



**Indian Institute of Information  
Technology Allahabad,**  
*Prayagraj UP(India)*

---

*Department of Information Technology*

*Project on*

**“3D Educational Game Using OpenGL”**



***“Chess”***

**Guide by**

Prof. Anupam Aggarwal

**Group - 01 by**

1. Shubham Kumar  
Bhokta(IIT2020007)

**Date: 8 th Marc 2023**

2. Rahul Mahato(IIT2020022)

---

# 1. Project Description

## 1.1

**Objective:** To develop a 2-player chess game that allows users to play against each other on a single computer

**Functionality:** The game will have the following features:

- *Display of the chess board:* The chess board will be displayed on the screen using graphics and will be updated in real-time as moves are made.
- *Piece movement:* Users will be able to select and move pieces on the board according to the rules of chess. The game will validate moves to ensure that they are legal.
- *Capture of pieces:* Pieces can be captured by moving a piece to a square occupied by an opponent's piece. The captured piece will be removed from the board.
- *Check and checkmate:* The game will detect when a player is in check and checkmate. If a player is in checkmate, the game will declare the opponent as the winner.
- *Undo/Redo:* The game will allow users to undo and redo their moves.
- *Saving and loading of games:* The game will allow users to save and load their games so they can resume playing later.
-

## 1.2 Overall Description

- **Product Functions**

S.N	Use Cases	Description of use cases
1	The KING	The King can move one square in any direction
2	The QUEEN	The Queen can move to any square in any direction as long as her path is not blocked
3	The ROOK	The Rook can move to any square along its file or row as long as its path is not blocked
4	The BISHOP	he Bishop can move to any square along its diagonals as long as its path is not blocked
5	The KNIGHT	The Knight is the only piece that can hop over other pieces in an L-shaped path
6	The PAWN	he Pawn may move only one square forward if its path is not blocked

## 2. Introduction

The current level of development in computer chess programming is fairly complicated, yet interesting as well. In this project, we were supposed to develop a chess-playing program. The program was supposed to play chess at a good level and have an adaptive property. The description of the project included the analysis of the evaluation of the pieces and board positions on the board. In addition, a literature survey on methods, techniques, and heuristics used in chess playing and chess programming was supposed to be done in order to analyze how the pieces, board, and positions on the board are evaluated.

## ○ **2.1 Purpose**

The purpose of this document is to provide a detailed requirements overview for the Chess application. It will also explain the purpose and features. Chess shall provide the user with a reliable, well designed chess game. The version of this product is 1.0 and this SRS shall cover the full scope of Chess.

