**Left Center Right (LCR) Game**

Developed by: Ryan St George

12/8/2023

**Table of Contents**

[**Special Features 2**](#_i17r8nmrmi7v)

[**Process Description 3**](#_xhrw2vmdyyxw)

[Compiler Command 3](#_drel0judg0yh)

[**Pseudocode 5**](#_aadvqt83ehds)

# **Special Features**

Throughout this LCR game there are multiple features which most are implied with a game/program of this level.

1. Error Handling
   1. Error handling in player names (must be letters)
   2. Error handling in number of players (must be a number)
   3. Error handling in Dice Rolling (see #2 for more details)
2. Rolling the Dice
   1. Its implied that 1 is L, 2 is R, 3 is C, etc.
   2. Hence only Letters are used to make the game more enjoyable
   3. Each “roll” when typed displays what the player had gotten such as L, L, C.
3. Rules
   1. The rules are displayed at the beginning of the game
   2. Ensure you have permissions to read the rule file locally
      1. In the future this can be adjusted and also added as a menu feature such as ‘Help’ or ‘display rules’.

# **Process Description**

This game was enjoyable to make. From each module slowly building on what was needed really took the stress out of building a complete game. After completing the modules I then read over the project guidelines again this week. After this I needed to understand the best file system layout for all the required files. I wanted to ensure it was scalable for any future additions.

So I began by adding a file system with resources, include, docs, and src. This allowed me to break everything apart. Keeping the header files, rules, and c++ files all within their own folders. From here I got to work. Starting with the rules which was my biggest struggle and I actually finished last. Sometimes when you struggle with a problem/part of code its good to leave it and come back to it, which is exactly what I did. I will get to this deeper momentarily. But I continued to work on the player option (previously partially completed), dice options, and then really threw it all together in the LCRGame.cpp and header file. After this I completed the main in running the game and compiling it.

### **Compiler Command**

I used this following compiler command FYI (likely can be cleaned up): g++ -std=c++11 -o LCRGame src/main.cpp src/LCRGame.cpp src/Dice.cpp src/displayRules.cpp -Iinclude

This is where I was almost complete.

I then wanted to address the rules issue. I first thought the rules file (previously able to be read in the module work) was not working due to a directory problem. So I moved it to the main directory outside resources and still to no luck. So maybe it was where it was being called so I tried directly in the main file with still no luck. So I sat on it for a day doing research and wanted to use some temporary error handling to see what directory I was in when running the executable, and also the exact error on why the file was not read. It turns out it needed permissions! So I dug deeper and ensured my macbook in this area had read & write permissions. It in fact did not! So in simply running sudo I was able to get it to work and I kept it that way.

Overall it has been fun and really a great project I am proud of. It may not be the most advanced to what real world day people may work on (heck I don’t know how much developers do in a day, but I would love to know). But it definitely carries a lot of principles and overall mindset to the real world and I hope that is portrayed in this assignment and also documentation.

# **Pseudocode**

BEGIN

class Player

Properties: chipCount, name

Methods: addChip(), removeChip(), getChips(), getName()

class Dice

method: roll(numberOfDice)

class LCRGame

Properties: players, dice, gameInProgress

Methods:

playGame(),

passChipToLeft(),

passChipToRight(),

checkGameEndCondition(),

announceWinner(),

isValidName(name)

function main()  
 will initialize LCRGame

call LCRGame.playGame()

function rollDice() returns an array of char

create array of char results

for i from 1 to 3

results = Dice.roll()

return results

LCRGame.playGame()

Display Rules

Ask and validate player count

Ask and validate for each player

While game in progress

for loop for each player

if have chips

roll dice

display each dice result

display remaining chips

Check if game has ended

If game ended

Announce winner

Else

continue

END