Lab 4:
I'm Out Of Othello Puns

Due March 22 by the start of lecture.

Overview

In this lab, you will complete the Othello WPF example from lecture, adding UI elements to support new options and tweaking a few other elements to make the application better overall. Note: you **must** use Visual Studio 2017 on Windows for this assignment.

Getting Started

Use Git to create a new branch of the lecture notes repository: `git checkout -b lab4`. (Alternatively, download the lecture notes directly from GitHub as a ZIP file.) Open the `Othello.sln` solution in **Visual Studio 2017**.

UI Modifications

Make the following modifications to the base Othello application:

1. Change the way that the player's tokens are drawn on the board. This doesn't have to be anything amazing, you just need to change something about the look of the tokens. Ideas: draw the circles with a gradient instead of a solid color; draw them with a texture; draw a different shape.
2. In the status bar in the MainWindow, change the **Score:** display so that instead of directly displaying the `Value` of the `OthelloBoard`, it displays a string describing who is winning, as in "**Score: white is winning by 3**" instead of "Score: -3".
   (a) To do this, you **must** write a new `IValueConverter` class that converts from an integer (bound to the `OthelloModelView`'s `BoardValue` property) to a string of the given format.
   (b) You will add your new Converter as a Static Resource of the `MainWindow` XAML, and modify the `Binding` in the status bar label to use the converter.
   (c) If the score is tied, show "Score: tie game".
3. Also in the status bar, add a label for whose turn it is. Add a property to `OthelloModelView` called `CurrentPlayer`, computed as the `CurrentPlayer` of the view model's `OthelloBoard` object. Write a converter that converts from an integer player to a string "Black" or "White". Bind your label content to the view model's `CurrentPlayer` property in the XAML.
4. Add a method `UndoLastMove` to `OthelloViewModel`. This method should called `UndoLastMove` on the board **if there is a move to undo**, then trigger `OnPropertyChanged` for any of the view model properties that changed because a move was undone.
5. Dock a `ToolBar` to the top of the main window. In this toolbar, add a button with the text `Undo`, which will call the `UndoLastMove` method you added to `OthelloViewModel`.

Deliverables

Turn in the following when the lab is due:

1. A printed copy of:
   (a) `MainWindow.xaml`
   (b) `OthelloViewModel.cs`

(c) `OthelloView.xaml`

(d) `OthelloView.xaml.cs`

(e) your new `IValueConverter`s

2. A screenshot of your application showing the changes that you made