

Queen's Chess

Ver 1.0

Developed by Queen Studios
In affiliation with UCI

Developers: Neel Sankaran , Oliver Amjadi , Jordan Queen, Sai Sandeep Reddy
B., AJ Smyth, Yi Sien Ku

Table of Contents

1. Glossary.....	3
2. Overview.....	4
3. Installation.....	5
4. Documentation of Functionality.....	6
5. Copyright and Licensing.....	7

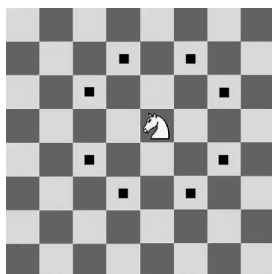
Glossary

Pawn - The pawn is a chess piece that can move forward one tile or two tiles on a player's turn or can capture a piece diagonally one spot forward towards the left or right.

Rook - A castle looking chess piece that can move forward or backward or side to side.

Bishop - A chess piece that can move diagonally

Knight - The knight is a chess piece that can move in an L shaped pattern. It can capture pieces that are two moves forward, backward or side to side and then left or right.

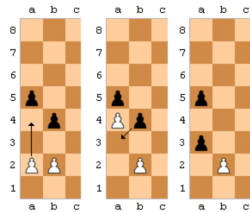


Queen - A chess piece that can move in the same way that both the rook and bishop can move

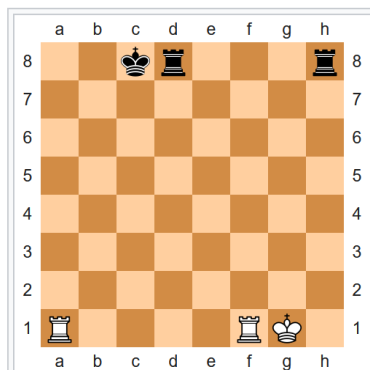
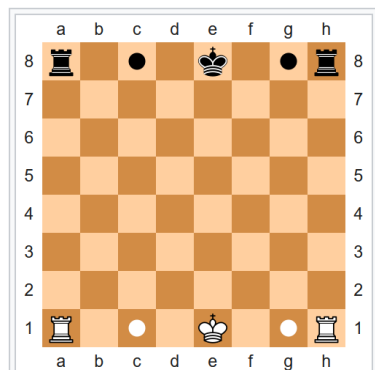
King - The main chess piece. If lost, indicates loss. The king can move left 1 position, right 1 position, up 1 position, or down 1 position.

Chess Algebraic Notation - A convention for notating the moves that occur in a game. Read sequentially it can be used to reproduce an entire match.

Capturing en Pessant - If a pawn moves forward twice, then an enemy pawn can capture it on the next term as if it was only moved one space forward.



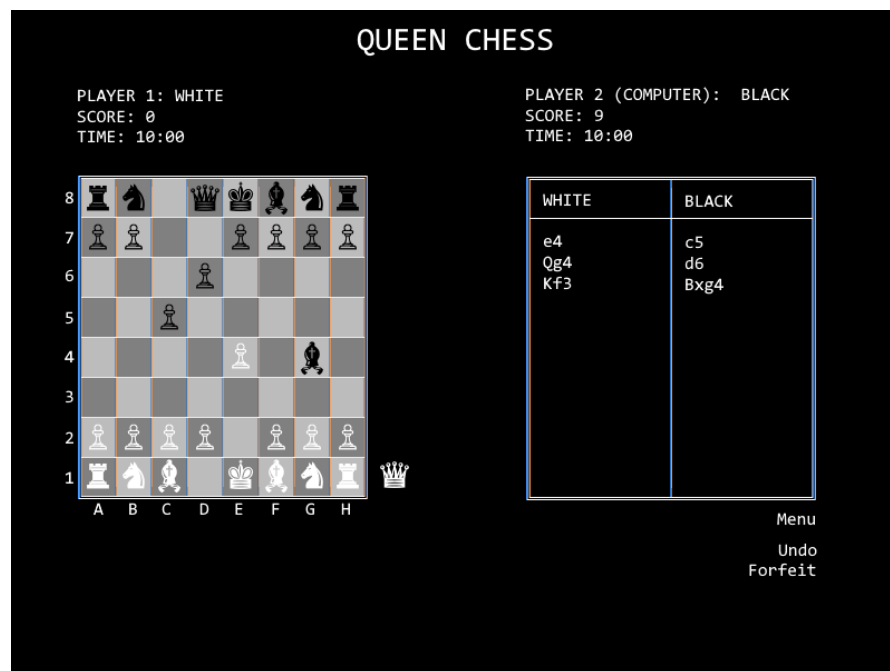
Castling - The rook can move beside the king and then the king and rook can switch pieces.



Overview

Introduction: Our app will be an AI based chess game that analyzes the moves from the human player and calculates the best move for the AI side to perform, in order to increase the chances of winning the game. Our app can be used to practice chess whenever required, without the need of a human opponent. It can also be used to compare the chess skill set of humans by using the AI to play games against different humans for a fair comparison.

GUI Demonstration: Our app will have a simple GUI written in [ncurses](#), which is a linux C library used to create in-terminal text based GUI. The following diagram shows the basic outline of our GUI.



(Made by AJ)

Main Features:

As can be seen from the diagram our main GUI features will include the following:

- Highlight valid moves(account for scenario where player is mated)
- Player score and timer
- Algebraic chess notation table for move history
- Current game board and sideboard consisting of captured pieces

Furthermore, our app will include a menu which will contain the following options:

- Set AI Difficulty and Match parameters(Game timer and starting side)
- AI vs AI mode
- Tutorial(Basic chess rule table)
- Exit program

Installation

System Requirements:

System Characteristic	Recommended specification
Processor	64-bit Opteron, EM64T
RAM	1 GB or Higher
Swap space	1 GB or Higher
Disk space	500 MB of Free Space
Operating System	Linux Kernel Stable Release 5.0+ (Latest: 5.17.1)

Installation:

Download the binary file, open a terminal window and navigate to the directory to which the binary was downloaded via “cd.” Configure the permissions to allow the binary to run with “sudo chmod +x queenChess.bin”. Now run the executable via ./queenChess.bin, taking care that you are still in the correct directory.

Uninstallation:

Simply delete the binary. To do this, open a terminal window, navigate to the directory containing the program via “cd” and remove the program with “sudo rm queenChess.bin”.

Documentation of functionality

Menu:

- Game Settings:
 - Match time limit: Players can choose the time limit to make each move.
 - Starting colors: Players can choose white, black, or random color
 - AI difficulty: Players can choose from Beginner, Intermediate, and Expert.
- Start match against AI
 - This function allows the user to start the match against the AI.
- Tutorial Page
 - We will have a mini tutorial page. This prints the rules of the game to the console so that the user can understand how the game works.

User Input/Output:



- User clicks on pieces on the board and clicks on the space they want the piece to move to and the piece will. Only valid spaces can be moved to. We will have a queue that contains all the pieces that were captured by each player. Each time something is captured it gets removed from the board and added to the queue. Furthermore, castling and capturing en passant will also be options for the user.

Copyright and Licensing

Copyright 2022 Queen

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Contact Information

info@queen.com