Online Examination System

Credits to:

- David Emil
- Rohim Mohamed
- Jonathan Maged
- Osama Nasser
- George Salama
- Ahmed Munir

Supervised By:

Eng/Youssef Al Baroudy

Python Chess Game Documentation

1. Overview/Introduction

Project Title: ***Python Chess Game***

Description:

- **A 2D chess game with both Player vs Player and Player vs AI modes**
- **Features multiple time controls with increment options**
- **Includes move history tracking and visual move hints**
- **Provides audio feedback for moves**

Tech Stack:

- **Frontend: Pygame**
- **Game Logic: python-chess library**
- **AI: Minimax algorithm with alpha-beta pruning**

Key Features:

- **Two game modes (PvP and vs AI)**
- **Configurable time controls (Bullet, Blitz, Rapid)**
- **Move validation using standard chess rules**
- **Visual indicators for check and legal moves**
- **Move history panel**

---

2. Features

2.1 Game Modes

Player vs Player: Two human players alternate turns

Player vs AI: Human plays against computer opponent

2.2 Time Controls

Mode	Base Time	Increment
Bullet	1-2 min	1 sec
Blitz	3-5 min	0-1 sec
Rapid	10-15 min	0-10 sec
Unlimited	No limit	N/A

2.3 Gameplay Features

Legal move highlighting

Check detection

Move sound effects

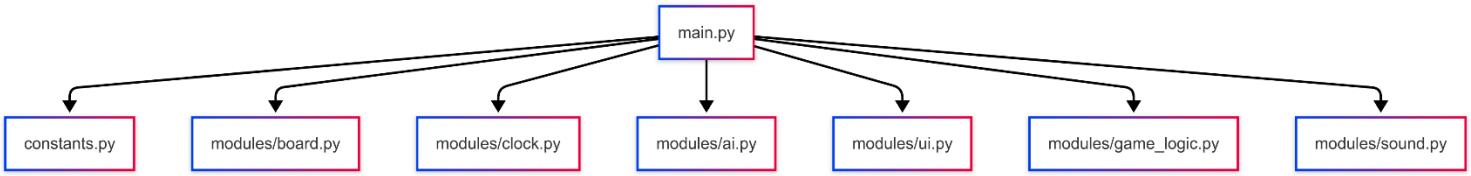
Move history tracking

Time control enforcement

---

3. Code Structure

3.1 Folder Organization



3.2 File Descriptions

- **main.py:** Entry point and game loop
- **constants.py:** All game constants and configurations
- **board.py:** Handles board rendering and display
- **clock.py:** Chess timer implementation
- **ai.py:** Computer opponent logic
- **ui.py:** Menus and interface elements
- **game_logic.py:** Core game rules and state
- **sound.py:** Audio management

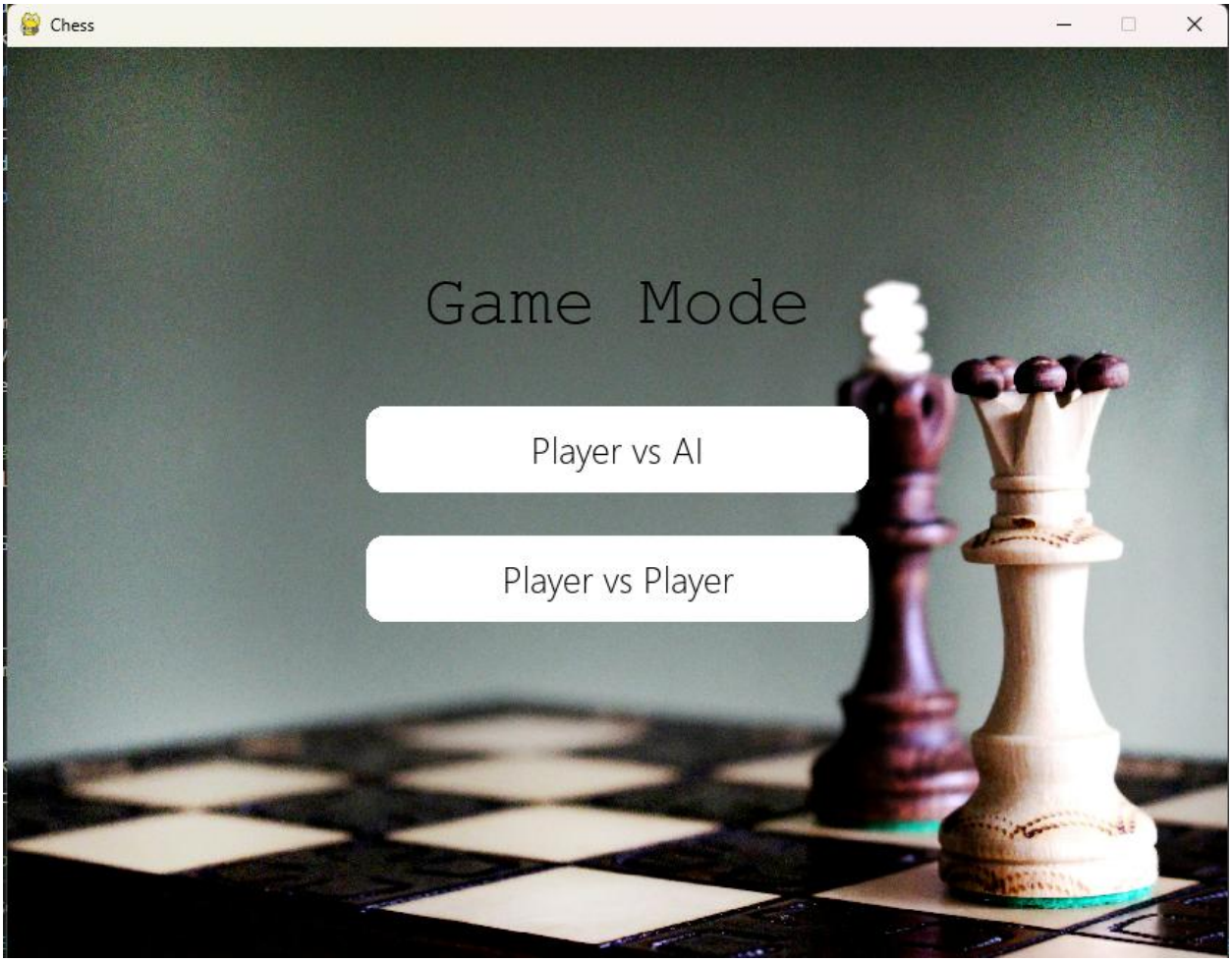
---

5. Testability

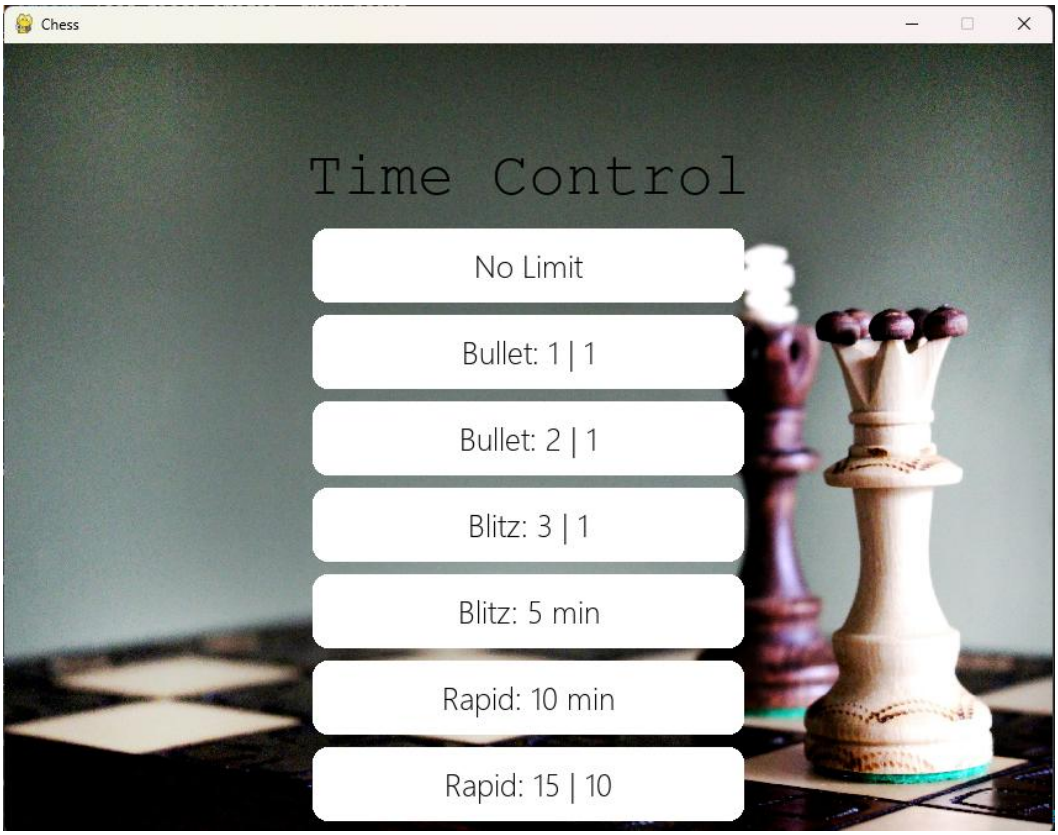
Test Case	Expected Results	Status
Legal move validation	Accepts valid moves, rejects invalid	Pass
Time control enforcement	Ends game when time expires	Pass
AI move generation	Always makes legal moves	Pass
Check detection	Correctly identifies checks	Pass
Move history	Accurately records all moves	Pass

