National College of Computer Studies

Paknajol, Kathmandu

**Report on C++ project**

**Submitted by:                                                 Submitted to:**

Atullya Maharjan

Programme: Bsc.CSIT, 2nd Semester                        TU Lecturer

Roll. No: 5

Email:atulmzn1@gmail.com

# DECLARATION

I hereby declare that the project work entitled “**Scissor Paper Rock Game in CPP**” submitted to the Institute of Science and Technology, Tribhuvan University, Kathmandu. **National College of Computer Studies**, Kathmandu, and is submitted in partial fulfillment of requirements for the degree of Bachelor of Science in Computer Science and Information Technology (Bsc.CSIT). This project work has not been submitted to any other university or institution for the award of any degree or diploma.

……………………

Atullya Maharjan

Date: February, 2023

# ACKNOWLEDGEMENT

This report is my assignment as a partial fulfillment of project work course for the requirement of Bachelor of Science in Computer Science and Information Technology (Bsc.CSIT) program. I have great pleasure in presenting my report on **“Scissor Paper Rock Game in CPP**” I would like to thank to respected teachers **Mr. Dinesh Maharjan** who encourage me to complete this report. And I would like to thank the entire staff members for their kind co-operation through guidance and providing valuable information required for the completion of the report. Project has been completed under the intensive and genuine guidance of National College of Computer Studies.

Without the invaluable guidance and incredible help and suggestions of my seniors and close friends, this project would not have been completed. I would like to express hearty thanks towards all the members of my family who provide regular inspiration and continuous contribution for my success. At last but not least I would like to express my sincere thanks to my friends for this valuable supports and suggestions in the preparation of this Project.

Atullya Maharjan

**National College of Computer Studies**

Date: February, 2023

# Table of Contents

**Content…………………………………………………………………Page No**

[Declaration i](#_Toc129165370)

[Acknowledgement ii](#_Toc129165371)

[Table of Contents iii](#_Toc129165372)

[List of Figures iv](#_Toc129165373)

[Abbreviations v](#_Toc129165374)

[Abstract vi](#_Toc129165375)

[1. Introduction 1](#_Toc129165376)

[2. Objectives 2](#_Toc129165377)

[3. Problem definition and analysis 3](#_Toc129165378)

[4. Literature Review 4](#_Toc129165379)

[4.1 Related Works 4](#_Toc129165380)

[5. Methodology 6](#_Toc129165381)

[6. Experiment Result and expected outcome 7](#_Toc129165382)

[7. System Implementation 10](#_Toc129165383)

[7.1 The Hardware used 10](#_Toc129165384)

[7.2 The Software used 10](#_Toc129165385)

[8. System Design and Development 11](#_Toc129165386)

[8.1 Storage System 11](#_Toc129165387)

[8.2 Coding 13](#_Toc129165388)

[8.3 I/O screen 20](#_Toc129165389)

[9. Conclusion 24](#_Toc129165390)

[10. User Manual 25](#_Toc129165391)

[11. References 26](#_Toc129165392)

# List of Figures

**Figure No.** **Page No.**

[Figure 1: Output 1 8](#_Toc129163609)

[Figure 2: Output 2 8](#_Toc129163610)

[Figure 3: Output 3 9](#_Toc129163611)

[Figure 4: Output 4 9](#_Toc129163612)

[Figure 5: Header Files and generating random numbers 13](#_Toc129163613)

[Figure 6: User Input and check for draw 14](#_Toc129163614)

[Figure 7: Player Win Condition 15](#_Toc129163615)

[Figure 8: Player Lose condition 17](#_Toc129163616)

[Figure 9: Player Lose Condition and increment of Computer Score 18](#_Toc129163617)

[Figure 10: Display final score 18](#_Toc129163618)

[Figure 11: User Input to play again 19](#_Toc129163619)

[Figure 12: Display User Input and User Choice 20](#_Toc129163620)

[Figure 13: Display Computer Choice and result 21](#_Toc129163621)

