Pig (pig)

Dr. Caleb Fowler
1/12/24

Problem.

You are coding a simple game called Pig. Players take turns rolling a die. The die determines how many points they get. You may get points each turn your roll (turn points), you also have points for the entire game (grand points). The first player with 100 grand points is the winner. The rules are as follows:
Each turn, the active player faces a decision (roll or hold):
Roll the die. If it's is a:
1: You lose your turn, no turn total points are added to your grand total.
2-6: The number you rolled is added to your turn total.
Hold: Your turn total is added to your grand total. It's now the next player's turn.

Problem Features.

These are additional details to help you
Make a simple AI with a random number. Each tun decide what solve the programming problem.
the computer will do by rolling a 6 sided die, a D6: on 1-3 hold, on a 4-6 roll again.

Requirements.

These are the requirements for the assignment. This means they are general and apply to the entire assignment, rather than one specific part. Not every assignment will have a requirements section.

- Your code must be readable inside Canvas (no garbage files).
- Your code must compile under the gcc compiler (g++ compiler is an alias to gcc).
- Your code must execute under Ubuntu 14.00 or greater.
- Your output must be logically correct.
- Do not optimize for computer performance.
- Global variables in any form are forbidden. This includes #define statement's, too. You can, and should, put classes and structs right under your include statements and using namespace std; statement.

- You cannot use any data structures from the standard template library. No <vectors> or <arrays> or anything like that. You can use C-Style arrays, however. You can also build your own dynamic memory structures.
- Include a **Source File Header**. I'll look for comments like this:
  // pig.cpp
  // Pat Jones, CISP 413
  // 12/34/56
- Include a **ProgramGreeting** function which runs at the very start of the program. You can display the same information as in your source file header on the terminal.

Specification Bundles.

You want to comment where your code implementing the individual specification begins. You want me to find them. Use the grep trick (style, below) to prove you can see them. I specification documenting the start of a feature is plenty - you do not need to put a comment everywhere the implementation code appears.

"C" Specification Bundle.

// Specification C1 - Fixed Seed
This program will need to generate random numbers. Set the random number seed to a specific integer. That way, it will always generate the same random number sequence - easier to fix bugs that way. Use the unsigned variable seed to hold the seed value.

// Specification C2 - Student Name
Allow name to accept first plus last name (ie 2 words). Do not use 2 cin's or 2 string variables, just 1 variable. Call this variable **hPlayer**. Read this in from the keyboard after your program greeting runs. Display it somewhere in the output.

// Specification C3 - Numeric Menu
Use a numeric menu to collect the human players actions. See figure 1 for an example menu.

// Specification C4 - Three Functions
Include at least three functions in your program. Put this specification comment above your function prototype(s).

"C" Bundle Menu Options.
1. Roll
2. Hold
3. Quit

Figure 1: Allowable menu options for this bundle.

"B" Specification Bundle.

// Specification B1 - Display Turn Stats
Keep track of the points each player scores each turn as well as the overall points both sides have each turn in the game.

//Specification B2 - Function Activity to Disk
Write a message when every function is called. Save this in a file called “log.txt”. Include a timestamp and a message string for each line you save in your log. Log when a function is called - at a minimum. If you also log the time the function ended, you can then compute the Elapsed Time (or have the program compute it!) for your Peer Review. It is also VERY helpful for debugging to log the incoming and outgoing parameter values.

// Specification B3 - hiScore on Heap
Store the game’s high score in a variable in the heap. Call this variable hiScore. Don’t forget to clean up after yourself before the program quits. This holds the high score for all the games you have played since the program began execution now. You do not need to store anything in a text file - that would be between games high score.

// Specification B4 – Display hiScore
Display the value you stored in B3 on the console at the end of the game. If you played multiple games, it is the highest score of all games played.

"A" Specification Bundle.

// Specification A1 - D6() function
This method returns a randomly generated int between 1 and 6 every time it’s called. This function is used to generate random values for the game. Make sure the function name is D6. The function prototype should look something like this:
int D6();

This function works by always calling RandomNumber(1, 6). That is, D6() wraps around RandomNumber() and always provides the same parameters to it.

// Specification A2 - RandomNumber() function
Create a function which generates a random number between arguments lo and hi. The function prototype should look like this:

int RandomNumber(int, int);
Make sure you call your function RandomNumber.

// Specification A3 - Games Played Counter

If you implemented specification A4, record who won each game played. Store this in some sort of data structure (stack, queue, array, etc. etc.). Display the game number and who won, as well as a count of the total number of games played.

// Specification A4 - Play Again Option
Once the game ends, provide the player an option to play another game.

Due Date.

This assignment is due by 11:59 PM on Sunday on the date on the calendar in Canvas. Example, if this assignment appears on the Canvas calendar during week 2, the assignment will be due that Sunday at 11:59 PM. All the assignments are open the first day of class and you can begin working on them immediately. I encourage you to start sooner rather than later with your homework, these always seem to take longer than you think. This first assignment due date is different from the other assignments; you must turn this assignment in on time to remain in the class. You cannot turn this assignment in late. This is different from the other class assignments.

How to Turn in your Homework.

Turn homework in by uploading to the appropriate Canvas Dropbox folder. Save your homework as a .cpp file. Then rename it to a .txt file. Upload this .txt file to Canvas. Don't zip or otherwise compress your files. Do NOT split your file up into multiple files. I know that is a standard industry practice, but it just get's in the way for this class. I ONLY accept homework through the Canvas Dropbox. Do not add it to the comments or email me - I will not accept it. If you are having trouble submitting the assignment, email me immediately. Make sure you upload it a few minutes before the assignment closes in Canvas. If you go over by just one second - you are late.