## ○ **2.2 Document Conventions**

*Acronyms and Abbreviations:*

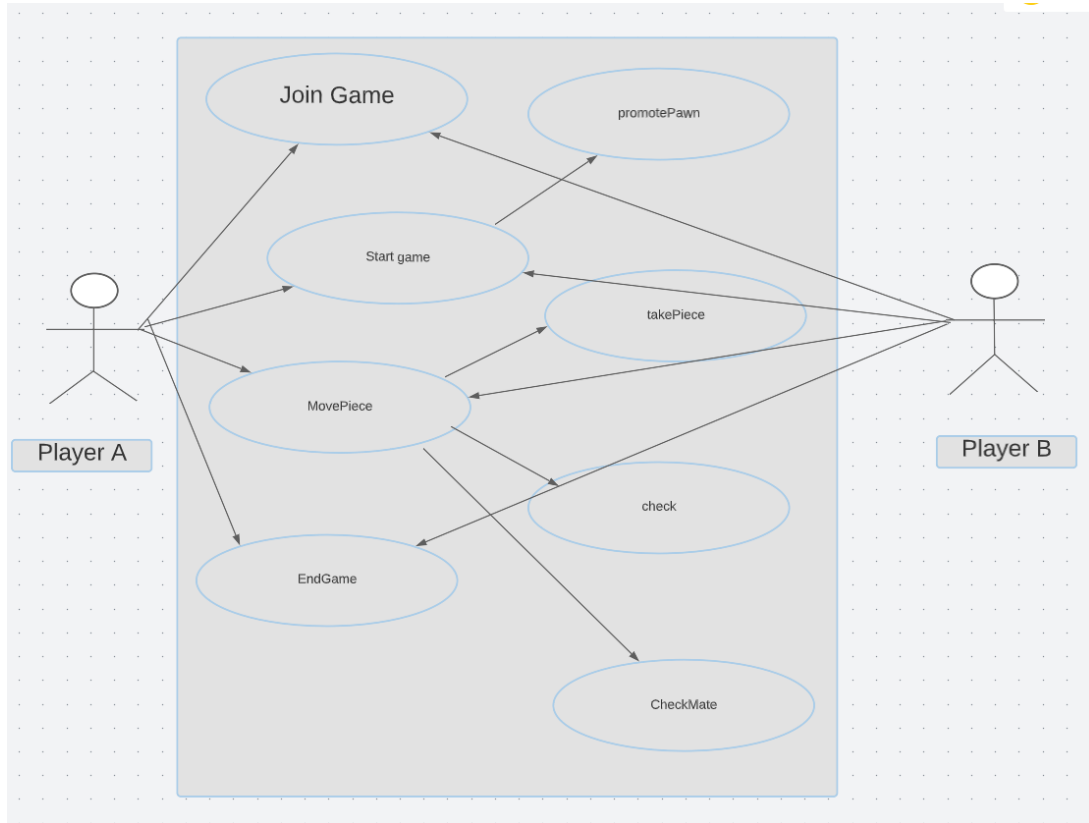
- SRS: Software Requirement Specification

*Definitions/Explanations:*

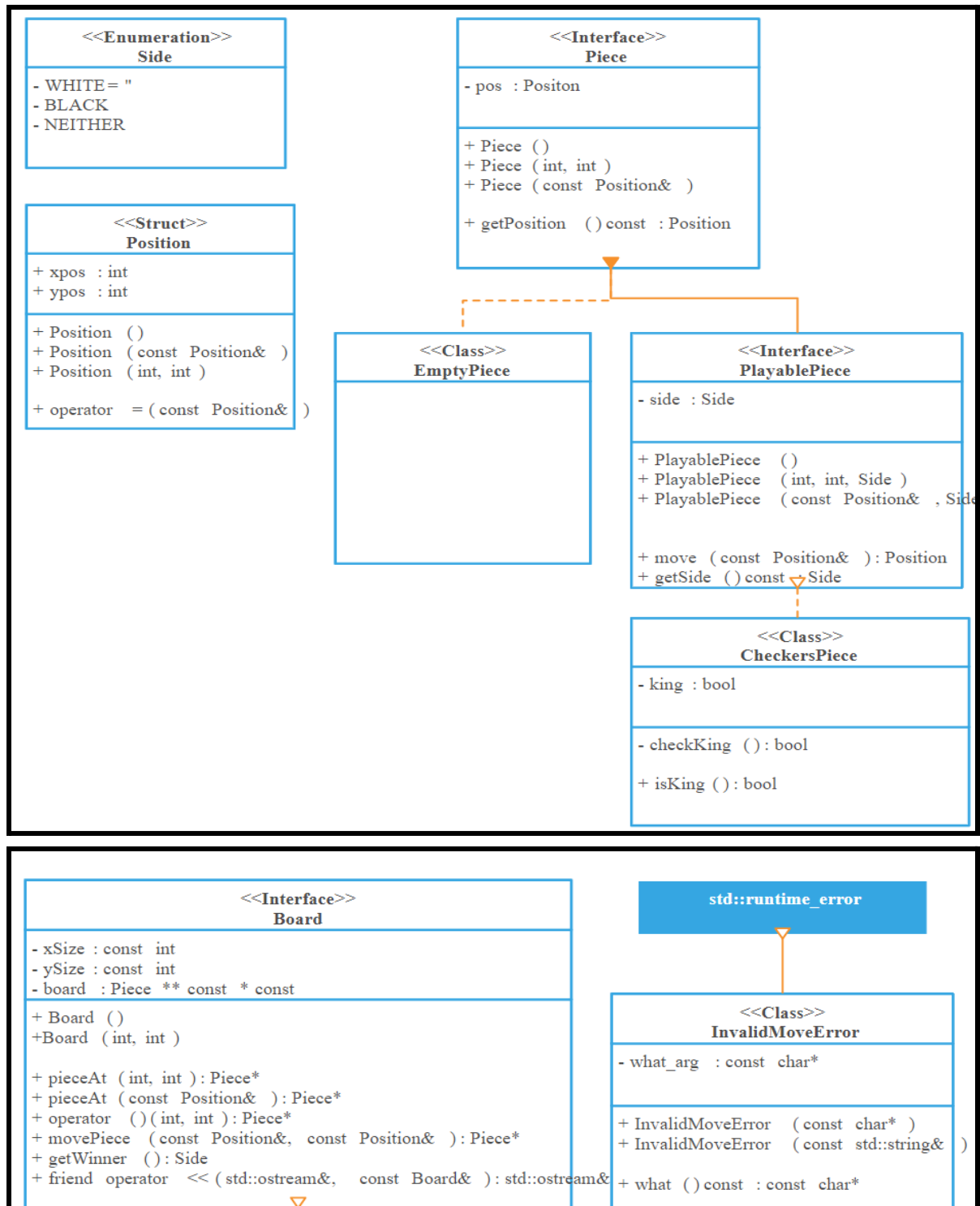
- The KING
- The QUEEN
- The ROOK
- The BISHOP
- The KNIGHT
- The PAWN

