Assignment 2 – LRC Report

Max Ratcliff

CSE 13S – Spring 24

Purpose

The Purpose of this program is to play a game of Left, Center, Right to completion, by simulating dice rolls and making player decisions based off of them

1 Questions

Please answer the following questions before you start coding. They will help guide you through the assignment. To make the grader's life easier, please do not remove the questions, and simply put your answers below the text of each question.

1.1 Randomness

Describe what makes randomness. Is it possible for anything to be truly random? Why are we using pseudorandom numbers in this assignment? **Randomness is when you can't predict what's about to happen next and you can't discern a pattern. Nothing is truly random, as there is some model that can predict it. We use pseudorandom numbers cause to the human eye they are basically random and impossible to re-generate without a seed which we can arbitrarily select from places such as the system clock.**

1.2 What is an abstraction

When writing code, programmers often use "abstractions". Define an abstraction in non computer science terms (Don't google it!) **abstractions breakdown complex problems into a series of simple steps that make it very easy to solve.**

1.3 Why?

The last assignment was focused on debugging. How can abstractions make debugging easier? What other uses are there for abstractions? Hint: Do you have to be the one to write the abstraction? **Abstractions can make debugging easier by helping you isolate a bug by breaking a program into functions, they also improve collaboration as many different people can work on their part of the program and as long as the individual parts work to the spec then the whole program should work.**

1.4 Functions

When you write this assignment, you can chose to write functions. While functions might make the program longer, they can also make the program simpler to understand and debug. How can we write the code to use 2 functions along with the main? How can we use 8 functions? Contrast these two implementations along with using no functions. Which will be easier for you? When you write the Program design section, think about your response to this section. **We can use 2 functions being a function to get the input for player number and seeds and handle errors and a function to return the move that happens due to the dice roll, to use 8 functions we could split the function to handle input into 4 separate ones,**

2 to get each input and 2 to check for errors, we could split the function to handle the roll into 2, and we could add functions to handle each turn and a win check. The first implementation would work well, but will have large blocks of code, the second implementation is a lot of extra coding for minimal return and no functions would make debugging hard.

1.5 Testing

The last assignment was focused on testing. For this assignment, what sorts of things do you want to test? How can you make your tests comprehensive? Give a few examples of inputs that you will test. **I will test to make sure the program properly handles too few or too many players, I will test to see when happens when it's given a non integer input such as a letter or a special character. I will also test to make sure the program handles an incorrect seed input properly.**

1.6 Putting it all together

The questions above included things about randomness, abstractions and testing. How does using a pseudorandom number generator and abstractions make your code easier to test? **using a pseudorandom number generator means that on two distinct program executions if the same player number and seed input is used than the winner should remain the same, and abstractions make bugs easier to identify and make it easier to test individual components of your program**

2 How to Use the Program

To use this program. First enter the number of players between 3 and 11 inclusive when prompted, and then enter an integer for a random seed when prompted... The program will then run automatically printing out player name, player roll and player chips each turn. Execution ends when there is a winner.

3 Program Design

3.1 Overall Pseudocode

Scan for the number of players (3 to 11) and check for bad input scan for seed and check for bad input enter while loop for round: enter for loop to loop through array: print name of first player if they have dice available for loop to roll all of player's dice: modulus 6 to get rand num between 0 and 5 update chips check for winner

3.2 Descriptions of Functions

Function 1:

- roll_dice

    - Inputs: none
    - Output: move (**DOT, LEFT, CENTER, RIGHT**)
    - Uses random number generator to simulate the roll of a dice and select the appropriate move

Function 2:

- no_winner

    - Inputs: array of chips and number of players
    - Outputs: True(1)/False(0)

– Used as a loop helper. Checks for only 1 player having chips, if so prints the name of the winner and returns false to break the loop, otherwise returns returns true to continue

Function 3:

- initchips

    – Inputs: array of chips and number of players
    – Outputs: initialized array of chips
    – helper function to initialize the starting array of chips to contain all 3s

Function 4:

- sim_game

    – Inputs: number of players
    – Outputs: printing number of tokens at the end of each players turn
    – simulates the Left, Center, Right game and prints the player name and number of tokens at end of turn