Assignment 3 Design Document

dreidel.c:

Includes all of the functions that contain the logic for the dreidel game.

spin_dreidel (void):

Generate random integer.
Take modulo 4 of integer
switch(integer):
case 0:
    return ‘G’
case 1:
    return ‘H’
case 2:
    return ‘N’
case 3:
    return ‘S’

play_game (num_players, num_coins, rounds pointer):

```
pot = 0
array play_coins [length:num_players] filled with int of num_coins
for (rounds =0; 1; rounds += 1):
    winner = -1
    for (i=0; i<num_players; i += 1):
        if play_coins[i] < 0: continue
        switch(spin_dreidel()):
        case ‘G’:
            play_coins[i] += pot
            pot = 0
        case ‘H’:
            play_coins[i] += pot/2
            pot -= pot/2
        case ‘N’:
            break
        case ‘S’:
            play_coins[i] -= 1
            if play_coins[i] > -1: pot += 1
            else if (print_mes): print elimination message
        if play_coins[i] == num_coins*num_players: return i
```

if winner = -1: winner = i
else: winner = -2
if winner != -2: return winner

play-dreidel.c:

Includes the main function which takes in command line options and plays the dreidel game, printing the winner, number of players and coins, and the seed.

main (argc, **argv):

names = [‘Aharon’, ‘Batsheva’, ‘Chanah’, ‘David’, ‘Ephraim’, ‘Faige’, ‘Gamaliel’, ‘Hannah’]
num_players = 4
num_coins = 3
seed = 613
print_mes = 0
take given arguments and run associated code:
-p n_players: num_players = n_players
-c n_coins: num_coins = n_coins
-s seed: seed = seed
-v: print_mes = 1
if (2 > num_players or 8 < num_players): return 1
if (1 > num-coins or 20 < num_coins): return 2
if (1 > seed or 9999999999 < seed): return 3
rounds = 0
result = play_game(num_players, num_coins, rounds)
print (names[result] num_players num_coins rounds seed)
return 0