Edison Chen

Assignment 1: Pass the Pigs DESIGN.pdf

Program Description:

The program implements a simplified version of David Moffat's dice game *Pass the Pigs*. The game can be played with k players, such that 2 <= k <= 10. The players are arranged in a cyclic fashion and will take turns rolling an asymmetrical die (the pig) to earn points. The pig has 5 different possible outcomes, although some outcomes are more likely than others. Rolling SIDE has a 2/7 chance of occurring and gives 0 points to the player in addition to immediately ending their turn. Rolling RAZORBACK has a 1/7 chance of occurring and earns 10 points for the player. Rolling TROTTER, like RAZORBACK, has a 1/7 chance of occurring and earns 10 points for the player. Rolling SNOUTER has a 1/7 chance of occurring and earns 15 points for the player. Finally, rolling JOWLER has a 2/7 chance of occurring and earns 5 points for the player. The game ends when any player has earned 100 or more points. A player keeps rolling the pig during their turn unless they roll SIDE or win the game.

The program first asks the player to input the number of players. If the user enters a number, k that is less than 2 or greater than 10, the program will print an error message and use a default of 2 players instead. The program will then prompt the user to enter a random seed. If the user enters a number outside of the range of a valid seed, (0 > rand_seed or rand_seed > MAX) then the program will print an error message and use 2021 as the default instead. The random seed will then be used as an argument for the function srandom(), which will set the seed to generate a sequence of pseudo random numbers called by the function random() unique to the seed. The same sequences of numbers from random() can be repeated by using the same seed inputted into srandom().

Files included:

- pig.c - source file that contains main() and implements the program.
- names.h - header file containing player names to be used in the game. Player names are indexed and the order of players will never change in the program. Ex. Willbur is always player 0 and so on.
- README.md - text file in markdown format that describes: the program, how to build the program, and how to run the program.
- Makefile - file that builds the program, runs the program, and formats all files to clang format.
- DESIGN.pdf - pdf that describes the program and design

Pseudocode:

Prompt user for number of players

If the input is anything other than in the range of 2 and 10 (inclusive)

    Return error message and use 2 as the default number of players

Prompt user for random seed

If the input is anything other than in the range of 0 and MAX of unsigned int (inclusive)

    Return error message and use 2021 as the default seed

Use the random seed input (or 2021 if invalid input was entered) as an argument to srandom to set the seed. (unsigned int is the type of argument the function srandom takes according to man srandom)

Set the players' "roll" as a random number modulo 7 to get a number between 0-6 (inclusive) and use the number as index of the pig array provided in assignment 1 pdf (mentioned further in

credits below) to randomly choose out of (SIDE, SIDE, RAZORBACK, TROTTER, SNOUTER, JOWLER, JOWLER).

Start at player zero's turn then advance in a circular fashion to player one and so on

While current player's points < 100

    Roll until either points >= 100 or player rolls SIDE

If player rolls SIDE

    Add 0 to the player's points

    Advance the player number by 1 so after player 0, is player 1, etc.

If player rolls RAZORBACK

    Add 10 to the player's points

If player rolls TROTTER

    Add 10 to the player's points

If player rolls SNOUTER

    Add 15 to the player's points

If player rolls JOWLER

    Add 5 to the player's points

If all the players have gone and no one has won

    Set the player number (tells who is currently rolling) to 0

If any player's points are greater than or equal to 100

    Stop rolling and print the current player's name and their points

Credit:

- I used the code provided in the assignment 1 pdf by Professor Long for the pig array containing all possible outcomes of a roll. Typedef enum {SIDE, RAZORBACK,

TROTTER, SNOUTER, JOWLER} Position defines the enumeration Position, and Position pig[7] = {SIDE, SIDE, …} is an array of Position to contain all 7 possible outcomes. I also used the code from the assignment 1 pdf containing the scanf, and print error statements in my program.

- I included the make file provided in the resources repo on GitLab.
- I attended Eugene's section on 9/28/21 in which he introduced the idea of using mod and random() to get a random value that could then be used to index the pig array. Another student attending also mentioned an int array as a way to track players' points which I also implemented in my code.
- I used the example DESIGN.pdf posted by user auds_o in the class discord as a reference when writing this.