**Capital University of Science & Technology**

**Term Project Proposal**

Department of Electrical and Computer Engineering

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Title** | | Rock-Paper-Scissors Game Against the Computer | |
| **Course Title** | | CPEG1611 | |
| **Sr. No.** | **Student Name** | | **Registration Number** |
| **01.** | MUHAMMAD SHAHMIR ABDULLAH | | BCPE243030 |
| **02.**  **03.** | ZUNAIRA CHAUDHRY  HANIYA | | BCPE243025  BCPE243015 |

|  |
| --- |
| **Idea:**  The Rock-Paper-Scissors game against the computer is a simple program where a player competes with an AI by selecting one of three choices: Rock, Paper, or Scissors. The computer randomly chooses its move, and the program determines the winner based on standard rules:   * Rock beats Scissors. * Scissors beats Paper. * Paper beats Rock.   The game announces the winner, declares a tie if both choose the same option, and may offer options to replay or track scores.  **Objectives:**   * **User Interaction**: Allow the player to input their choice (Rock, Paper, or Scissors). * **Randomized AI Behavior**: Implement a system for the computer to randomly select its choice. * **Game Logic**: Apply the rules of Rock-Paper-Scissors to determine the winner. * **Result Display**: Show the player's choice, the computer's choice, and the game outcome (win, lose, or tie). * **Replay Option**: Provide the player with an option to play multiple rounds. * **Score Tracking**: Keep track of wins, losses, and ties across multiple games. * **Input Validation**: Ensure the player's input is valid and handle invalid entries gracefully. * **User Experience**: Deliver clear instructions and an engaging interface.   **Applications:**   * **Entertainment**: Provide a fun and simple game for users to play. * **Learning Tool**: Serve as an educational project for beginners to learn programming concepts like randomness, conditional logic, and user input handling. * **AI Experimentation**: Use it as a platform to explore AI behavior, such as pattern recognition or predictive algorithms. * **Practice Interface Design**: Create text-based or graphical interfaces for user interaction. * **Game Development Basics**: Offer an introduction to creating interactive games. * **Stress Testing**: Test random number generators or algorithms in programming environments. * **Team Competitions**: Implement multiplayer modes or leaderboards for friendly challenges.   **Block Diagram:**  **Generate Computer's Choice**  **Get User Input (Choice)**  **Display Instructions**  **Start the Game**  **Restart Game Loop**  **Offer Replay Option**  **Display Results (Choices and Winner/Tie)**  **Apply Game Logic (Compare Choices)**  **Exit Game** |

|  |  |
| --- | --- |
| **Instructor Remarks** | **Student 1 Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |
| **Student 2 Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |
|  |
| **Instructor’s Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** |