

Devan O'Boyle

Lab 1: Left, Right, & Center

Problem: Some number of players, $1 < k \leq 14$, sit around a table. Each player has in their hand \$3. There are three dice and each die has 6 faces and is labeled: $3 \times$, $1 \times L$, $1 \times R$, or $1 \times C$. As a result, we know that there is a 50% chance of rolling \circ , & 16.66% chance of rolling L, R, or C.

if player has \$3 or more, they roll 3 dice

if player has \$2 or more, they roll 2 dice

if player has \$1 or more, they roll 1 die

otherwise they must pass

if player rolls L, then they give \$1 to the player on the left

if player rolls R, then they give \$1 to the player on the right

if player rolls C, then they put \$ in the pot in the center

if the player rolls \circ , then they ignore it

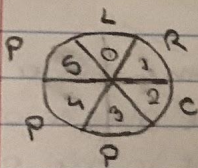
First step: ask for the number of players playing the game as well as a value for a random seed by taking user input - also check that the inputs are valid
using the array of philosopher names, pick out the first n number of names w/ n being the number of players entered by the user

iterate through each turn an infinite number of times.
the loop should only stop once there is a single player left in the game which happens when they are the only ones left with money

* values to keep track of: ^{both total & currently playing} # of players, position of current player, the amt of money that each player has, the amt of money in the center

on a player's turn, first check how much money they have, if they don't have any, skip them by iterating to the next turn, if they do, have them roll the number of dice corresponding to the amount of money that they have

- the rolling of dice will be done by iterating through the seed's predetermined pseudorandom values so that the program's output is reproducible
- in order to set each probability appropriately, each number from 0 to 5 will correspond to a given value of the die: { Left, Right, Center, Pass, Pass, Pass }



after the die is rolled, \$1 will either be passed to the left, right, center, or not at all

- the rolling of the die will be looped through for as much money that the player has, capping at 3 times for \$3 or more

Passing Money: if the die rolls Left, right, or center, then \$1 is subtracted from that player's total

Passing left: to pass money to the left, \$1 must be added to the player before the current one, which is done by taking the position of the current player & adding the # of total players subtracted by 1, then you take the remainder of this value divided by the total # of players

- once you have that player's position, you can add \$1 to their total

Passing Right: similar to passing money to the left except you find the position of the person to the right by taking the position of the current player, add 1 & take the remainder of it divided by the total # of players

Passing Center: simple, just add \$1 to the center pot

- after each dice roll, check to see if the player is out of money, if they are, then stop passing & subtract 1 from the # of players currently in the game, after that, move on to the next player's turn
- the while loop should check to see if there is only the 1 player left in the game with each iteration, once the loop breaks, the winner's name should be printed by looping through the players to find which one still has money, then that player's index will be used to print their name & the amount of money they one with - also the \$ in the center pot will be printed
- to keep track of the amount of money that each player has, create a separate array, the length of the total amt of players and set each element to 3 for \$3
ex. 4 players array $\rightarrow \{3, 3, 3, 3\}$
length of 4
each index of the array corresponds to a name in the list of names
- to keep track of the players that are still in the game & which one's aren't, another array will be used with each player having a default value of 1, meaning that they are still in the game
ex. 4 players array $\rightarrow \{true, true, true, true\}$

a player will have their value set to false once they run out of money, meaning that they are no longer playing.

- at the beginning of every iteration of the loop there will have to be a check to see if the player has money, but is still marked with false. this means that the player was out of the game, but was passed some money & is now back in so their value should be set back to true to indicate that they are playing.

- each time a player goes from a true \rightarrow false then the # of current players is subtracted by 1
- each time a player goes from a false \rightarrow true the # of current players is added by 1

if there is one player remaining, there will be a check before the infinite while loop exits to see if there was another player that was brought back into the game during that person's turn.

- this will be done by looping through all of the players and see if any of them have money but have not yet been marked with true in the boolean array.
- if there is such a player, then that player will be marked as true, & the program will continue the while loop.
- otherwise the while loop will break.