# HUG-CHESS

HUGH ANDERSON (May 2023)

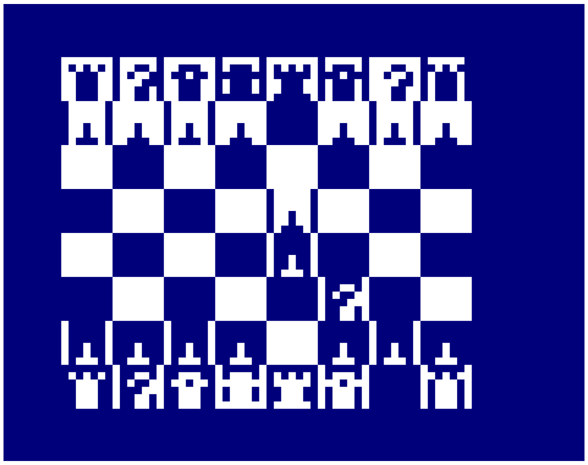
## HUG-CHESS

Hug-Chess is a tiny chess game that ran on a HUG1802/ETI-660 computer wih 3K RAM. Its core "move" algorithms were based on Peter Jennings "MicroChess", which was 6502 code that ran on the KIM1 microcomputer. In 1977 I re-implemented the 6502 chess-move algorithms in 1802 machine code, and later that year developed a graphical display on the HUG1802[1] in CHIP8 and 1802 machine code[2].

It was sold for $10 or so back in the late 1970s in NZ and Australia, to both HUG1802 and ETI-660 people, on a cassette tape. I also sold a tiny basic, and a "games" tape. When a friend recently prompted me to see if I could get my leftover HUG1802s working, I thought most of this software was lost to the depths of time.

However, while looking for HUG1802 related items, I found a mouldy CHESS tape from 1978, with multiple copies of the binary. When I read the tape, each copy was different, but similar enough to identify that they were all copies of the code. Through a process of diff, disassembly and so on[3], I have resurrected what may be a nearly clean working copy of the game. It is slow, plays a terrible game of chess, and will probably make mistakes...

When you load and run it, it looks like this:



[1] Note that the HUG1802 was re-engineered by ETI as the ETI-660 in 1980. The circuitry is very similar to the HUG1802, and the monitor rom/memory map is identical, so any code that runs on the HUG1802 runs on the ETI-660.

[2] Note also that this Basic is no relation to Tom Pittman's tiny basic - it shares no code, no history, and has different operations.

[3] Tape converted to MP3 using a cassette-to-mp3 converter; viewed using Audacity to convert to mono and deduce the tape format; wrote a python scanner to pull data from the MP3; disassembly using dasmx.exe on my M1 mac (Win10/Parallels converts the X86 code on-the-fly); and then trying to imagine what the 1802 code should do, and what was what; testing under emulation on EMMA-02/ETI-660.

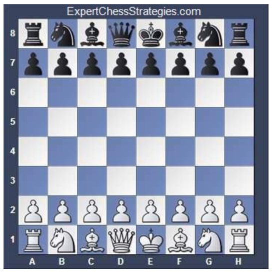
## GENERAL NOTES

The binary is to be loaded at **#0600** , and then run as a CHIP8 program using `reset` '8'.

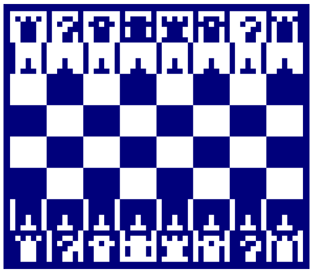
To play the chess game, the following commands/keycodes are relevant:

1. Key 'C': Clear the board back to the starting position. White is at the bottom, black facing at the top of the screen.
2. Key 'D': Do a move for the player at the bottom of the screen (initially white).
3. Key 'E': Swap the board around. Note that if you Do-Swap-Do-Swap - you can get the machine to play itself.

To make your own piece move, you key the *from* and *to* coordinates followed by Key 'F', according to the following layout:



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 7,7 | 7,6 | 7,5 | 7,4 | 7,3 | 7,2 | 7,1 | 7,0 |
| 6,7 | 6,6 | 6,5 | 6,4 | 6,3 | 6,2 | 6,1 | 6,0 |
| 5,7 | 5,6 | 5,5 | 5,4 | 5,3 | 5,2 | 5,1 | 5,0 |
| 4,7 | 4,6 | 4,5 | 4,4 | 4,3 | 4,2 | 4,1 | 4,0 |
| 3,7 | 3,6 | 3,5 | 3,4 | 3,3 | 3,2 | 3,1 | 3,0 |
| 2,7 | 2,6 | 2,5 | 2,4 | 2,3 | 2,2 | 2,1 | 2,0 |
| 1,7 | 1,6 | 1,5 | 1,4 | 1,3 | 1,2 | 1,1 | 1,0 |
| 0,7 | 0,6 | 0,5 | 0,4 | 0,3 | 0,2 | 0,1 | 0,0 |



For example:

| KeyPress | Move | Comment |
|---|---|---|
| C | | Clear the board |
| D | P-K4 | Get program to make a move for white |
| 6343F | P-K4 | A traditional response |
| D | N-KB3 | Program moves quickly... |
| 7655F | N-QB3 | This is the Giuoco Piano opening |
| D | B-B4 | And the computer has this encoded as an opening |
| 6444F | P-Q4 | Now we deviate from the traditional |
| D | P-K5 | And the response takes a while.... |
| ... | ... | ... |

Anyway - this is all I have. If I ever find more documentation...