VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

ACADEMIC YEAR 2025 – 2026

Stone Paper Scissor Game

Technology : Pyhton , CSV file storage

> Stone Paper Scissor Game

Course : Python Essentials

Submitted By : Yug Maheshkumar Patel

Reg No : 25BAI10675

Branch : AIML(CSE)

Faculty : DR PREETAM SUMAN

Vellore Institute of Technology , Bhopal

Here you go!

I prepared complete PPT text slide-wise for the Multi-Game Python Program you provided (Number Guessing, Tic-Tac-Toe, Hangman, Rock–Paper–Scissors).

PPT: MULTI-GAME PYTHON PROJECT

Slide 1: Title Slide

- Multi-Game Python Project
- Number Guessing • Tic-Tac-Toe • Hangman • Rock-PaperScissors

Slide 2: Introduction

- This project contains four different games in a single Python file.
- Users can select any game from a menu.
- Helps beginners understand loops, conditions, lists, and game logic.

Slide 3: Games Included

- Number Guessing Game
- Tic-Tac-Toe (Two Player)
- Hangman
- Rock-Paper-Scissors

Slide 4: How the Program Works

- User chooses a game from the main menu.
- Based on choice, the code runs that specific game.
- Program uses loops so user can keep playing until exit.

FUNCTIONAL REQUIREMENTS

Slide 5: Functional Requirements – Overview

- These describe what the system must do.

Slide 6: Game Selection

- User must be able to choose from 4 games.
- Invalid input must ask for re-entry.
- System must run the selected game.

Slide 7: Number Guessing – Requirements

- System generates a random number between 1–100.
- User enters guesses until correct.
- Program must show:
- Higher / Lower hints
- Guess count
- Must stop when correct number is guessed.

Slide 8: Tic-Tac-Toe – Requirements

- 2-player game (X and O).
- Board must display after every move.
- System must check:
- Win condition
- Tie condition

- Must allow replay.

Slide 9: Hangman – Requirements

- System chooses a random word.
- User guesses letters.
- Must track:
- Correct letters
- Wrong attempts
- Remaining chances
- Must show win or lose message.

Slide 10: Rock-Paper-Scissors – Requirements

- Player inputs choice.
- Computer randomly selects choice.
- System compares choices and shows:
- Win
- Lose

- Tie
- Score must be updated and shown at end.

NON-FUNCTIONAL REQUIREMENTS

Slide 11: Non-Functional Requirements – Overview

- These describe how well the program works.

Slide 12: Usability

- Menu is simple and beginner-friendly.
- Instructions displayed clearly for each game.
- Easy to understand input prompts.

Slide 13: Performance

- All games must run instantly.
- Random number and word generation must be fast.

Slide 14: Reliability

- Program must not crash on wrong inputs.
- Must handle repeated plays without errors.

Slide 15: Maintainability

- Code divided into sections for each game.
- Easy to add more games in future.
- Functions used in Tic-Tac-Toe improve readability.

Slide 16: Error Handling

- Invalid menu input → Ask again.
- Tic-Tac-Toe: invalid position → Error message.
- Hangman: invalid letter → Error message.

Slide 17: Scalability

- More games can be added easily.
- Hangman word list can be expanded anytime.

CODE EXPLANATION SLIDES

Slide 18: Main Menu Code

```
game = input("ENTER 1 FOR NUMBER GUESSING,
2 FOR TIC-TAC-TOE,
3 FOR ROCK-PAPER-SCISSORS,4 hangman : ")
```

- Allows user to choose which game to play.

    - Loop ensures valid input.

Slide 19: Number Guessing Logic

- Uses random number.
- Gives hints (higher/lower).
- Counts guesses.
- Ends when correct.

Slide 20: Tic-Tac-Toe Logic

- Uses a list of 10 elements for the board.
- Checking win conditions using combinations.
- Alternates between X and O.

Slide 21: Hangman Logic

- Selects random word from list.
- Replaces letters with underscores.
- Decreases attempts on wrong guess.

Slide 22: Rock-Paper-Scissors Logic

- Computer chooses randomly.
- Compares choices to decide outcome.
- Score increases/decreases.