Research Inventy: International Journal of Engineering And Science
Vol.10, Issue 8 (August 2020), PP 50-54
Issn (e): 2278-4721, Issn (p):2319-6483, www.researchinventy.com

Guessing DingKe Game Design based on Module Division

Guo Yan-fang
China, Chongqing,Yubei District,Taoyuan Avenue, No.1000
Chongqing Industry Polytechnic College, Chongqing, 401120

Abstract:*Guessing DingKe game is also known as the "rock, scissors, paper" game. It can realize the game's winning or losing function through using C language to write a programon the computer. The program is composed of main module, selection module, comparison module, output module and so on. The whole program uses the thinking mode of module division, defines global variables, enumeration types and function prototypes. According to the established rules, the program decides the winner by the choice between the player and the computer.*

Keywords:*C language;module division;"rock, scissors, paper"*

Date of Submission: 30-07-2020 Date of Acceptance: 14-08-2020

I. INTRODUCTION

Guessing DingKe game is a popular game that is easy to use, suitable for all ages, and is full of fun and fascinating. It has appeared in many occasions in daily life and is deeply loved by people. In this paper, we use the process oriented programming language- C language to design and write a simple "rock scissors paper" game, in order to deeply understand the application of C language.

II. GAMEFUNCTIONANALYSIS

The program design is divided into two sides: computer and player. Each side has three choices: rock, scissors and paper. One of them is randomly selected in the process of the game. Then the computer judges the result according to the rules, outputs the result, and continues the game until the end of the player's choice. In the process, players can also read the game guide or see the current situation.

Rules of the game: rock smashing scissors; scissors cutting paper; paper covering rock. The whole program is tested in DEV-C + + software environment.

III. GAME MODULE DIVISION

According to the functional requirements, the program is divided into four modules: main module, selection module, comparison module and output module.

3.1. Main module

Describes the main function that completes the entire game task. This function exists as an independent module. It can also be said that it is the first level task decomposition of the program, from which six functions to realize the game function are extracted.

3.2 Selection module

There are two functions in this module: player selection and computer selection. The keyboard receives user input and returns this input value, so no parameters are required, but there is a p_r_s enumeration type. The computer also generates a value for stone, scissors, and cloth and returns this value, so there is no need for a parameter, but there is also a p_r_s enumeration type.

3.3 **Comparison module**

The function in this module compares the value input by the user with the value generated by the machine to determine the win or loss. So it has to have two parameters, which are both p_r_s type, it should also have a return value, which is the result of judgment. This result is also an enumeration type. When the player wins, win is returned. When the player loses,it returns to lose. It returns to tie when tied.

3.4 Output module

There are three functions related output in this module. It is required to be able to display the user's input guide and tell the user how to enter his choice. It is required to be able to report the win and lose results and record the number of win and lose. It is required to be able to report the progress of a battle so far.

IV. REALIZATION OF GAME MODULE FUNCTION

For convenience, we write all Symbolic Constant definitions, type definitions and function prototype declarations in a header file, so that each module includes this header file. Then, there is no need for each module to write prototype declarations of those functions.

In order to improve the readability of the program, two enumeration types are defined in the program: the user's legal input p_r_s and outcome. The two enumeration types are defined as follows:

enum p_r_s {paper, rock, scissor, game, help, quit} ;

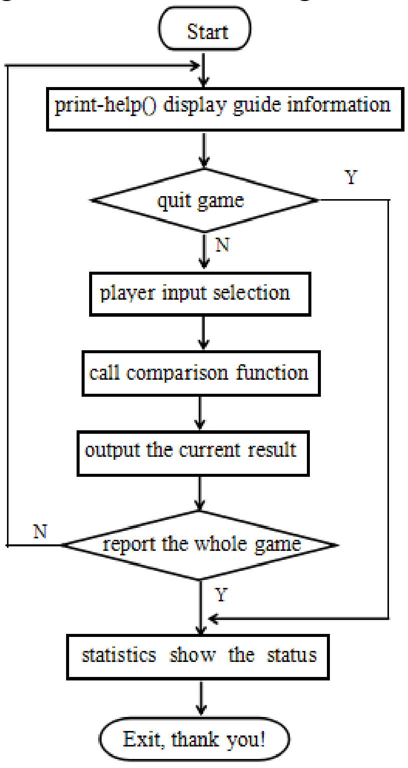
enum outcome {win, lose, tie} ;

The six values in the p_r_s type indicate that the selection of user table is paper, rock, scissors, view game situation, view game guide and exit game. The three values of outcome correspond to win, lose and draw respectively. Next, the program contains a detailed analysis of each function.