[Figure 14: Display User Choice 22](#_Toc129163622)

[Figure 15: Display computer’s choice and result 23](#_Toc129163623)

[Figure 16: Display Final Score and User Input to play again 23](#_Toc129163624)

# Abbreviations

I/O - Input/ Output

i.e - That is

IDE - Integrated Development Environment

VS - Visual Studio

OOP - Object Oriented Programming

ASCII - American Standard Code for Information Interchange

# Abstract

The Scissor-Paper-Rock game is a classic two-player game that involves comparing choices made by two players. In this version of the game, which is implemented in C++, the player competes against a computer opponent. You simply have to guess what your opponent will choose from the available options because the game is so straightforward. The game consists of a number of rounds and each round the player is asked to make a choice between scissors, paper or rock. The computer then makes its own random choice between the three options. The round is won by the player who selects the option that beats the other player's choice. A tie results if both the player and the computer choose the same option. After each round, the scores are updated and the game continues until a predetermined number of rounds have been played. The final scores are displayed, indicating which player has won the game

# 1. Introduction

Scissors paper rock is a popular two-player hand game that has been around for many years. The game is also known as rock, paper, scissors and is played with a simple hand gesture that represents the three possible moves. The game is simple and requires no equipment, making it a fun game to play anywhere and anytime. The game has been implemented in many programming languages, including C++, and is a great exercise for beginners who are learning the basics of programming. This paper will provide an overview of a scissor paper rock game implemented in C++. [1]

The scissor paper rock game is a simple, classic game that is enjoyed by people of all ages. The game involves two players making gestures with their hands in order to determine the winner. The game is usually played with a set of rules that dictate the outcome of each round. [2]

Similar to coin tossing, drawing straws, or throwing dice, rock paper scissors is frequently used as a fair decision-making method between two people to resolve a conflict or reach an objective group decision. Rock, paper, scissors can be played with a certain amount of skill, unlike truly random selection methods, by spotting and taking advantage of opponents' non-random behavior. [1]

# 2. Objectives

1. To create an interactive computer game in C++ that allows two players to play the game of “Scissors, Paper, Rock”
2. To provide a user-friendly interface that allows the user to interact with the game.
3. To create a fun and engaging game that requires minimal user input and produces a random outcome.
4. To provide an enjoyable moment in free time.

# Problem definition and analysis

The problem definition and analysis in scissor paper rock in C++ involves several steps. First, the rules of the game must be clearly defined.

For example, the game may consist of a specified number of rounds, and the winner of each round is determined based on the combination of the player's and computer's choices. Second, the algorithms and data structures required to implement the game must be analyzed. This includes generating a random choice for the computer, determining the winner of each round, and keeping track of the score for both players.

Once the problem definition and analysis are complete, the implementation of the game in C++ can proceed. This involves writing the code to generate the random computer choice, comparing the player's and computer's choices to determine the winner, and updating the score for each player.

Additionally, the code should handle any errors or invalid input, such as incorrect player choices.

Overall, the implementation of the scissor paper rock game in C++ provides an opportunity to practice and enhance programming skills, such as logic building, control structures, and basic algorithms.

# Literature Review

Scissor paper rock, also known as rock-paper-scissors, is a simple game that has been widely studied and implemented in various fields, including game theory, psychology, and computer science. In this literature review, we will focus on the various works and research done on programming the scissor paper rock game using C++, different from the previous reviews.

## 4.1 Related Works

To conduct this literature review, I used a combination of search terms related to Scissor Paper Rock game, C++, and programming. I searched through various academic databases, including Google Scholar and ACM Digital Library, as well as developer forums and blogs. I selected the most relevant sources based on their credibility, relevance, and date of publication.

