Pass the Pigs Design.pdf

Program Description

This program represents a game in which a certain number of players roll an asymmetrical dice, referred to as the pig. There are five different sides to the pig, and each side represents a different action taken. Four of the five sides grant the player a given number of points, while the other side ends the player turn. Each turn goes until the turn end side is rolled, or the player reaches a winning score. For the purposes of this assignment, the winning score is 100. The program takes two inputs, the number of players between 2 and 10 inclusive, and the seed for random number generation.

Files included

pig.c: Code that contains main() and the majority of the code for the simulation of the game

names.h: File that contains an array of length 10, with names of players.

Makefile: Formats program into clang format and compiles it into an executable.

README: File in markdown format that describes building and running the program

DESIGN.pdf: Describes the design for the program with pseudocode and illustrations.(This document)

Pseudocode/Structure:

Get user input for the number of players, between 2 and 10 inclusive. If input is not specified, print an error message and use 2.

Get user input for random seed, between 0 and maximum value of unsigned integer, inclusively. If invalid input is given, use 2021 instead. The constraints on this input are to make sure seed is a valid input for the srandom function.

```
While the maximum player score is less than 100.
    Iterate through the number of players in a cyclical fashion and begin player turn.
    Print that the current player rolls.
    The below code of the player rolling the pig is in the turn() function.
        Player rolls the pig.
            If the player rolls SNOUTER.
                Add 15 points to the player’s score.
                The player rolls again.
            If the player rolls SIDE.
                End the player’s turn.
            If the player rolls JOWLER
                Add 5 points to the player’s score.
                The player rolls again.
            If the player rolls RAZORBACK
                Add 10 points to the player’s score.
                The player rolls again
            If the player rolls TROTTER
                Add 10 points to the player’s score.
                The player rolls again.
            In all cases above, print what the player rolls.
Once exiting the loop print the name and score of winning player
```

More specific notes about Pseudocode

- typedef enum { SIDE, RAZORBACK, TROTTER, SNOUTER, JOWLER } Position; const Position pig[7] = { SIDE, SIDE, RAZORBACK, TROTTER, SNOUTER, JOWLER, JOWLER };
    - This code is to create an enumerated type containing the various rolls of the pig
    - The constant array pig[] creates the asymmetrical dice used.
    - This code was provided in the assignment documentation (© 2021 Professor Long).
- Score tracking
    - The player scores are tracked via an array of ints.
- turn()
    - This function encompasses the entirety of the player turn.
    - It returns the score at the end of the player's turn.
    - The player's turn ending is tracked by if SIDE is rolled or if the player score is 100 or greater.
- Outer while loop.
    - The program will use a do-while loop that tracks if the score of the current player is 100 to see if the game will end.
    - This loop will increment the current player number in a way that stays within the input player count so that it does not get out of the indices of the scores or player name arrays.
- Inner while loop
    - This loop is used within the turn function.
    - This loop tracks if the individual player turn ends.
    - The loop uses two conditions, if the current score of the player is under 100, and if SIDE is not rolled to keep rolling a player turn.

- Header files
    - Includes names.h, which is given in the resources document (© 2021 Professor Long). This document provides the names of players.
    - Includes limits.h, which is used for UINT_MAX to check seed input.
    - Include stdio.h which is used for the scanf() and printf() functions, take user input and print output.
    - Include stdbool.h, which is used to make boolean values to track if a player turn ends.
    - Include stdlib.h, which is used to generate random values via the random() function.

Error Handling.

- If the user inputs an invalid (outside of the range of 2 and 10) number of players, the program should show an error message and use a player count of 2.
- If the user inputs an invalid seed (outside of the range of 0 and the maximum value of unsigned integer), the program should show an error message and use a player seed of 2021.
- If the used inputs a non integer input for the player count value, the seed will also be an invalid input. This is due to the functionality of scanf() not clearing input when the type does not match and cannot be converted. This functionality is ok, as stated by Professor Long in lecture.