4.1 The main() function

The main function main () is the entrance of the whole program, which is responsible for the declaration of variables and the call of each sub function. Each function is finally called and executed directly or indirectly in the main function, and finally realizes the function of guessing DingKe game.

Figure1. Main flow chart of game design



4.2 The selection_by_machine() function

selection_by_machine() function is mainly used to get the choice of the computer. There is a p_r_s enumeration type. In the process of the game, the computer randomly generates a random number between 0 ~ 2, which represents the three choices of paper, rock and scissors.

4.3 The selection_by_play() function

selection_by_play() function is mainly used to obtain user input, and its return is also a p_r_s enumeration type. The switch-case-default loop structure is used to determine the function structure. The user's choice is represented by 'p','r','s','g','q' respectively to represent paper, rock, scissors, view the game situation and exit the game.

4.4 The compare() function

Since the computer has three choices and the player has three choices, there should be nine situations when comparing. But the compare() function compares with an equality the player_ choice == machine_ choice, which contains the results of three cases at once, and each clause in the switch-case structure can return the comparison result through the structure of a ternary operator, and the result is also an outcome enumeration type. The core sentence of judgment is as follows:

```
if (player_choice == machine_choice)   return tie;
switch(player_choice)
{
        case paper: result = (machine_choice == rock) ? win : lose;  break;
        case rock: result = (machine_choice == scissor) ? win : lose;  break;
        case scissor: result = (machine_choice == paper) ? win : lose;  break;
        }
```

4.5 The print_game_status() function

When the user selects' g ', the main program will call automatically in print_game_status() function,which is used to display the current combat situation. It is necessary to know the statistical results so far. Therefore, three parameters are required, namely, the number of wins, the number of losses and the number of ties, but there is no return value.

4.6 The print_help() function

In the process of running the game, the first output can guide the user how to make input selection, so it has no parameters and no return value.

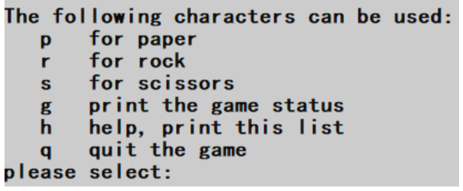
4.7 The report() function

The report() function is mainly used to report the results of the whole game, so we need to provide an information, that is, the comparison results of this game. As a matter of fact, there is also a function to realize the war situation of the game. In order to display the war situation of the game, there must be a record of the war situation. The most suitable recording place is in the report() function. When the current battle situation is reported, the statistical result information of the war situation is recorded at the same time. Therefore, in this function, in addition to displaying the results of this time, the results of this competition should also be recorded in the statistical information. This can be achieved by using the prefix increment operator.

V. GAME TEST RESULTS

Figure 2 shows that when the program runs the test, it first displays the guide information and prompts the player to input and select.

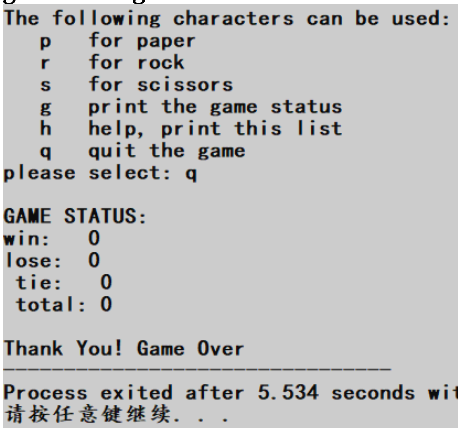
Figure2.The running guide results of the program



```
The following characters can be used:
   p   for paper
   r   for rock
   s   for scissors
   g   print the game status
   h   help, print this list
   q   quit the game
please select:
```

Figure 3 shows that when the game selector selects 'q' at the beginning, the program will output the battle information and exit the game.

