

Zack Traczyk
ztraczyk@ucsc.edu
03/30/21

CSE13S Spring 2021
Assignment 1 - Left, Right and Center

DESIGN DOCUMENT

OBJECTIVE:

Simulate a simple game called Left, Right and Center. The game involves $1 < k \leq 14$ players who all start with \$3 dollars. The players go around a circle rolling as many dice as dollars they hold. However, the dice are not 1-6, but are labeled with 3 x **•**, 1 x **L**, 1 x **R**, 1 x **C**.

GIVEN:

```
typedef enum faciem {PASS, LEFT, RIGHT, CENTER} faces;
faces die [] = {LEFT, RIGHT, CENTER, PASS, PASS, PASS};

char *philosophers[] = {"Immanuel Kant ",
                        "Martin Heidegger",
                        "David Hume",
                        "Georg Wilhelm Friedrich Hegel",
                        "Arthur Schopenhauer",
                        "Ludwig Wittgenstein",
                        "Karl Wilhelm Friedrich Schlegel",
                        "Friedrich Nietzsche",
                        "Socrates",
                        "John Stuart Mill",
                        "Plato",
                        "Aristotle",
                        "Thomas Hobbes",
                        "Rene Descartes"};
```

In the lab doc an enumerated list for the faces and the representation is already taken care of. This enumeration means that PASS = 0, LEFT = 1, RIGHT = 2, and CENTER = 3.

However, there is no need to use the integer value consciously since I can use faces to check equality.

DATA TYPES:

There are a few things that need to be kept track of during gameplay:

1. Player <ul style="list-style-type: none"> a. Name b. Balance 	Using a structure: <pre>typedef struct player = { char name[35]; int balance = 3; } player;</pre>
2. Number/ Order of players	Using an array of pointers pointing to structs: <pre>int numplayers; int *players[] = { &player1, &player2, ... }</pre>
3. Pot	Using an integer: <pre>int pot = 0;</pre>

APPROACH (PSEUDOCODE):

seed = ? input ?

numplayers = ? input ?

FOR numplayers:

 create struct with name = philosophers[idx]

 Players[idx] = address of struct

WHILE moneyleft?:

 FOR players:

 FOR player.money:

 roll = random(values of die)

 IF roll != faces.PASS:

 player balance --

 IF roll == LEFT:

 left player balance ++

 ELSE IF roll == RIGHT:

 right player balance ++

 ELSE IF roll == CENTER:

 pot ++