# Minor Project Report

# On

# A Multiplayer Chess

## Submitted by:

| Name | Roll No | Branch |
|------|---------|--------|
| Sarthak Jain | R171218088 | CSE-DevOps |
| Muskaan Madan | R171218063 | CSE-DevOps |
| Siddharth Chopra | R178218045 | CSE-MAD |
| Manvi Kulshrestha | R178218022 | CSE-MAD |

**Under the guidance of**

Dr. Monit Kapoor

Associate Professor

HOD - Department of Cybernetics

**UPES**

## School of Computer Science and Engineering

**University of Petroleum & Energy Studies**

**Dehradun - 248001**

**2020**

# School of Computer Science and Engineering
## University of Petroleum & Energy Studies, Dehradun

## Project Proposal Approval Form (2020)

### PROJECT TITLE:

### A Multiplayer Chess

**ABSTRACT:**

In this project we aim to implement the standard chess game using C, a basic and a broadly used language, supporting structured programming. The chessboard will be 2D and will have a grid size of 8 * 8. The chessboard and the pieces are going to be the essential elements of the project, on which further, we will be implementing the logic of the game. Without these two parts, all the functionalities cannot be achieved as the board must restrict the movement of pieces at any time, and the game cannot run without the pieces. [1]

The game provides human to human capability for both playing and learning, in which they need to be present in the same geographical location physically. It is a offline game which asks the user to choose their set of color for further playing and thus helps in deciding who wants the make the first move.

**Keywords**: chessboard and pieces, human to human capability.

# INDEX

# 1. INTRODUCTION

In the last five years the strength of computer chess programs has grown immensely. They have a place with easygoing games, and it requires two players to continue on a chessboard of size 8 * 8. The guidelines of this game are straightforward yet have boundless ways to move pieces, thus each step is urgent to the player.

Initially we are working with 2 player game. Later we will step up with multiplayers with similar algorithm. This program for chess will be the most efficient one till now in C language. The program supports a reasonable number of functions and the game will be in 2D. This game will be played in same system by the two different players.

## 2. PROBLEM STATEMENT:

How might we create new computer chess model so that chess engines merge the prerequisites on knowledge implementation with a maximum of efficiency?

Our challenge is to develop multiplayer chess, for which we require three things:

- First is the chess knowledge with the unlimited implementation,
- Second is the higher performance or the higher computing speed,
- The last is to minimize the unnecessary overhead.

## 3. LITERATURE REVIEW:

A relationship between chess skill and intelligence has long been discussed in the literature and popular culture. The notion that playing chess makes you smarter goes something like this: chess requires concentration and intelligence, and as mathematics and literacy require the same general skills, then practicing chess must also improve one's intelligence. [2]

This involves having the whole knowledge of chess including its special as well as the basic rules, which may surprise the less experienced players. People should make sure that they know each and every rule, so they don't get caught off guard.  Some of the special rules include the En passant rule, which is a special pawn capturing move in chess. Pawns can usually capture only pieces that are directly and diagonally in front of it on an adjacent file. It moves to the captured piece's square and replaces it. With en passant, though, things are a little different.

This type of capture is the only one in chess where the capturing piece doesn't land on the same square as its victim. To perform this capture, you must take your opponent's pawn as if it had moved just one square. You move your pawn diagonally to an adjacent square, one rank farther from where it had been, on the same file where the enemy's pawn is, and remove the opponent's pawn from the board. There are also a few requirements for this move to be legal! [3]

Occasionally chess games do not end with a winner, but with a draw. One of the reasons why a chess game may end in a draw is, Fifty moves. This means that, when fifty consecutive moves have been played where neither player has moved a pawn or captured a piece, the game gets draw. [4]
Keeping all these things in mind can enhance the user experience while playing and can help in winning.

