**CS 4500, Spring 2019**

**HW4: Visualizing the Game**

HW4 is the final assignment having to do with the Circles and Arrows game. The rules of the game are the same. This time…

1. There can be as few as 2 circles, and as many as 20 circles.
2. There can be at most 20 out arrows for each circle.
3. You should assume that the input file (if it exists) is formatted correctly, and that the circles and arrows in that file describe a fully connected digraph.
4. You should prompt the interactive user for the name of a file that holds the data that defines one circles and arrows game. If the user enters the name of a file that exists in the same file directory within which the program is executing, then the program continues (see #3). If no file exists with that name in the directory, then the program should give an appropriate error message, and reprompt for a file name.
5. After an existing file is selected, the program should run the game once.
6. As the game proceeds, the program should show the game on the screen using colored circles and arrows. Circles that have not yet been visited should have one color, and circles that have been visited should colors different from circles that haven’t been visited. It would be nice if colors are used to show circles that have been visited more with different colors than circles that have been visited less. Be creative. Make sure that the unvisited circles stand out from circles that have been visited. Arrows should be visible, and should show direction; but arrows needs not change color when they are traversed, although you might add that feature if you want.
7. Make the visualization go slowly enough that the interactive user can see the individual moves, but fast enough that the game moves right along. It is a judgment call how fast that is, but make that judgment after some experimentation. Each move should take the same amount of time, as much as possible.
8. When the game finishes, and you may assume that it will finish, display the final visualization, and beneath that picture, write on the screen the total number of checks, the maximum number of checks on any one circle, and the average number of checks on a circle. You do not have to keep a running total of these things as the game proceeds, though you may; but the final stats must be displayed when the game is finished.
9. Prompt the interactive user to push ENTER to finish the program.

As usual, document your program carefully with a thorough opening comment, and appropriate internal comments. If you use anyone else’s code, give credit where credit is due.