Assignment 1 – LRC Report Template

Nithin Duvvuru

CSE 13S – Winter 24

Purpose

Audience for this section: Pretend that you are working in industry, and write this paragraph for your boss. You are answering the basic question, “What does this thing do?”. This section can be short. A single paragraph is okay.

Do not just copy the assignment PDF to complete this section, use your own words.

This is a basic game called Left, Right, and Center implemented into C. In short, players take turns rolling a die, and depending on the number they roll, they pass around chips, the player that gets all the chips wins the game.

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.

Randomness

Describe what makes randomness. Is it possible for anything to be truly random? Why are we using pseudorandom numbers in this assignment?

I think randomness is based on being impossible to guess the result of something because there is no logic involved. I believe nothing can be truly random because something will always weigh the result in one direction. For example, if it is a coin flip, it would be a 51 percent chance to land on whatever side you start on. Even using a random function in Computer languages is not truly random because Computers rely on logic to provide random numbers. Pseudorandom numbers will always be numbers based on the initial seed value so using them in this game can help us debug and test our code with ease.

What is an abstraction

When writing code, programmers often use ”abstractions”. Define an abstraction in non computer science terms (Don’t google it!)

Abstractions refer to generalizing complex scenarios making them easier to understand.

Why?

The last assignment was focussed 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 make debugging easier by letting the programmer focus on the essential details while temporarily ignoring the unnecessary details that make the program complex. The abstraction can be apparent when working on the same program with a group of people. By dividing the program into using abstractions sections and working on it individually, you collaboratively create abstractions with your team.

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.

Two functions that would be helpful in the implementation of this program would be a function that simulates rolling a die and another function that would handle each player taking turns. Using eight functions would just be separating various steps implemented in these two functions further such as having a different function for handling left and right passes, and managing the pot. I think having two functions would be the easiest way for me to implement this program because I think having eight functions is just unnecessary and having zero functions would result in a lot of redundant lines of code.

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.

Some things I would test are if the program manages the chips properly, making sure turns are taken in order and if the pseudorandom number generator produces the correct output depending on the given seed. I can make my tests comprehensive by covering a variety of scenarios in the same test.

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 generator and abstractions makes testing easier because pseudorandom numbers are not completely random so you can keep debugging the same numbers. Meanwhile, abstractions make testing general parts of the code easier so you can debug the parts with problems instead of the whole program.

How to Use the Program

Audience: Write this section for the user of your program. You are answering the basic question, “How do I use this thing?”. Don’t copy the assignment exactly; explain this in your own words. This section will be longer for a more complicated program and shorter for a less complicated program. You should show how to compile and run your program. You should also describe any optional flags that your program uses, and what they do.

To show “code font” text within a paragraph, you can use `\lstinline{}`, which will look like this: `text`.

For a code block, use `\begin{lstlisting}` and `\end{lstlisting}`, which will look like this:

```
Here is some code in lstlisting.
```

And if you want a box around the code text, then use `\begin{lstlisting}[frame=single]` and `\end{lstlisting}`

which will look like this:

```
Here is some framed code (lstlisting) text.
```

Want to make a footnote? Here's how.[1]

Do you need to cite a reference? You do that by putting the reference in the file `bibtex.bib`, and then you cite your reference like this[1][2][3].

The program starts off by asking the user to input the number of players and a random seed for the PRNG, giving errors and using default values if the user inputs invalid arguments. The program will proceed to roll dice for each player specified, eventually ending in a winner.

Program Design

Audience: Write this section for someone who will maintain your program. In industry you maintain your own programs, and so your audience could be future you! List the main data structures and the main algorithms. You are answering the basic question, "How is this thing organized so that I can have a chance of fixing it?". This section will be longer for a more complicated program and shorter for a less complicated program.

The program will include two lists, one for chips and one for player names. Index 0 of the chips list will correspond to index 0 of the player list and so on. I plan to implement two functions, one that manages to simulate the dice roll using PRNG and another function that handles what happens during each player's turn while the main function handles cycling between turns and checking for winners.

FINAL DESIGN CHANGES: While coding the assignment, I found it easier to add another function that checked if someone won the game. The roll die function also took two arguments for the intended use case I had.

Pseudocode

Give the reader a top down description of your code! How will you break it down? What features will your code have? How will you

rolldie function:

uses PRNG to simulate rolling die

returns die value

taketurn function (takes in index number for player and chip):

checks if player is eligible to roll calls rolldie function handles different rolling outcomes changes values in lists

main function:

cycles between players for turns and uses two functions defined above to do so keeps going until winner is found

Function Descriptions

For each function in your program, you will need to explain your thought process. This means doing the following

- The inputs of every function (even if it's not a parameter)

[1] This is my footnote.

- The outputs of every function (even if it's not the return value)
- The purpose of each function, a brief description about a sentence long.
- For more complicated functions, include pseudocode that describes how the function works
- For more complicated functions, also include a description of your decision making process; why you chose to use any data structures or control flows that you did.

Do not simply use your code to describe this. This section should be readable to a person with little to no code knowledge.

References

[1] Wikipedia contributors. C (programming language) — Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/C_(programming_language), 2023. [Online; accessed 20-April-2023].

[2] Robert Mecklenburg. *Managing Projects with GNU Make, 3rd ed.* O'Reilly, Cambridge, Mass., 2005.

[3] Walter R. Tschinkel. Just scoring points. *The Chronicle of Higher Education*, 53(32):B13, 2007.