سلسلة الخوارزميات وحل المشاكل - المستوى الثاني

PROGRAMMING ADVICES

LEARN THE RIGHT WAY

26+ Years of Experience

Mohammed Abu-Hadhoud

MBA, PMOC, PgMP®, PMP®, PMI-RMP®, CM, ITILF, MCPD, MCSD

ProgrammingAdvices.com

```cpp
#include<iostream>
#include<cstdlib>

using namespace std;

enum enGameChoice { Stone = 1, Paper = 2, Scissors = 3 };
enum enWinner { Player1 = 1, Computer = 2, Draw = 3 };

struct stRoundInfo
{
    short RoundNumber = 0;
    enGameChoice Player1Choice;
    enGameChoice ComputerChoice;
    enWinner Winner;
    string WinnerName;
};

struct stGameResults
{
    short GameRounds = 0;
    short Player1WinTimes = 0;
    short Computer2WinTimes = 0;
    short DrawTimes = 0;
    enWinner GameWinner;
    string WinnerName = "";
};

int RandomNumber(int From, int To)
{
    //Function to generate a random number
    int randNum = rand() % (To - From + 1) + From;
    return randNum;
}

string WinnerName(enWinner Winner)
{
    string arrWinnerName[3] = { "Player1","Computer","No Winner"
};
    return arrWinnerName[Winner - 1];

}
```

```
enWinner WhoWonTheRound(stRoundInfo RoundInfo)
{

    if (RoundInfo.Player1Choice == RoundInfo.ComputerChoice)
    {
        return enWinner::Draw;
    }

    switch (RoundInfo.Player1Choice)
    {
    case enGameChoice::Stone:
        if (RoundInfo.ComputerChoice == enGameChoice::Paper)
        {
            return enWinner::Computer;
        }
        break;

    case enGameChoice::Paper:
        if (RoundInfo.ComputerChoice == enGameChoice::Scissors)
        {
            return enWinner::Computer;
        }
        break;

    case enGameChoice::Scissors:
        if (RoundInfo.ComputerChoice == enGameChoice::Stone)
        {
            return enWinner::Computer;
        }
        break;

    }

    //if you reach here then player1 is the winner.
    return enWinner::Player1;

}

string ChoiceName(enGameChoice Choice)
{
    string arrGameChoices[3] = { "Stone","Paper","Scissors" };

    return arrGameChoices[Choice - 1];

}
```

```cpp
void SetWinnerScreenColor(enWinner Winner)
{
    switch (Winner)
    {
    case enWinner::Player1:
        system("color 2F"); //turn screen to Green
        break;

    case enWinner::Computer:
        system("color 4F"); //turn screen to Red
        cout << "\a";
        break;
    default:
        system("color 6F"); //turn screen to Yellow
        break;
    }
}

void PrintRoundResults(stRoundInfo RoundInfo)
{

    cout << "\n____________Round [" << RoundInfo.RoundNumber << "]
____________\n\n";
    cout << "Player1  Choice: " <<
ChoiceName(RoundInfo.Player1Choice) << endl;
    cout << "Computer Choice: " <<
ChoiceName(RoundInfo.ComputerChoice) << endl;
    cout << "Round Winner   : [" << RoundInfo.WinnerName << "]
\n";
    cout << "__________________________________\n" << endl;

    SetWinnerScreenColor(RoundInfo.Winner);

}

enWinner WhoWonTheGame(short Player1WinTimes, short
ComputerWinTimes)
{

    if (Player1WinTimes > ComputerWinTimes)
        return enWinner::Player1;
    else if (ComputerWinTimes > Player1WinTimes)
        return  enWinner::Computer;
    else
        return enWinner::Draw;

}
```

```cpp
stGameResults FillGameResults(int GameRounds, short
Player1WinTimes, short ComputerWinTimes, short DrawTimes)
{

    stGameResults GameResults;

    GameResults.GameRounds = GameRounds;
    GameResults.Player1WinTimes = Player1WinTimes;
    GameResults.Computer2WinTimes = ComputerWinTimes;