6. Functionality Description & GUI

- Main Menu:
    - Screen Dedicated for beginning the game with a catchy background

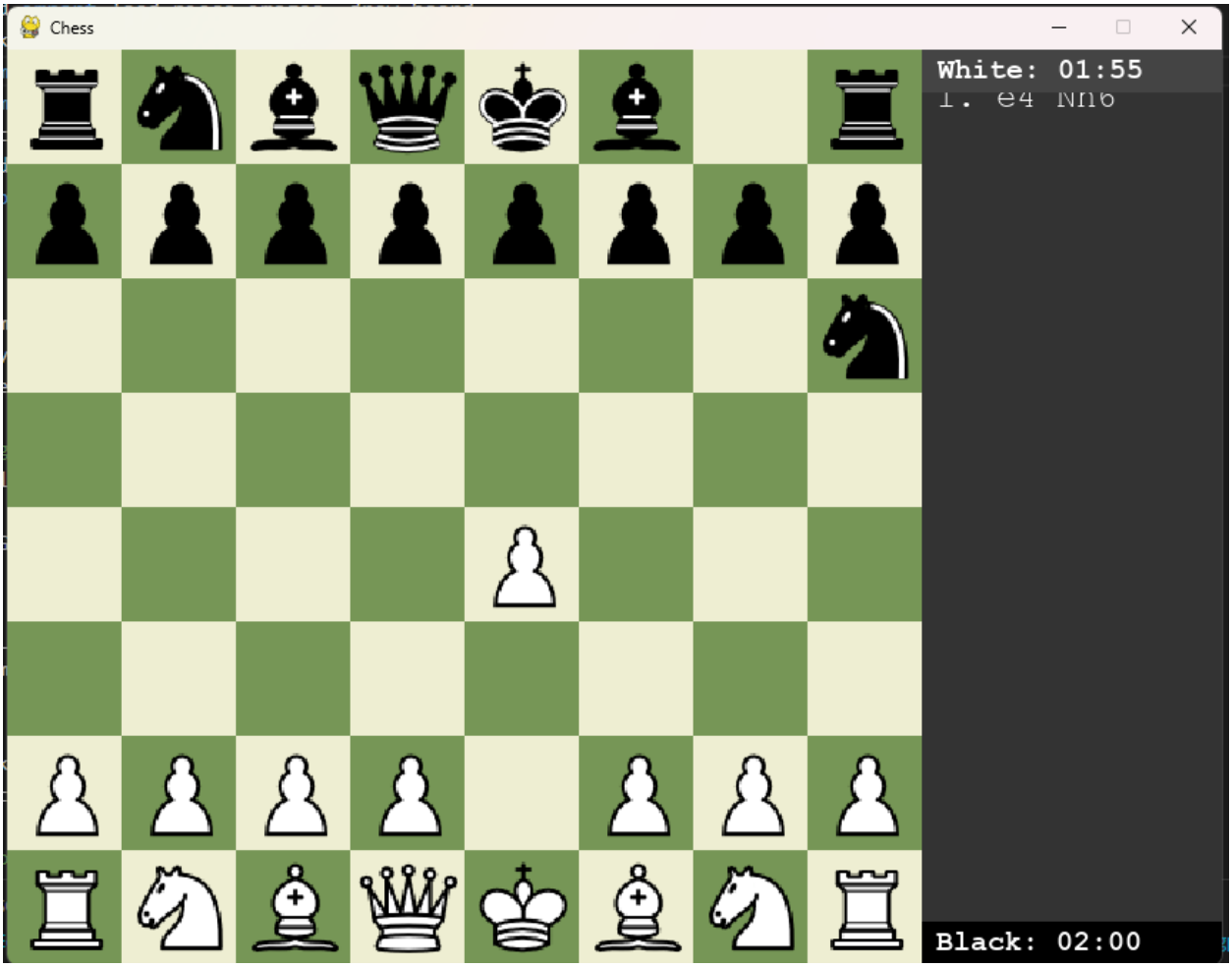


- Choosing Mood:
    - Screen Dedicated for choosing the game mood and time from specified options



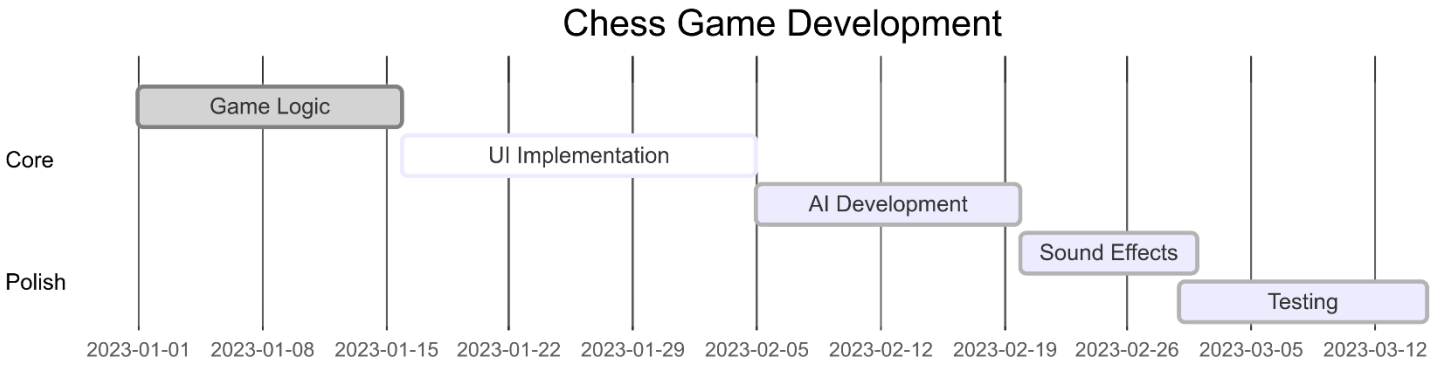
- End Game Screen:
    - Win/loss/draw announcement
    - Option to restart

- Game Screen:
  - Chess board with pieces
  - Move history panel
  - Dual timers for players
  - Visual move hints

