A SYNOPSIS ON

CHESS WITH ARTIFICIAL INTELLIGENCE (A.I.)

Submitted in partial fulfillment of the requirements
for the award of the degree of

BACHELOR OF COMPUTER APPLICATIONS

To

Guru Gobind Singh Indraprastha University, Delhi

Under the Guidance of:
Dr. Barkha Bahl

Submitted by:
N. Krishna Khanth
BCA-VI Sem
02220602018
Sharad Jain
BCA-VI Sem
04020602018

Session 2018 – 2021

TRINITY INSTITUTE OF PROFESSIONAL STUDIES

(Affiliated to Guru Gobind Singh Indraprastha University, Delhi)
Ranked "A+" Institution by SFRC, Govt. of NCT of India
Recognized under section 2(f) of the UGC Act, 1956
NAAC Accredited "B++" Grade Institution

Table of Contents

1. Introduction

Chess is a game of strategy and tactics for two players, played on an 8x8 chequered board. Chess is fun and leisure activity that develops perspective, deepens focus, boosts planning skills, etc. Our project implements the chess game with graphical user interface. The chess game follows the basic rules of chess and all the chess pieces only move according to valid moves for that piece. We propose an application which will allow users to play chess against other players as well as computer AI. Our motive is to make computer think the possible moves and decide a suitable move depending on the difficulty level in the constraints of the game.

1.1 Idea and Purpose

We suggest an application with GUI which will allow users to play chess with other players as well as a computer AI. This application will provide premium features like high level AI free of cost and minimal computational loads. AI will use several complex algorithms to decide moves and provide a challenging experience to users.
This project will allow us to explore field of artificial intelligence and its scope.

2. Methodology

2.1 Existing Methods

Presently, the following methods are used in chess application:

1. Limited or no UI customisation – Present applications for chess provide almost no customisation to its UI.
2. Expensive features – Advanced features like high level AI are not free and expensive.
3. Poor AI – Many free of cost AI provide no challenge for even an average player.

2.2 Motivation

This application's motivation arose due to the current money-making trend in game industry:

1. Players are stuck with provided UI for a given application.
2. Game becomes tedious if it does not provide enough of a challenge.
3. Post-game notation is not available in many applications.
4. Players cannot set maximum duration of game.

As players, we feel that these problems cause discomfort in gaming experience.

2.3 Suggested Solutions

We suggest the following solutions to the above-discussed problems:

1. A free of cost chess application with AI functionality.
2. Several option to customise UI of the application.
3. Multiple levels of difficulty for AI.

2.4 Feasibility Study

An Assessment of the feasibility of the project

Economic Feasibility

The project is economically feasible as it works with functions with low-cost services such as laptops and desktops.

Free-ware, only cost of running computer

Technical Feasibility

The current project is technically feasible as the application requires:

1. Any python supported IDE
2. Server-Side Services
3. GUI development tools

All these are readily available and can be successfully deployed on any available computer.

Behavioural Feasibility

The application is behaviourally feasible since it requires no technical guidance; all the modules are user friendly.

Operational Feasibility

The application is operationally feasible as:

1. Complete GUI-Base, which is user friendly.
2. Inputs to be taken are self-explanatory.

2.5 Software Requirement Specifications

Software Requirement Specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements and may include a set of use cases that describe user interactions that the software must provide. It establishes the basis for an agreement between customers and the software providers on what the software product is to do and what it is not expected to do so that there is no room for confusion in the future. If used appropriately, SRS can help prevent software project failure.

Our proposed system has the following requirements:

1. Game will follow rules of chess as provided by International Chess Federation FIDE.
2. Players should be able to move chess pieces using mouse.
3. PGN notation of each move should be displayed at time of gameplay.
4. A file should be saved after each game containing PGN (Post Game Notation) of that game.
5. Players should be able to play with other players.
6. Players should be able to play with computer AI.
7. Players should be able to choose from at least 3 difficulty levels.
8. Proper message should be displayed for every special event like check, checkmate, stalemate and draw.
9. System should display time elapsed for every move.
10. A clock should be displayed for total time remaining or total time elapsed since beginning of the game.
11. Players should be able to set a time limit of the game.