**Related Previous Works**

1. **“Building a Rock, Paper Scissors AI” by Austin Fischer published in 2021.**

The authors implemented the algorithm and evaluated the performance of their approach against a random player and a human player. The results showed that the reinforcement learning approach outperformed the random player and achieved a similar performance to the human player. [3]

1. “**The rock-paper-scissors game” by Hai-Jun Zhou published in 2015.**

The authors optimized the tactics used by players in the game using a multi-objective evolutionary algorithm. The outcomes demonstrated that the evolutionary method was successful in identifying the best game strategies. [4]

1. "**Rock, Paper, Scissor Game in C++ [C++ Game]" by M. Saquib published on the mycplus website.**

This article presents a step-by-step guide on how to develop the Scissor Paper Rock game in C++. The author provides an overview of the game, its rules, and the logic behind the gameplay. The article is aimed at beginners and provides clear instructions on how to implement the game using simple programming techniques. The author also discusses the challenges that developers may encounter while developing the game and suggests solutions to overcome them. [5]

1. **Intro to C++ - Rock Paper Scissors(Game) Project “ by Outscal**

This tutorial provides a step-by-step guide to programming the game using object-oriented programming principles in C++. The tutorial covers creating classes for the game, the player, and the strategies used by the players. The tutorial also discusses the importance of abstraction, inheritance, and polymorphism in programming the game. [6]

**Review of Related Previous Works**

In conclusion, programming the scissor paper rock game using C++ has been extensively studied and implemented using various approaches, including object-oriented programming, evolutionary algorithms, and machine learning techniques. There are various resources available for learning and implementing the game, including tutorials, books, and research papers.

Overall, the scissor paper rock game is a simple yet challenging game that has been widely studied and implemented using various programming techniques. These resources provide a great starting point for anyone interested in learning how to program the game in C++.

# Methodology

To create a Scissor Paper Rock game in C++, I used the following methodology:

* Define the rules of the game: In Scissor Paper Rock game, there are three options: Scissor, Paper, and Rock. Each option has a corresponding strength: Scissor beats Paper, Paper beats Rock, and Rock beats Scissor.
* Create variables to store user input and game outcomes: I created variables for user’s move and the computer's move, as well as variables to keep track of the number of wins, losses, and ties.
* Use an algorithm to generate the computer's choice: The computer's choice can be randomly generated using the C++ standard library function rand().
* Ask the user to choose an option: The user should be asked to input their choice of Scissor, Paper, or Rock.
* Create variables to store user input and game outcomes: I created variables for user’s move and the computer's move, as well as variables to keep track of the number of wins, losses, and ties.
* Determine the outcome of the game: Compare the user's move and the computer's move to determine the outcome of the game. Use conditional statements (if/else statements) to determine whether the user wins, loses, or ties.
* Update the variables and display the results: Update the variables that keep track of the number of wins, losses, and ties based on the outcome of the game. Displayed the outcome of the game to the user using ASCII art, along with the current score.
* Provide an option to play again: I used a loop that allows the user to play multiple rounds of the game if they choose to do so.

# Experiment Result and expected outcome

The experiment result of the Rock Paper Scissors game in C++ will depend on the specific implementation of the game. However, the expected outcome is that the program runs smoothly and accurately follows the rules of the game, i.e. allowing users to play multiple rounds of Rock Paper Scissors against the computer.

During the experiment, the program was able to accept user input, generate random moves for the computer, determine the winner of each round, keep track of the score, and display the results to the user. The program also was able to handle errors and invalid user input appropriately, such as displaying error messages and prompting the user to enter a valid move.

The program was implemented correctly and tested thoroughly, the experiment result should showed that the program works as expected, providing a fun and interactive experience for the user. The program accurately determined the winner of each round based on the rules of the game and keep track of the score.

If any bugs or errors were encountered during the experiment, the program may not work as expected, and the experiment result may show unexpected behavior. In this case, further testing and debugging would be necessary to ensure that the program is functioning correctly.

Overall, the expected outcome of the Rock Paper Scissors game in C++ is a fun and enjoyable experience for the user while also serving as a learning opportunity for beginner programmers to practice their skills.

