

# Rock-Paper-Scissors Game Report

Firas El Khesheh

## **I. Introduction**

The purpose of this report is to provide an overview of the MIPS assembly language game developed by our team. The game is a simplified version of "Rock, Paper, Scissors" and was created as a project for CSC312 class. This report will discuss the features of the game, implementation details, future improvements, how the work was organized among team members, and a concluding remark.

## **II. Features**

The MIPS Rock, Paper, Scissors game offers the following features:

Interactive gameplay: The game prompts the user to input their move (rock, paper, or scissors) and generates a random move for the computer.

Score tracking: The user's score and the computer's score are displayed after each round, allowing players to keep track of their progress.

Multiple rounds: The game allows a maximum of five rounds, after which the winner is determined based on the scores.

End game messages: Depending on the final scores, appropriate messages are displayed to indicate whether the user or the computer won the game.

## **III. Implementation Details**

The game was implemented in MIPS assembly language using the MARS simulator. The code follows a structured approach with separate sections for data declarations and the main program logic. The program utilizes branching instructions, system calls, and load/store instructions to handle user input, generate random moves, compare moves, update scores, and display messages.

## **IV. Future Improvements**

While the current implementation of the MIPS Rock, Paper, Scissors game fulfills the basic requirements, there are several areas for potential improvement:

Enhanced graphical interface: Currently, the game relies solely on text-based prompts and messages. Adding a graphical interface could enhance the user experience and make the game more visually appealing.

Sound effects: Incorporating sound effects for different events, such as the user's win or tie, would add an extra layer of immersion to the game.

Difficulty levels: Introducing different difficulty levels or strategies for the computer opponent could increase the game's replayability and challenge.

## **V. How Work Was Organized**

The development of the MIPS Rock, Paper, Scissors game was a collaborative effort among the three team members. The workload was divided as follows:

- Firas: Firas was responsible for implementing the core game logic, including user input, generating random moves, and determining the winner of each round.

- Gabriel: Gabriel focused on score tracking and displaying the user's score and the computer's score after each round. He also contributed to debugging and code optimization.

- Jean-marie: Jean-marie played a key role in implementing the game flow, including the round count, maximum rounds, and end game conditions. He also assisted in testing and documentation.

The team members maintained regular communication throughout the development process, sharing code snippets, discussing implementation strategies, and conducting thorough testing to ensure the game's functionality.

## **VI. Conclusion**

In conclusion, the MIPS Rock, Paper, Scissors game provides an interactive and entertaining experience for players. Through our collaborative efforts, we successfully implemented the game using MIPS assembly language and the MARS simulator. The game features user input, score tracking, multiple rounds, and end game messages. While there is room for further improvement, such as adding a graphical interface and sound effects, the current version serves as a solid foundation. The teamwork and coordination among the team was crucial in delivering a functional and enjoyable game. Overall, this project allowed us to deepen our understanding of computer architecture concepts, MIPS assembly language, and collaborative development. We gained practical experience in designing and implementing a game, which required careful attention to detail and effective teamwork.