3. Tools: Software and Hardware Requirements

3.1 Tools and Technologies

The following tools and technologies are expected to be used in development. Further may be added as the operations are implemented.

1. Language: Python
2. Libraries: Chess, Tkinter, pygame

3.2 Hardware Requirements

The following hardware requirements are recommended to be fulfilled in order to run this software.

1. CPU: Intel core i3 3rd Generation / AMD FX-6100
2. RAM: 2 GB
3. GPU: Integrated Graphics
4. Storage: 1 GB

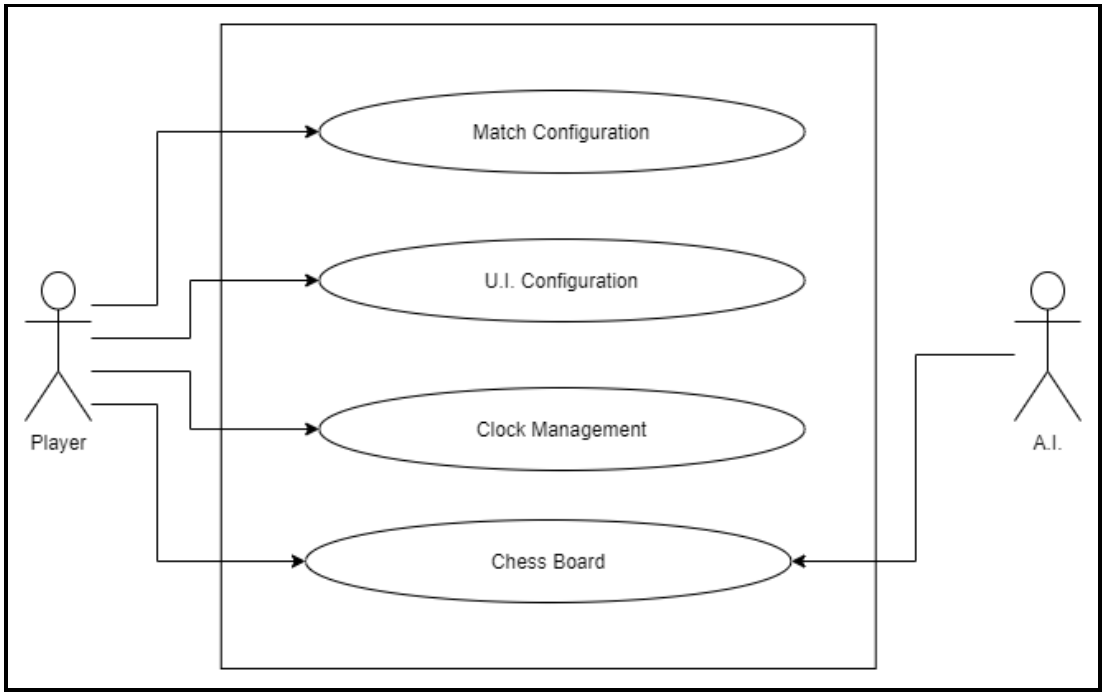
3.3 Software Requirements

The following software requirements are recommended to be fulfilled in order to run this software.