4. **OBJECTIVE:**
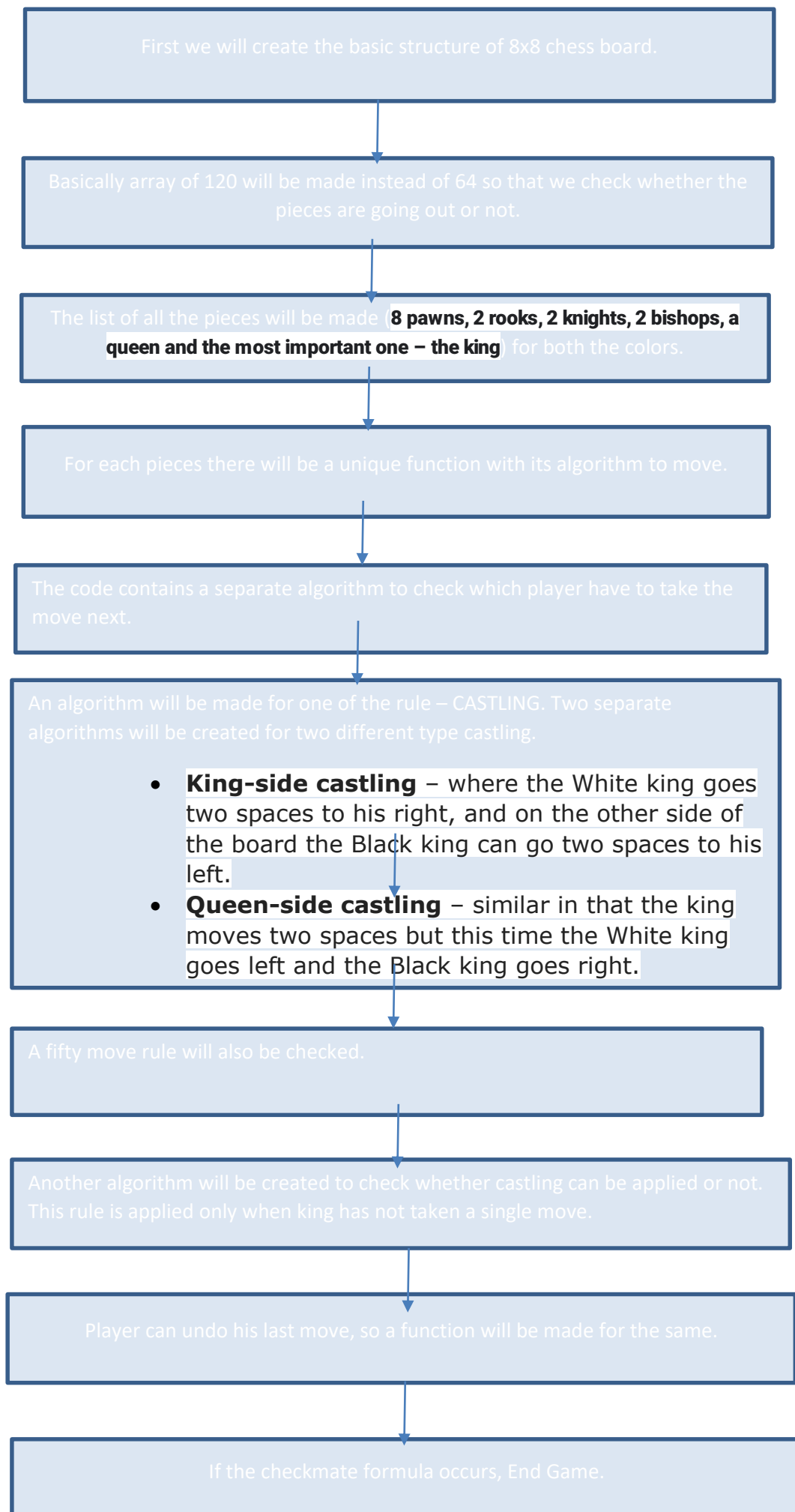
Our aim is to make a multiplayer chess engine with the better efficiency.

**Sub-objective:**

- Designing the 8 * 8 chess board.
- Setting up the pieces with their names.
- Putting algorithms for implementing the basic as well as the special rules of chess.
- Keeping the count of every move.
- Declaring the winner of the game or displaying "DRAW" if the game gets draw due to some condition.

# 5. METHODOLOGY:

First we will create the basic structure of 8x8 chess board.

Basically array of 120 will be made instead of 64 so that we check whether the pieces are going out or not.

The list of all the pieces will be made (**8 pawns, 2 rooks, 2 knights, 2 bishops, a queen and the most important one – the king**) for both the colors.

For each pieces there will be a unique function with its algorithm to move.

The code contains a separate algorithm to check which player have to take the move next.

An algorithm will be made for one of the rule – CASTLING. Two separate algorithms will be created for two different type castling.

- **King-side castling** – where the White king goes two spaces to his right, and on the other side of the board the Black king can go two spaces to his left.
- **Queen-side castling** – similar in that the king moves two spaces but this time the White king goes left and the Black king goes right.

A fifty move rule will also be checked.

Another algorithm will be created to check whether castling can be applied or not. This rule is applied only when king has not taken a single move.

Player can undo his last move, so a function will be made for the same.

