**OOP Program #7**

**Fall, 2014**

**Due Fri, Oct 31**

**This is not a group project; do your own work**

**30 points (see below)**

Create a GUI program to allow two users to play the game of “Connect 4”.

* The GUI should show the current game grid, along with an indication of whose turn it is (identify player turn by “black’s turn” or “red’s turn”).
* The game should have a button to allow the user to start over (when they press this button, the user should be prompted with a “Are you sure”? before clearing the game grid).
* The game grid has 6 rows and 7 columns. Draw the grid with a yellow background. Draw white circles over the background to represent open spaces; draw red/black circles to represent game pieces.
* The user should be able to click anywhere in a column to indicate that they wish to move in that column.
* You do not need to animate the game piece dropping. The correct color game piece should just “show up” in the correct position. Note that the user should not have to click in the exact position where the game piece will go; the user just has to click somewhere in the desired column, and then your program determines where the game piece will be placed. Clicks in a full column should be ignored (i.e., the player doesn’t lose their turn and nothing changes in the state of the game).
* The game should detect a win and display a MessageBox indicating the winner. Once the user clicks OK on this message box, the game board should be cleared.
* Minimize/restore should work for your program (i.e., the game gets repainted when the window is restored). Your program does not need to respond to window resize events by changing the size of the grid. You can just draw a fixed-size grid (you decide the size); if the user makes the window smaller or bigger than the grid, we’ll ignore that problem for now.
* Your program should have a menu that lets the user save/restore the state of the game. This should work in a similar way to your Circle-drawing program.
* Think carefully about your design/implementation and ensure that you are using appropriate language features. You should not do everything in the Form1 class! Split the functionality among multiple classes (similar to the way we have used a Game class and Circle class in previous assignments). Specifically, Form1 should contain the code that deals with the user interface, but should have a minimum of “game logic” in it. It should initiate things, but not contain the code that does everything. If you do not have at least one meaningful, well designed class other than Form1, you will receive a zero on the assignment.

**Turn-in Instructions**

Turn in hardcopy in class and softcopy of entire zipped solution folder to bboard.

**Grading Guidelines**

* Does it satisfy all requirements in this assignment sheet? 50%
* Did you use good style and object-oriented structure? 40%
* Did you follow instructions properly on submitting the program? 10%

**Late Policy:**

Flat 25% late penalty if not turned in on the due date. You have one week beyond the due date to submit the program. Beyond this you will receive a zero on this program grade. If there are extenuating circumstances that warrant additional time on the program, please talk to me (BEFORE the due date). I will handle requests for additional time on a case by case basis.