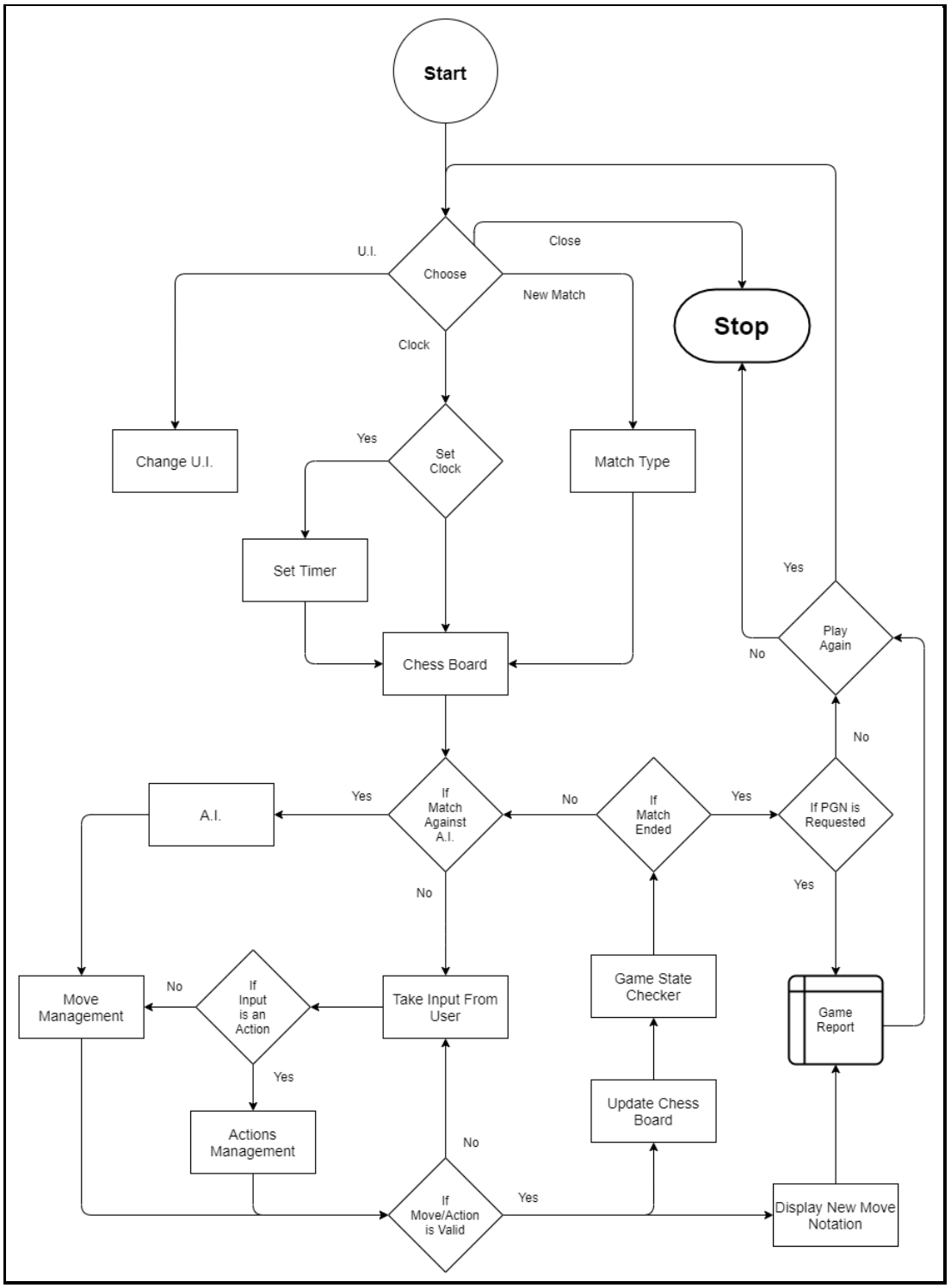
1. OS: Any Operating System
2. Programming Language: Python

