methods

## Step 5

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

## Step 6

|  |  |
| --- | --- |
| A screenshot of a cell phone  Description automatically generated | Choose **View** then **Designer** from the **Menu** in **Visual Studio 2019** |

## 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>  <Grid Name="Display" Margin="50"  HorizontalAlignment="Center"  VerticalAlignment="Center"/>  </Viewbox>  <CommandBar VerticalAlignment="Bottom">  <AppBarButton Icon="Page2" Label="New" Click="New\_Click"/>  </CommandBar> |

The first block of **XAML** the main user interface features a **Grid** to represent the game and the second block of **XAML** is the **CommandBar** which contains **New** to setup and start the 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);  } |

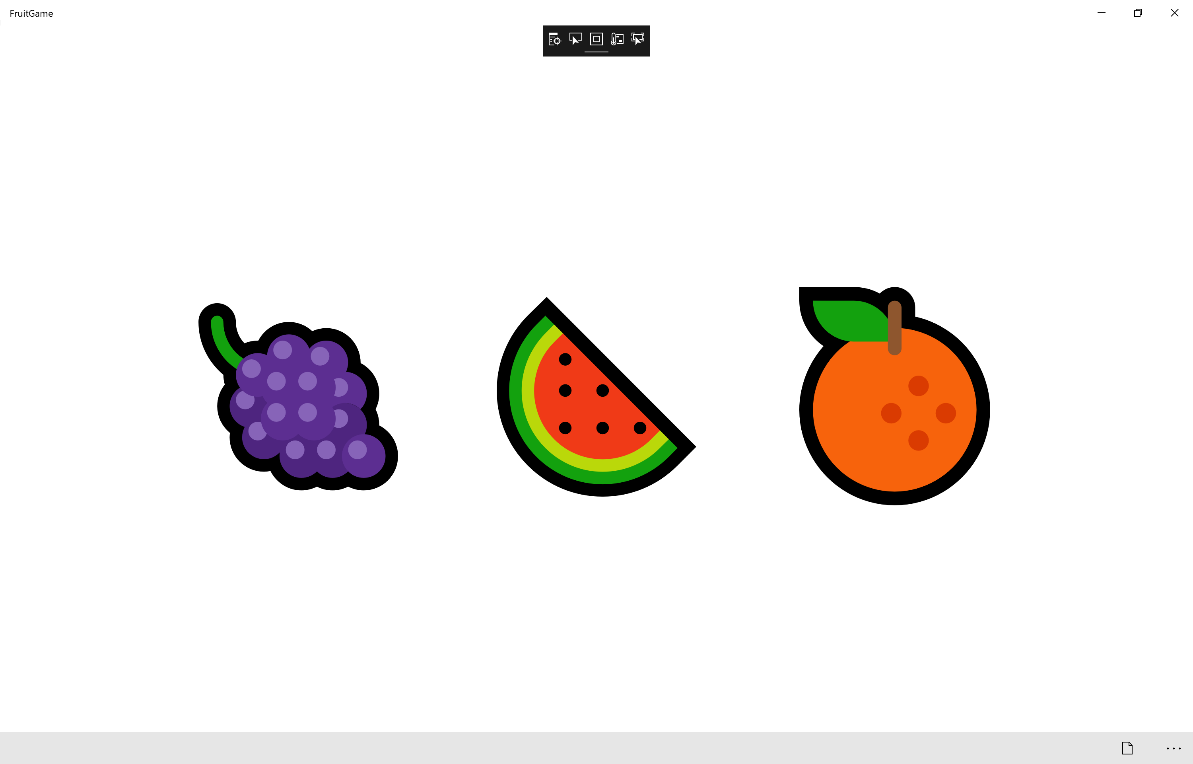
Below the **MainPage** method an instance of the Library **class** is created. In the New\_Click(...) **Event** handler will setup and play the game using the New **method** in the Library **class**

## 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 and you can win by getting three identical **fruit**



## Step 12

|  |  |
| --- | --- |
| A picture containing object  Description automatically generated | To Exit the Application, select the **Close** button in the top right of the Application |