**CSCE 155N Matlab Programming Project 1 – Summer 2016**

**Assigned: Tuesday 6/14/2015**

**DUE DATE:**

**Friday 6/24/2015 at 11:59 PM**

**(hardcopy in class Monday)**

**Transylvanian Transformational Transportation Taskforce**

**Problem Statement:**

Transylvania has been a haven for a host of “transformational” creatures. Many have been captured and have been put on display in zoos. Because of the perceived danger, a new zoo is being constructed with improved safety features. After all, we don’t want wayward children slipping into enclosures with these creatures. The TTTT is being formed with the task of transporting them to their new home.

At this point the TTTT has designed cages that will be placed on a flatbed trailer in a rectangular grid of some number of rows and columns. Each cage will house one creature. The creatures being transported include some mix of vampires, werewolves, and zombies. The problem for you to deal with is that they pose a danger to each other. They can reach through the bars of their cages to grab and bite the creatures immediately to their front, back, left, and right. When a werewolf fights with a vampire, the werewolf will invariably bite the vampire; when a vampire fights with a zombie, the vampire will invariably bite the zombie; when a zombie fights with a werewolf, the zombie invariably bites the werewolf. They are all nocturnal creatures. During the daytime they sleep, but the bitten ones become transformed into whatever bit them. So a vampire bitten by a werewolf becomes a werewolf, etc.

They do all recover from their wounds during the day, and resume biting each other the following night. We don’t really mind all the biting and transforming, but we really would like to have representatives of each creature to deliver to the new zoo. You are to consider a number of scenarios of packing the creatures on the flatbed trailer, simulating their transformations over the days of the trip, and verifying that at least one of each exists upon delivery.

**Problem input:**

You need to get the number of rows and columns for the cage arrangement. You need to get the initial placements of the creatures into the cages. You need to get the number of days for the trip. How you do this is up to you, but you need to justify your design in the report that accompanies your program. You may choose inputting from the keyboard, prompted for each datum, or prompted for an entire matrix. You may choose to input from a data file. Perhaps you can think of other ways to get the data into the program. If you upgrade this project to become project 4, you might use a GUI approach to data entry.

**Problem output:**

Similarly, you may design how the results are reported. We should see the final pattern and numbers. We may appreciate seeing the progress each night brings. Again, discuss your choices in the report.

**Problem options:**

You are welcome to propose variations, such as having random mutations, extending the trip, stacking the cages on top of each other, etc. Yes, discuss your choices in the report.

**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 ***tttt1.m***, ***tttt2.m***, etc.

By the deadline hand in electronically the two files, tttt.m (the Matlab function file for the game), and tttt.doc (which contains summaries, documentation, and sample runs). In the next class period, 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 simulation. (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%