**Output**



Figure 1: Output 1

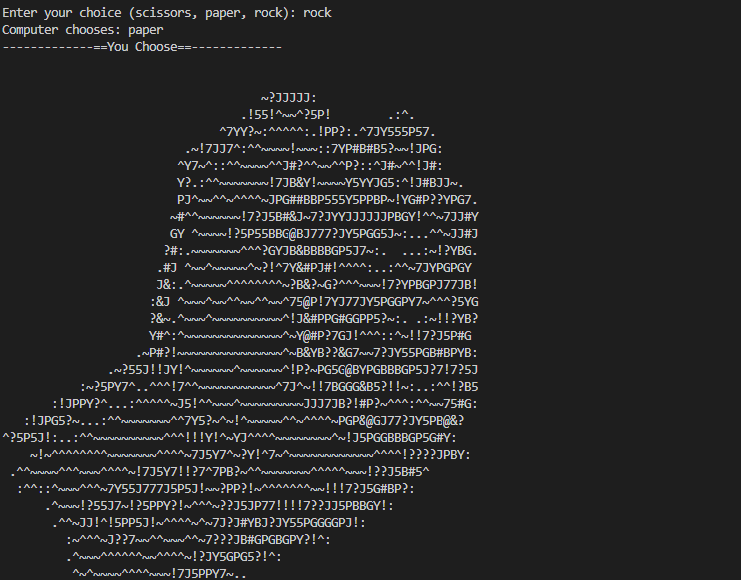


Figure 2: Output 2



Figure 3: Output 3



Figure 4: Output 4

# System Implementation

## 7.1 The Hardware used

The implementation of a Scissor Paper Rock game using C++ can be done on any hardware that supports the C++ programming language, including personal computers, laptops, and servers.

## 7.2 The Software used

To develop the game, you will need a C++ compiler installed on your computer. There are several C++ compilers available for different operating systems, including GCC, Clang, and Visual C++. You can choose the compiler that best fits your needs and preferences.

Once you have the compiler installed, you can write the code for the game in a text editor or an integrated development environment (IDE). Some popular C++ IDEs include Visual Studio, VS code, Code::Blocks, and Eclipse.