Figure3.The game interface when select exit



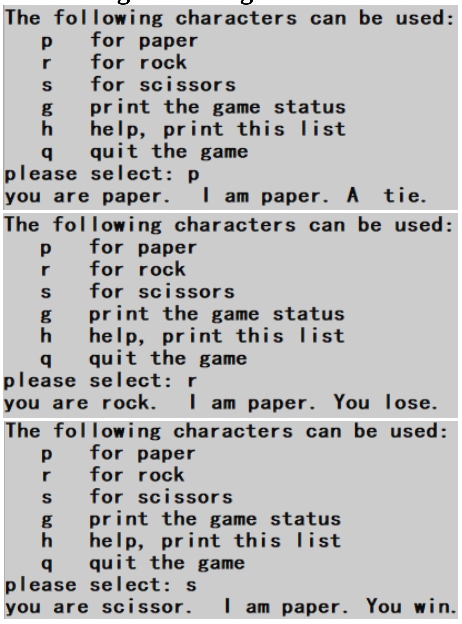
```
The following characters can be used:
   p   for paper
   r   for rock
   s   for scissors
   g   print the game status
   h   help, print this list
   q   quit the game
please select: q

GAME STATUS:
win:   0
lose:  0
 tie:   0
 total: 0

Thank You! Game Over
--------------------------------
Process exited after 5.534 seconds wit
请按任意键继续. . .
```

Figure 4 shows that when the game selector selects 'p','r','s', the program will automatically call the compare() function and output the winning or losing result.

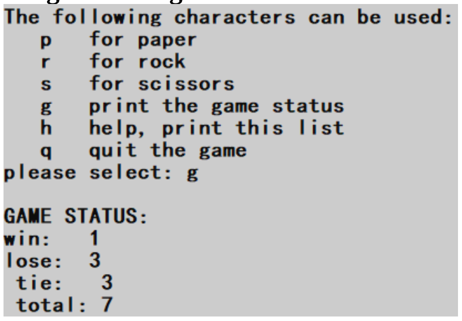
Figure4.The winning and losing interface of three situations



```
The following characters can be used:
   p   for paper
   r   for rock
   s   for scissors
   g   print the game status
   h   help, print this list
   q   quit the game
please select: p
you are paper.  I am paper. A  tie.
The following characters can be used:
   p   for paper
   r   for rock
   s   for scissors
   g   print the game status
   h   help, print this list
   q   quit the game
please select: r
you are rock.  I am paper. You lose.
The following characters can be used:
   p   for paper
   r   for rock
   s   for scissors
   g   print the game status
   h   help, print this list
   q   quit the game
please select: s
you are scissor.  I am paper. You win.
```

Figure 5 shows that when the player selects 'g', the program will give the game's overall combat information.

Figure5.The game statistics interface



```
The following characters can be used:
   p   for paper
   r   for rock
   s   for scissors
   g   print the game status
   h   help, print this list
   q   quit the game
please select: g

GAME STATUS:
win:   1
lose:  3
 tie:   3
 total: 7
```

VI. CONCLUSION

This game is based on C language, in DEV-C + + environment to complete the simple programming of guessing DingKe game, and the whole game process is tested. The results show that the program can pass the compiled syntax logic without errors, and the running results and theoretical predictions have no big difference, and can achieve the desired effect.

REFERENCES

[1]. Tan Haoqiang. C programming [M] (5th Edition). Beijing: Tsinghua University Press, 2017.
[2]. Su Xiaohong. Study Guide for practical course of C language university [M] (4th Edition). Beijing: Electronic Industry Press, 2017.
[3]. Yan Weimin. Data structure (C language version) [M]. Beijing: Tsinghua University Press, 2018.

Guo Yan-fang. " Guessing DingKe Game Design based on Module Division." *International Journal of Engineering and Science*, vol. 10, no. 08, 2020, pp. 50-54.