**CSCE 155N Matlab Programming Project 2 – Summer 2012**

**Assigned: Thursday 7/26/2012**

**DUE DATE:**

**Electronic: Thursday 8/2/2012 at 11:59 PM**

**(hardcopy in class on Friday)**

**Reversi Dots & Boxes**

**Problem Statement:**

Reversi (also called Othello) is a well-known game of strategy. Go to <http://www.gamesforthebrain.com/game/reversi/> for a site where you can play the game (along with a bunch of other good “thinking person games”). The general idea is that two players alternate placing bi-colored markers on a checkerboard, and flipping to his/her color all opponent’s markers which become surrounded on two sides.

Another game designed for a grid is Dots and Boxes. Go to <http://www.mathsisfun.com/games/dots-and-boxes.html> where you can try it out. Starting with a grid of dots, two players alternate drawing lines connecting adjacent dots vertically or horizontally. If a player completes a square, it is marked for that player and he/she gets to play again. The player completing the most squares wins.

We are merging these two games into a single game I am calling Reversi Dots & Boxes (RDB). RDB begins as a Dots & Boxes game on a blank grid. However, any time a square is completed, any adjacent sequences of opposing squares between this and another of ones own squares will be flipped. The effect is that once claimed by a given player, squares might flip back and forth between the two players as they are surrounded, ala Reversi. The winner is the player who ends up with the most squares once all boxes are completed.

Similar to the first project, you are to work in teams of two or three students to design prototypes in Matlab. You do not have to be in the same teams, but you may. Larger teams may be allowed by permission, but there would be higher expectations. Possibilities for enhancements (which can earn bonus points) include variable dimensions, maintaining a “top ten” list in a file, allowing either random computer play, smart computer play, or play against another person, etc.

As this is a prototype, having a fancy graphics display and event-driven interface is not expected. A simple “ASCII graphics” display and text-based I/O is quite sufficient. However, you may wish to consider this for the later GUI project!

**Collaboration:**

Work together as a class on any or all aspects of the research and design. Ideally take advantage of the talents of each member of the team. Form small teams to finalize each of the multiple design options. It is essential to keep track of who did what and where any useful information was found. You need to keep track when you help someone and when you receive help from someone. This includes students from other teams, those outside the class, TAs and me (the instructor). Keeping a log is highly recommended.

**What and How to Submit:**

Read and have your program conform to the “Program Documentation Guidelines” which were provided previously.

As you make progress with the project, periodically handin (electronically) updates numbered as mine are on my site. How many updates really depends on what seems natural. It might be 5 or it might be 15. Anyway, we should see a progression of “working” programs handed in as ***rdb1.m***, ***rdb2.m***, etc.

By the deadline hand in electronically the two files, rdb.m (the Matlab script file for the game), and rdb.doc (which contains summaries, documentation, and sample runs). In class the day, hand in hardcopy versions, stapled together with the cover page in front.

Each team member should submit his/her own analysis of the relative contributions of all the members toward the project. This should be submitted electronically using each member’s handin account. This is in addition to the acknowledgement section of the main report. Assuming allocation is fairly even, all will receive the same grade.

The Word document should contain the following, all carefully labeled:

* Cover page with name(s) and the account under which it is submitted, title, date submitted, etc.
* A discussion of the features you implemented in the project. Describe how they work and what Matlab options were used to program them. This should be at a fairly high level, not a line by line analysis of the code.
* An “instruction manual” that a non-programmer can use to run the Ricochet program.
* An annotated cut and paste sample dialog sampling of the running of the program. (Hint: Use the ‘diary’ command or cut & paste as appropriate.)
* A discussion of the testing that was performed. This should include testing of each component as it was being built, and testing of the final program ensuring that it works properly under a comprehensive range of conditions.
* An annotated cut and paste of a sample dialog, demonstrating how your program responds to extreme and faulty input. (This could be combined with the previous section.)
* Acknowledge all collaborations (both internal to the team and external), detailing what each person contributed individually, and what was done jointly. Indicate approximate percentages of the work contributed by each person in design, coding, testing, documentation, and report preparation.

**Grading Criteria:**

* Properly running features – 30%
* Program logic is well designed – 20%
* Progress versions – 20%
* Documentation guidelines are followed – 10%
* Handin Documents formatted and arranged as specified – 10%
* Testing is comprehensive – 10%