Assignment 3 Design

Shirin Rokni
shrokni

October 15, 2022

1 Purpose

The purpose of this program is to simulate a single game of dreidel. The simulation of dreidel is played with up to 8 players where each player, in alphabetical order, spin the dreidel and take one of the four actions: Nun (nothing), Gimel (the player takes the entire pot), Hayh (the player takes half of the pot, rounding down) and Shin (the player puts a coin into the pot). If a player is required to put a coin into the pot and they have no coins, they are out of the game and can no longer participate, meaning that they are eliminated from the game. After the game ends, the program prints out the name of the winning player, how many people that were playing, the amount of coins they all started with, how many rounds it took to complete the game, and the random number seed.

2 dreidel.h

purpose: contains the definitions for functions defined in dreidel.c and used by other code.
defines the spin_dreidel function
defines the play_game function

3 dreidel.c

purpose: contains the dreidel game logic

3.1 include statements

```
# include <stdio.h>
# include <stdint.h>
# include <stdbool.h> // so that I could use Boolean operators
# include "dreidel.h" // calling the dreidel header file
# include "mtrand.h" // calling the Mersenne Twister pseudorandom number generator functions
```

3.2 variable definitions

defining the players as an array that has a pointer
defining the bool 'elim' as false

3.3 definition of the spin_dreidel function

purpose: returns one of the four NGHS dreidel options randomly
assign an int variable 'value' to the random value generator function from mtrand, modulo 4
a switch statement with the expression being the 'value' variable
- case 0 (if value == 0)
  - return the 'G' character then break
- case 1 (if value == 1)
  - return the 'H' character then break
- case 2 (if value == 2)
  - return the 'N' character then break
- case 3 (if value == 3)
  - return the 'S' character then break
- default case

3.4 definition of the player_count function

purpose: make a function that counts the number of non-negative elements in the array
note: when calling the array that the number of coins are stored, use a pointer
- set a counter variable to 0
- a for loop that increments by 1 as long as it is less than the number of players in the game
  - if a person's coin is greater than or equal to -1
    - increment the counter
- return the counter

3.5 definition of the stop_dreidel function

purpose: carries out the commands of each dreidel side and prints the elimination statement
note: when calling the array that the number of coins are stored and the pot value, use a pointer
a switch statement with the expression being the spin_dreidel() function
- case 'G'
  - increment the player's coins by the amount in the pot
  - decrement the pot's value to 0
- case 'H'
  - increment the player's coins by half other pot's value
  - decrement the pot's value by itself divided by 2
- case 'N'
  - a break statement
- case 'S'

if the player's coins are greater than 0, decrement it by 1 and add to the pot's value with 1
if the player's coins are less than -1 (player is eliminated), and if the Boolean statement is called, print the elimination statement

3.6 defining the play_game function

purpose: actually plays a single dreidel game using all of the prior functions and returns the number of the player who won
define an array that will keep track of each player's coins and will have the length of how many players there are
a for loop that incrememnts by 1 for as many times as there are players
    filling the empty array that keeps track of coins with the beginning number of coins
defining a pot variable that represents the pot
a while loop that executes as long as there is more than one person in the player_count counter
    a for loop that increments by 1 for as many times as there are players
        if the player's coin value is greater than -1
            call the stop_dreidel function
    increment the number of rounds by 1
a for loop that increments by 1 for as many times as there are players
    if the player has more than 0 coins return their array index

4 play-dreidel.c

purpose: parses command line options to call functions in dreidel.c to play the dreidel game. It prints the name of the winning player as well as other metrics.

4.1 include statements

```
# include <stdio.h>
# include <stdbool.h> // for the boolean statements
# include <stdint.h>
# include <unistd.h>
# include "dreidel.h"
# include "mtrand.h"
```

4.2 declarations and definitions

define the flags "vp:c:s:"
declare the players array which will contain all of the players' names
declare the bool that will turn on and off the elimination print statement

4.3 main function

define the opt, people, coins and rounds variables set to default values
define the seed variable using uint64_t
a while loop that executes for as long as the flags are used
    a switch statement with the expression being the 'opt' variable
        case p (if the user does -p value)
            redefine p with the user's command line input
            if the user enters a number < 2 and >8, return a nonzero integer
            break
        case c (if the user does -c value)
            redefine c with the user's command line input
            if the user enters a number < 1 or > 20. return a nonzero integer
            break
        case s (if the user does -s value)
            redefine s with the user's command line input
            if the user enter a number < 0 or > a 10 digit value, return a nonzero integer
            break
        case v (if the user does -v)
            make the Boolean equal true
            break
        default case
            return a nonzero integer
call the mtrand_seed function to create the seed integer
define a variable with the play_game function to return the array value of the winner
print a line which has the winner's name, how many people were in the game, how many coins they started out with, how many rounds it took them to win, and the random number seed