Assignment 1 – Pass the Pigs

Sebastian Avila

CSE 13S – Fall 2023

Purpose

This program implements a version of David Moffet's dice game[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`. 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. 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 and later used in a switch statement. 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 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.

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

Algorithms

```
game loop
    loop while game is not set to over
        loop through the players
            loop while player's turn is not set to over
                set roll to random number
                switch through possible cases
                    if roll is 0 or 1 (side)
                        end turn
                    if roll is 5 or 6 (jowler)
                        player score + 5
                    if roll is 2 or 3 (trotter or razorback):
                        player score+ 10
                    if roll is 4 (snouter)
                        player score +15
                    if none
                        error
                if player's score >= 100
                    end turn loop
                    end game loop
                    print winner's name
```

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

I have completed a rough draft of the program. I still need to go back and refine the output. I have several print statements that were used to test features and debug. These need to be removed before the final version of this program is submitted. Additionally, I still need to test my program thoroughly and make sure that the output is correct. Fig 1 1 shows the current output of my program. The program runs and a winner is shown, but I have yet to check if the game is running correctly and that the winner shown is the correct winner.

References

```
savila35@cse13s-vm:~/cse13s/asgn1$ ./pig
Number of players (2 to 10)? 2
Random-number seed? 3
Margaret Hamilton
rolling
4check
rolling
5check
rolling
0check
Katherine Johnson
rolling
0check
Margaret Hamilton
rolling
6check
rolling
0check
Katherine Johnson
rolling
1check
Margaret Hamilton
rolling
5check
```

Figure 1: Screenshot of the program running.