Assignment 1 - Pass the Pigs

Asaveri Rao

CSE 13S - Fall 2023

1 Purpose

The main purpose of this game is to serve as a dice game and be a simplified version of Pass the Pigs. This game helps build C programming skills. The skill set that is being tested is primitive types, loops, conditionals, arrays, and user input. The objective of the game is to accumulate 100 points through rolling a die. Each side is labeled with different names, and two of them have a higher probability of getting chosen.

2 How to Use the Program

The program consists of multiple steps to execute the game. The program asks the user for a few different inputs, the number of players and a random seed. After attaining that information, the program enumerates through an array, and runs a pseudo-random number generator that simulates the rolling of a die.

3 Program Design

The program's design is set up in a fairly simple way. There are a few user-input formatted strings that get information from the users for the game. Libraries and files like stdio.h, stdlib.h, and names.h are also included in the program for immediate access. There are also if statements that place conditionals for certain aspects of the game like, the number of players and and seed number to keep it in check. Lastly, there are arrays present in the program as well. The array Position signifies the every side the die could land on, which is why some sides are repeated more than once.

3.1 Data Structures

The program for this game consists of a few different data structures. The major data structure utilized was an array. This was used as a way to access the names of the players and the sides of the die. If any other data structure was used over an array it would over complicate the code. An array's general structure also makes it easier to access each element within the files provided. Enumerationss were also used to assign a pig position to a specific value. There were other ways this could have been programmed, however enum already assigns each value in the array to the chronological order of 0, 1, 2, etc, thus making it more efficient to code. There were also strings that were used, for the other basic characters that needed to get defined. Strings are one of the more convenient and easier to use structures for this program.

3.2 Algorithms

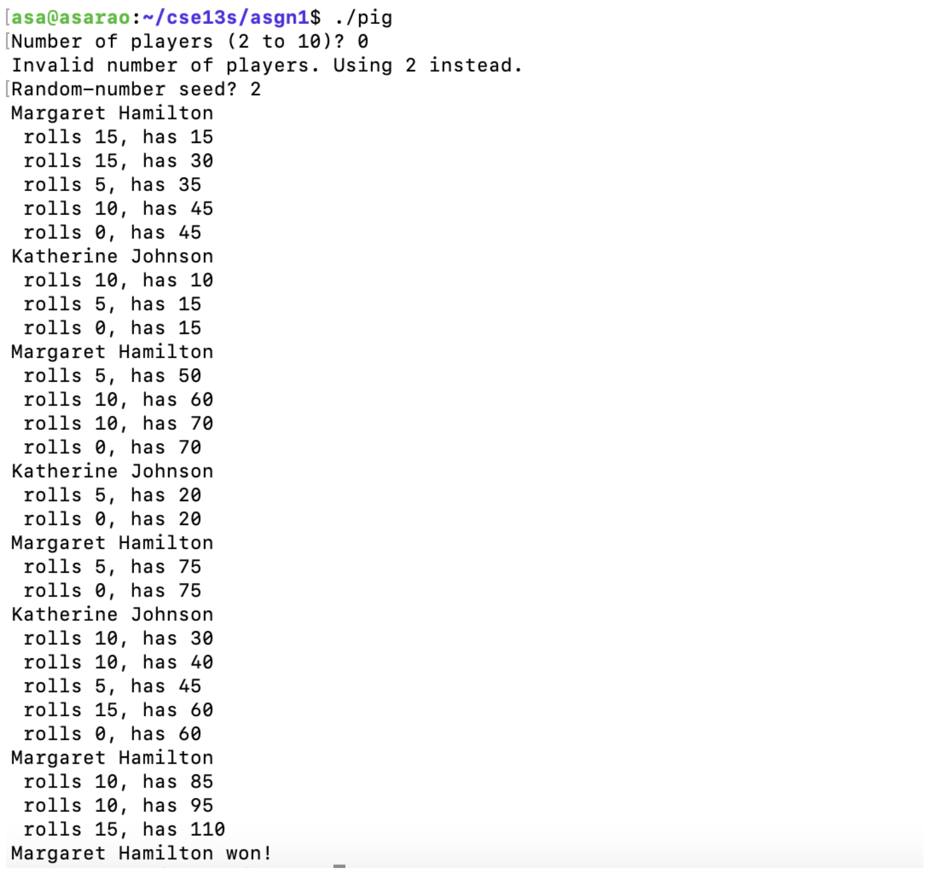
```
Declare a character variable random_elem
 Generate a random integer between the range of [0,6]
 Utilizing rand() % 7, which is stored into a variable
 The character is then accessible through an array,'pig'
 Then stored into 'random_elem'
 Integer value of 'random_elem' is then printed through printf
```

3.3 Function Descriptions

While coding the rest of the game with the code I had planned out, functions did not seem to be the most efficient way to calculate the scores or move on to the next player, so I did not utilize any. Instead I used while and for loops and the switch command to alternate through various cases for each side of the die.

4 Results

The intended purpose of the program was to successfully simulate the Pass the Pigs game, which is what was achieved with this code. There were three main aspects of the program that were important to execute in order for the game to run properly. The first one being defining variables and assigning the correct values to variables, arrays, and strings. The second major component being the die simulation, where we had to generate random numbers/sides to be rolled. This was also successfully executed through a series of integers and arrays. The final component that was essential for the program to execute properly was the adding the current player's score to their cumulative score and moving on to the next player. This part was where I was having a few challenges. Most of the challenges were minute errors like the formatting, whether that be an extra space or the loop for the next player not working correctly. This was fixed through debugging and rewriting certain parts of the code for the if statements in regards to the score calculator and next player.



```
[asa@asarao:~/cse13s/asgn1$ ./pig
[Number of players (2 to 10)? 0
Invalid number of players. Using 2 instead.
[Random-number seed? 2
Margaret Hamilton
 rolls 15, has 15
 rolls 15, has 30
 rolls 5, has 35
 rolls 10, has 45
 rolls 0, has 45
Katherine Johnson
 rolls 10, has 10
 rolls 5, has 15
 rolls 0, has 15
Margaret Hamilton
 rolls 5, has 50
 rolls 10, has 60
 rolls 10, has 70
 rolls 0, has 70
Katherine Johnson
 rolls 5, has 20
 rolls 0, has 20
Margaret Hamilton
 rolls 5, has 75
 rolls 0, has 75
Katherine Johnson
 rolls 10, has 30
 rolls 10, has 40
 rolls 5, has 45
 rolls 15, has 60
 rolls 0, has 60
Margaret Hamilton
 rolls 10, has 85
 rolls 10, has 95
 rolls 15, has 110
Margaret Hamilton won!
```

Figure 1: Results after invalid input of 0

5 References

Kernighan, Brian Wilson, and Dennis M. Ritchie. The C Programming Language. Prentice-Hall, 1978.