

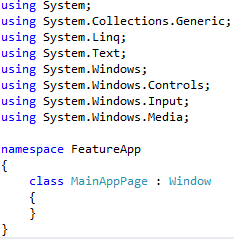
Systems Development: Object Oriented Programming

(H171 35)

WPF Application

Step 2 – creating the main application page

* Step 2 is creating the applications main page, which will be launched from the initial “welcome” screen created in step 1. We are going to take a different approach here and create the applications main page using C sharp code rather than dragging and dropping elements or editing XAML (both of which are perfectly acceptable alternatives). This method is being used to demonstrate how you can control the creation of the entire page through C sharp code, and it also is useful if you have a collection of pages and you may want to create different layouts rather than always navigate to the same fixed layout.
* Add a new class called MainAppPage.cs
* As we are going to be using this class to define our “Window” in code rather than XAML, then we inherit from class “Window”. Remember to add the additional “using” statements as well.



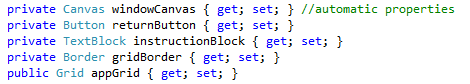
* I would like this main app window to have a layout similar to the following:

Use arrow keys to move around the screen

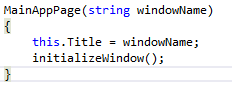
10 x 10 grid

Return to start page

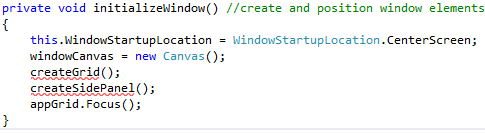
* This involves using the objects:
  + Canvas
  + Button
  + Textblock
  + Grid
  + Border
* Which means each of these being a member of the page MainAppPage.cs i.e. MainAppPage.cs “**has a”** Canvas, Button, Textblock, Grid and Border



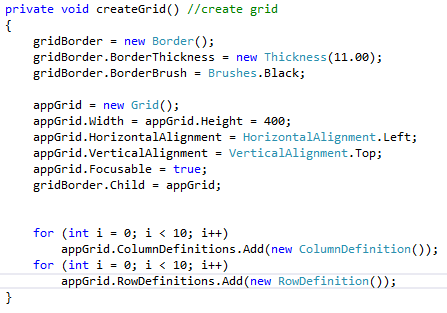
* As with all classes, we need to have a constructor that is called as soon as an instance of the class is created. Our constructor will have a single parameter, which will be used to set the title of the window, and will call the initializeWindow() method.



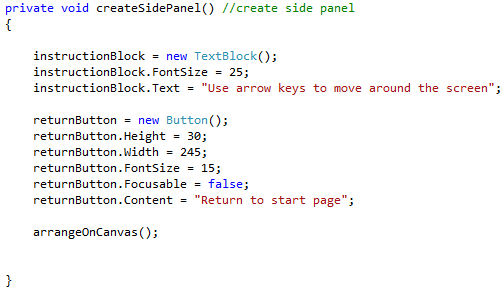
* We now need to write the initializeWindow() method which will create and position window elements.
  + Firstly, we will create the Canvas layout which will be the main container for our window elements
  + then we will call a separate method where we will have the instructions on how to create the grid portion of the main app page
  + then another separate method for creating the side-bar portion of the main app page.



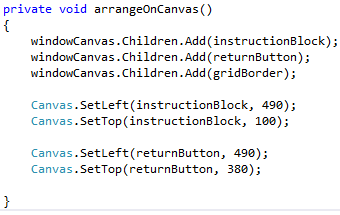
* Create the grid portion of the main app page
  + Create a border and set the border properties (this will surround the grid)
  + Create a 400x400 grid, aligned to the top left of its containing object
  + Set the grid as the child element to draw the border around
  + Add 10 columns and 10 rows to the grid



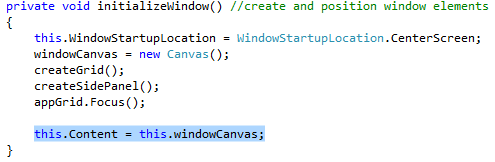
* Create the side-bar portion of the main app page
  + Create the textblock containing the instructions
  + Create the “return to start page” button
  + Call method to position all elements on the canvas



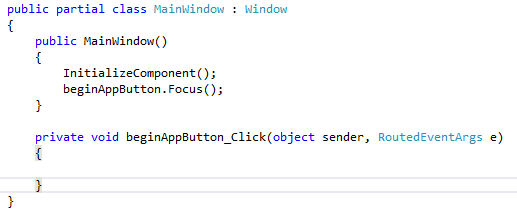
* + Add and position each element on the canvas



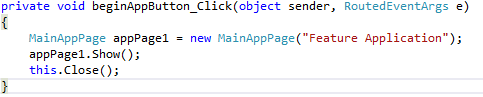
* Now that we have created elements and positioned them on a canvas we need to set this newly created canvas to be the “Content” of the “Window”. This is done back in initializeWindow()



* Now we will go back to the initial page (MainWindow.xaml) and write code for the “Begin” button. Double-click on this button to create an event handler in the MainWindow.xaml.cs file. (This event will launch this MainAppPage we have just created):



* We need to create an instance of the MainAppPage which we want to open/launch when this button is clicked
* Then display the new MainAppPage created
* Close the current initial page we have open (MainWindow.xaml)



**STOP AT THIS POINT AND LAUNCH THE APP!**

* The initial page should launch and you should be able to use the button to navigate to the second MainAppPage. They should look similar to the following:

