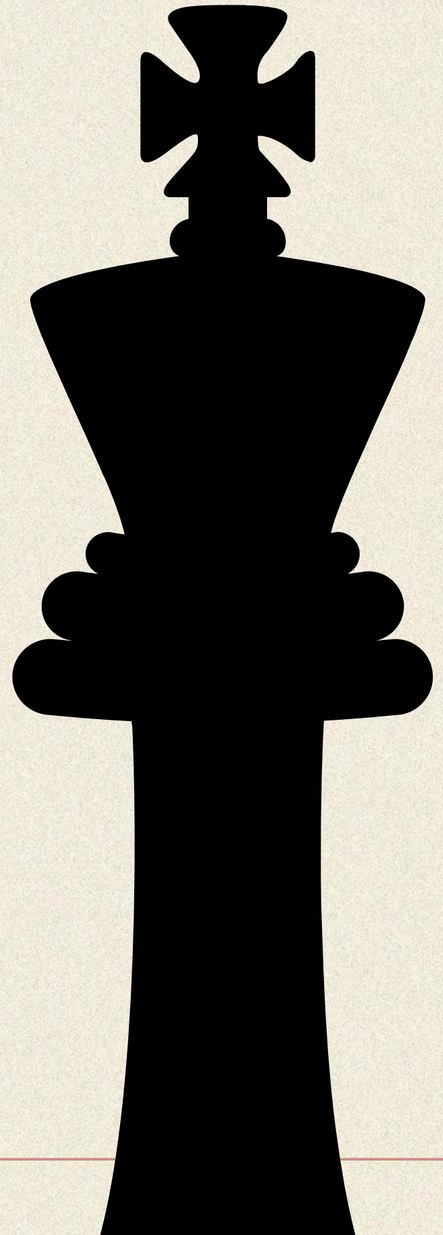
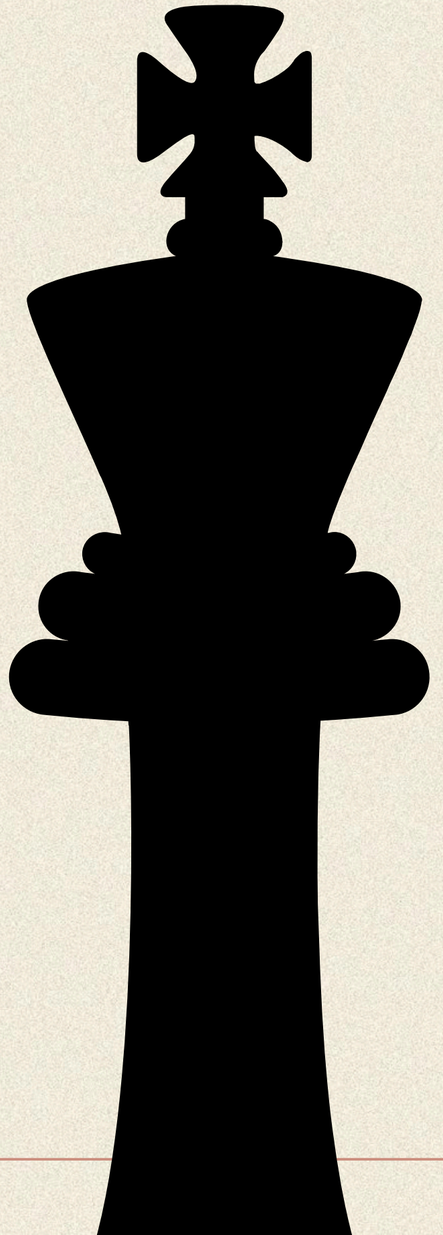
Python Mini Project

# CHESS GAME






- Ankit Mittal 24055
- Krish Garg 24042
- Tanisha Singhania 24057
- Sakshi Saraiya 24011
- Khushi Goyal 24034

# Problem Statement

Objective:

To create a functional and interactive text-based Chess Game using Python, allowing two players to play against each other following the standard chess rules.