### 3. Proposed UML Diagram for Chess Game

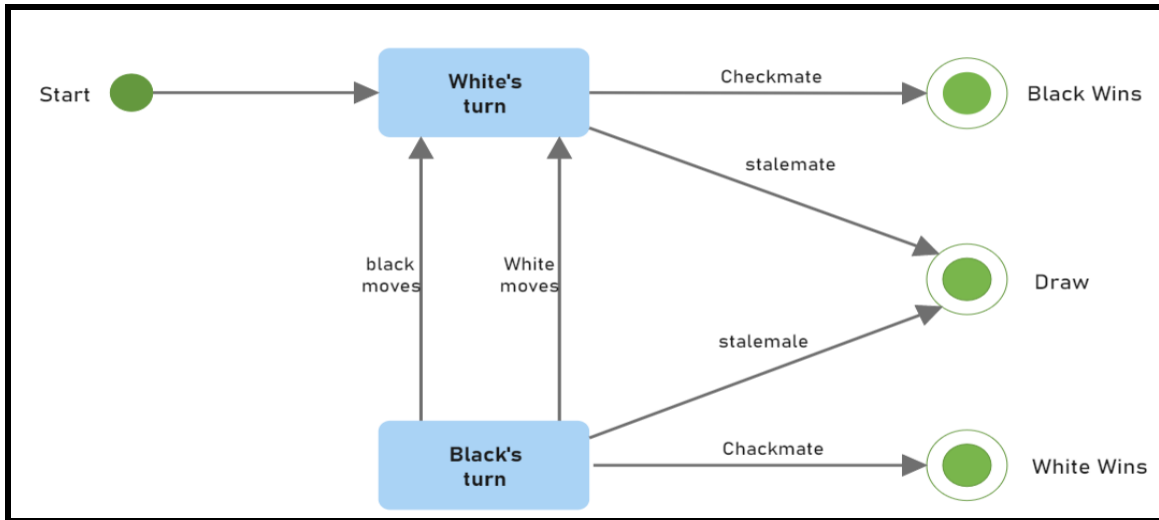
#### 3.1 Use Cases Diagram:



