**Introduction:**

Our project is about creating minigames in Java. The games we chose are Sudoku, Tic-Tac-Toe and Rock, Paper Scissors. These games is very popular and is fairly simple by itself. They are a two player games.

Sudoku is a captivating number puzzle game that challenges your logical prowess. Played on a 9x9 grid, the objective is to fill in the empty cells with numbers from 1 to 9, ensuring that each row, column, and 3x3 sub grid contains every digit exactly once. Sudoku's elegant simplicity hides intricate patterns and strategies, making it a brain-teasing pastime enjoyed by millions worldwide. It sharpens cognitive skills, fosters patience, and offers endless variations, making it a timeless and addictive mental exercise.

Tic-tac-toe (also known as nougats and crosses or X’s and O’s) is a paper-and- pencil game for two players, X and O, who take turns marking the spaces in a n x n grid. The player who succeeds in placing n of their marks in a horizontal, vertical, or diagonal row wins the game.

Rock, Paper, Scissors is a universally recognized hand game that transcends cultures and generations. Players simultaneously form one of three shapes with an open hand: a rock, paper, or scissors. The rules are simple: rock crushes scissors, scissors cut paper, and paper covers rock.

This quick and casual game relies on chance, making it a fun and spontaneous way to settle disputes or make decisions. Its charm lies in its accessibility and the thrill of predicting your opponent's choice, making it a timeless and enjoyable social diversion for all ages.

**Aim:**

The aim of this basic Java project is to create versatile and engaging terminal-based gaming application that offers three classic games: Sudoku, Tic Tac Toe, and Rock Paper Scissors. This project aims to provide users with an enjoyable and interactive gaming experience, allowing them to test their logical skills with Sudoku, strategic thinking with Tic Tac Toe, and quick decision-making in Rock Paper Scissors, all within a single cohesive program. The project should offer intuitive gameplay, clear instructions, and a user-friendly interface, providing entertainment and mental challenges for players of all ages and skill levels.

**Motivation:**

The motivation behind undertaking this project lies in the following factors:

1. Educational Value: Developing minigames in Java provides a practical opportunity to apply key concepts learned in programming courses, including data structures, loops, and conditional statements.

2. Problem-Solving: Designing and implementing the game logics challenges students to think critically and devise efficient algorithms for checking win conditions and validating player moves.

3. User Interaction: Creating a user-friendly interface fosters skills in user input validation and output presentation, essential for any software development endeavor.

4. Fundamental Programming Skills: This project serves as a foundation for more complex game development and software engineering projects, helping students build essential programming skills.

**Project objective:**

The primary objective of this project is to develop a Java-based minigames that provides an engaging and interactive gaming experience. Below is a table summarizing the pros and cons of existing methods or implementations of minigames:

|  |  |  |
| --- | --- | --- |
| Existing Methods | Pros | cons |
| Text Based Approach | Simplicity and ease of implementation | Limited user interface and aesthetics |
|  | Quick development time | Lack of Visual appeal |
|  | Minimal resource requirements | Limited scalability and Features |
| Gui Based approach | Enhanced user interface with graphics | Requires additional libraries and tools |
|  | Improved user experience | May involve a steeper learning curve |
|  | Can be customized and Themes | Development time may be longer |
|  | Can accommodate advanced features | May consume more system resources |

**Facilities required for proposed work:**

Software Requirements:

1. Java Development Kit (JDK): Java is the primary programming language for this project. Installed the latest version of JDK to compile and run Java programs.

2. Integrated Development Environment (IDE): Used VS Code by Microsoft for the development of the program.

3. Version Control System: Used GitHub to manage our project's source code and collaborate with team.

4. Project Documentation Tools: Used Microsoft Word for preparation of the synopsis and required documentation.

Hardware Requirements:

1. Computer: You will need a personal computer or laptop capable of running the chosen Java IDE and software development tools.

2. Operating System: Java is platform-independent, but ensure your chosen IDE and other tools are compatible with your operating system (e.g., Windows, macOS, Linux).

3. Memory and Storage: Ensure your computer has sufficient RAM and storage space for software development. A minimum of 4GB RAM is recommended.

4. Input/Output Devices: Standard input/output devices, including a keyboard and mouse, are essential for coding and testing.