**CS 4500**

**HW3**

HW3 assumes that you have successfully accomplished HW1 and HW2. Therefore, we assume that you have a program that plays the strange game (red marker searching randomly for a way out), and that you have devised a way to display that for users.

HW3 requires you to revise and extend your HW2 program. You will again use graphics to show the user what is happening. This must be done using Javascript in a single file. You will also be making TWO markers move instead of one.

We’ll go back to fixing the grid size as 10 X 10. This time a red marker starts in the lower left cell of the grid, and a blue marker starts at the upper right cell of the grid. Each “turn” has each of the two markers taking a random “step” in the same way as in HW1 and HW2. The difference is that red is trying to get to the upper right cell, but the blue marker is trying to get to the lower right cell. When either marker gets to its “goal” cell, the game stops. If a marker lands on a cell already occupied by the other marker, the marker that gets landed upon is sent all the way back to its “home cell,” where it started the game. This only happens for the FINAL cell of a marker’s move, not any intermediate cell.

You have a great deal of latitude in deciding what graphical representation would be most enlightening to viewers. You will decide how long each step will last, how many pixels and what colors will represent the cells. Also, think creatively about what information you will show to the viewer, and how you will make that information visible. Be careful about screen geography. Part of your grade will be based on whether or not I think your graphics are helpful in understanding what is happening with this double random walk simulation. When the simulation has run its course, output to the screen the information you collect about the simulation. That information is noted as follows…

Note #1. The red marker always moves first.

Note #2. Keep track of who “wins” (that is, who gets to the goal cell first).

Note #3. Keep track of the average number of times each cell is touched by either of the markers, and the maximum number that a cell is touched.

Note #4. Keep track of how many times each of the markers moved before the game ended.

Note #5. If both markers have made a million moves (some of which will land on the same cell, which should be counted), then the game ends with appropriate reports to the screen.

Note #6. Keep track of how many times each marker was sent back to its home square by the other marker.

This is an individual assignment. However, you will be working with two other students who will be assigned in class. You can talk about the details of the assignment with your study partners, and no one else. You may NOT share code directly. That is, you should be typing in your own code, NOT cutting and pasting your partner’s code into your source code. You can walk through your code with your study partners, and each should try to learn from the other. If one of you knows graphics better than the others, share that knowledge, but don’t directly share the code, either electronically or on paper. You can show your partners your code on the screen of your computer.

Make sure to include your partners’ names in your code’s opening comments. You should also tell how your partners helped you with this assignment. Always give credit where credit is due.

Documentation is central. The file you submit should have your name, your partners’ names, the date, and other helpful information as necessary. There should be a beginning comment that gives an overview of the game being simulated, and gives an overview of the program itself. Inside the code, use “paragraphing” comments to break up code. Subprograms (such as functions) should include a comment telling the reader the purpose of the code and any assumptions inherent in the code. Any tricky parts of code should be commented to enhance human readers’ understanding of your intent, and the details of the implementation.

For HW3, your program MUST be a web-based program using Javascript (and HTML and CSS as needed). Your program has to be in a single, unzipped RTF file. With a web-based program there is no need for downloading files, linking, compiling, or any such complications. You will be submitting an RTF file with your source code, and that’s all.

Be sure to get clarifications from the customer (your instructor…, me…) when necessary.

Keith