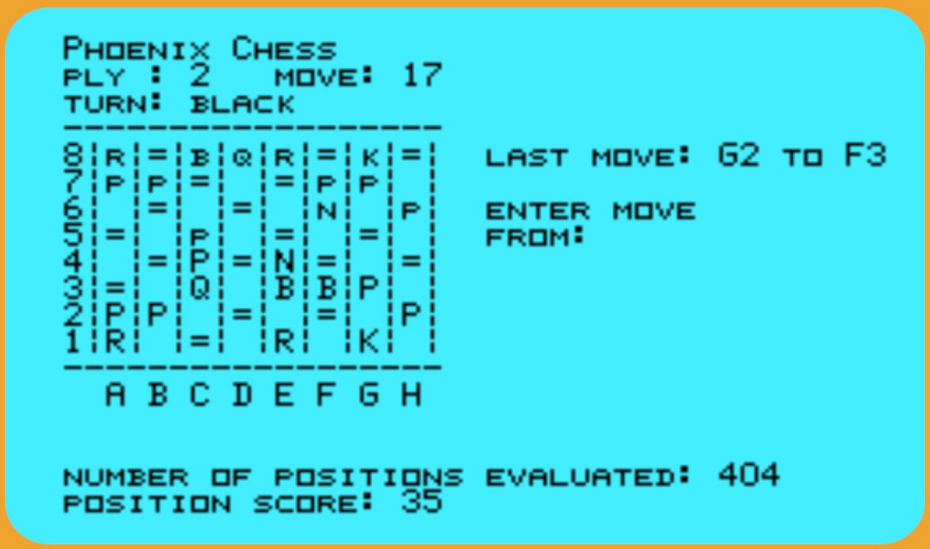
Texas Instruments Home Computer

Phoenix Chess

UCSD PASCAL DISK SOFTWARE

A full featured chess engine for both beginner and advanced players



Programmed by Walid Maalouli

2026

Quick Reference Guide

System Requirements

TI 99/4A Computer
Peripheral Expansion Box
SAMS card of any size
Floppy Disk Controller
Pcode Card
Single Floppy Disk Drive
PIO printer optional for printing the game record

Game Startup

1. Place the game disk in the root disk drive
2. Turn the Pcode card switch to ON
3. Turn the computer display ON
4. Power ON the Peripheral Expansion Box
5. Switch the TI 99/4A computer ON
6. The Pcode system will boot and the game will automatically load and run.

Game Keys

[LETTER|NUMBER]	Piece board coordinates
[R]	Redo move entry
[U]	Utility menu

TEXAS INSTRUMENTS **HOME COMPUTER**

PHOENIX CHESS



Have you outgrown Video Chess and are looking for the next challenge? Look no further: Phoenix Chess goes to the next level, offering you a worthy opponent regardless of your skill level. Are you up to the challenge?

Programming:	Walid Maalouli
Testing and debugging:	Thorsten Czub, Michael Thomas and Arne Bailliere

Copyright: This game is released under the GNU General Public License v3.0

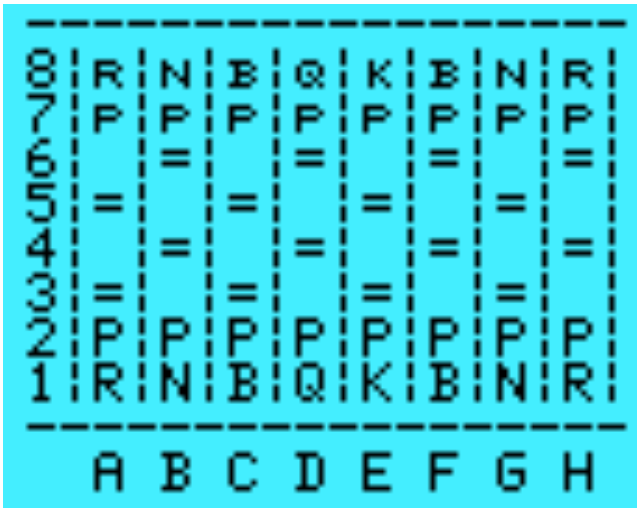
INTRODUCTION

Phoenix Chess is an attempt at writing a chess program on the TI 99/4A under the UCSD Pascal environment, with the goal of consistently beating the Video Chess program. It's one of those bucket list projects initially conceived 11 years ago and nearly abandoned until revived in late 2024, a personal learning journey littered with frustrations as well as triumphs and in the end I think successful.

CHESS ENGINE HIGHLIGHTS

- Phoenix chess follows all the official chess rules
  - castling
  - pawn promotion to any piece
  - en-passant capture
  - 3-fold repetition draw
  - it will also resign the game under certain circumstances
- Variable difficulty levels (ply) from 1-6
- Quiescence searching where the game tree search depth is selectively increased when piece captures are possible
- Opening book covering the major opening lines up to 4 ply deep
- Board representation with 64-bit bitboards, a unique feature among chess engines for the early microcomputers

GAME REPRESENTATION



The chess board consists of 64 squares with rows labeled from 1 - 8 and columns labeled from A – H. The squares alternate between light and dark squares, with the latter being represented by a **= symbol** inside the square.

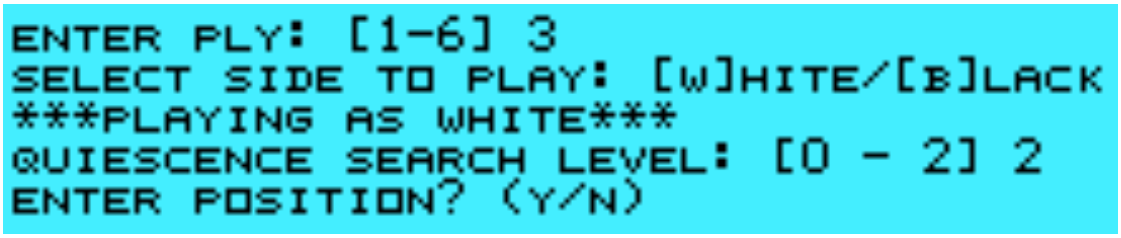
The chess pieces are represented as below:

- K or k : king
- Q or q : queen
- R or r : rook
- N or n : knight
- B or b : bishop
- P or p : pawn

with upper case letters indicating the white pieces and lower case letters the black ones.

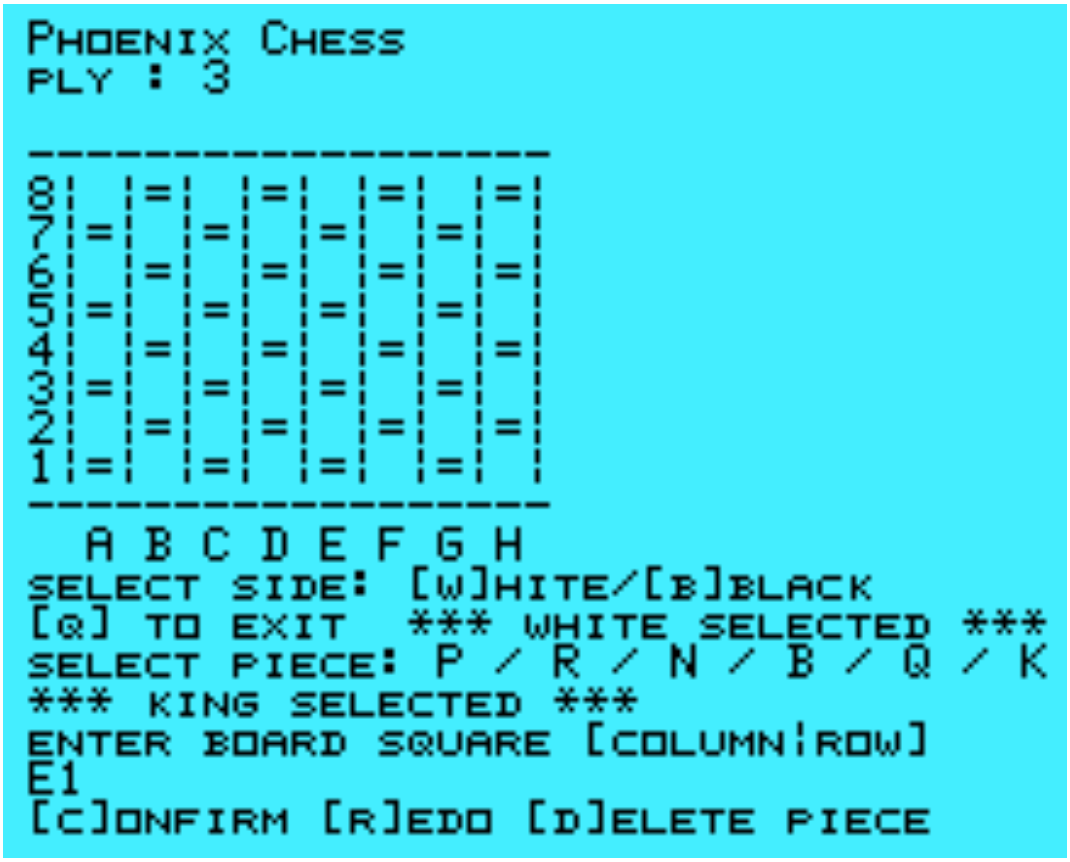
HOW TO PLAY

When the game launches, you will be presented with a splash screen as the program initializes and loads its databases. It will look for the presence of a SAMS card, *and if it is not found then an error message will be displayed and the program will terminate,* otherwise the number of available SAMS pages will be displayed. Once initialization is complete, the main program will launch and you will be asked to pick a set of options:



- ply level: enter a number from 1 to 6. The higher the level, the stronger the engine, but also thinking time increases exponentially. Each ply is a half-move.
- side to play (white or black). The white pieces are always on the bottom of the board and start first.
- quiescence search level from 0 to 2. This indicates how many *additional plys* the game engine will search when captures are possible on the board. While this option does improve play quality, it also lengthens thinking time accordingly.
- option to set up the board. Allows you to create custom positions for analysis or mate search.

POSITION SET UP

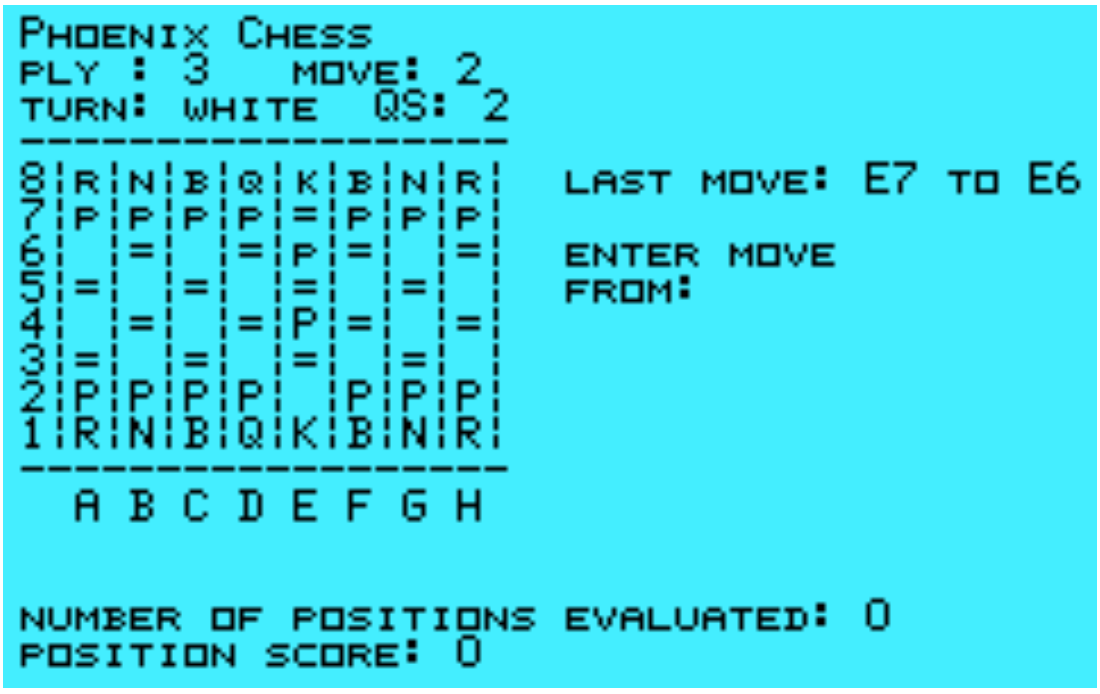


If you choose to set up a position, you will be presented with an empty chessboard and prompted to place pieces on the board. First select the color by pressing either **W** for white or **B** for black. Next select the type of piece by pressing the appropriate letter (**K, Q, R, N, B, P**). Finally enter the destination square coordinates (column|row: ex. D5). You will then be asked to either **C**onfirm, **R**edo, or **D**elete. **Confirm** places the piece on the board, **Redo** will let you re-enter the destination square, and **Delete** will delete the type of piece indicated on the selected square.

Warning: the computer does not check the legality of the pieces placed on the board, so illegal positions will be accepted but will likely result in strange behavior from the chess engine.

One you have entered all the pieces, press **Q** to end the set up and you will be prompted to indicate if you want to allow castling for white and black, which side will be playing first, and the starting turn number. From there, the game will proceed as usual.

PLAYING THE GAME



The user interface consists of 4 sections:

1. Three lines of text at the top showing the game title, ply level, move number, color of the current side playing and the quiescence search level (***QS***).
2. The board representation in the middle reflecting the current state of the game.
3. The last move played and either a ***thinking…*** indicator if the engine is searching for a move, or a prompt to enter a move if it's the human player's turn are displayed on the right.
4. The number of moves evaluated by the engine during the last move and the current game score are shown on the bottom of the screen.