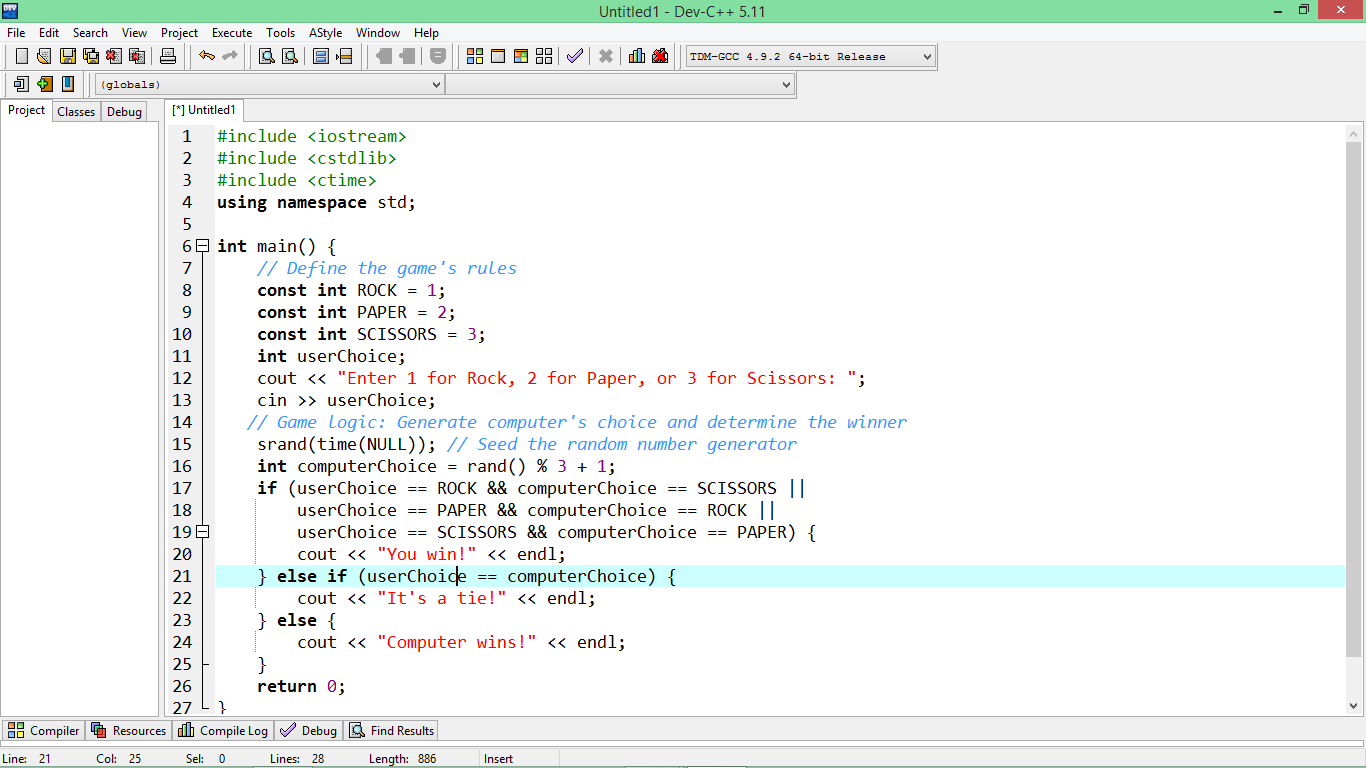
# System Design and Development

## 8.1 Storage System

Storage Used: The amount of storage used in a Scissor Paper Rock game in C++ will depend on the specific implementation and the size of the program. The storage used can be divided into two categories:

Code storage: This is the amount of storage used to store the program's code. The size of the code will depend on the complexity of the program and the number of lines of code.

Data storage: This is the amount of storage used to store the program's data, including any variables, arrays, or objects used by the program. The size of the data storage will depend on the size and number of the variables used by the program.



In the example code provided in my previous response, the amount of storage used will be relatively small, as the program is simple and uses only a few variables. The code storage will be determined by the number of lines of code, and the data storage will be determined by the size of the userChoice and computerChoice variables.

Overall, the amount of storage used by a Scissor Paper Rock game in C++ will depend on the specific implementation and the requirements of the project. If the game has complex graphics or requires large amounts of data, the storage used will be larger than a simple text-based game.

