Carlise Moreland and Joshua Pollock

CS 122L-1

Lab 6: Risk Dice

12 March 2018

**Task Description**

In this lab, we are responsible for simulating a risk style dice games. The lab will run through the probabilities of the attack dice winning, defense dice winning, or a tie game. There is an extra credit opportunity to create the same program using different numbers of attack and defense dice. The output of these probabilities will be displayed within a pie chart.

**Learning Objectives**

The learning objective for this lab are as follows:

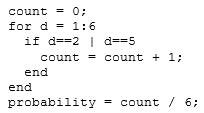
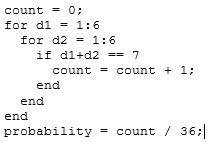
* Calculate the probabilities of each possible outcome for a risk-style dice battle
* Graph the outcomes with a pie chart.
* Practice using loops, conditionals, graphs, and if statements.

**Approach**

For this lab we will be looking at a risk type simulation using MATLAB. Using the Wikipedia Risk entry, we should be able to conduct this and show the probabilities of each outcome. We first started out by reading through the instructions and setting up the initial nested for loops. From here all we would need to add is all the if statements and logical statements to manipulate the dice values and count attacker wins, defender wins, ties, and number of games played.

**Mathematical Concepts**

There are a few mathematical concepts used in this lab. These concepts are:

* The probability of rolling a 1 through 6 on a D6 is 6/6, or 1.0 (100%)
* The probability of rolling a 1 or a 5 is 2/6, or 0.33 (33%)
* program that will calculate the probability of a 2 or a 5 being rolled on a D6
  + 
* program that calculates the probability of a roll on 2D6 adding up to 7
  + 

**Program Description**

Lab 6

Uses 4 nested for loops to simulate 2D6 dice. These loops will iterate 1 through 6. There will be two attacker dice and two defender dice. The dice will be sorted from greatest to least. The dice will then be compared to each other. If all the attacker die is greater than the defense, the attackers win. Else if the sum of the defense dice is greater than or equal to the sum of the attack dice, the defense wins. Else the game is a tie. The resulting data is shown in a pie chart.

EC 1

Uses 4 nested for loops to simulate 2D6 dice. These loops will iterate 1 through 6. There will be three attacker dice and one defender dice. The dice will be sorted from greatest to least. The dice will then be compared to each other. If all the attacker die is greater than the defense, the attackers win. Else if the sum of the defense dice is greater than or equal to the sum of the attack dice, the defense wins. Else the game is a tie. The resulting data is shown in a pie chart.

EC 2

Uses 3 nested for loops to simulate 2D6 dice. These loops will iterate 1 through 6. There will be two attacker dice and one defender dice. The dice will be sorted from greatest to least. The dice will then be compared to each other. If all the attacker die is greater than the defense, the attackers win. Else if the sum of the defense dice is greater than or equal to the sum of the attack dice, the defense wins. Else the game is a tie. The resulting data is shown in a pie chart.

EC 3

Uses 3 nested for loops to simulate 2D6 dice. These loops will iterate 1 through 6. There will be one attacker dice and two defender dice. The dice will be sorted from greatest to least. The dice will then be compared to each other. If all the attacker die is greater than the defense, the attackers win. Else if the sum of the defense dice is greater than or equal to the sum of the attack dice, the defense wins. Else the game is a tie. The resulting data is shown in a pie chart.

**Source Code**

Lab 6

function [attack\_wins,defense\_wins,tie] = Lab6

attack\_wins=0;

defense\_wins=0;

tie=0;

gamesPlayed=0;

for a1 =1:6

for a2 = 1:6

for d1=1:6

for d2=1:6

attackers = sort([a1 a2]);

defenders = sort([d1 d2]);

if (attackers(1) > defenders(1)) && (attackers(2) > defenders(2))

attack\_wins = attack\_wins + 1;

elseif ( attackers(1) + attackers(2) <= defenders(1) + defenders(2))

defense\_wins = defense\_wins + 1;

else

tie=tie+1;

end

gamesPlayed=gamesPlayed+1;

end

end

end

end

attack\_wins = attack\_wins / gamesPlayed

defense\_wins = defense\_wins / gamesPlayed

tie=tie/gamesPlayed

pieChart=[attack\_wins defense\_wins tie];

labels = {'Attacker Wins 2','Defender Wins 2', 'Both Win 1'};

pie(pieChart,labels);

end

EC 1

function [attack\_wins,defense\_wins] = Lab6EC1

attack\_wins=0;

defense\_wins=0;

tie=0;

gamesPlayed=0;

for a1 =1:6

for a2 = 1:6

for a3=1:6

for d1=1:6

attackers = sort([a1 a2 a3]);

defenders = d1;

if attackers(1) > defenders(1)

attack\_wins = attack\_wins + 1;

elseif ( attackers(1) + attackers(2) + attackers(3) <= defenders(1))

defense\_wins = defense\_wins + 1;

else

tie=tie+1;

end

gamesPlayed=gamesPlayed+1;

end

end

end

end

attack\_wins = attack\_wins / gamesPlayed

defense\_wins = defense\_wins / gamesPlayed

tie=tie/gamesPlayed

pieChart=[attack\_wins defense\_wins tie];

labels = {'Attacker Wins 1','Defender Wins 1', 'Both Win 1'};

pie(pieChart,labels);

