Home * Engines * **Crafty**

Crafty Owl [8]

Crafty, [1]
a portable open source engine supporting the Chess Engine Communication Protocol written by Robert Hyatt in ANSI C starting in the early 90s, loosely derived from Cray Blitz, winner of the 1983 and 1986 World Computer Chess Championships [2]. Crafty pioneered in using Rotated bitboards , parallel search and probing Nalimov Tablebases. It performs a principal variation search, null move pruning, LMR as well as a SEE swap algorithm for move ordering and to prune "bad" captures in quiescence search. In 2006/2007, Crafty switched from rotated to Magic bitboards [3], according to Robert Hyatt because it was not faster but simpler [4]. Crafty **25.1**, released in October 2016, not only includes an increase in playing strength [5] but support for Syzygy bases by Ronald de Man aided by the coding contributions of Basil Falcinelli [6]. Crafty **25.3** features playing strength adjustment between 800 and 2600 Elo [7].

Table of Contents

Team Members

as mentioned at WCRCC 2010 [9] and CCT15, 2013 [10]

- Peter Berger, Crafty's book author at WCCC 2004, WCCC 2005 and WCCC 2006
- Michael Byrne
- Robert Hyatt
- Tracy Riegle
- Peter Skinner

Tournaments

Crafty participated at four World Microcomputer Chess Championships, the WMCCC 1996, WMCCC 1997, WMCCC 2000, and WMCCC 2001, three World Computer Chess Championships, the WCCC 2004, WCCC 2005 and WCCC 2006, the ACCA Americas' Computer Chess Championships, the ACCA World Computer Rapid Chess Championships, CCT and various other tournaments. Crafty won the CCT1 in 2000, CCT5 in 2003 and CCT6 in 2004. At the Fifth Annual ACCA Americas' Computer Chess Championships in 2010 Crafty was runner-up behind Thinker [11] .

Descriptions

1997

from the ICGA page [12] :

```
Crafty is a "bitmapper" using 64 bit words to represent the chess
board, along the lines of the famous Chess 4.x program from
Northwestern University. It uses a traditional alpha/beta search with
the PVS (null-window) enhancement, along with null-moves (R=2) and
lots of search extensions including "fractional ply extensions" to
drive the search deeper along interesting lines. It has a very simple
quiescence search that only considers capture moves and is fairly
selective about which captures are included. It does a full endpoint
evaluation, with no root pre-processing nor incrementally updated
scoring terms. It is currently about 37,000 lines of ANSI C with about
3,000 lines of that being evaluation.

Since Crafty uses bitmaps, much of the evaluation is significantly
shorter than it would be in a more traditional (array-based) board
representation, so that this 3,000 lines of code is somewhat
misleading (for example, to ask "can this pawn run and promote before
the opposing king can get there?" only takes one line of code in
```

Crafty. It is still very fast, searching around 100,000 moves per second. At 60 seconds per move, it solves 297/300 of the Win At Chess tactical positions. It has a large opening database composed from 250,000 GM games, and uses 3-4-5 piece endgame databases during the search (not just at the root of the tree.) It has played over 100,000 games during the past two years, playing on various chess servers around the world, and has maintained ratings on these servers that are always near the very top.

2008

from a CCC post [13] :
Crafty's basic search is pretty simple. Just PVS + null-move + LMR + check extension (all others have been removed after testing showed they hurt performance rather than helped), + simple q-search checks (I am playing with this as I write this however and it might change) + simple q-search. I've used killers and hashing since middle 70's so those are not new. Bitboards were around in the middle 70's so that's not new. I don't think there is anything remarkable in my evaluation, certainly nothing we were not doing 15 years ago or longer, other than tuning adjustments.

As far as tuning goes on LMR and null-move, I have not gone very far afield there. I use R=3 everywhere. LMR is a 1-ply reduction although I have plans to try a variable reduction so that for moves that look really ugly I reduce them even further (white playing Na1 for example, with king on the other side of the board, no passed pawns, etc...)

I have even run without the check extension, the only one that is left. Many think the purpose of this extension is to find deep mates. That's wrong. The purpose is to try to expose horizon-effect moves and avoid them when possible. I also want to test a restricted check extension, where it is only applied in the last N plies where the horizon effect is most notable, where right now I apply them everywhere with no limit, always +1 ply added to depth.