To enter a move, type the coordinates of the starting square in the format *column*| *row* such as **D5** when the ***from*** prompt is displayed. *No need to press the* ***Enter*** *key.* If the selected square contains a valid piece, then the ***to*** prompt will appear asking for a destination square, otherwise you will continue to be prompted for a valid starting square. Enter the destination square as before, and it will only be accepted if it is a valid move. Should you want to re-enter the whole starting-ending square sequence, then press ***R***. Once the move is accepted, the board display will be updated to reflect it and the computer will immediately start searching for its best move, shown by the ***thinking…*** indicator. Your entered move will be shown after the ***last move*** indicator. Once the engine finds a move, the board display will update accordingly

along with the number of positions evaluated and the current game score. *When still in the opening library, the number of positions evaluated and score will remain 0.*

The computer will not accept an invalid move such as:

- moving an opponent's piece
- illegal destination square
- moving a piece which does not take your king out of check if it is currently checked
- castling attempt when the king has already moved, one or more of the interposing squares are under attack or blocked, or if the rook on the side of the castle has already moved

To castle, simply move the king to the new destination square and the rook will be automatically moved accordingly.

The ***check!***, ***checkmate!***, ***stalemate!***, ***resign!***, or ***draw by 3-fold repetition*** indicators will appear above the last move display depending on the game status. Except for a check situation, the game will immediately end at that point and you will automatically enter the utility menu.

UTILITY MENU



When it is the human player's turn to move and the ***from*** prompt is displayed, pressing **U** will display the utility menu on the right side of the screen. Press the appropriate number or letter next to the option to select it.

- **Load game** : load a previously saved game from disk. Enter the name of the save file including the drive prefix if the file is not on the root disk and it will load the game and resume where it last left off. Ex: **#5:MEPHISTO.SAVE**
- **Save game** : save the current game to disk. Enter the name of the save file including the drive prefix if the file is not on the root disk and the game will be stored to that file name. Ex: **#5:MEPHISTO.SAVE**

- **Backup** : backup 1 *full move*. A warning beep will let you know when you reach the first move.
- **Forward** : advance the game 1 *full move*. A warning beep will let you know when you reach the last move.
- **First move** : jump to the first move of the game.
- **Last move** : jump to the last move of the game.
- **Switch sides** : the computer will play your move immediately. Used repeatedly, this function allows you to have the computer play itself.
- **Change ply** : change the game ply between 1 and 6.
- **Play** : resume the game at the *currently displayed position*. Any positions beyond that will be lost.
- **End game** : quit the game. The game record will be lost if not saved to file first. You will be prompted to confirm.
- **Print game** : the game record will be printed the the PRINTER: device. You will be prompted for the human player's name and the date.

The opening book will no longer be available if the game is modified.

TECHNICAL COMMENTS

- Phoenix Chess is the only chess engine for the early microcomputers that makes use of 64-bit bitboards to represent the chessboard, thanks to the use of the fast extra storage provided by the SAMS card. This allowed for much more efficient board manipulations than otherwise would have been possible. The drawback however was that each bitboard operation requires 4 successive sub-operations given that the TI can only process 16-bit words, and therefore the engine speed was negatively affected.

- There is likely a memory leak in the OS not generally apparent except when memory becomes very tight as in the case of Phoenix Chess at the higher ply levels since it makes heavy use of recursion. *This will eventually lead to a stack overflow crash* at some point in the game dependent on the ply level. *This is not an issue with the engine, but rather a bug in the pcode system itself*.

  I strongly recommend keeping a record of the game on a side chess board or saving frequently. If the game crashes, restart Phoenix Chess, use the position entry facility to reset the position , or load the saved game, and continue from there.

- Thinking times are very long, ranging from **2 minutes** for ply 1, **30 minutes** for ply 2, **4 hours** for ply 3 and **days** for subsequent plys. *Adding quiescence searching levels also considerably lengthens the thinking times and is best limited to plys less than 4*. For higher plys, accelerated emulation is recommended for practical play.

- Phoenix Chess was programmed entirely within the native UCSD Pascal environment on the TI 99/4A, albeit under emulation for disk reliability and speed of compilation, using the Editor, Compiler, Linker, Assembler and various UCSD Pascal integrated utilities. It spans over 4500 lines of code and yet still fits within the memory confines of the TI at runtime thanks to the use of UNITS which are swapped in and out of memory in real time as needed, highlighting the impressive modularity of the UCSD Pascal system.

- The estimated ELO level of the engine is around 1300 based on games played against many different legacy chess computers and software engines as tested by Thorsten Czub. This is very respectable and should provide a decent challenge to most beginner and intermediate chess players.

NOTES

Texas Instruments Home Computer



PHOENIX CHESS - 2026