**Project Report: Multi-Game Python Application**

**Project Overview**

This project is a Python application that implements four classic games: **Number Guessing Game**, **Rock, Paper, Scissors**, **Tic-Tac-Toe**, and **Hangman**. The application is designed to provide an engaging and interactive user experience, allowing users to choose from different games and play them in a console environment.

**Objectives**

* To create an interactive multi-game application using Python.
* To provide a simple and user-friendly interface for playing games.
* To implement basic game mechanics and logic for each game.

**Features**

1. **Number Guessing Game**:
   * Randomly selects a number between 1 and 100.
   * Allows the user to guess the number and provides feedback on whether the guess is too low or too high.
   * Tracks the number of attempts taken to guess correctly.
2. **Rock, Paper, Scissors**:
   * The user can choose between rock, paper, or scissors.
   * The computer randomly selects its choice.
   * Displays the outcome: win, lose, or tie.
3. **Tic-Tac-Toe**:
   * A two-player game where users take turns placing their marks (X or O) on a 3x3 grid.
   * Checks for winning combinations after each turn.
   * Declares a winner or a tie.
4. **Hangman**:
   * Chooses a random word from a predefined list.
   * Allows the user to guess letters until they either guess the word or run out of attempts.
   * Displays the current state of the word and the number of attempts remaining.

**Implementation**

The application is structured into functions, each handling a specific game:

* **Main Function**: Manages user input and navigates to the chosen game.
* **Game Functions**:
  + number\_guessing\_game(): Handles the number guessing logic.
  + rock\_paper\_scissors(): Manages the rock-paper-scissors gameplay.
  + tic\_tac\_toe(): Implements the tic-tac-toe game mechanics.
  + hangman(): Facilitates the hangman word-guessing game.

**Code Structure**

The code is organized for clarity and ease of use:

* Each game function contains its own logic, making it easy to understand and modify.
* User input is validated to ensure proper gameplay.
* The main loop allows users to play multiple games until they choose to exit.

**Usage**

To run the application:

1. Ensure Python is installed on your system.
2. Copy the provided code into a Python environment or script file.
3. Execute the script in the terminal or command prompt.
4. Follow the on-screen instructions to choose and play games.

**Conclusion**

This multi-game Python application provides a fun and interactive way to engage with classic games. It showcases fundamental programming concepts such as loops, conditionals, and functions, making it a great project for beginners learning Python.