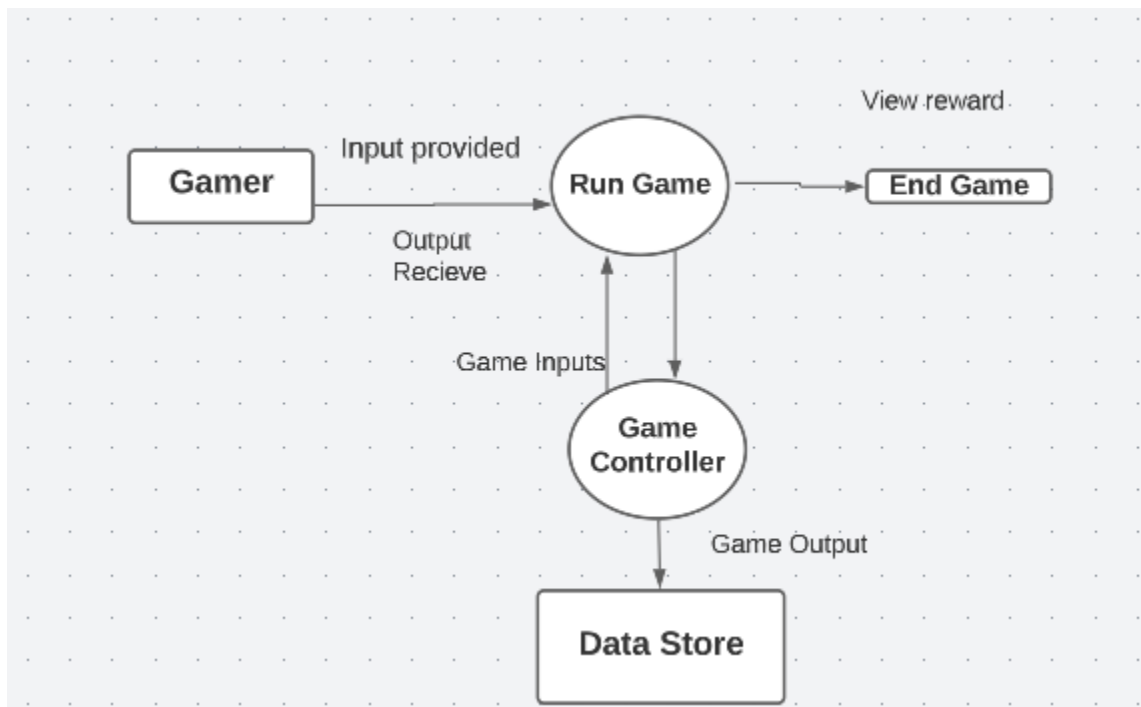
## 3.2 Class Diagram



## 3.3 State Diagram:)



### 3.4 Simple flow of game diagram



### 3.5 User Classes and characteristics

The expected user shall be anyone with access to a computer which can run the gcc. There are no privilege levels/educational level/expertise nor experience required. The user is not expected to know Chess, as the rules and tips/strategies shall be included in the application.

### **3.6 Operating Environment**

This shall be a c++ based application, so the user must have c++ 17

### **3.7 Design and Implementation Constraints**

#### **3.8 Design Constraints:**

Game rules: The 3D chess game should adhere to the traditional rules of chess, with some modifications to accommodate the additional dimension.

Board design: The game board should be designed to represent a three-dimensional space, with squares or cells on each level to represent the different heights of the pieces.

Piece design: The pieces should be designed to reflect their traditional counterparts, but with an added dimension to accommodate the third axis of movement.

User interface: The game should have an intuitive user interface that makes it easy for players to navigate and make moves within the 3D environment.

Game AI: The game should have a computer opponent or AI that can play a competitive game of chess.

#### **3.9 Implementation Constraints:**



Performance: The game should run smoothly and efficiently, with fast load times and no lag during play.

Compatibility: The game should be compatible with a range of devices, including desktop computers, laptops, and mobile devices.

Storage space: The game should have a reasonable file size to allow for easy downloading and installation.

Network connectivity: If the game has online multiplayer capabilities, it should work smoothly over a variety of network connections.

Security: The game should be secure, with appropriate measures in place to prevent cheating and protect player data.

### **3.10 Assumptions and Dependencies**

Since this will be a standalone application, there shall not be any other dependencies aside from the requirement of having the c++environment installed in the computer.

## **4. Specific Requirements:**

### **4.1 Functional Requirements: The Pieces and The Rules:**

#### **Use Case 1:**

##### **The KING:**

The King is the most important piece. When it

is trapped so it cannot move without being captured, then the game is lost. This trap is called checkmate.

