Carlise Moreland and Joshua Pollock

CS 122L-1

Lab 11a: Alien Invasion

30 April 2018

**Task Description**

In this two-part lab, we are tasked with creating five separate programs that provide various applications. These five functions we will be creating are: randomFloatValue, getUserValue, getFileName, calculateBounds, and writeToFile. Each of these functions has specifications that the function should fulfill.

**Learning Objectives**

In this lab we will be practicing further with MATLAB functions. We will learn about using MATLAB for simulations and modeling. We will be practicing using various functions to solve the problem given to us (i.e. the rand() function).

**Approach**

To begin this lab, we started by downloading the given .zip file from BBLearn and extracted it to the Lab 11 folder. We looked over the specifications for each function, and if we didn’t understand something, we looked it up online. We took a look at the rand() function, in order for us to generate a number within a given range with lower and upper bounds. This proved to be quite easy, and we were able to get a function done. Next, we looked up how to see if a string ended in ‘.txt’. This proved more difficult, but we were able to get the function completed. The rest of the functions were quite easy to complete and all we needed was to read through the lab specifications.

**Program Inputs**

**randomFloatValue**

Takes in a lower bound of the range for random numbers and the upper bound of the range for random numbers.

**writeToFile**

Takes in a filename string for a file that holds all the data for the lab. Takes in a parameter num\_sim which is the number of random numbers to generate and write to the file. Takes in a lower and upper range for the random numbers to be generated.

**calculateBounds**

Takes in the parameter a\_rate, which is the rate of alien attackers, and the variable var, which is the percent variance of the attackers.

**getFileName**

Does not take in any parameters. Asks the user to input a file name. Checks that the filename is correct and follows lab specifications.

**getUserValue**

Takes in lower and upper bounds for the value to be generated. Has a prompt parameter that takes in a string that prompts the user to enter a value between the min and max.

**Program Outputs**

**randomFloatValue**

Outputs the generated random value that is within the bounds.

**writeToFile**

Outputs the information to the file that is opened.

**calculateBounds**

Outputs the lower\_bound and upper\_bound found by the equations given by the lab specifications

**getFileName**

Outputs the filename if the user inputted a correct and valid filename (must end in ‘.txt’).

**getUserValue**

Outputs the value the user enters if the number is in the bounds.

**Source Code**

**randomFloatValue**

function val = randomFloatValue(lower, upper)

val=lower+rand()\*(upper-lower);

end

**writeToFile**

function writeToFile(filename,num\_sim,lower,upper)

fileID = fopen(filename,'w');

while(num\_sim>0)

randVal = randomFloatValue(lower,upper);

fprintf(fileID,'%f\n',randVal);

num\_sim = num\_sim-1;

end

fclose(fileID);

end

**calculateBounds**

function [lower\_bound, upper\_bound] = calculateBounds(a\_rate,var)

variationAmount = a\_rate\*(var/100);

lower\_bound = a\_rate - variationAmount;

upper\_bound = a\_rate + variationAmount;

end

**getFileName**

function filename = getFileName

validName=false;

compareTo='.txt';

while(~validName)

filename = input('Input a file name: ','s');

if length(filename) >= length(compareTo)

comparedString = filename(end-3:end);

validName = strcmp(compareTo,comparedString);

end

end

end

**getUserValue**

function val = getUserValue(prompt, lower, upper)

numberInBounds=false;

while(~numberInBounds)

val = input(prompt);

if(val >= lower && val <= upper)

numberInBounds = true;

end

end

end

**Conclusions**

This lab proved to be quite simple. The hardest part was the randomFloatValue and the getFileName. These functions proved to be a little more difficult than the rest. If we were to refactor the lab, we would choose to refactor the getFileName. At the time we did not know that MATLAB had a strsplit() function for strings. We could have easily split at the ‘.’ And then check if the second string was ‘txt’. The randomFloatValue did not work for us at first as we forgot to multiple rand() by the upper bound minus the lower bound. After these few hiccups, we completed the lab very easily and effectively.