**CSCE 150E Matlab Programming Exercise 3 – Summer 2010**

**Assigned: 8/6/2010; Due: 8/12/2010 at 11:59 PM**

**Objectives:**

* Design and implement more intricate MATLAB programs
* Use of structures, functions, and GUI

**Problem Statement (Option 1):**

On my web site (<http://cse.unl.edu/~riedesel/pub/cse150>) there are folders for Latrunculi, Sudoku, and Mancala. Each contains the iterations of the development of a board game that demonstrates top-down programming with iterative development. Your task is to choose one of them and make is “GUI”. You are welcome to select another game or entirely different project (clear it with me first), but with these three you have the advantage of the code already done.

The expectation is that you take the program, which uses the traditional non-graphical input and fprintf I/O functions, and use hand-coding to make it event-driven. For example, with Sudoku, you might create a grid of buttons representing the 81 squares of the puzzle; perhaps incorporate a feature to load in from a file the starting configuration of numbers; and have call-back functions for all the buttons. The user might poke on a square, which results in either a text-box query for the desired numeral, or pull down a menu containing the choices. These are just ideas. You may have alternatives.

**Problem Statement (Option 2):**

Design and program a GUI electronic “cheat sheet”, as was presented in class on Friday. Each line could consist of a property (such as speed, volume, definition), a description (such as light in vacuum, sphere, triangle), formula/value (such as 186,000, 4/3pi r^3, three sided-polygon), units (such as mi/sec, cc, ---), and type (such as constant, formula, ---). Capability to select one of the lines and having the formula run on user supplied data would be really nice to have. Maintaining an alphabetical ordering as lines are added, edited, and deleted would be good. Switching to different subjects, each represented by a different data file, is a good feature. These are all ideas. Add, amend, or delete as you wish.

**What and How to Submit:**

Read the “Program Documentation Guidelines” and “Web Handin Instructions” posted in the Programming Exercises section in Blackboard, and follow their specifications, just as for the first two homeworks.

By the deadline hand in electronically the source files (possibly multiple .m and .fig files), and the .doc report file (which contains summaries, documentation, and sample runs). You may submit them separately or zipped up.

A paper copy (due the next day, as usual) is preferred. Please let me know by email if you are off campus and doing this is inconvenient, and I will rely on the electronic version to produce a printout.

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

* Cover page with name(s), title, date submitted, etc. The title should indicate the game you are implementing.
* Documentation of the subject research you did to prepare for the project. Cite sources and explain the formulas.
* Provide a discussion of the features you implemented in the game. Describe how they work and what Matlab options were used to program them.
* Rules for playing the game. Include sample screens from runs of the code.
* Acknowledge all collaborations, detailing what each person contributed individually, and what was done jointly.

**Grading Criteria:**

* Program functions as intended – 25%
* Program logic is well designed – 20%
* Documentation guidelines are followed – 25%
* Handin Documents formatted and arranged as specified – 15%
* Testing is comprehensive – 15%

Grading Standards:

* Grades will be weighted to some extent by how challenging the project is. This is not designed to preclude introductory programmers from potentially making good grades.