The King can move one square in any direction. A King can never move into check, or onto a square where it can be captured by an opponent's piece. If a King is not in check, and no other legal move is possible, then the position is said to be in stalemate. A stalemated game is a draw, or a tie.

## **Use Case 2:**

### **The QUEEN:**

The Queen is the most powerful piece.

The Queen can move to any square in any direction as long as her path is not blocked. Her range and the ability to attack many pieces at once are the source of her power.

## **Use Case 3:**

### **The ROOK:**

The Rook is the next very powerful piece after

Queen. The Rook can move to any square along its file or row as long as its path is not blocked. Its range is the source of its power

## **Use Case 4:**

### **The BISHOP:**

The Bishop comes next to Rook in terms of power. The Bishop can move to any square along its diagonals as long as its path is not blocked. Its range is the source of its power

## **Use Case 5:**

**The KNIGHT:**

The Knight is nearly as powerful as the Bishop. The Knight is the only piece that can hop over other pieces in an L-shaped path. This ability makes it particularly powerful in the early stage of a game when the board is crowded with pieces

**Use Case 6:****The PAWN:**

The Pawn is the least powerful piece because of its poor mobility. The Pawn may move only one square forward if its path is not blocked. However, it may move as an option one or two squares forward on its first move only. It may capture only one square diagonally. It may not capture forward. It may not move backward. The lowly Pawn usually does not last long, but if it is able to reach the 8th row or rank, then it can promote itself to any other piece except the King. A Pawn thus promoted is replaced by that piece. Therefore, it is possible to have more than one Queen, or two Rooks, Bishops, or Knights on the board at one time.

**4.2 Hardware Interfaces:**

- Laptop, computers with minimum 2gb ram and 10gb hard disk
- Battery Backup for emergencies

**4.3 Software Interfaces:**

- Graphical User Interface (GUI): The 3D chess game should have a graphical user interface that allows players to interact with the game, make moves, and view the game board. The GUI should be intuitive and easy to use.

- **Move Input Interface:** The game should have an interface for receiving player moves. This can be achieved through mouse clicks, touch events, or keyboard inputs.
- **Piece Movement Engine:** The game should have a piece movement engine that interprets player moves and updates the game state accordingly.
- **AI Engine:** If the game has an AI opponent, it should have an AI engine that generates moves for the computer player.
- **Sound and Music Interface:** The game may also have a sound and music interface that adds sound effects and background music to enhance the player experience.

#### **4.4 Communication Interfaces(Proposed):**

**Network Interface:** If the game has online multiplayer capabilities, it should have a network interface that allows players to connect and play games with each other over the internet. The network interface should handle data transmission between players, ensuring a stable and secure connection.

**Online Matchmaking:** The game may have an online matchmaking interface that allows players to find and join games with other players.

**Chat Interface:** If the game has multiplayer capabilities, it may have a chat interface that allows players to communicate with each other during games.

**Notifications Interface:** The game may have a notifications interface that provides players with updates on game events, such as an opponent's move or the end of the game.

**User Account System:** If the game has an online component, it may have a user account system that allows players to create profiles, save progress, and access leaderboards. The user account system may also handle player authentication and authorization.

#### **4.4 Some Special Moves and Cases:**

##### **1) Castling:**

Here Black is castled short or on the King side. White is uncastled. Here Black is castled long or on the Queen side. White is uncastled. Castling is an important move in chess. It allows a player to quickly move both the King to safety and the Rook to the center for battle. For this reason, wise players carefully guard their ability to castle and usually castle early in the game. Likewise, clever players will attempt to prevent their opponent from castling.

When castling the player moves his King two squares toward one of the player's Rooks and moves that Rook to the opposite side of the King. A player may not castle if either the King or the Rook involved have already moved. Also, the King may not castle out of, through, or into check. There must be no pieces between the King and Rook when castling.

## **2) En-passant:**

A player may capture another player's pawn in passing (En Passant) under very specific circumstances. This move is designed to prevent a player from taking advantage of the two-square first move rule for pawns which might allow them to pass their opponent's pawn(s) without a chance to capture.

