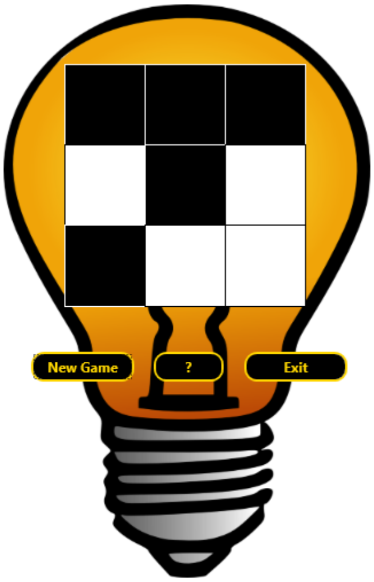
WPF Commands and Control Templates

GUI Programming

10 points

You are to modify your Lights Out application in the following ways:



1. *(2 pts)* Use a transparent window style with a customized image to replace the standard window your game is currently using. See the section below that describes how to implement this. Remove the main menu so only the New Game and Exit buttons remain.

2. *(1 pt)* Add some code so the user can still move the window when he or she clicks down on the image part of the window. You will need to do some research to figure this out.

3. *(2 pt)* Add an About button that has just a question mark in it. Pressing this button should launch the About dialog box. The New Game, About, and Exit buttons should use the `New`, `Help`, and `Close` Commands, respectively. Therefore your buttons should no longer use `Click` callbacks.

4. *(1 pt)* Disable the Exit button until the game is successfully completed. Use the `CanExecute` callback to make the `Close` command available only when the entire grid is off. (Yes, this will probably frustrate the user.)

5. *(2 pts)* Use a Control Template to modify the visual appearance of all three buttons. Use Triggers to visually change the buttons when the mouse is placed on top of them and when the mouse button is pressed down on top of them. Change at least *two properties* for each trigger.

6. *(2 pts)* Create a trigger for the Exit button so when the Close command is not available (the property `IsEnabled` is `False`), the look of the button is altered so it is obvious that the button is in a disabled state.

Zip up your entire project and submit your zip file to Easel before it is due.

Transparent Window

To create a customized window which deviates from the standard look, set the `Window` element's `WindowsStyle` property to `None` which removes the standard frame around your window. To make your window irregularly shaped, set `AllowsTransparency` to `True` and the `Background` to `Transparent`. If you add controls, only the space within the boundaries of the controls behave as part of the window, and clicks within the transparent area of the window are not responded to.

To make a customized background, set the background image to a PNG that contains transparent parts:

```
<Window.Background>
   <ImageBrush ImageSource="circle.png" />
</Window.Background>
```

where `circle.png` is an image which has been added to your project. The transparent parts of the PNG will allow the background to show through.