

CHESS

Purpose

This document specifies all the requirements for the Chess game software. These requirements relate to the functionality, constraints, performance and the system interface.

The Chess program is a program used to play game. First goal is to allow two users or players to play the game interactively from remote locations. And the second goal will be that the program should be working and allow the users to play the game.

Scope

This document describes the software requirements for the Chess program. This document will be used by the end-users, tester, and developers of the game.

Current Problem and proposed idea

Currently for developing games, a lot of highly sophisticated engines, languages and tools are used for animation and graphics purpose.

We can implement CHESS in simple languages like python, etc with pretty much similar graphics like the existing ones and can also reduce the size of application to a greater extent.

Networking can be added to let additional features like spectators in the game, etc.

Definitions

Bishop:

One of two pieces of the same color that may be moved any number squares diagonally, as long as no other piece blocks its way. One piece always remains on White squares and the other always on Black.

Castling:

To move the king two squares horizontally and bring the appropriate rook to the square the king has passed over.

Check:

To make a move that puts the opponents King under direct attack.

Checkmate:

A situation in which an opponent's king is in check and it cannot avoid being captured. This then brings the game to a victorious result.

Chess Board:

A board you need to play Chess. Have 64 black and white square.

Chess:

A game played by 2 people on a chessboard with 16 pieces each.

En Passant:

A method by which a pawn that is moved two squares can be captured by an opponent's pawn commanding the square that was passed.

King:

The main piece of the game, checkmating this piece is the object of the game. It can move 1 space in any direction.

Knight:

This piece can move 1 space vertically and 2 spaces horizontally or 2 spaces vertically and 1 space horizontally. This piece looks like a horse. This piece can also jump over other pieces.

Pawn:

One of eight men of one color and of the lowest value usually moved one square at a time vertically and capturing diagonally.

Player/User/Client:

A user or a player will be the person that is playing the chess game

Queen:

This piece can move in any number of spaces in any direction as long as no other piece is in its way.

Rook:

One of two pieces of the same color that may be moved any number squares horizontally or vertically, as long as no other piece blocks its way.

Stalemate:

A situation in which a player's king is not in check, but that player can make no move. This then results is a stalemate, which is a [draw](#).

Product perspective

An entertainment tool to enjoy and play with friends over the network. Software games allow playing games with people that you may not know.

System interfaces

CHESS software integrates two internal systems to provide functionality

i) Client CHESS software has an interface to the user's client to receive user input and moves selections for the game.

ii) Network CHESS software has an interface to the network in order to transmit information and connect players.

User interfaces

CHESS includes an interface resembling a common chessboard. CHESS requires python3 installed, memory space and storage space on the user computer to save data.

However, CHESS is not portable and the clients will need to install one time the chess application on each computer that will be used to play.

Connect interface is used by game players and display player information.

As players make moves, the screen is updated to reflect the moves made in the game.

Hardware interfaces

CHESS runs on any computer hardware meeting the following criteria:

- Capable to use TCP connections
- Includes Memory Storage
- Includes a mouse
- Includes a keyboard

Software interfaces

CHESS software integrates some external software to provide functionality

Client: CHESS software interfaces with the user computer and expect that it has python3 environment installed.

Network: CHESS software interfaces with the user computer and expect that it is capable of use TCP connections.

Communications interfaces

Communication between the clients is facilitated by common network protocols using TCP/IP.

Memory constraints

CHESS software requires 1 gigabyte of RAM memory.

Operations

CHESS not provide backup or recovery operations.

Product functions

CHESS system will provide the following functions:

- Record of Move timers
- Records of moves made in this game so far.
- Records of pieces that each player killed.
- Records of valid moves around selected pieces.
- Players are able to send messages to each other.
- Spectators can be allowed to watch other players play.

User characteristics

The user of CHESS need experience and be able to play chess at least a basic level. Furthermore, user needs to be familiar and the comprehended chess rules.

Constraints

CHESS may experience hardware limitations constrain for graphics and python language requirements if installed in a not compatible computer.

The interface may appear different on different Operating systems in terms of graphics, dimensions, etc

Assumptions and dependencies

CHESS is not platform dependent and can be installed in any operating system capable to run python3 environment.

Requirements Apportioning

The priority levels for the requirements are:

<u>Priority 1</u>	<u>Priority 2</u>	<u>Priority 3</u>
Highest priority. All items of this level must be implemented and verified or the program is unacceptable.	Requirements of this priority are expected to be implemented, but their omission will not result in an unacceptable program; so long as their omission does not affect higher priority components.	Lowest priority and not expected to be implemented.

Priority 1:

- i. Users shall be able to connect via IP address.
- ii. Users shall be able to start a game once two users are connected and ready.
- iii. Users shall be given the choice of who plays black and white.
- iv. The player playing white is first to move.
- v. A player may forfeit at any time during gameplay.
- vi. Forfeiting shall end the game immediately.
- vii. The active player shall select a piece by clicking it.
- viii. When a piece is selected, all legal moves for that piece are highlighted.
- ix. When a piece is selected, the active player may select another piece by clicking it.
- x. A selected piece must always belong to the active player.
- xi. The active player shall move the selected piece by clicking on any legal square.
- xii. The active player shall capture a piece by moving onto a legal square containing an opposing piece.
- xiii. An undo request shall be ratified by the active player.
- xiv. When an undo request is accepted the game is reverted to the state of the board prior to the request.

Priority 2:

- i. A player must be given a confirm dialog before forfeiting.
- ii. Captured pieces shall be displayed in a captured pieces box.
- iii. The inactive player may request to undo the prior move.
- iv. There shall be no more than one undo request per turn.
- v. A player shall be able to save a log of the moves at any time.