**CSCE 155N Matlab Programming Project 2 – Fall 2013**

**Assigned: Friday 10/29/2013**

**DUE DATES:**

**Electronic: Wednesday 11/13/2013 at 11:59 PM**

**(hardcopy in class on Thursday)**

**Ricochet**

**Problem Statement:**

Picture a closed “black box” about two inches deep and about a foot and a half on each side. You cannot open it to see what is inside. Around the sides are holes numbered from 1 – 40 (ten per side) just large enough to shoot ping pong balls into. Inside the box are 5 to 10 randomly placed baffles, oriented at 45 degree angles either right or left. A ping pong ball entering a hole directly in line with a baffle will ricochet off audibly at a 90 degree angle. Each baffle is only large enough to be hit from the hole directly in line with it on each of the sides of the box. A ping pong ball will go straight through the box in one hole and out the hole directly opposite, unless it hits a baffle. It is possible that the ping pong ball will ricochet off multiple baffles in succession before finally exiting the box. Each bounce can be heard as it happens.

This is a single player game. The idea is to shoot ping pong balls one at a time into any desired holes, and observe from which holes the ping pong balls eventually emerge, noting the sound of any ricochets. As the player determines the location and orientation of each baffle, he/she marks it on the top of the box. The goal is to find all the baffles using the fewest possible moves.

Similar to the first project, you are to work in teams of two or three students to design prototypes in Matlab. Larger teams may be allowed by permission, but there would be higher expectations. Possibilities for enhancements (which can earn bonus points) include maintaining a “top ten” list in a file, allowing either random computer generated baffle positions or an opponent to set the baffles, 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.

**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. 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.

One week before the deadline electronically handin a Word document in which you report on the makeup of your team, the features and any enhancements you are planning to implement, the progress made (and difficulties faced) to date, and the schedule and work assignments for the coming week.

By the two deadlines hand in electronically the two files, ricochet.m (the Matlab script file for the game), and ricochet.doc (which contains summaries, documentation, and sample runs). In class the day after each deadline, 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 of the running of the program. (Hint: Use the ‘diary’ command.)
* 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:**

* Program functions as intended – 30%
* Program logic is well designed – 20%
* Documentation guidelines are followed – 20%
* Handin Documents formatted and arranged as specified – 10%
* Testing is comprehensive – 10%
* Progress report – 10%