

Assignment 1 DESIGN.pdf

About Program:

This program is a simplified version of the game of David Moffat's dice game called Pass the Pigs. Its purpose is to introduce the C language.

The rules of this game are as follows:

- A game played with k players such that $2 \leq k \leq 10$
- Players take turns in cyclic fashion rolling "pig" dice
 - Rolling Side yields 0 points and ends current players turn
 - Rolling Razorback or Trotter earns 10 points
 - Rolling Snouter earns 15 points
 - Rolling Jowler earns 5 points
- Game ends when any player has earned 100 or more points

Files included in asgn1:

- names.h
 - The array of player names provided by assignment doc
- pig.c
 - The implementation of the game
- Makefile
 - Directs the program compilation, building pig from pig.c
- README.md
 - Describes how to use the program and includes errors or bugs known
- DESIGN.pdf
 - Describes the design and design process for the game

Pseudocode/Structure:

Prompt user for number of players and scan input from stdin (if input invalid, then print an error to stderr and default to 2 as number of players)

Prompt user to enter random seed (if input invalid then print error to stderr and default to 2021 as random seed)

Set random seed and set each player to 0 points \rightarrow `int arrayOfPoints[amt of players] = {0}`

Array of players[amt of players] **use given names**

While var = true *#loop that runs the checks of winning score and adds scores of each player based on rolls*

For players \rightarrow `(int t = 0; t < amtOfPlayers; i++)` *#loop through all the players' scores*

If the current player won \rightarrow `(score[t] >= 100)` *#check each score*

Print congrats + player name (`player[t]`)

```

    Var = false
    Break
Switch cases for rolls
Case side:
    No points added and go to next player
    Break
Case razorback or trotter
    Add 10 points → (score[t] += 10) and roll again
Case snouter
    Add 15 points → (score[t] += 15) and roll again
Case jowler
    Add 5 points → (score[t] += 5) and roll again

```

Design Process/Brainstorming:

What's needed + how to tackle:

- User input for number of players
- Input for seed (figure out later)
- An array for dice/pig (alr have)
- An array for players (player[numOfPlayers])
- Array for player scores (score[numOfPlayers] = {0})
 - Loop through all the players (for int i = 0; i < num of players; i++)
 - For each player roll the dice using random and depending on roll:
 - Add points and roll again or no points and go to next roll
 - If roll = side
 - Check points (see last bullet)
 - No add, continue → next player rolls
 - If roll = anything else
 - Score[i] += points of roll
 - Check points (last bullet)
 - Roll again for same player → figure out how to do
 - After each roll, check the amount of points they have to see if they won
 - If >= 100 points then exit

Errors:

The only error known in this program is that the output (depending on seed) sometimes differs from the output of the binary file given in the assignment's resources folder.