4. System Design

4.1 Use Case Diagram:

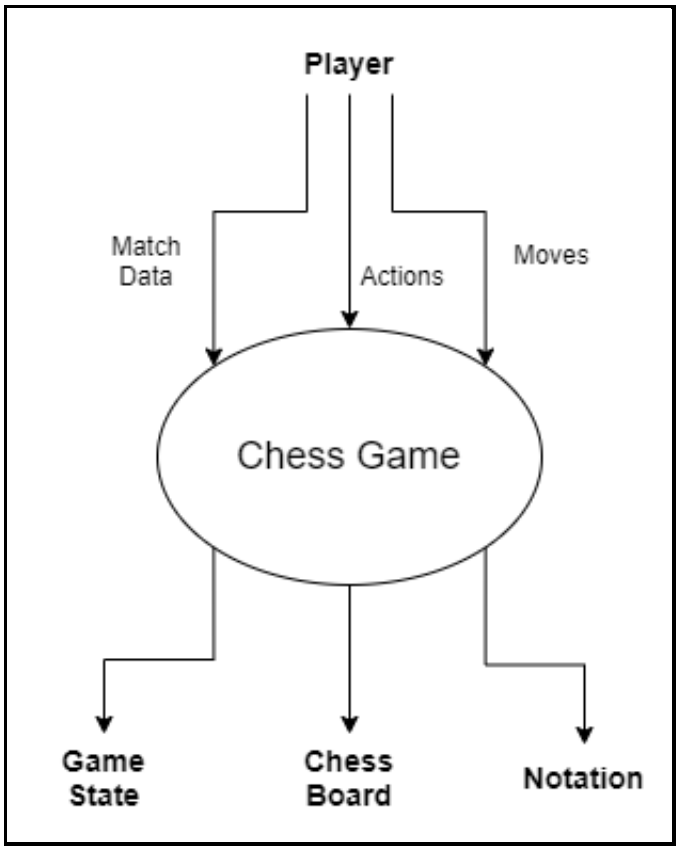


4.2 Flow Chart:

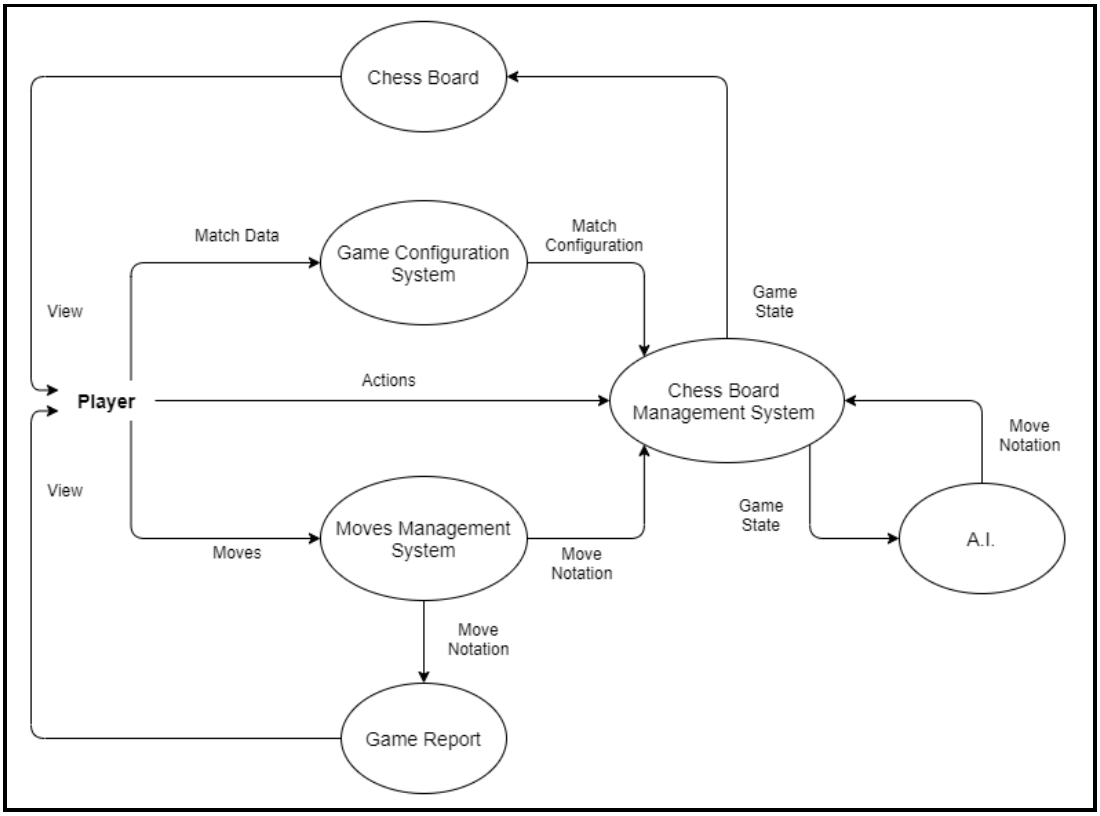


4.3 Data Flow Diagram (DFD):

Level 0:

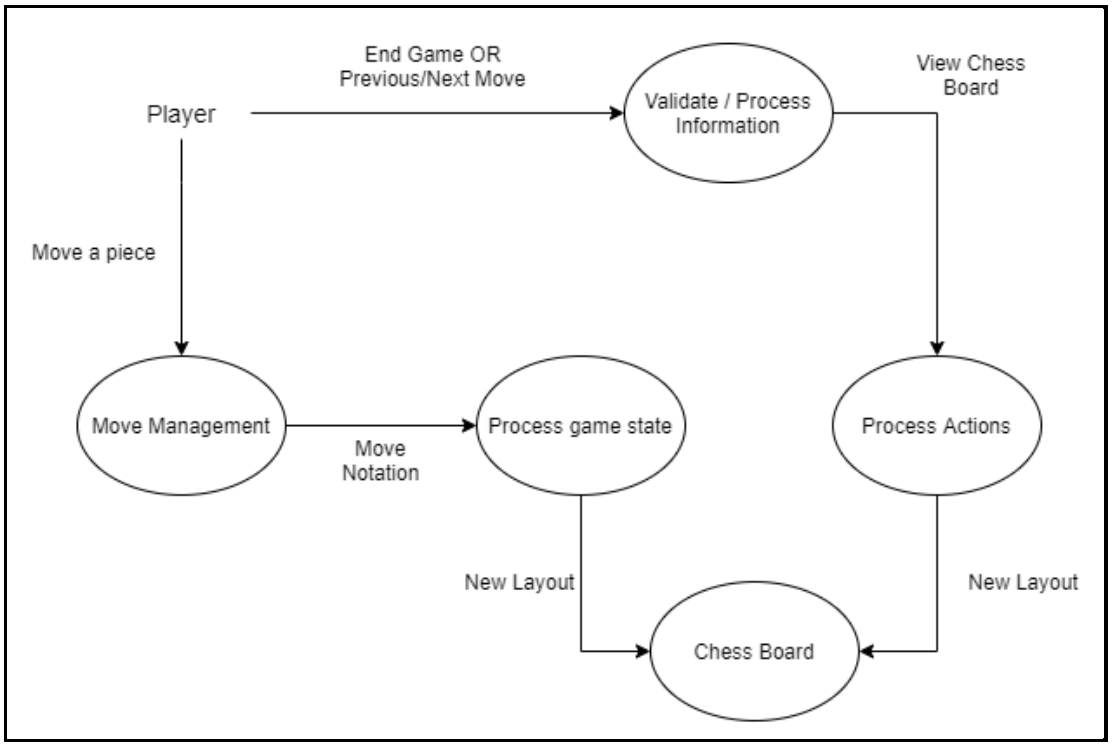


Level 1:

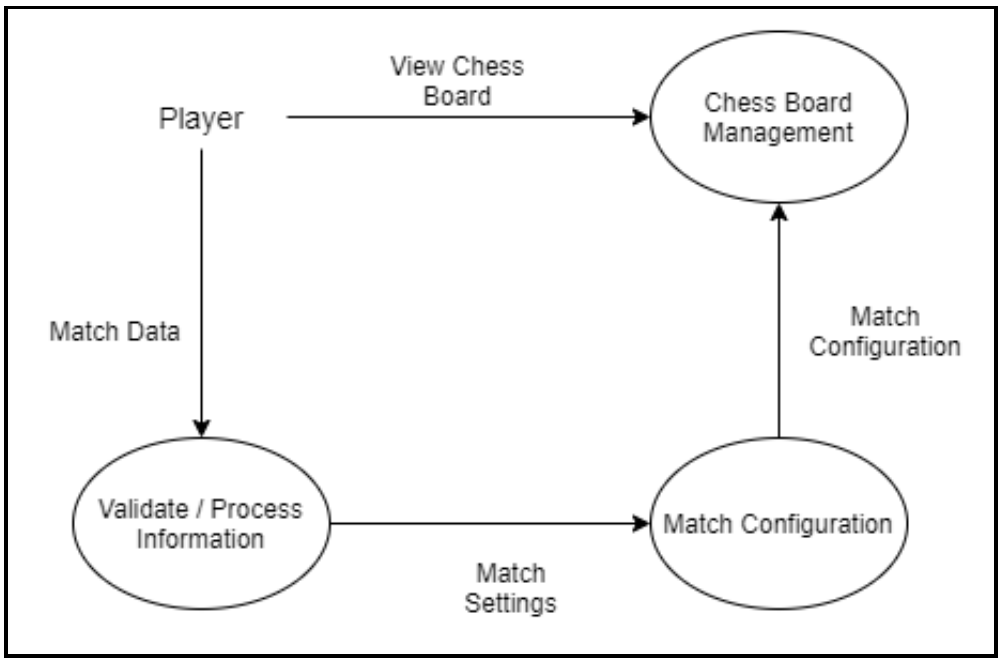

Chess Board
Match Data
Game Configuration System
Match Configuration
Game State
View
Player
Actions
Chess Board Management System
Move Notation
View
Moves Management System
Moves
Move Notation
Game State
A.I.
Move Notation
Game Report

Level 2:

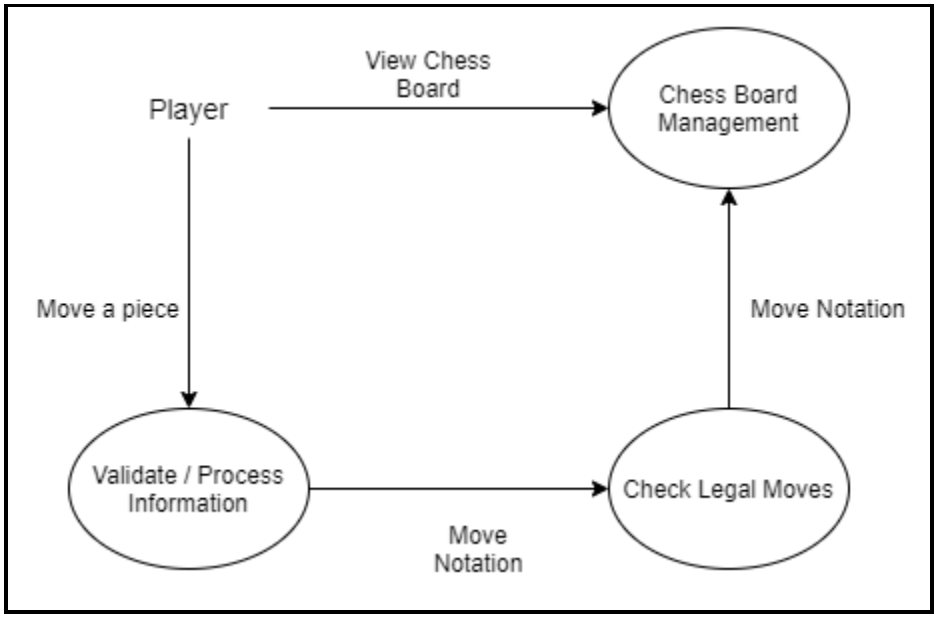
Chess Board Management:



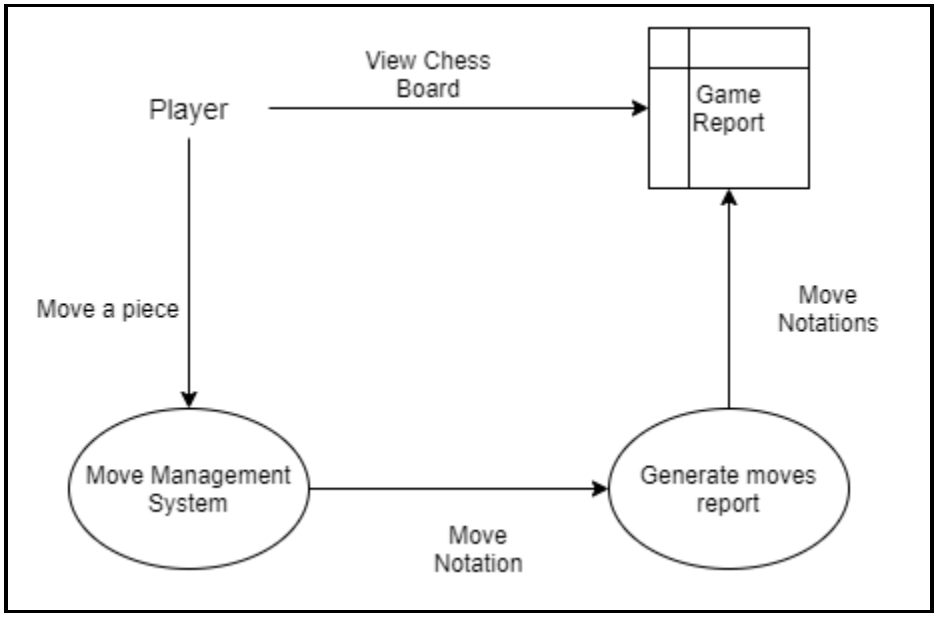
Game Configuration System:



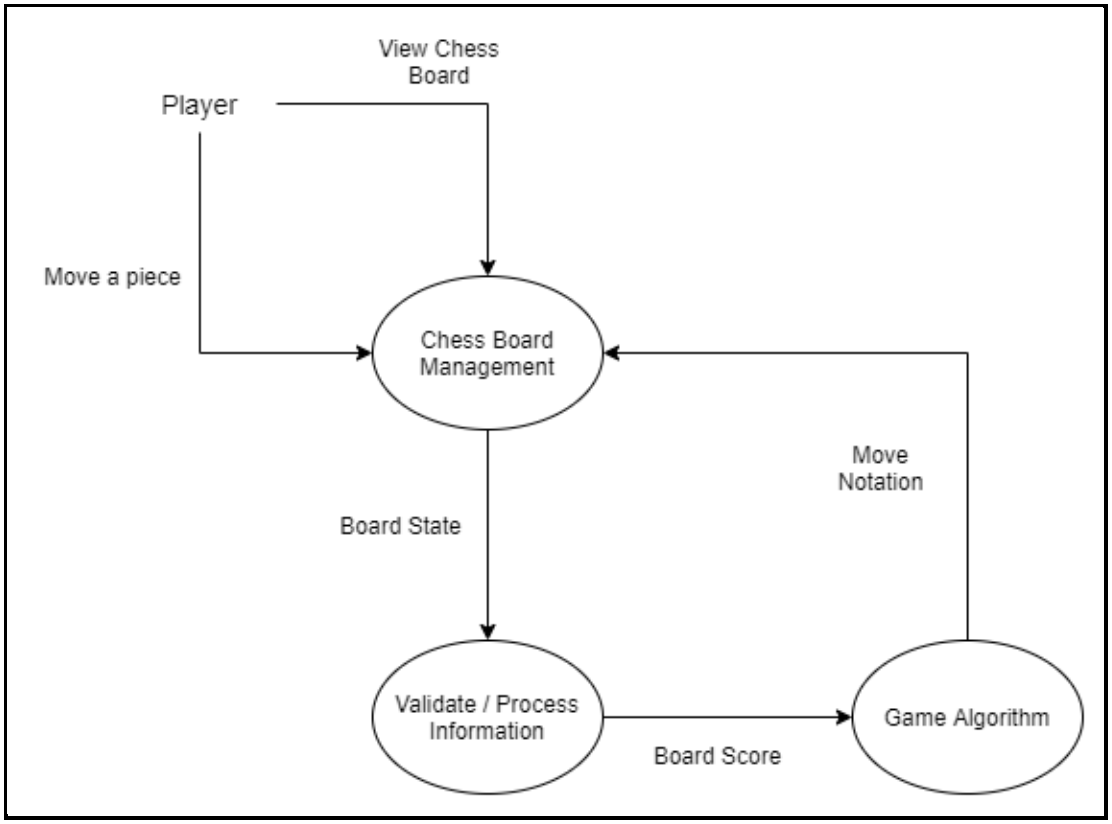
Move Management:



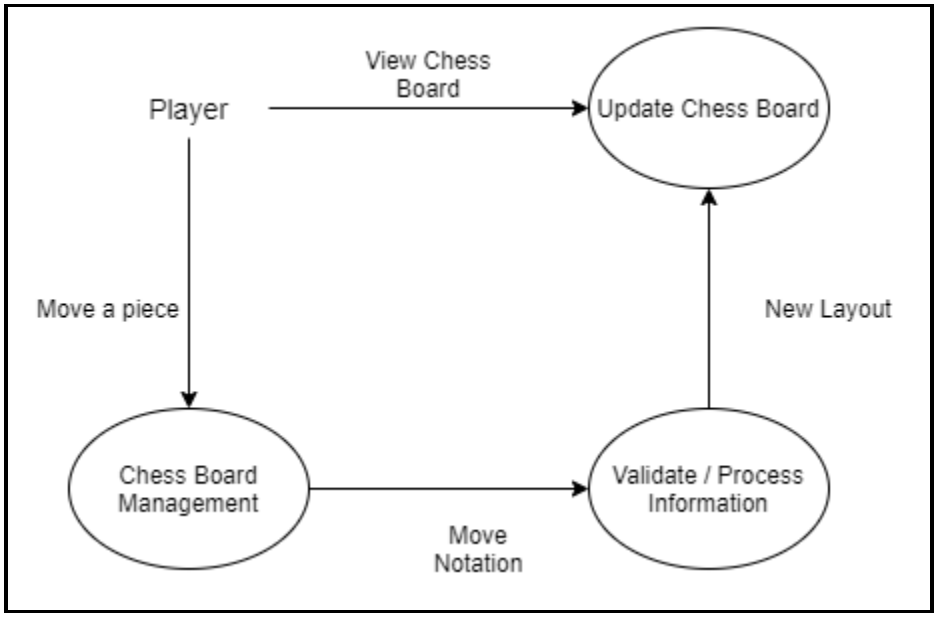
Game Report:



A.I.:



Chess Board:


Player
View Chess Board
Update Chess Board
Move a piece
New Layout
Chess Board Management
Validate / Process Information
Move Notation

5. Future Scope

As of now, this project is a desktop application. The players have to be on the same computer to play against each other.

In the near future:

1. This application could be redesigned as an internet connected or web-based application to make online matches possible.
2. Login and scoring functionality can be added. Friend list functionality can be added so players can easily play with specific players. Functionality to maintain overall score of players can be added.
3. This will make the match making process easier for the players and encourage more people to participate.
4. Functionality to analyse a chess match can be added.
5. AI can be further improved.

6. Conclusion

The application would provide a simple chess playing experience to users. A free application with good A.I. for chess enthusiast. We hope it would provide a satisfactory experience playing against provided A.I. player.

7. References

[1] https://pypi.org/project/pygame/

[2] https://pypi.org/project/tkintertable/

[3] https://pypi.org/project/python-chess/

[4] https://stackabuse.com/minimax-and-alpha-beta-pruning-in-python/

[5] https://www.fide.com/FIDE/handbook/LawsOfChess.pdf

[6] https://handbook.fide.com/

[7] https://www.w3schools.com/python/python_modules.asp

[8] https://www.chessprogramming.org/Minimax

[9] https://www.chessprogramming.org/Negamax

[10] https://pypi.org/project/chess/

[11] https://medium.com/the-innovation/the-anatomy-of-a-chess-ai-2087d0d565#:~:text=The%20minimax%20algorithm%20takes%20advantage,other%20tries%20to%20minimize%20it.

[12] https://www.chessprogramming.org/NegaScout

[13] https://www.chessprogramming.org/Alpha-Beta

[14] https://www.chessprogramming.org/Quiescence_Search

[15] https://medium.com/dscvitpune/lets-create-a-chess-ai-8542a12afef