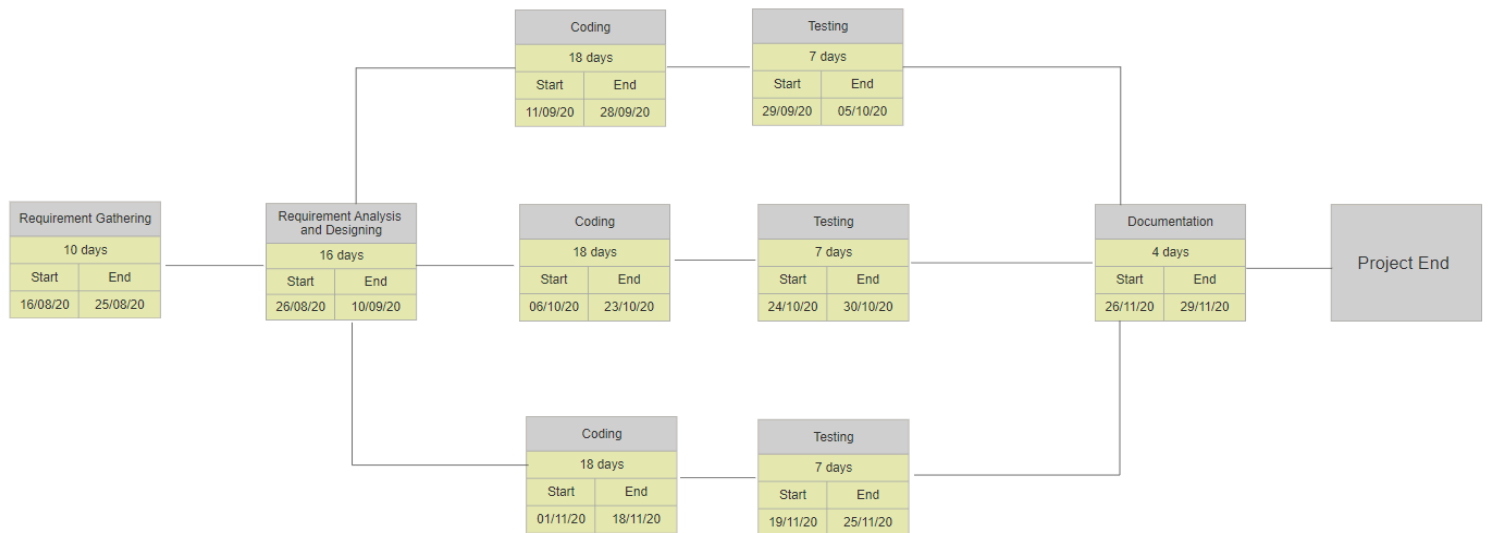If the checkmate formula occurs, End Game.

# 6. SYSTEM REQUIREMENTS:

Our program empowers two clients to play chess, which implies there is no AI engaged with our work. The game needs to make various pieces, imagine the chessboard, acknowledge different clients input controls, and know when to end the game cycle.

- Hardware Requirements
  - Computer system
  - Minimum 8GB RAM
- Software Requirements
  - Any OS i.e. Windows, Linux and macOS
  - Any text editor such as Notepad++ and Console
  - Internet Access

# 7. PERT CHART

| Coding | |
|---|---|
| 18 days | |
| Start | End |
| 11/09/20 | 28/09/20 |

| Testing | |
|---|---|
| 7 days | |
| Start | End |
| 29/09/20 | 05/10/20 |

| Requirement Gathering | |
|---|---|
| 10 days | |
| Start | End |
| 16/08/20 | 25/08/20 |

| Requirement Analysis and Designing | |
|---|---|
| 16 days | |
| Start | End |
| 26/08/20 | 10/09/20 |

| Coding | |
|---|---|
| 18 days | |
| Start | End |
| 06/10/20 | 23/10/20 |

| Testing | |
|---|---|
| 7 days | |
| Start | End |
| 24/10/20 | 30/10/20 |

| Documentation | |
|---|---|
| 4 days | |
| Start | End |
| 26/11/20 | 29/11/20 |

Project End

| Coding | |
|---|---|
| 18 days | |
| Start | End |
| 01/11/20 | 18/11/20 |

| Testing | |
|---|---|
| 7 days | |
| Start | End |
| 19/11/20 | 25/11/20 |

# REFERENCES:

[1] **Essential elements of chess available at** https://iopscience.iop.org/article/10.1088/1742-6596/1195/1/012013/pdf

[2] **Chess and intelligence available at** https://www.straitstimes.com/singapore/education/does-playing-chess-make-you-smarter-not-really

[3] **En passant rule available at** https://www.chess.com/terms/en-passant

[4] **Fifty move available at** https://www.chess.com/forum/view/general/please-explain-50-moves-rule-of-draw

**Approved By:**

(Name & Signature)                                    (Name & Signature)

**Project Guide**                                    **Head of Department**

**DR. MONIT KAPOOR**                                 **DR. MONIT KAPOOR**