end

EC 2

function [attack\_wins,defense\_wins] = Lab6EC2

attack\_wins=0;

defense\_wins=0;

tie=0;

gamesPlayed=0;

for a1 =1:6

for a2 = 1:6

for d1=1:6

attackers = sort([a1 a2]);

defenders = d1;

if attackers(1) > defenders(1)

attack\_wins = attack\_wins + 1;

elseif ( attackers(1) + attackers(2) <= defenders(1))

defense\_wins = defense\_wins + 1;

else

tie=tie+1;

end

gamesPlayed=gamesPlayed+1;

end

end

end

attack\_wins = attack\_wins / gamesPlayed

defense\_wins = defense\_wins / gamesPlayed

tie=tie/gamesPlayed

pieChart=[attack\_wins defense\_wins tie];

labels = {'Attacker Wins 1','Defender Wins 1', 'Both Win 1'};

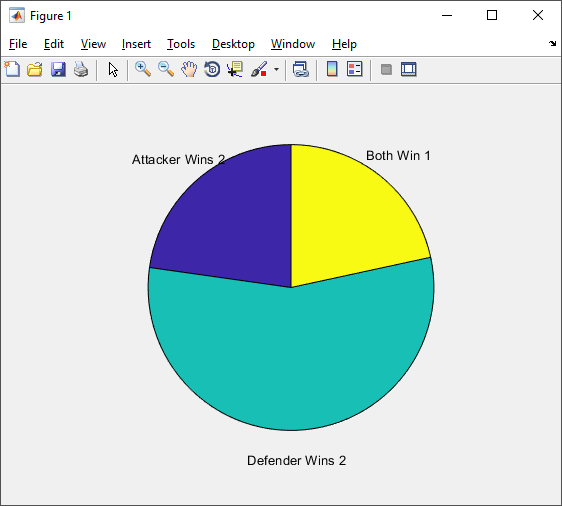
pie(pieChart,labels);

end

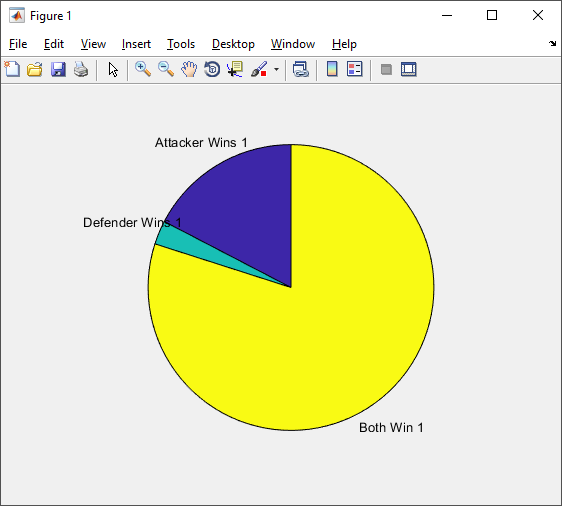
EC 3

**Code Execution Results**

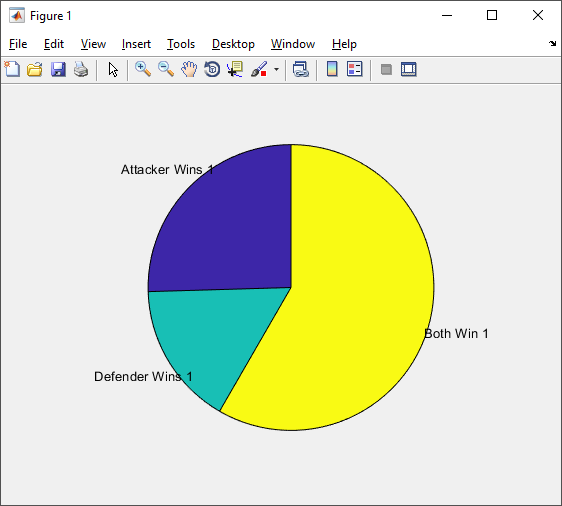
Lab 6



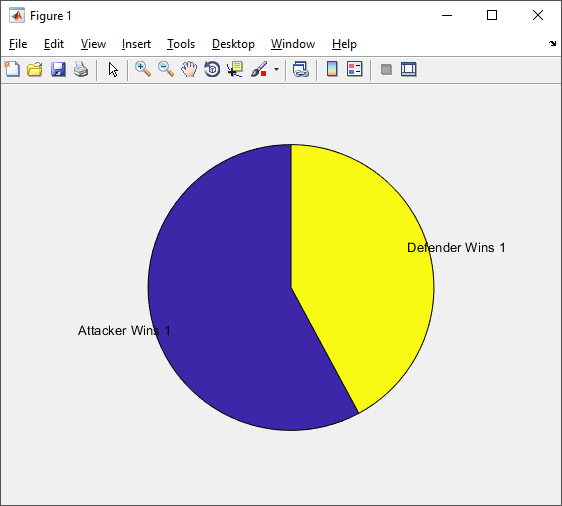
EC 1



EC 2



EC 3



**Conclusions**

This lab was quite confusing to get completed. The instructions did not provide clear directions on what to do. We were confused on what the if statements should have been. However, we did get pie charts to display. This lab has taught us how we can use the graphing functions of MATLAB to display statistical information. This lab has also helped us understand what for loops do and how we can use them. While it was very brief, we had also begun to learn about how nested for loops work.