**CSCE 155N Matlab Programming Project 3 – Spring 2012**

**Assigned: Friday 3/16/2012**

**Revised: Wednesday 4/18/2012**

**DUE DATES:**

**Code and Report files: Wednesday 4/25/2012 at 11:59 PM**

**(hardcopy in class on Friday)**

**I’m Drowning!**

**Problem Statement:**

Picture yourself having fallen into a cave and being injured. Storm waters are pouring into the cave and threaten to flood everything. All you have is a laptop with Matlab running on it and a camera on which you have snapped a picture that shows the configuration of the cave. While this may not be sufficient to save you, at least you can get an idea of how much time you have left.

Suppose that the cave is essentially 2-dimensional, though likely of a quite wildly irregular shape. There is only one opening straight above you, and the flow of water is steady. Your task is to simulate the filling of the cave with water on the computer.

In reality, you are quite safe, and there is no handy cave of which to snap a picture (assuming you could even get an angle that captures the entire side view!). So as an alternative, draw the outline of a crazy shaped cave using a dark marker, and take a picture of that. Store the picture as a jpeg file in your working directory. Also, store a picture of yourself (or another person whom you might wish to drown). Your program will prompt you for the files to use, the scaling (or otherwise indicate the actual dimensions), and the rate of water flow (square meters per second?). You may need to reposition/resize your self-portrait so that it looks right.

Animate the figure to show water pouring in from the entry hole, hitting the floor, and spreading out horizontally, realistically filling in the low points, eventually spilling over high points, and rising higher and higher. As a bonus, deal with the possibility that there may be air pockets into which the water physically would not rise into, and for further bonus simulate some rise, considering that air is actually compressible. Finally, at the point where the water rises above your nose, redraw the eyes as x’s.

At the end of the simulation (or optionally throughout), display the amount of time that elapsed and the quantity of water that poured into the cave.

Similar to the first two projects, 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. There are many possibilities for enhancements and variations in how the images are captured and processed. Please pursue such ideas!

**Collaboration:**

Work together as a class on any or all aspects of the research and design, ideally taking 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 two deadlines electronically hand in the two files drowning.m (the Matlab script file for the game) and drowning.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), the account under which it is submitted, title, date submitted, etc.
* A discussion of the features you implemented and assumptions you made in the project. Describe how they work. 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 drowning simulation. Be careful to specify how the user should prepare the input files and any tricky points.
* 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 designing, coding, testing, documenting, and preparing the report.

**Grading Criteria:**

* Program functions as intended – 50%
* Program logic is well designed – 30%
* Documentation guidelines are followed – 10%
* Handin Documents formatted and arranged as specified – 10%

**Base Grade:**

The base grade is derived from the features your team implements. A team begins with needing one feature in order to earn a grade of C. Each member is expected to contribute one additional feature. So, for example, a three member team should have four features to earn a base grade of C. Each additional feature is work a full grade.

Possible features include:

* Fill a straight sided and bottomed cave with water
* Fill a curved bottom, showing a flowing stream to the lowest point if incoming water hits off-center
* Fill an irregular bottom cave with water, allowing spilling over pools
* Allow waterfall effect over protruding cliff ledges
* Allow multiple active streams spreading out and rising
* Allow variable rates of water flow
* Handle variable ceiling situations that create air pockets
* Implement compressibility of air in air pockets
* Include a victim that can be positioned and “drowned”
* Others? Check with me!