Jessie Zhu jelzhu@ucsc.edu CSE 13s, Spring 2021 Sunday, April 4, 2021

Assignment 1: Left, Right, and Center

Design Document

Left, Right, and Center is a game where there are 2-14 people in a circle all start with $3 and everyone takes turns rolling a dice 0-3 times corresponding their bank balance. ($3+ balance rolls 3 times)

1. LEFT means player gives person to the left $1
2. RIGHT means player gives person to the right $1
3. CENTER means player puts $1 into the center pot
4. PASS means player doesn't do anything

The game repeats until there is only one person that has $, which would be the winner.

The purpose of this lab is familiarize us with and use basic loops and conditional statements that we have learned about in class to build a simple game. A main part running my code is having an infinite while loop to keep the game going, properly handling each player's turn, as well as many if/else statements to manage the small details.

Design Process:

- This particular lab wasn't too complex so I just used the basic loop types we learned in class and wrote a pseudocode structure that made sense in my head (below), and then went back to add more details about particular parts, such rolling the dice and performing said action or checking win condition.
- I didn't think very far about optimizing my code/implementing functions, so there was **a lot** of copy and pasting that I could have avoided (that I should have avoided, I am deeply sorry for the mess), but everything was working so I didn't want to mess that up.
- I was able to smoothly turn the pseudocode on this doc into code, but had one issue, which I posted on piazza. I basically had my game running fine, except that it seemed like my dice sides were out of order (i copied and pasted the dice code from the doc). The dice had matching patterns with the given binary file but the sides seemed like they were out of order. What happened was that I had a line that did: side = die[index] and then after I compared the numbers like this: if (die[side] == LEFT)….. So the issue was that I used "side" as an index of the die, where I should have just competed "side" directly to LEFT, RIGHT, etc..
- There weren't any major adjustments in code structure when I turned the pseudocode into C code, except for small details special to how c code is structured. I had to double check my variable types to make sure I was declaring/ printing the correct types, which I had never done before with python.

Pseudocode:

Main function, the skeleton of my code:

In **main**, the infinite while loop keeps the game going, playing each players turn. During each players turn, the amount of money they have at the beginning of the turn determines how many times the roll action will be happen. The while loop's condition keeps the game running until there is only one person that is active in the game.

Sides of dice = [LEFT , RIGHT , CENTER , PASS , PASS , PASS]

Main:

Ask for random seed, validate, and set the seed

Ask for number of all players, num_players, and validate

active_players = players still in game, begins as number of all players

money = [3, 3, 3, 3, 3, 3, 3, 3, 3, 3] (aka bank, actual length is number of all players)

curr = 0 (index of current player)

center = 0 (center pot money)

While (there is at least more than 1 active players):

if the curr player has $3 or more:

ROLL* (3 times)

if curr player ends with $0, active players -1

elif player has $2:

ROLL* (2 times)

if curr player ends with $0, active players -1

elif player has $1:

ROLL* (1 time)

if curr player ends with $0, active players -1

else player has $0:

do nothing, skip

next player's index: curr player's index +1 mod 10

Get and Display **WINNER***

Below are helper parts: (I didn't make them sub-functions and just wrote the code out instead)

Roll: simulates rolling the dice, and then performs the corresponding action to the rolled side. On a LEFT roll, the player loses $1, person to the left gains $1 and afterwards if they have $1(meaning they used to have $0), then we add 1 to the active_player counter, which helps keeps track of our game status. Same process for RIGHT roll. For CENTER, the current player loses a dollar to the center pot.

***ROLL:**

Roll side = sides of dice [random number 0-5]

If side rolled LEFT:

player to left +$1

curr player -$1

if current $ of left person is now $1, active players +1 (left player was revived)

Else if side rolled RIGHT:

person to right +$1

curr player -$1

if current $ of right person is now $1, active players +1 (right player was revived)

Else if side rolled CENTER:

curr player -$1

center +$1

Else: (rolled a pass)

do nothing

Winner: the infinite while loop ends when there is only one person with money left. The next part in main will be to find winner. Winner uses a for loop to find that person and then prints the winner message accordingly, displaying said player's money balance and the amount of money won from the pot balance.

***WINNER:**

for person in total players:

if this person has balance of not $0, then they are the winner

print winner info, bank balance, and balance of center pot

***END OF GAME, RETURN 0 ***