The capture is made exactly as if the pawn moved only one square on the first move. In the picture, Black's pawn moved up two squares as is its right. White captured the pawn by removing it from the board and placing the passed white pawn on the square marked ep before playing another move. This move, like any other, is optional and can occur as often as a similar situation arises between pawns.

## **5. Other Nonfunctional Requirements**

### **5.1 Performance Requirements**

Chess shall not crash. Other than that, there are not any specific performance requirements. The time complexity and space complexity have not been specified. The overall goal shall be to get the program running without any bugs or user-related problems.

### **5.2 Hardware Requirements**

- Standard PC, Computers
- Internet connection with good speed.
- Pentium IV 1.7Ghz dos or better processor.
- 1 GB ram or more
- At least 10 Gb Hard Disk Space.

### 5.3 Software Requirements

To develop a chess game using OpenGL, you would need the following software requirements:

Operating system: Linux , windows

A development environment: This can be an Integrated Development Environment (IDE) such as Visual Studio or Code::Blocks, or a text editor such as Sublime Text or Notepad++.

OpenGL library: You would need to have the OpenGL libraries installed on your computer. You can obtain these libraries from the OpenGL website or from your operating system's package manager.

A graphics library: To create the pieces and board of the chess game, you would need a graphics library. One option could be the GLUT library, which is a simple and easy-to-use graphics library for OpenGL.

A chess engine: To make the game functional, you would need to implement a chess engine that can handle the rules of the game and generate moves. There are open-source chess engines available online that you can use or you can write your own.

A version control system: To manage the code and collaborate with other developers, it is recommended to use a version control system such as Git.

Note that the specific software requirements may vary depending on the platform you are developing on and the specific needs of your project.

#### 5.4 Safety Requirements

There are no safety requirements.

#### 5.5 Security Requirements

There are no security requirements. Chess has no database, no user information, or any private information that could be used to compromise an individual.

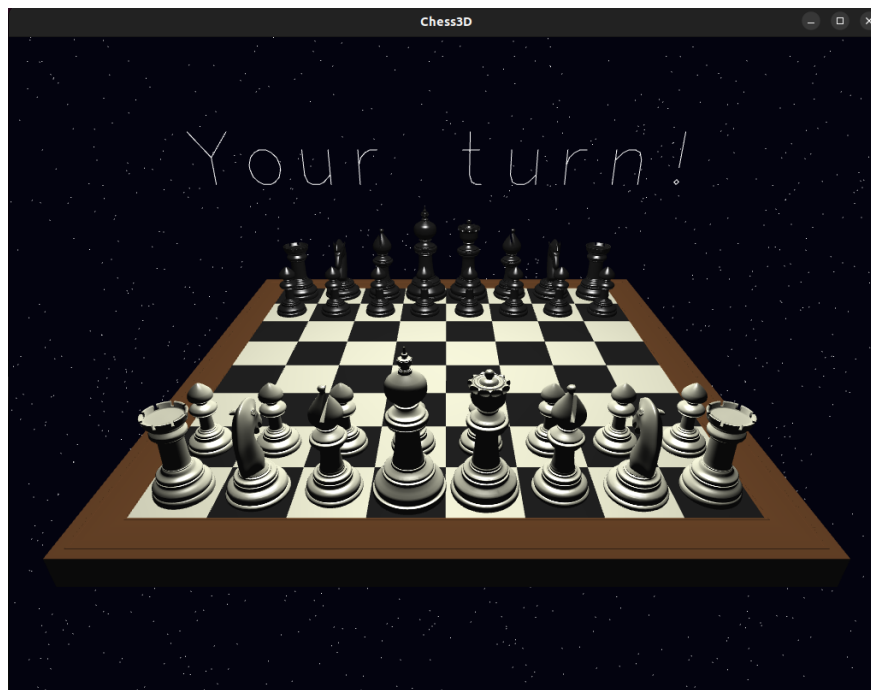
#### 5.6 Software Quality Attributes

Chess shall run smoothly and give the user a somewhat enjoyable experience playing chess at a basic level.

