Brian Nguyen
Assignment 1 DESIGN.pdf

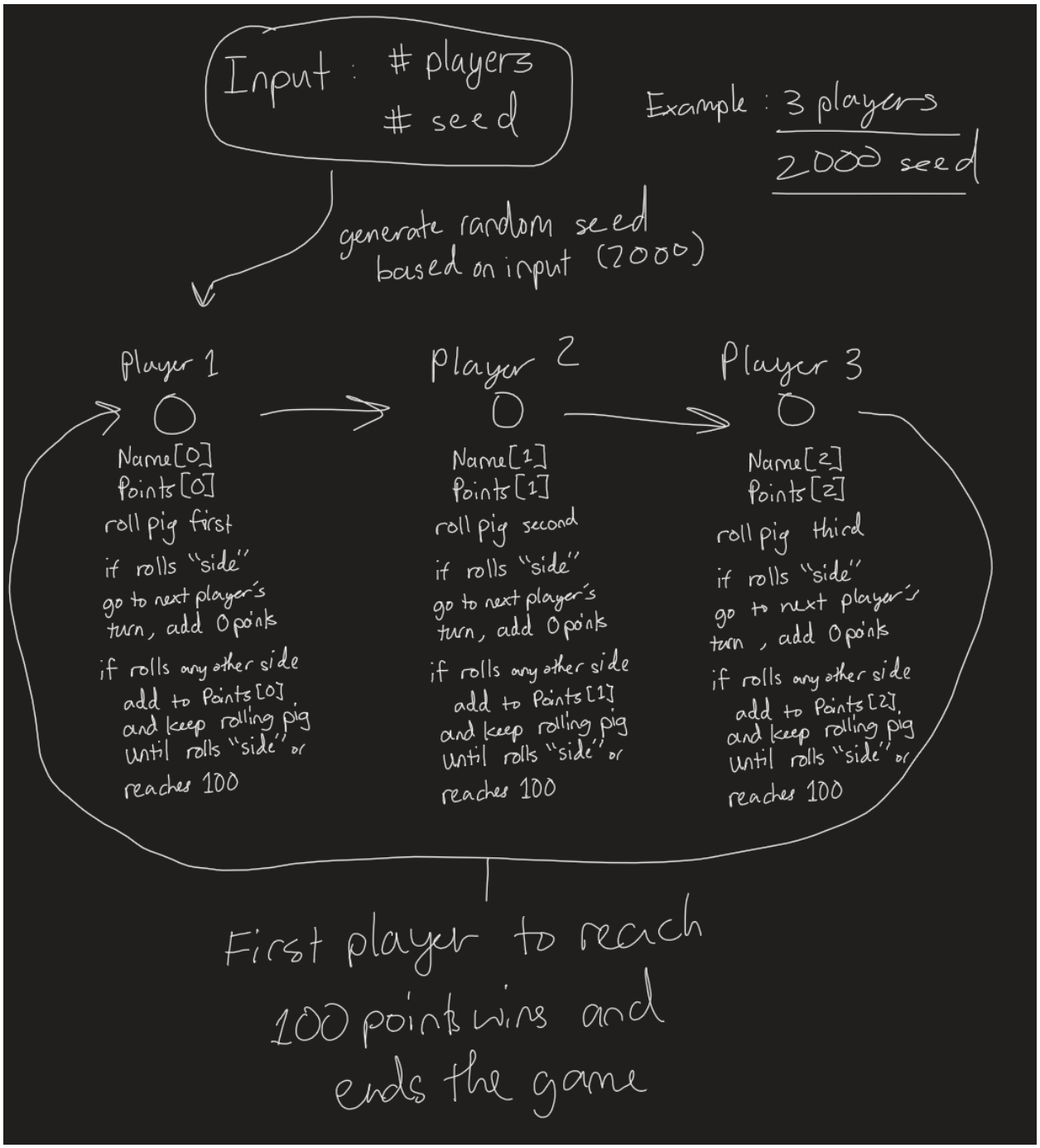
Description

This program simulates a game of “Pass the Pigs” based on a random positive 32-bit integer seed with a minimum of 2 and a maximum of 10 players. All players start with 0 points with the objective of trying to reach 100+ points first. To earn points, players must each cyclically take turns “rolling” the pig. Much like dice, a pig has 7 sides: 2 “side” sides, 1 “razorback” side, 1 “trotter” side, 1 “snouter” side, and 2 “jowler” sides. Each different side corresponds to an additional amount of points added with rolling either “razorback” or “trotter” earning 10 points, “snouter” earning 15 points, and “jowler” earning 5 points. A player stops rolling and forfeits their turn when they roll a “side,” which gives them 0 points and gives the turn to the next player in the circle.

Files

- Pig.c
    - The source code file for the program which contains all the logic that allows the game to simulate.
- Names.h
    - The header file that provides the names of the players that will be “playing” the game in an array.
- Makefile
    - The file that formats the program to be used in clang-format and compiles it to be run.
    - Also, the make clean function must remove compiler-generated files.
- README.md
    - The markdown text file that contains the description of the project and how to run it.
- DESIGN.pdf
    - The manual of the program expressed thoroughly with pseudocode and visualizations.

Layout/Structure



Pseudocode/Description

#Initializes needed files and default value of seed if error occurs

Include/import all files (including names.h)

Define default 2021 random seed

#Associates Positions of pig to numbers to give numbers meaning

Define enumeration of positions

#**Sets each position in an array**

Define pig positions array

#Asks for total players and prints error if invalid

Initialize total players variable

Scan for input from user for amount of players (between 2 and 10 inclusive)

Print error if it is not in range or invalid and leave error msg + default to 2 players

#Asks for random number for seed and prints error if invalid

Initialize seed variable

Scan for user input for random seed

Print error if seed is not a integer and leave error msg + default to 2021 seed

#Sets seed based on input

Set random seed based on input

#Makes an array of points to keep track of each players points based on index which correlates to their names

Make points array based on players

#Initialize variable that will keep track of current player rolling

Make current player variable

#Starts rolling and keeps making players "roll" until 100 points is reached by some player

While points of current player < 100

Print current player rolling

Roll number between 0 and 6 based on seed

If pig lands on x side

Add points to array index of current player based on side

If pig lands on "side"

Increment Current player variable to go to next player's turn

Don't add any points

#Prints victory message for player that won

Print stdout msg that tells current player they won and celebrate

NOTES about Pseudocode:

- For checking for invalid input, it will both look for if it is within range of 2-10 for players and use scanf to detect if an int was inputted by checking scanf()'s return value
- Same with seed except it just checks if an int was inputted

Error Handling

- I tried to stay close and tried to replicate the reference binary program in asgn1 resources:
    - When inputting invalid input for players, it errors for both players and random seed but doesn't crash and uses default values
    - When inputting 123asdf or 123.11135 it will take "123" and work as normal
    - Random seed takes in input as a %d int rather than an unsigned integer so some edge cases may vary, but all things I tested as well as pipeline has worked well. The program does not crash when things are inputted out of its int range so I left it.
- The errors will be correct in that it will never accept a data type that excludes ints entirely such as asdf.

Credits

- The Assignment 1 PDF that Professor Long has put out for us
- The C Programming Language book
- The recording of the lab section Eugene held on 9/28
- man scanf() for showing the return value of scanf to handle invalid input (also google search about scanf() return values)