**CSCE 155N Matlab Programming Project 3 – Fall 2013**

**Assigned: Friday 11/18/2013**

**DUE DATES:**

**Electronic: Wednesday 12/11/2013 at 11:59 PM**

**(hardcopy in class on Thursday)**

**SuperPong**

**Problem Statement:**

Pong was one of the earliest video games, involving a ball bouncing off the rectangular “walls” and “ceiling” of the screen and a user-controlled “racket” that slides back and forth at the bottom to hit the ball back upward before it hits the “floor”. Our implementation in Matlab will have a computer-controlled racket, so that the user simply observes the action after specifying some initial conditions. These initial conditions include a choice of racket (various shapes, colors, sizes) stored in disk files, the elasticity of the ball (percentage of the kinetic energy retained after a collision with the wall or racket), scaling factor for size of the screen, maximum speed of the racket as it slides, and strength of gravity.

Scoring of the game could be by total time in the air prior to hitting the floor, total number of bounces off the racket (perhaps adding in some factor for the number of hits against the wall), etc.. Note that non-horizontal surfaces for the racket are essential for adding energy back into the ball (by attacking the ball), otherwise gravity and loss of kinetic energy will soon bring it to a halt.

It will be necessary to apply formulas related to energy, trajectories, partially elastic collisions, derivatives (to find slope at point of impact on the racket), etc.. You may need to do some research on this (and hopefully share what you learn, according to our collaboration policy).

Options may include sound at impact, putting English on the ball (involves more initial data including mass, friction of the surface, etc.), various algorithms for computing optimal strategies for bouncing the ball, letting the ball roll to a stop as it runs out of kinetic energy, keeping track of scores, etc..

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.

Having a GUI with event-driven programming would be a natural extension of this project, but that also applies to the previous two projects. If you would do this for SuperPong, you could hand it in just once as project 4 and have the project count for both 3 and 4. But please let us know!

**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.

By the first deadline hand in electronically the file pong.m (the Matlab script file for the game). By class time the next day, make sure that the file pong.doc (which contains summaries, documentation, and sample runs) is also electronically submitted. Also in class 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 Pong 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%
* Quality of the user’s manual – 10%