

Programming Logic - Beginning

152-101

Unit 4 – Welcome Screen Lab

5 points

Name	
Score	/ 5
Update Value	
Make all corrections and resubmit to earn update points	

Overview

Many apps include a Welcome screen. To make our app more professional we'll add these to our Bowling program.

1. Use your Bowling program from Lab 1 or download a copy of the Bowling starter program from OneDrive or my website.
2. Note that in addition to the Bowling solution folder, this folder contains a generic welcome screen.
3. Open the Bowling project, open the form in Design View and then open another tab in Code View.

Creating a Welcome Screen

To make the welcome screen accessible to our app, we first need to create a new window. For your assignments (not this lab), you should use your own welcome screen. The welcome screen can be largely the same for each project, but you will need to build it each time.

4. Right click on the project name in the Solution Explorer and select Add..window and name the window Welcome.xaml or something else that is appropriate to the project.
5. In the xaml/designer you will add the welcome message, launch button, and exit button. Ensure that you name the buttons, and then create event handlers for them.
6. Move to the WelcomeScreen.xaml.cs file and add long variables for initialTicks and finalTicks to handle the timer methods

```
public partial class WelcomeScreen : Window
{
    long initialTicks, finalTicks;

    public WelcomeScreen()
    {
        InitializeComponent();
    }
}
```

- a.
7. Inside the btnExit event handler you will have:

```
this.Close();
Application.Current.MainWindow.Close();
```
- a.
8. For the Launch event handler, you will have the following:

```
Application.Current.MainWindow.Show();
this.Close();
```
- a.
9. In the WelcomeScreen.xaml.cs file you will need to add a couple of methods to activate the timer that we are using. Add these methods directly below the public WelcomeScreen() method

```
private void Timer()
{
    initialTicks = DateTime.Now.Ticks;
    System.Windows.Threading.DispatcherTimer dispatcherTimer = new System.Windows.Threading.DispatcherTimer();
    dispatcherTimer.Tick += new EventHandler(dispatcherTimer_Tick);
    dispatcherTimer.Interval = new TimeSpan(0, 0, 1);
    dispatcherTimer.Start();
}

```

a.

```
private void dispatcherTimer_Tick(object sender, EventArgs e)
{
    //update the lable which displays the current second
    lblTimer.Content = DateTime.Now.Second;

    // Forcing the CommandManager to raise the RequerySuggested event
    CommandManager.InvalidateRequerySuggested();

    finalTicks = DateTime.Now.Ticks;
    long elapsedTicks = finalTicks - initialTicks;
    if (elapsedTicks > 50000000) //10000000 ticks per second
    {
        Application.Current.MainWindow.Show();
        this.Close();
    }
}

```

b.

10. The last item in the WelcomeScreen.xaml.cs file is to call the Timer() method in the WelcomeScreen Constructor.

```
public WelcomeScreen()
{
    InitializeComponent();
    Timer();
}

```

a.

11. Now move to the MainWindow.xaml.cs file

- a. In the MainWindow constructor add the following:

```
public MainWindow()
{
    InitializeComponent();

    //open the welcome screen
    WelcomeScreen ws = new WelcomeScreen();
    ws.Show();

    //hide the mainwindow until later on
    this.Hide();
}

```

12. That should be all that you need for the Welcome Screen to work.

Note that this welcome screen includes two buttons. The Start Program button allows the user to close the welcome screen and start the main program. The Exit button allows the user to close the welcome screen **without** running the main program in case they accidentally launched this program.

13. Add a label at the top of the form and ensure it has your name on it. You should customize your welcome screen for each program.
14. In order to link our welcome screen to the main form, they must be part of the same *namespace*. Remember from Unit 2 that namespaces contain classes. In order for classes to communicate with each other, they must be in the same namespace. You should not have to change anything in this regard, but verify that both windows have the same namespace.
15. If you do need to change the namespace to be correct....After pasting (retyping), a red box appears below the new namespace name `namespace Bowling`. Hover over this red box with your mouse and click the menu button that appears. Choose the first rename option. This ensures the old namespace name is replaced everywhere (designer.cs and resx) it may have appeared.
16. Now, we're ready to link the welcome screen to our main form. In C# (unlike some other languages like VB), one form in the project must serve as the *master form* (my name for it). All other forms in the project are launched from the master form and those forms ultimately return control to the master form.
17. Open wndBowling code view.
18. Run the program. Click the Start button. The main program should launch.
 - a. A 5 second timer should run and then launch the program automatically. Alternatively, if you select the Launch button the program will begin sooner.
19. If you are in the lab with the instructor present, demonstrate the following (in order). If the instructor is not available, zip and email your copy of the Bowling application.

- Run the program. Ensure your name is on the Welcome Screen.
- Wait—demonstrate the Timer.
- Close the program—run it again. Demonstrate the Start button.
- Close the program—run it again. Click Exit in the Welcome Screen

Optional:

- Demonstrate the flashing bowling picture.