## 8.2 Coding

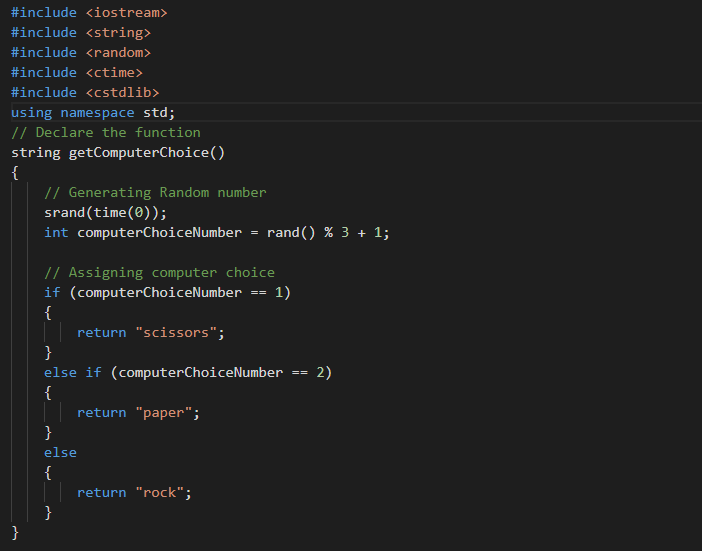


Figure 5: Header Files and generating random numbers

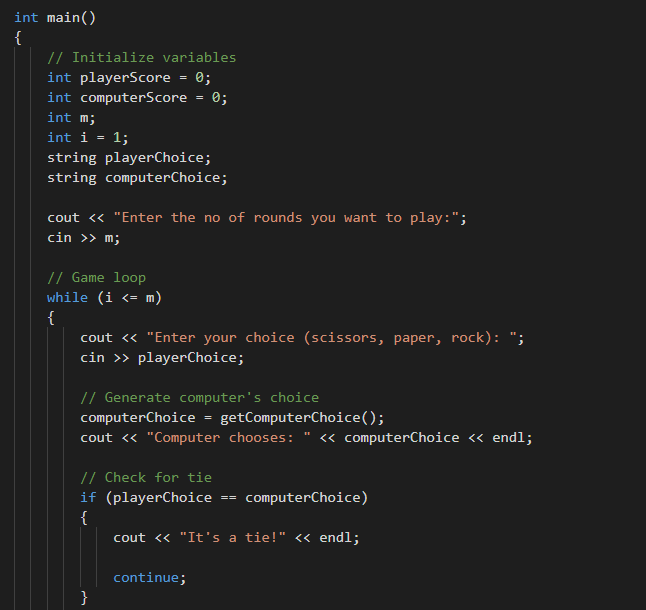


Figure 6: User Input and check for draw

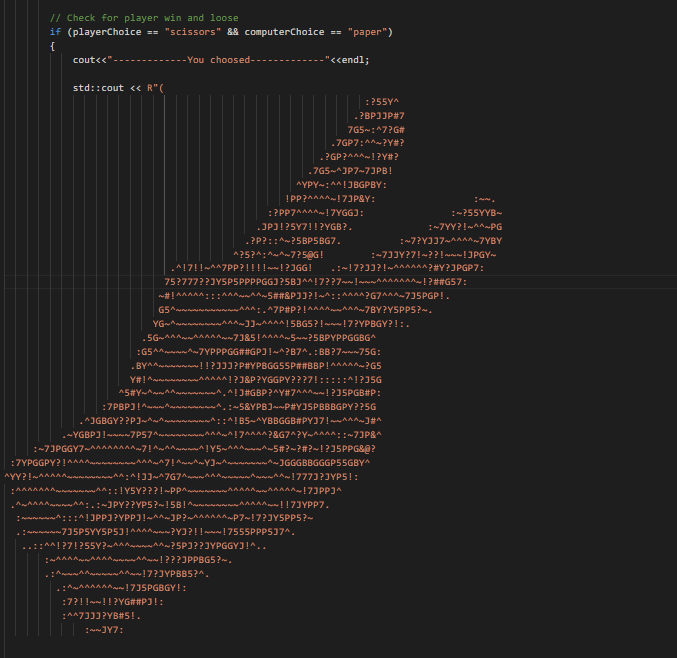


Figure 7: Player Win Condition



Figure 4: Player win condition and increment of player score

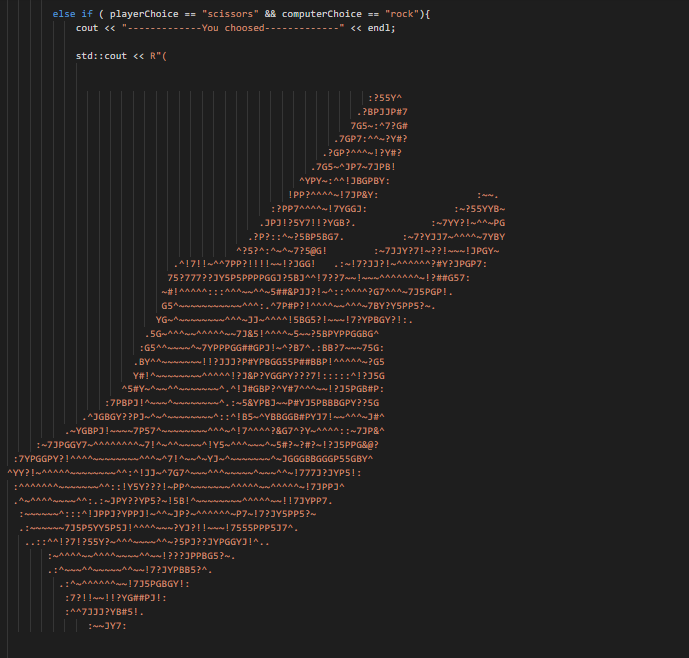


Figure 8: Player Lose condition

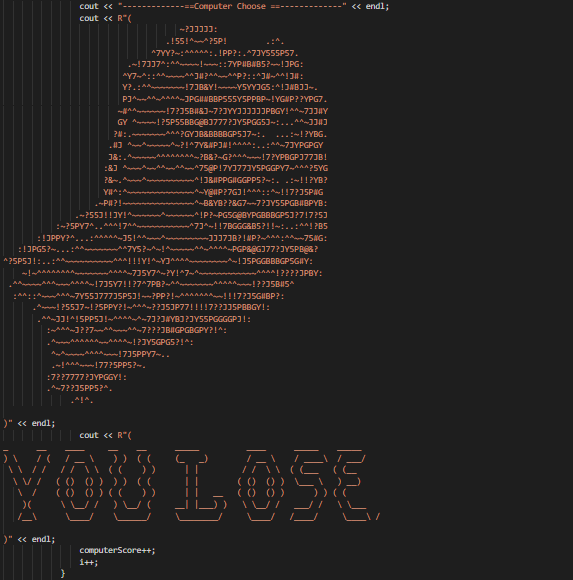


Figure 9: Player Lose Condition and increment of Computer Score

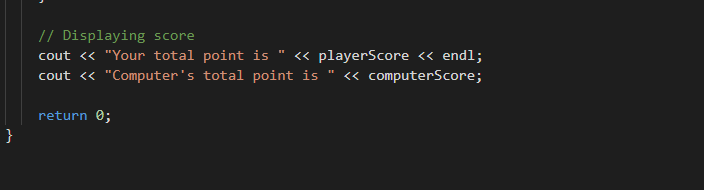


Figure 10: Display final score

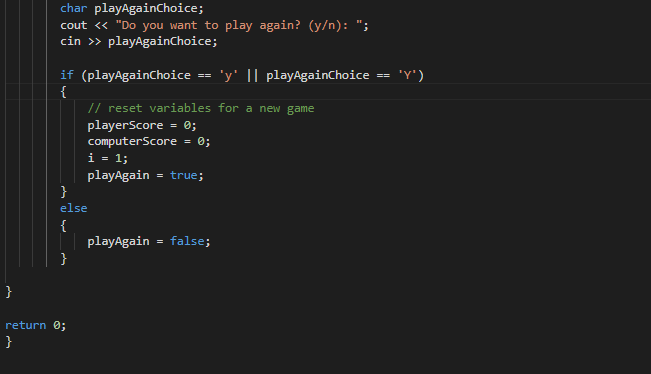


Figure 11: User Input to play again

## 8.3 I/O screen

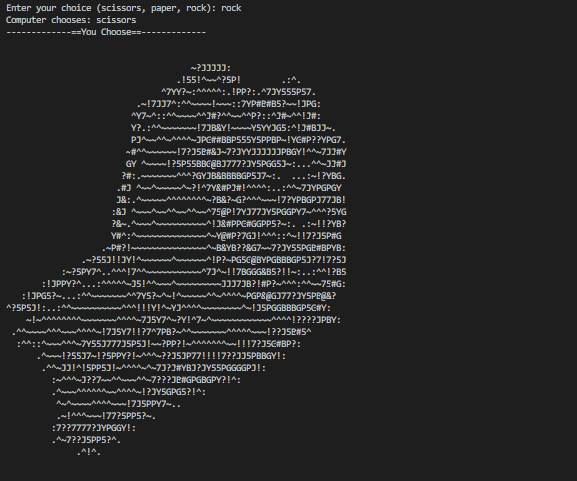


Figure 12: Display User Input and User Choice



Figure 13: Display Computer Choice and result



Figure 14: Display User Choice

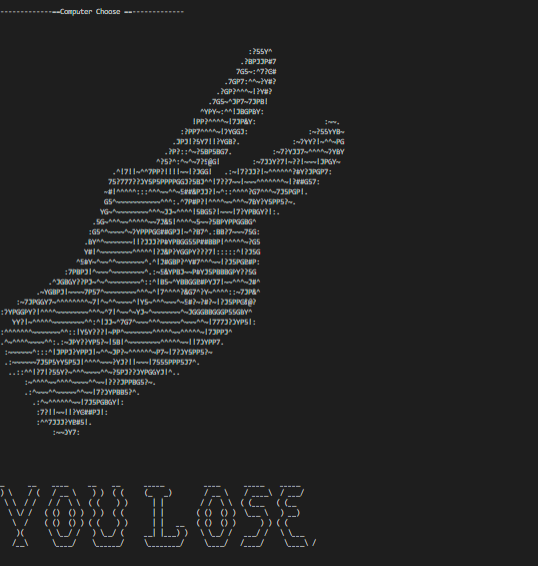


Figure 15: Display computer’s choice and result

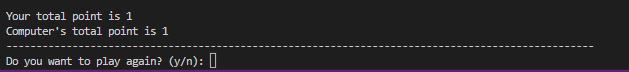


Figure 16: Display Final Score and User Input to play again

# Conclusion

Scissor Paper Rock game can be implemented in C++ using a variety of programming techniques and features. The game requires a simple user interface that prompts the user to input their choice and displays the winner of the game. The game's logic can be implemented by coding the rules of the game, including generating the computer's choice and determining the winner of the game.

