**CSCE 155N Matlab Programming Project 2 – Summer 2015**

**Assigned: Sunday 8/2/2015**

**DUE DATE:**

**Electronic: Sunday 8/9/2015 at 11:59 PM**

**(hardcopy in office by Monday)**

**Reflections**

**Problem Statement:**

*Note: There are similarities between the game of Reflections and the Ricochet project from a previous semester. You may wish to refer to that description and the project development located at my cse site.*

Reflections is a game for 2 or 4 players. In the game of Reflections, we begin with an empty game board that is divided into a grid like a checkerboard. This may be 8 x 8 but could plausibly be of other dimensions. Each player chooses one of the four sides of the board to be their “home” base. In the case of 2 players, these would not have to be opposite each other.

Each player will have a supply of small mirrors that can be mounted vertically on the squares of the board. A mirror is to be placed along either diagonal of a selected square. Players take turns placing one of the mirrors from their supply onto the board. Play continues until the board is covered with mirrors, or the supply of mirrors is exhausted.

After the mirrors have been placed, each player aims a laser directly into any of the mirrors on his/her home base. The number of reflections before the laser beam emerges from the board is counted. The player with the highest number of reflections wins the round. Ties are possible.

Strategy may include attempting to extend the number of ones own reflections or to disrupt an opponent’s string of reflections.

For this project the game should visually display the current state of the board, allow for different size boards, allow 2 to 4 players to select their home bases, control the sequence of play beginning with a rotating starting player for each round, maintain a tally of wins and losses, provide a convenient way to select the orientation and square for a mirror, save a partially played round for a later day, and visually simulate the path of the laser beam.

**Collaboration:**

Work together as a team on any or all aspects of the research and design. Ideally take advantage of the talents of each member of the team. Use your 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 and 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 ***reflections1.m***, ***reflections2.m***, etc.

By the deadline hand in electronically the two files, reflections.m (the Matlab function file for the game), and reflections.doc (which contains summaries, documentation, and sample runs). In class the day after, 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 set up and run the game.
* An annotated cut and paste sample dialog sampling of the running of the game. (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%