#### 5.7 Business Rules

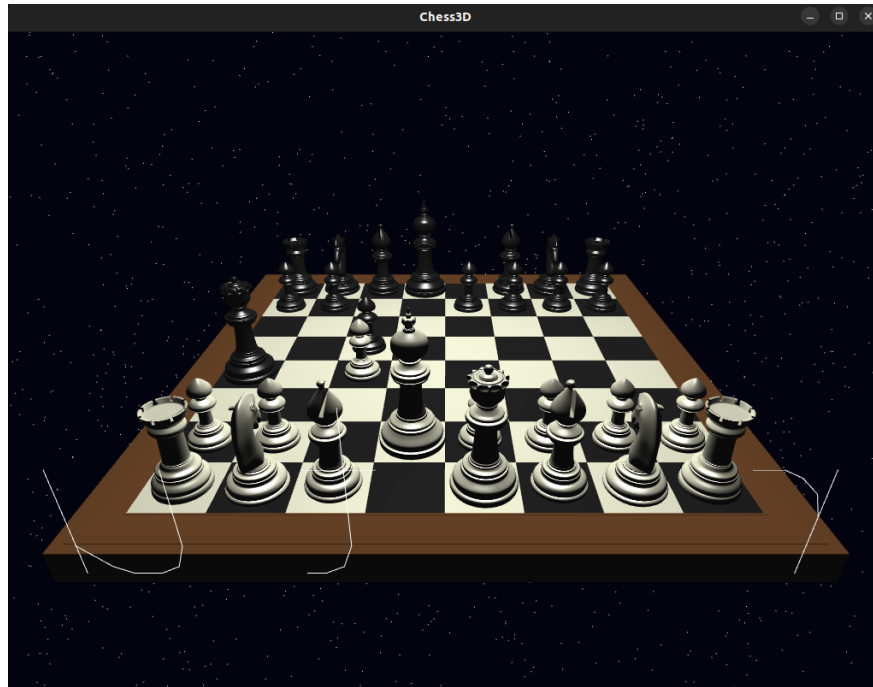
There are no business rules at this time.

