Assignment 1-Pass the Pigs

Kevin Gallegos

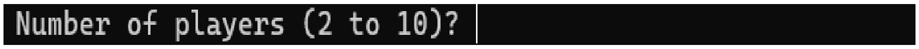
CSE 13s - Spring 2023

Purpose

Pass the pigs is a conventional dice game that revolves around competing against players to get to 100 points the fastest while rolling a die that determines the amount of points awarded on each roll. This program aims to simulate passing the pigs with the option of having 2, up to 10 players in the simulation along with choosing a seed to better simulate the randomness of rolling dice.

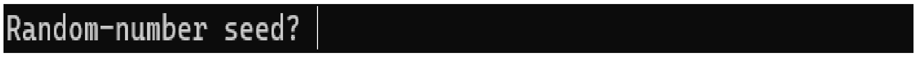
How to use the program

Once the program starts running you will be asked to answer a series of questions that set parameters for the simulation you will run with this program.
First you will be asked:



```
Number of players (2 to 10)?
```

Simply enter any number ranging from 2 to 10. This will set the amount of players that will be used in the execution of the program all containing different names and tracking their scores. The program will default to “2” if the user fails to input a number from this range or an invalid character.
The program then will ask:



```
Random-number seed?
```

Here the user will be allowed to input any integer number as long as it is not an invalid character or a negative number. The program will default to “2023” if the input is invalid.
After these are set, the simulation will execute accordingly without further input from the user.

Program Design

The program will start by defining all variables and arrays that will be used within the program. This will be followed by including a local header file containing the names of the players in the game and a prototype function acknowledging the function containing the loop of the program. The code will start by asking the user for a number of players (2-10) and for a seed, both of which go through checks to ensure proper input, defaulting to “2” players and “2023” as a seed if the user fails the input checks. From there a loop will run iteratively going through a list of players with a limit to the maximum number of players set by the user earlier, generating a random number each time to simulate a roll. The program will make the same player roll until a

0 is rolled or they hit a total points of 100 or more, In the event of rolling a 0 the next player will continue this process and so on. Once a player hits 100 they will be congratulated and the program ends. Because arrays will be used to identify players and such the variable containing the set number of players by input (defaultplay) will be subtracted by 1 as the arrays all start at position “0”. This assures that the input syncs up with arrays such that if “2” players was selected it would iterate through 2 array positions as “0, 1”, two positions. The loop that continues after the user inputs sets a variable (ID) to zero which correlates to the arrays’ positions that have the player names and scores will check (ID) making sure that after enough iterations it wont go over the maximum number of players the user imputed. From there a random number is stored in “r” and is fed into an array to determine the points they will earn for this iteration, this will continue until the player reaches 100 or gets 0 points on an iteration, in that event ID will be added by 1 signifying the next player's turn.

Data Structures

-playscore[] Contains the scores of every player, starts off as 0,0,0,0,0,0,0,0,0,0
Each position in the array correlates to a player, i.e. playscore[0] correlates to player_name[0]
The ‘ID’ variable syncs these properly
-points[], contains all possible points that can be obtained from a dice roll, will be 5,5,10,10,15,0,0

Algorithms

Pseudo code of the loop is as follows
Assuming token is the position for points array containing all the possible rolls you can get

```
while(ID <= maxplayers)
   while(playerscore[ID] not 100)
      if(ID greater than max players defined)
          Set ID back to 0
      Token  = generate random number 0 - 6
      player score = current player score + points[token]
      Print “I rolled this () current score: ())
      If player score greater or equal to 100
            Print ‘you win!’
      Else if point earned is 0
            Go to next player
```

Function Descriptions

Pig function is the only function that will be runned through main.

It accepts ‘defaultplayers’ as input for x amount of players that will be simulated playing, As Well as ‘defaultseed’. From there, the function will continue to the loop that goes through every player based on how many will play. A random number will be generated and from that number will pick a position in an array to determine the amount of points awarded and they will continue to generate random numbers and get points until they roll 0 or reach 100. The loop will move on to the next player if 0 is reached. If 100 is accumulated by a player the loop will break and congratulate the winning player.

Results

The code successfully compiles and outputs the desired function properly, displaying who wins and properly keeps track of scores and the players attached to them. Some issues were at one point outputting large numbers due to using ‘&myvar’ on some printf functions, this was changed. The output is as of writing this, varying compared to the test output. This is the only concurrent issue with the code.

References

Jessie tutor sections McHenry 0345 M 12:00pm-2:00pm, TH 10:00am-12:00PM