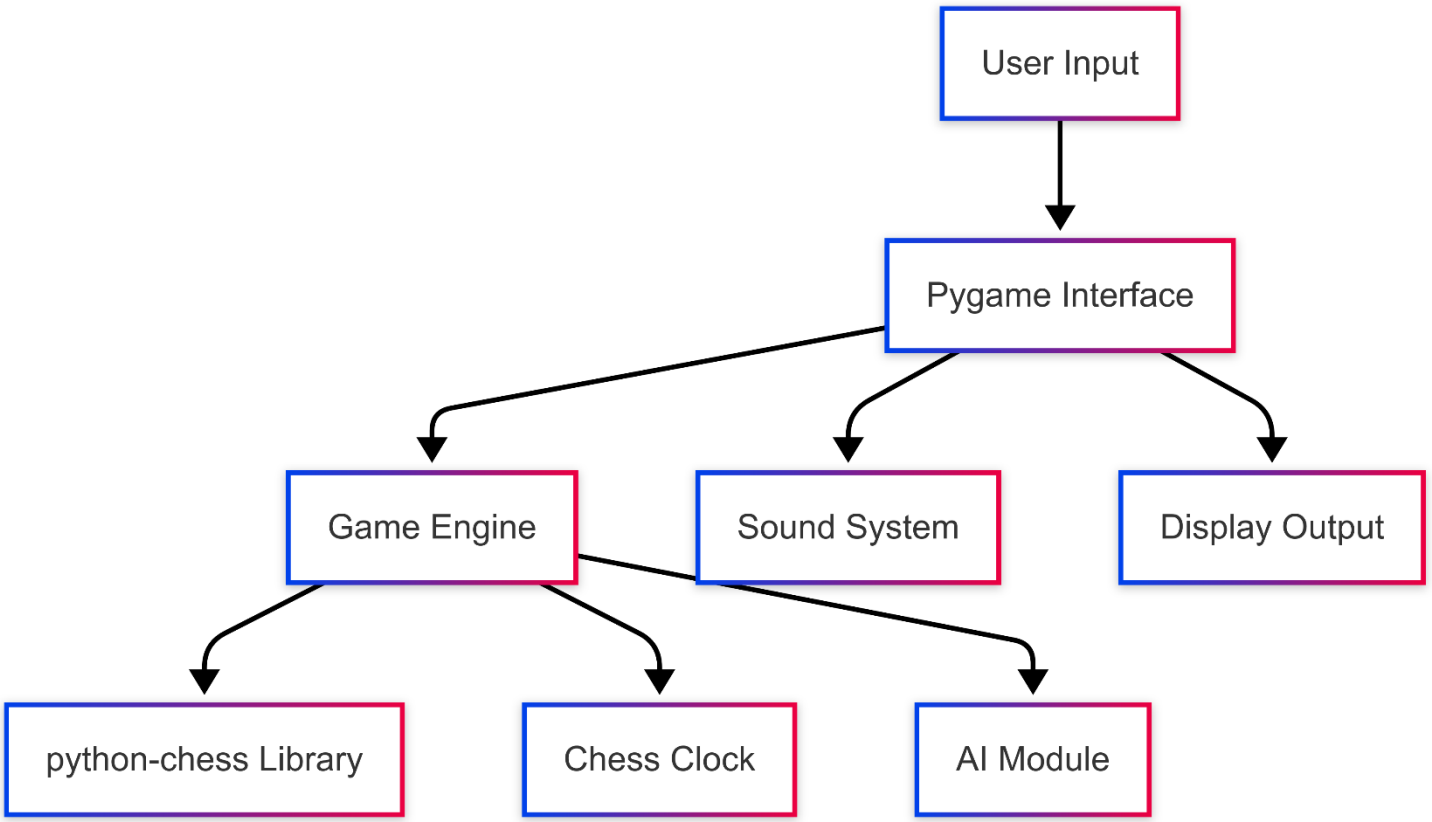


7. Planning

7.1. Development Time line

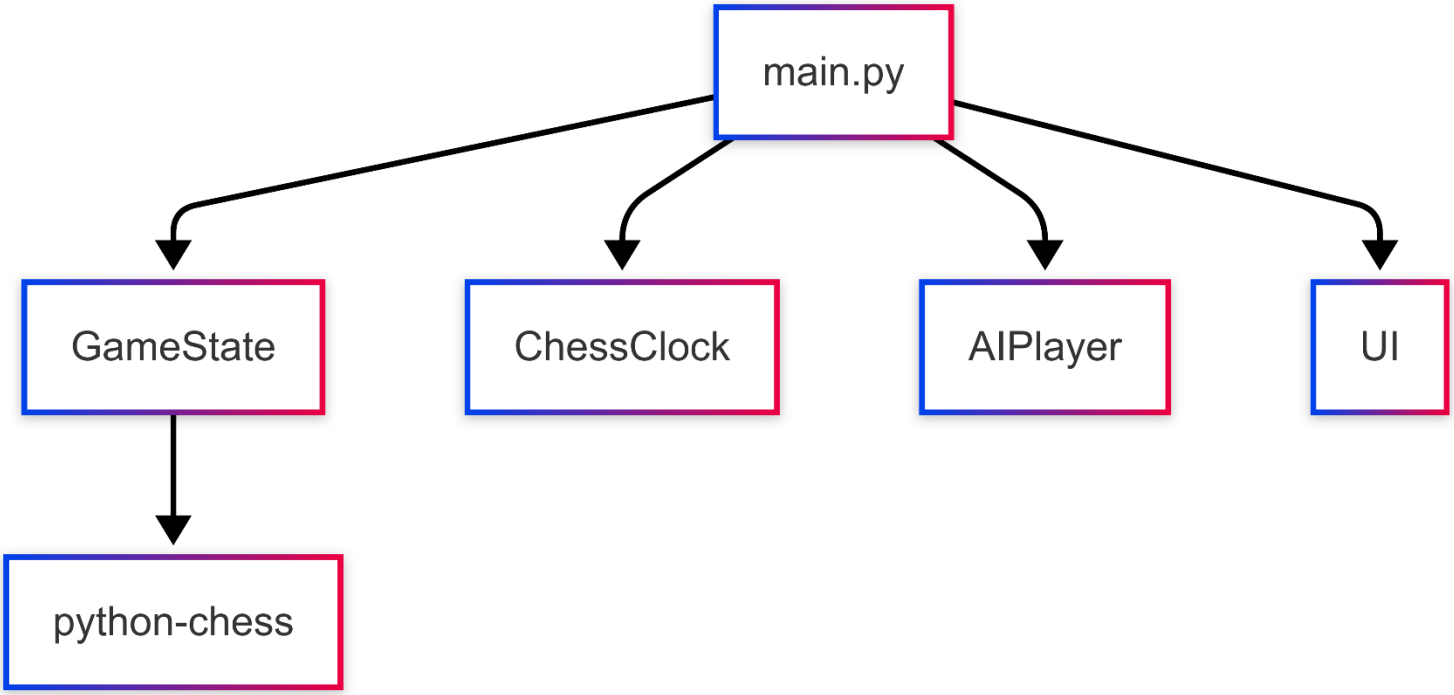


7.2 Network Diagram

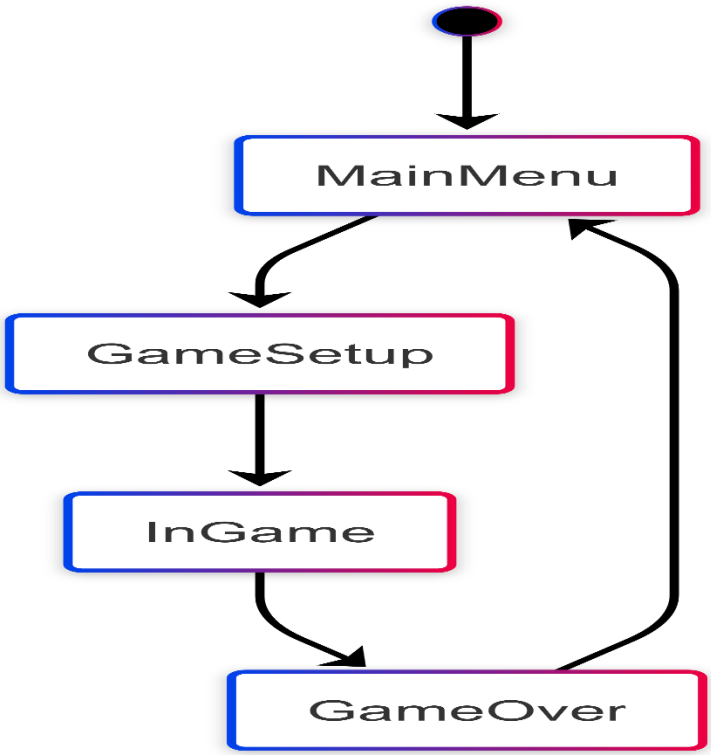


8. Designing

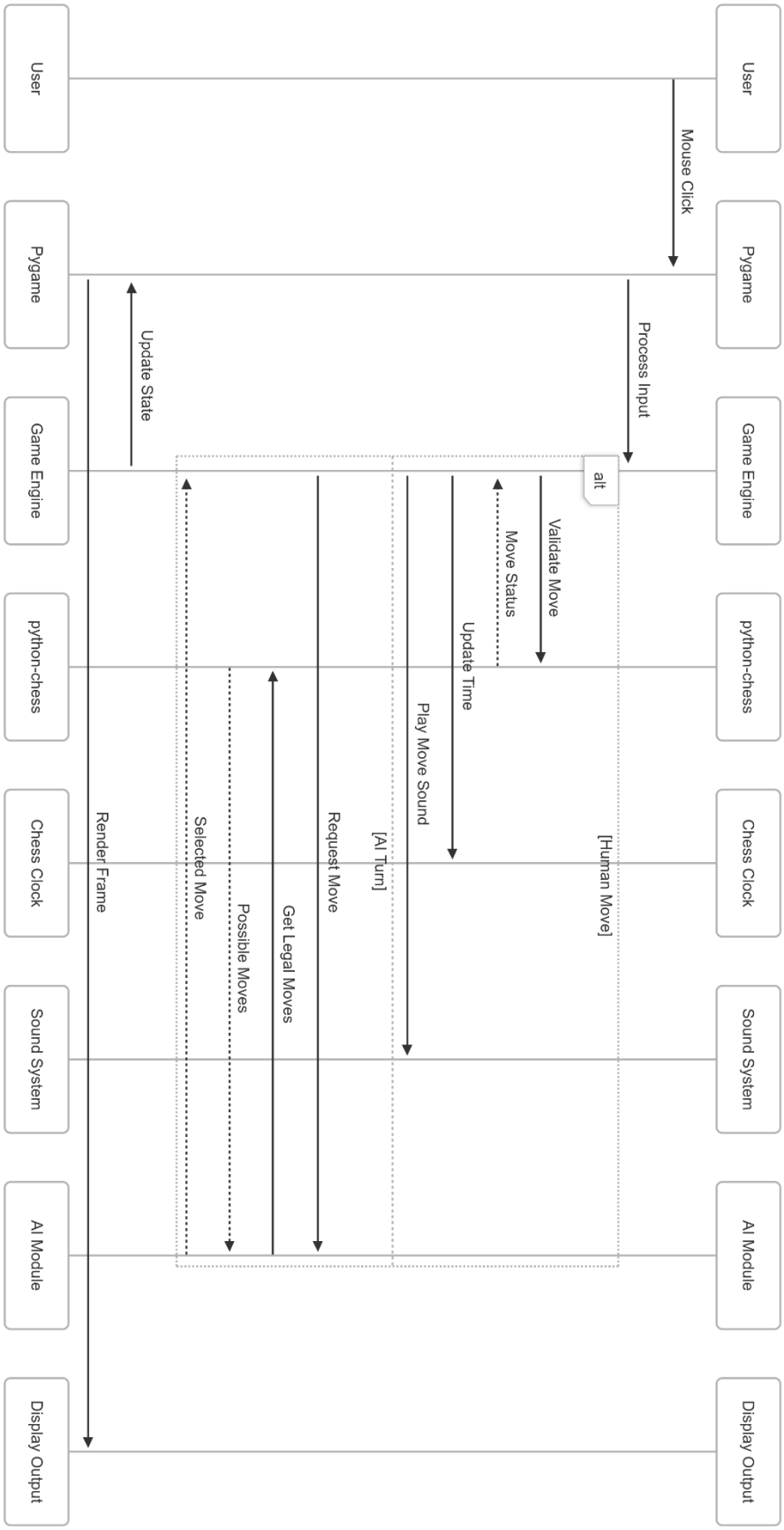
8.1.Architectural Diagram



8.2. State Diagram



8.3. Sequence Diagram



9. Future Improvements

1. **Enhanced AI**: Improved algorithms and difficulty levels
2. **Game Replays**: Save and review past games
3. **Customization**: Board/piece themes
4. **Tutorial Mode**: Learning tools for beginners
5. **Tournament Support**: Round-robin and elimination modes

---

10. Conclusion

The **Python Chess Game** provides a complete chess experience with both human and computer opponents.

Built with Pygame and python-chess, it offers:

- Authentic chess gameplay following all standard rules
- Flexible time controls for different play styles
- Clean, modular codebase for easy maintenance
- Engaging visual and audio feedback

This project demonstrates effective use of:

- Game development with Pygame
- AI implementation using minimax
- Object-oriented design principles
- Team collaboration through modular architecture

The system serves as both an entertaining game and a foundation for future chess-related projects. Potential enhancements could include online play, advanced AI, and additional training features.

we welcome any feedback or contributions to help us improve further.