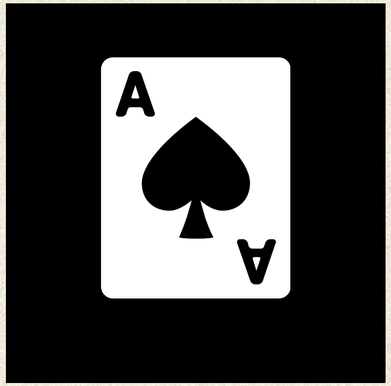
# Reason for making Chess Game


1. Chess is a classic strategy game, ideal for applying OOP (Object-Oriented Programming) concepts.
2. Great project to demonstrate:

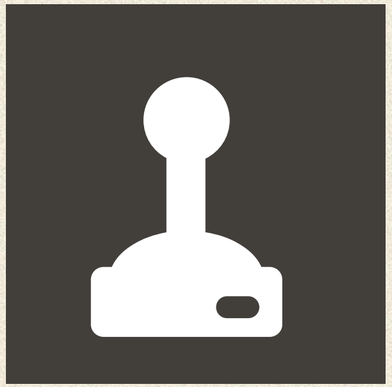
- Class hierarchies
- Inheritance and polymorphism
- Game logic and board management
- Reinforces programming concepts through a real-world use case.

# Tools and Technologies

- Programming Language: Python 3
- Development Environment: Jupyter Notebook

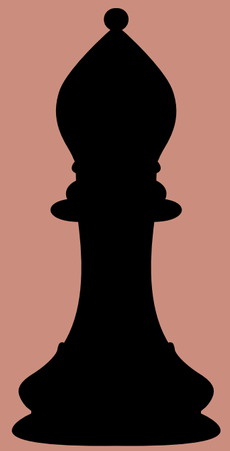



## Concepts Used:



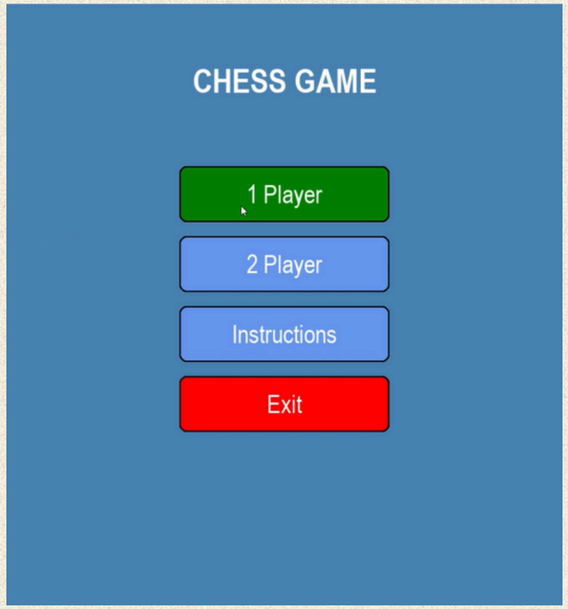
- Classes and Objects
- Inheritance and Method Overriding
- Dictionaries and Lists
- Input validation and control loops

# Key Functionalities

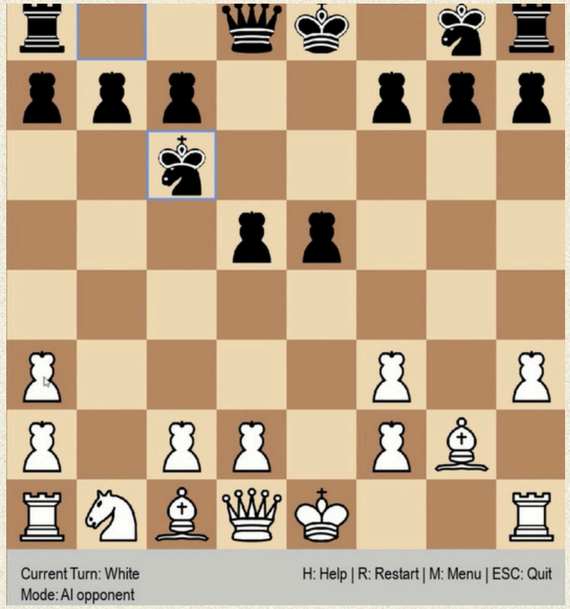


- **Board Initialization:** Standard 8x8 grid with correct pieces.
- **Move Validation:** Only legal moves allowed.
- **Turn Handling:** Alternating turns between White and Black.
- **Check Detection:** Basic check mechanism.
- **Game Continuity:** Prompting moves till checkmate or exit.

