

Assignment 1 – LRC Report Template

Debi Majumdar

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.

The purpose of this assignment is to implement the Left, Right and Center game in C. We are going to make a simplified version of the game in real life and it is entirely a game of chance. Something important for us is that the people playing are sitting around in a circle and can have a certain number of chips. So each player takes turns rolling the dice which in turn chooses the amount of chips they can keep.

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?

Randomness is when two events truly have no connection or relation between them. The probability of one event is in no way effecting the others. So there is no pattern or predictability. We are using pseudorandom numbers in this assignment as true randomness is not something which can be predicted or replicated. Which is why we are using algorithm made numbers to replicate the dice.

What is an abstraction

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

In non computer science terms an abstraction is like something non specific or an idea that is not finished yet. A streamlined version of something more complex. I think abstractions have a lot of different meanings like abstract art or reading the abstract of a scientific paper as a few examples.

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 help debugging by allowing us to simplify strings of code into different functions. So it helps us make the testing process easier by identifying the bugs while we are developing the code. Hence, providing a different way to look at the code. Other then debugging, abstraction is used for hiding a lot of external details in a program. Abstractions are a programming technique. You don’t always have to make

them yourself. You can often use ones that already exist, which saves time and helps you put together more complex programs more efficiently.

Functions

When you write this assignment, you can choose 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.

With structuring the dice game for my program, the choice between two functions with the main code and utilizing eight functions (initializePlayers, rollDice, moveChips, updateGameState, checkWinner, displayResults, processPlayerTurn, and handleChipDistribution) can reflect a sort of trade-off between a more simplified and enhanced design. However, this approach may lead to a more complex function than expected. When it comes to the eight-function approach, they are more modular and have specifications within them. This would let each of the functions focus on each part of the game. This design would be better for flexibility. According to me I would consider which is more readable and maintainable to be easier.

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.

In this assignment, testing plays an important role in making sure that the dice game is correctly simulated. The program should be tested for all its inputs and if the number of players are in the range. Such as, if there are a minimum of 3 players and a maximum of 10 players. Along with that we know the logic and if the chips are distributed correctly. Lastly, is just making sure that the scenarios are all tested and that all of the random values work with those scenarios. Some examples are if there are 3 players with a random seed of 0, 6 players with a random seed that is a large value and if there are 4 players with a high seed value.

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?

By using a pseudorandom number generator and abstractions my testing becomes easier as it can be moderated to some degree. First the Pseudorandom number generator ensures that the sequence of the numbers is generated consistently. So that the same sequence of random numbers will be produced each time which in turn helps with debugging. Secondly the abstractions can also provide a clear structure of the code.

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 use this program, you will need to compile the program and then run it. First, you will need to open your terminal and navigate to the correct directory, which is `asn1`. After which you will run the program, which will then prompt the user by asking how many players in the game. The user will answer either a number between 3-10. After, the user will provide a random number. Some optional flags are not required, but when there is, the user can explore additional features that can modify certain aspects of the simulation like, for example, a help menu for more information. Overall, the program will be compiling and running the program within the terminal.

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]}.

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.

For those who have to maintain this program, the first thing I would have to mention is to make sure to understand how it is organized as a whole as it can help with any modifications that will be needed. Now this program will be running on data structures (array), algorithms, and the organization. One thing is to understand enums (names given to constants) as it will be used to represent the outcomes of a dice roll. Along with that is the algorithms that we will be using with the pseudorandom numbers as `random()` function will be setting the random seed. Lastly, the program will be following a more organized way as it will help with maintenance. The key algorithms involve the logic for rolling dice, distributing chips based on dice outcomes, and determining the game's end condition.

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

I will organize the code in a modular manner, as it will help with its readability. The main features that I will be coding are user inputs, which involve the user putting in the number of players. Next will be the random seed input, which will prompt the user to input the numbers. This random seed is set with `random()`, which is a function that helps with testing. Each player will start off with three chips. Loops will be implemented as well to simulate the rounds of the game. Things like the user's name or number of chips will be printed. The winner will also be shown after each round. Following that, is the error handling, which will print out error messages if the user inputs incorrectly. Overall, this modular design will help both the users and the maintainers for future use. So in a summed up way it is initialize the game settings, loop through each player's turns - so roll dice, move the chips based on the dice roll, and then checking for the game-end condition, lastly, declaring the winner.

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)
- 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.

¹This is my footnote.

-
- 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.

1. `initializeGame`: It Configures initial game settings. 2. `playerTurn`: It Manages actions during a player's turn. 3. `checkWinner`: It Checks if the game has ended and identifies the winner.

References

- [1] Wikipedia contributors. C (programming language) — Wikipedia, the free encyclopedia. [https://en.wikipedia.org/wiki/C_\(programming_language\)](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.