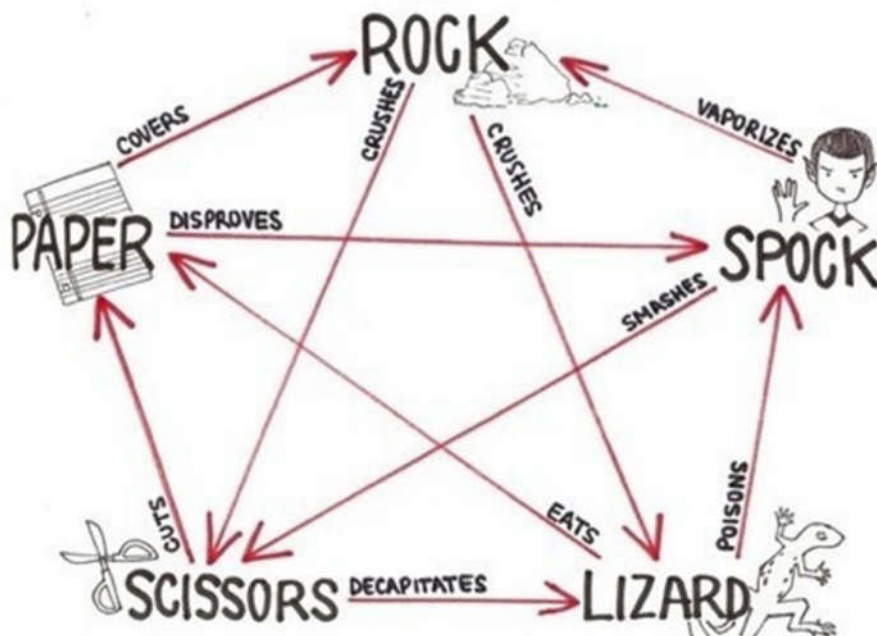


Module	CS4012 – Representation and Modelling
Day	16
Lab	10
Topic	Game: Rock - Paper - Scissors - Lizard - Spock

Summary

Rock - Paper - Scissors - Lizard - Spock (RPSLS) is a variant of the popular hand game Rock - Paper - Scissors (RPS). Game [inventors](#) Sam Kass and Karen Bryla noted that the odds of a draw in a game of RPS are pretty high if you know someone well enough. In a game of RPSLS, however, the odds of a draw are reduced by virtue of the additional weapons. Each choice in a game of RPSLS wins and loses against two others, and draws with itself. The following lab session focuses on developing a Lua implementation of the game played through the command-line interface.



Determining the winner

With five different weapon choices and five outcomes, simply coding up a solution involving “if-then-else-if-else” statements would require (5×5) 25 different conditional blocks. A better solution involves assigning to each weapon a number (between 1 and 5) and performing a comparison with a little modular arithmetic.

A game is played by each player picking a number; the second player being the computer. The number chosen by player one is subtracted from that of player two, and the difference modulo 5 is the result. If the result is an odd number player one wins, if even player two wins. If the result is zero the game is a draw.

Tip-of-the-hat: The use of RPLS as a programming exercise in Lua is inspired by the online [course](#): An Introduction to Interactive Programming in Python, from Rice University.

Calculating the result

The weapons can be numbered as follows:

Rock – 5, Paper – 1, Scissors – 2, Spock – 3, Lizard – 4

In this order, every weapon will lose against the next and win against the one after that. For Instance Rock - 5 will lose to Paper - 1, but win against Scissor - 2. This wraps around from Lizard to Rock like going round a circle. Take the following game as an example:

Player one chooses Rock – 5

Player two chooses Paper – 1

Taking the difference of the numbers chosen by player one and two, and then obtaining the difference modulo 5, results in:

$$(5 - 1) \% 5 = 4$$

As the result number (4) is even, player two wins. Note that if the difference is negative, obtaining the difference modulo 5 will result in a positive number. In all scenarios, a number between 0 and 4 will result.

Coding it up

A code template has been provided to guide the development process; you are strongly encouraged to use it. The general approach can be stated as follows. Read the name of player one's choice from the command line, and convert this to a number between 1 and 5. Generate the number chosen by player two as a random number, and convert this to a name. Calculate the result of the game using the chosen numbers and with the equivalent names output an appropriate message. A typical game output is given below:

New Game

Rock, Paper, Scissors, Spock... enter your move!
Rock

Player chose Rock and Computer chose Paper
Rock beat Scissors

Player wins!

Rock, Paper, Scissors, Spock... enter your move!
Rock

Player chose Rock and Computer chose Paper
Paper beats Rock

Computer wins!
Game over!

Tip: Don't write large tracks of code before seeing if it works. Test your code as you go.

Exercise

1. Starting with the code template, populate the array “names” with string elements corresponding to the weapon names. In this case, the indices are numbers i.e. the weapon numbers - see previous page.

Hint: Consider using the so-called list constructor to initialize an array on creation.

2. In a similar way, populate the array “numbers” with elements corresponding to the weapon numbers. In this case, the indices are strings i.e. the weapon names.
3. Develop the skeletal function printResults in such a way that on receiving the result of a game (as a number) an appropriate message is outputted for a draw, loss and win.

Hint: Make use of the conditional block within the function

4. Develop the skeletal “while” loop so that it continues to loop until a game over condition. Note code that is to be repeatedly executed should be placed here.
5. Write the code required to generate player two’s weapon number and convert this to the corresponding name.

Hint: A weapon number is randomly generated and its corresponding name can be determined using the “names” array. Make use of the variables “computerNumber” and “computerName”.

6. Similarly, write the code required to read in player one’s weapon name and convert this to the corresponding number.
7. Using the chosen numbers from both players, determine a result.
8. Output the result of the game with a call to the printResult function, by passing the determined result as an argument,
9. Test your game. Run the program repeatedly to generate different computer guesses and then evaluate the output.