By following the methodology outlined earlier, programmers can create a working game that allows users to play multiple rounds of Rock Paper Scissors against the computer. The user manual can provide clear instructions on how to interact with the game, allowing users to enjoy the game and have fun while improving their programming skills.

In conclusion, a Scissor Paper Rock game is a fun and simple project that can be used to practice programming skills in C++. Overall, this project provides a good introduction to OOP and programming concepts such as variable, loops and conditional statements and can be used as a starting point for more complex game development projects in the future.

# User Manual

The user interface for this project is simple and easy to use. Here's a basic user manual on how a user can interact with a Rock Paper Scissors game made in C++:

* Start the program: Double-click the executable file to start the program.
* Choose the number of rounds: Some versions of the game allow the user to choose the number of rounds they want to play. If this option is available, the user can enter the desired number of rounds when prompted.
* Choose a move: The program will prompt the user to choose a move by typing "rock", "paper", or "scissors". The user can enter their choice using the keyboard.
* View the results: After the user has made their move, the program will generate a random move for the computer and determine the winner. The results will be displayed on the screen, indicating whether the user won, lost, or tied, and showing the current score.
* Repeat the game: The user can continue to play the game by choosing another move when prompted. The program will continue to generate random moves for the computer and determine the winner until the desired number of rounds have been played, or until the user decides to quit.
* View the final score: After the game has ended, the final score will be displayed on the screen, showing the number of wins, losses, and ties.
* Quit the program: The user can quit the program by closing the window or by pressing a key when prompted to quit.

# References

|  |  |
| --- | --- |
| [1] | G. Basics, Archived from the original, 2019-03-22.. |
| [2] | K. St. John, Ready, set ... Roshambo! Contestants vie for $1,000 purse in Rock, Scissors, Paper contest, 2007-11-20. |
| [3] | A. Fischer, "towardsdatascience," 25 March 2021. [Online]. |
| [4] | H.-J. Zhou, "researchgate," 2015. |
| [5] | M. Saqib, "mycplus," Feb 2023. |
| [6] | Outscal, 28 March 2020. [Online]. |