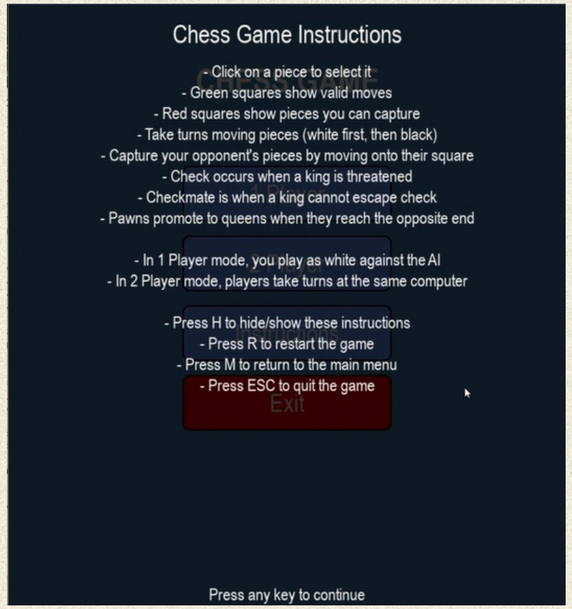
# Sample Output




Initial Page




After a few moves




Instructions Page

# Code Architecture

**Piece (Base Class)**

Common methods like valid_move()

**King, Queen, Bishop, Knight, Rook, Pawn**

Subclasses with specific rules

**Board Class**

Initializes and manages board

**Game Logic**

Turn-based input, move validation, board display.

# Challenges Faced

- Implementing complex movement logic.
- Avoiding bugs in turn-based input.
- Ensuring valid board state after every move.
- Maintaining user-friendly prompts and error handling.

# Learnings

- Deepened understanding of OOP.
- Importance of modular code and debugging strategies.
- Experience in developing interactive applications.

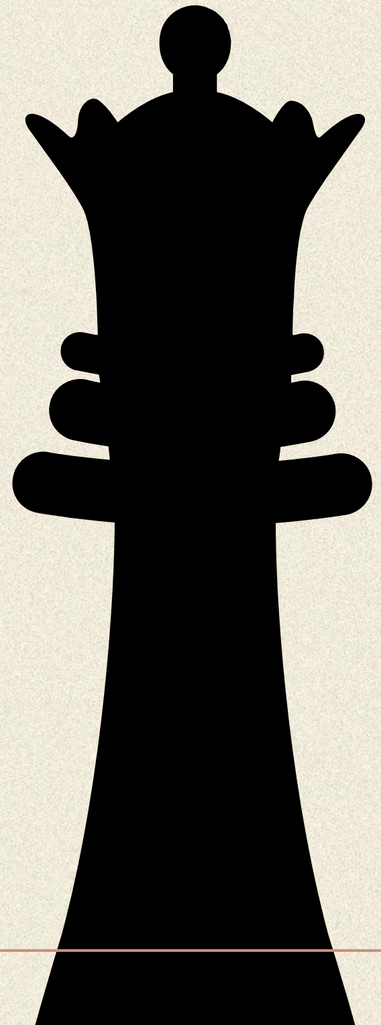
# Future Scope

•Add GUI using Tkinter or Pygame.

•Include features like:

- Timer
- AI opponent (Minimax algorithm)
- Checkmate and stalemate detection
- Save/Load game feature

# Conclusion



- Successfully implemented a functional chess game in Python.
- Reinforced OOP and logic-building skills.
- Foundation for advanced game projects.

# THANKS