Assignment 1 – Pass the Pigs

Sebastian Avila

CSE 13S – Fall 2023

Purpose

This program implements a version of David Moffat's dice game: "Pass the Pig."[1]. The program allows for any k players, such that $2 \leq k \leq 10$, to take turns rolling a dice to earn points. The dice's asymmetrical sides have corresponding point values and actions. The first player to reach 100 points is the winner of the game.

How to Use the Program

To use this program, compile using `clang pig.c -o pig` and run using the command `./pig`. You must have the file names.h in the same directory. The program will then prompt you to enter the number of players. To do this, type a number into the command line and press enter. The program will validate your entry and then prompt you for a number to use for the random seed. Enter this number in the same manner as before. This seed is the starting point from which the random numbers will be generated. After submitting input for the seed, the program will validate your entry then run the game, announcing each player's turn and eventually the winner.

Program Design

The program first defines an enum called Position with 5 named constants each with a specified value. A constant array named pig with each pig position assigned to an index 0-6. The program then declares an integer variable for the number of players and gives it the value 2. It then prompts the user for the number of players and checks the result. If the user's entry is valid, the program gives the number of players variable the value of the user's entry; if not, then the value remains equal to 2. An unsigned integer variable is then declared and initialized to 2023. This variable stores the random number seed. A prompt is printed telling the user to input a seed. This entry is checked. A valid entry is then stored in the seed variable. An invalid entry causes the seed value to remain equal to 2023. An integer array with length equal to the number of players is created to store each of the players score. 3 integer variables are declared and initialized to 0. These variables store the state of the game, turn, and the current roll. To simulate the game, the program uses 3 nested loops. The outermost loop continues until the a player reaches the score 100 or greater. The second loop loops through the list of players allowing each player to take their turn. The innermost loop continues until the current players turn is over.

Data Structures

This program contains an enum named `Position`, a constant array named `pig` and two arrays named `player_name` and `player_scores`. I chose to use an enum to create the different pig positions because it allowed me to set constant values that correspond to the sides of the pig/dice which could then be used in the pig array. I chose a constant array for the pig positions because each position would be assigned an index and they would not change. I chose to use an array to hold the player's scores because the values would be

[1] https://en.wikipedia.org/wiki/Pass_the_Pigs

easily accessible and I would be able to iterate through them. Because the players' names were given in an array, this meant that the index value of a player in the array of player names would correspond with the index value of that players score in the array of player scores.

Algorithms

```
game loop
    loop while game is not set to over
        loop through the players
            print players name
            set turn to not over
            loop while player's turn is not set to over
                set roll to random number
                add points to players score
                print roll value and player's points
                if roll == pig's side then
                    set turn to over
                if player's score >= 100 then
                    print players's name
                    set turn to over
                    set game to over
                    jump to end of players list
```

Function Descriptions

- This program takes two inputs. Both are integer values. The first is used for the number of players and will be checked to make sure it is a valid integer and that it is between 2 and 10. The second will be checked to make sure it is valid integer and will be changed to unsigned.
- This program outputs two prompts to the user; the first tells the user to input the number of players and the second tells the user to enter a number for the random seed. The program also prints the players name on their turn and prints the winners name once the game is over.
- The purpose of the main function is to simulate the pass the pig game with the amount of players and the random seed that the user entered.
- I chose to use to use a while loop for the outermost loop because it allows the game to continue for as long as necessary. I chose a for loop to loop through the players because it allowed me to use the increment variable as an index to access both the players score and the players name. I chose a while loop for each players turn because it allows the turn to continue until the player rolls the value needed to end its turn.

Results

My program works as intended as shown in Fig. 1 1 One thing that could be improved is when a user enters a non-numerical value when prompted for the number of players, the program will default both the number of players and the seed, skipping the prompt for the random seed entry. This is shown in Fig. 2 2 The program still achieves the correct output, but it would be an improvement if the program would only default the number of players and still prompt the user to enter a number for the random seed. This issue is also present in the pig_ref binary (Fig. 3) 3.

```
savila35@cse13s-vm:~/cse13s/asgn1$ ./pig
Number of players (2 to 10)? 90
Invalid number of players. Using 2 instead.
Random-number seed? 12
Margaret Hamilton
 rolls 15, has 15
 rolls 10, has 25
 rolls 0, has 25
Katherine Johnson
 rolls 15, has 15
 rolls 10, has 25
 rolls 10, has 35
 rolls 10, has 45
 rolls 15, has 60
 rolls 10, has 70
 rolls 0, has 70
Margaret Hamilton
 rolls 5, has 30
 rolls 10, has 40
 rolls 10, has 50
 rolls 15, has 65
 rolls 5, has 70
 rolls 10, has 80
 rolls 15, has 95
 rolls 10, has 105
Margaret Hamilton won!
```

Figure 1: Screenshot of the program running.

```
savila35@cse13s-vm:~/cse13s/asgn1$ ./pig
Number of players (2 to 10)? ]
Invalid number of players. Using 2 instead.
Invalid seed. Using 2023 instead.
Random-number seed? Margaret Hamilton
 rolls 5, has 5
 rolls 10, has 15
 rolls 10, has 25
 rolls 0, has 25
Katherine Johnson
```

Figure 2: Screenshot of the seed prompt being skipped.

```
savila35@cse13s-vm:~/cse13s/asgn1$ ./pig_ref < input2.txt > expect2.txt
Invalid number of players. Using 2 instead.
Invalid seed. Using 2023 instead.
savila35@cse13s-vm:~/cse13s/asgn1$ ./pig < input2.txt > outputt2.txt
Invalid number of players. Using 2 instead.
Invalid seed. Using 2023 instead.
savila35@cse13s-vm:~/cse13s/asgn1$ diff outputt2.txt expect2.txt
savila35@cse13s-vm:~/cse13s/asgn1$
```

Figure 3: Screenshot of the pig_ref and pig having the same output after error.