    GameResults.DrawTimes = DrawTimes;
    GameResults.GameWinner = WhoWonTheGame(Player1WinTimes,
ComputerWinTimes);
    GameResults.WinnerName = WinnerName(GameResults.GameWinner);

    return GameResults;
}

enGameChoice ReadPlayer1Choice()
{

    short Choice = 1;

    do
    {

        cout << "\nYour Choice: [1]:Stone, [2]:Paper, [3]:Scissors
? ";
        cin >> Choice;

    } while (Choice < 1 || Choice >3);

    return (enGameChoice)Choice;


}

enGameChoice GetComputerChoice()
{
    return  (enGameChoice) RandomNumber(1, 3);
}
```

```
stGameResults PlayGame(short HowManyRounds)

{
   stRoundInfo RoundInfo;
   short Player1WinTimes = 0, ComputerWinTimes = 0, DrawTimes = 0;

for (short GameRound = 1; GameRound <= HowManyRounds; GameRound++)
    {
        cout << "\nRound [" << GameRound << "] begins:\n";
        RoundInfo.RoundNumber = GameRound;
        RoundInfo.Player1Choice = ReadPlayer1Choice();
        RoundInfo.ComputerChoice = GetComputerChoice();
        RoundInfo.Winner = WhoWonTheRound(RoundInfo);
        RoundInfo.WinnerName = WinnerName(RoundInfo.Winner);

        //Increase win/Draw counters
        if (RoundInfo.Winner == enWinner::Player1)
            Player1WinTimes++;
        else if (RoundInfo.Winner == enWinner::Computer)
            ComputerWinTimes++;
        else
            DrawTimes++;

        PrintRoundResults(RoundInfo);
    }

    return FillGameResults(HowManyRounds, Player1WinTimes,
ComputerWinTimes, DrawTimes);

}

string  Tabs(short NumberOfTabs)
{
    string t = "";

    for (int i = 1; i < NumberOfTabs; i++)
    {
        t = t + "\t";
        cout << t;
    }
    return t;

}
```

```
void ShowGameOverScreen()
{
    cout << Tabs(2) <<
"__________________________________________________________\n\n";
    cout << Tabs(2) << "                 +++ G a m e  O v e r
+++\n";
    cout << Tabs(2) <<
"__________________________________________________________\n\n";

}

void ShowFinalGameResults(stGameResults GameResults)
{

    cout << Tabs(2) << "_____________________ [Game Results
]_____________________\n\n";
    cout << Tabs(2) << "Game Rounds        : " <<
GameResults.GameRounds << endl;
    cout << Tabs(2) << "Player1 won times  : " <<
GameResults.Player1WinTimes << endl;
    cout << Tabs(2) << "Computer won times : " <<
GameResults.Computer2WinTimes << endl;
    cout << Tabs(2) << "Draw times         : " <<
GameResults.DrawTimes << endl;
    cout << Tabs(2) << "Final Winner       : " <<
GameResults.WinnerName << endl;
    cout << Tabs(2) <<
"___________________________________________________________\n";

    SetWinnerScreenColor(GameResults.GameWinner);
}

short ReadHowManyRounds()
{
    short GameRounds = 1;

    do
    {
        cout << "How Many Rounds 1 to 10 ? \n";
        cin >> GameRounds;

    } while (GameRounds < 1 || GameRounds >10);

    return GameRounds;
}
```

```cpp
void ResetScreen()
{
    system("cls");
    system("color 0F");

}

void StartGame()
{
    char PlayAgain = 'Y';

    do
    {
        ResetScreen();
        stGameResults GameResults = PlayGame(ReadHowManyRounds());
        ShowGameOverScreen();
        ShowFinalGameResults(GameResults);

    cout << endl << Tabs(3) << "Do you want to play again? Y/N? ";

    cin >> PlayAgain;

    } while (PlayAgain == 'Y' || PlayAgain == 'y');

}

int main()
{
    //Seeds the random number generator in C++, called only once
    srand((unsigned)time(NULL));

    StartGame();

    return 0;
}
```