### Screenshot



When we have start the game

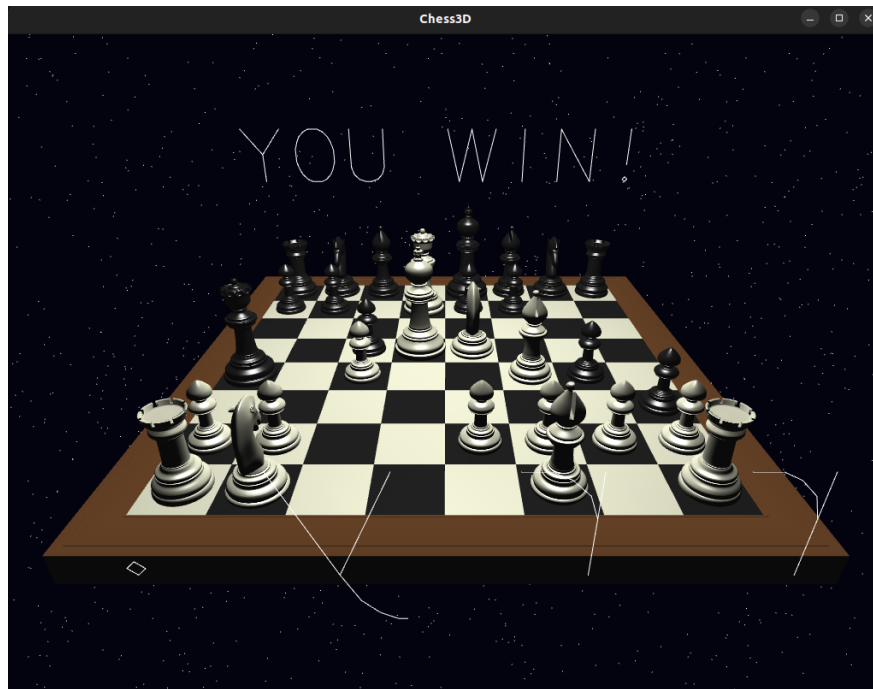




When we have playing the game



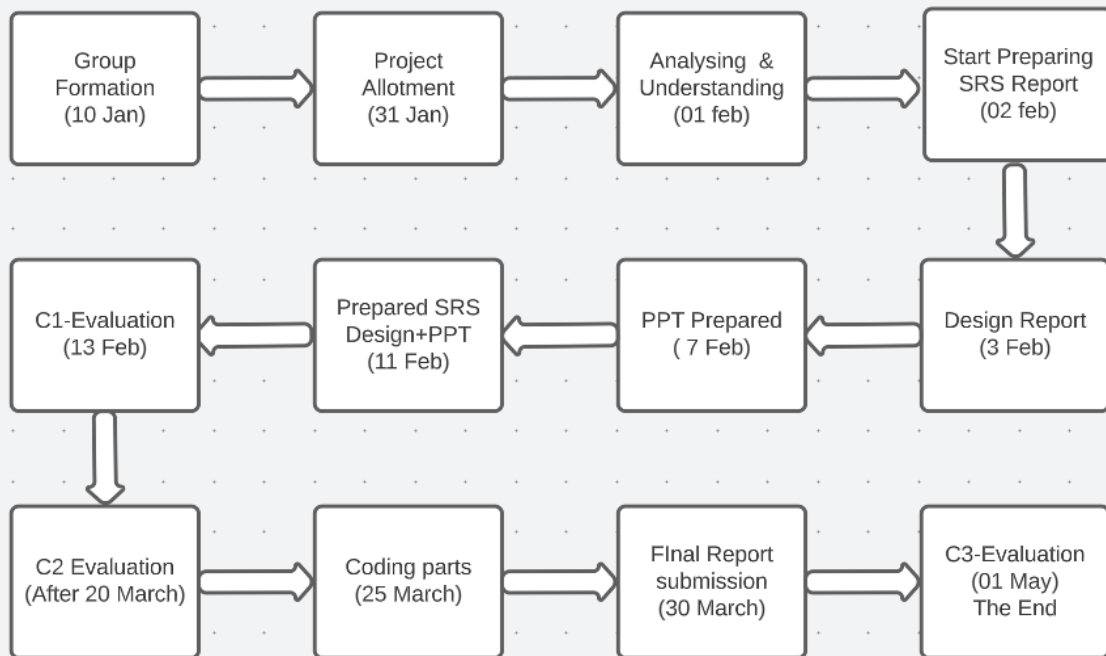
When we have taken wrong step



When we have win the match

**Activity Chart:)**

## Activity Chart for Game Development in OpenGL



## 6. Conclusion

The conclusion of a 3D chess game, like any chess game, is determined by the outcome of the play. The game ends when one player's king is checkmated, meaning that the king is in a position to be captured (in check) and there is no way to move the king out of capture (mate). A player may also resign, forfeit the game, or the game may end in a draw if neither player has a chance of winning. In the case of a draw, this can occur when neither player has enough pieces to checkmate the other, when a stalemate occurs (where a player cannot make a legal move), or if the same position is repeated three times.

## 7.Future Scope

The future scope of 3D chess games depends on the advancements in technology and the changing interests of the players. Currently, 3D chess games are mostly played on

computers and mobile devices, but with advancements in virtual and augmented reality technology, there is potential for 3D chess to be played in a more immersive and interactive environment.

In terms of the game itself, there is potential for new variations and rule sets to be introduced, as well as the integration of artificial intelligence to create more challenging opponents. There is also the possibility of adding more dimensions to the game, making it even more complex and challenging.

Overall, the future of 3D chess games will likely be shaped by the continued development of technology and the evolving interests and preferences of players. As long as there is interest in the game and its variations, it is likely that new and innovative versions of 3D chess will continue to be developed and enjoyed by players around the world.

## **8. References:**

**[1] Chess Gaming and Graphics Using Open-Source Tools**

<https://ieeexplore.ieee.org/document/5328146>

**[2] geeksforgeeks**

<https://www.geeksforgeeks.org/draw-a-chess-board-using-graphics-programming-in-c/>

**[3] Chess Gaming and Graphic**

<https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=bddeb8c213b6078880621977cdca95cd7f18b940>

