

# Assignment 1 – Pass the Pigs

Johann Lee

CSE 13S – Fall 2023

## Purpose

The purpose of this program is to emulate a simplified version of David Moffat's dice game, Pass the Pigs. Through the usage of randomized numbers, players are given points which are counted until a total of 100 points has been scored by a player. Once fulfilled this program will end with the 100-point player having won.

## How to Use the Program

Compiling the program through terminal requires the usage of the make command conjoined with the make-file.

This program is for the most part automatic however it does require some inputs by the user. First, is the total number of players and second is the seed for the specified run of the game. Once these values are accepted the program runs

## Program Design

### Data Structures

The main data structures within this program include enumerations of an array called pigs under the name Position. The reason for enum was to hold the values of the pig within a char which would aid in stopping any confusion while coding. There are also two scanf() statements which are the only user inputs of the program. Then the main game is encased around a do-while loop that waits for a player to reach 100 points. The reason for this while for once the game exited the inner layers it would fully exit based on a condition that had been achieved at the bottom. Furthermore, a for loop designating the current player is used as it holds a looping pattern within the while loop outside and can control which player is going. Within this for loop is another while loop which has the exit condition of rolling a side pig which is when the player should change. Finally, this innermost loop has a switch case meant to output based on the pig roll that the player does as this is the most efficient in creating outputs based on inputs of the pig roll.

Describe your data structures here. Data structures include things like arrays, enums, strings, and structs. You should also mention why you chose the data structures that you did.

---

## Algorithms

Algorithms to show key components are as defined. This pseudo code shows the main components.

```
Player Swap Algorithm
  loop from i = 0 to number of players
    while pig side is not rolled
      roll random pig
      switch (rolled pig)
        case side pig
          player changes and no points given
        case any other
          points given then rolls again
```

All the pseudo code here was explained in the previous section regarding usage and reason for use.

## Function Descriptions

My code used no functions besides the main function. The reason for this was because there was no need for specified methods to be utilized on multiple occasions. All the code in the main function has its one case usage.

## Results

The results of my program were a success and gave the expected output. Since there were no faults in my code it can be said that it was not lacking anything key. However, this does not mean it could not be improved. I think foremost I could have utilized fewer nested loops as that drastically decreases computing efficiency and increases the time for execution.

During my process, there were some steps that had to be taken before getting a successful result that I would like to mention. Foremost, I had issues regarding the lining up of my player array and points array. This was due to the for loop and how I lined up the counting algorithm. This in turn caused players to not be looped correctly and instead loop the first player over and over. Another key issue that I faced was initializing my player points array. In this case, I had copied the method the array was initialized in player.h which left the length of the array blank. This caused my array to not initialize properly. These were the main problems that I faced when coding the final iteration of my project.

## References