Generation II

Crafty **25.0**, Generation II, December 2015 [14]

- This version contains a major rewrite of the parallel search code, now referred to as **Generation II**. It has a more lightweight split algorithm, that costs the parent MUCH less effort to split the search. The key is that now the individual "helper" threads

do all the work, allocating a split block, copying the data from the parent, etc., rather than the parent doing it all. Gen II based on the DTS "late-join" idea so that a thread will try to join existing split points before it goes to the idle wait loop waiting for some other thread to split with it. In fact, this is now the basis for the new split method where the parent simply creates a split point by allocating a split block for itself, and then continuing the search, after the parent split block is marked as "joinable". Now any idle threads just "jump in" without the parent doing anything else, which means that currently idle threads will "join" this split block if they are in the "wait-for-work" spin loop, and threads that become idle also join in exactly the same way. This is MUCH more efficient, and also significantly reduces the number of split blocks needed during the search. We also now pre-split the search when we are reasonably close to the root of the tree, which is called a "gratuitous split. This leaves joinable split points laying around so that whenever a thread becomes idle, it can join in at these pre-existing split points immediately. We now use a much more conservative approach when dealing with fail highs at the root. Since LMR and such have introduced greater levels of search instability, we no longer trust a fail-high at the root if it fails low on the research. We maintain the last score returned for every root move, along with the PV . Either an exact score or the bound score that was returned. At the end of the iteration, we sort the root move list using the backed-up score as the sort key, and we play the move with the best score. This solves a particularly ugly issue where we get a score for the first move, then another move fails high, but then fails low and the re-search produces a score that is actually WORSE than the original best move. We still see that, but we always play the best move now. One other efficiency trick is that when the above happens, the search would tend to be less efficient since the best score for that fail-high/fail-low move is actually worse than the best move/score found so far. If this happens, the score is restored to the original best move score (in Search()) so that we continue searching with a good lower bound , not having to deal with moves that would fail high with this worse value, but not with the original best move's value.

- We also added a new method to automatically tune the new SMP parameters. The command is autotune and "help autotune" will explain how to run it.
- In addition , we did a complete re-factor of pawn evaluation code. There were too many overlapping terms that made tuning difficult. Now a pawn is classified as one specific class, there is no overlap between classes, which simplifies the code

significantly. The code is now easier to understand and modify. In addition, the passed pawn evaluation was rewritten and consolidates all the passed pawn evaluation in one place. The evaluation used to add a bonus for rooks behind passed pawns in rook scoring, blockading somewhere else, etc. All of this was moved to the passed pawn code to make it easier to understand and modify.

- Added a limited version of late move pruning (LMP) for the last two plies. Once a set number of moves have been searched with no fail high, non-interesting moves are simply skipped in a way similar to futility pruning.
- We had a minor change to history counters that now rely on a "saturating counter" idea. I wanted to avoid the aging idea, and it seemed to not be so clear that preferring history moves by the depth at which they were good was the way to go. I returned to a history counter idea I tested around 2005 but discarded, namely using a saturating counter. The idea is that a center value (at present 1024) represents a zero score. Decrementing it makes it worse, incrementing it makes it better. But to make it saturate at either end, I only reduce the counter by a fraction of its distance from the saturation point so that once it gets to either extreme value, it will not be modified further avoiding wrap-around. This basic idea was originally reported by Mark Winands in 2005. It seems to provide better results (slightly) on very deep searches. One impetus for this was an intent to fold this into a move so that I could sort the moves rather than doing the selection approach I currently use. However, this had a bad effect on testing, since history information is dynamic and is constantly changing, between two moves at the same ply in fact. The sort fixed the history counters to the value at the start of that ply. This was discarded after testing, but the history counter based on the saturating counter idea seemed to be OK and was left in even though it produced minimal Elo gain during testing.
- We change to the way moves are counted, to add a little more consistency to LMR. Now Next*() returns an order number that starts with 1 and monotonically increases, this order number is used for LMR and such decisions that vary things based on how far down the move list something occurs. Root move counting was particularly problematic with parallel searching, now things are at least "more consistent". The only negative impact is that now the move counter gets incremented even for illegal moves, but testing showed this was a no-change change with one thread, and the consistency with multiple threads made it useful.
- Added the "counter-move" heuristic for move ordering (Jos Uiterwijk, IJICCA) which simply remembers a fail high move and

the move at the previous ply. If the hash, captures or killer moves don't result in a fail high, this move is tried next. No significant cost, seems to reduce tree size noticeably. Added a follow-up idea based on the same idea, except we pair a move that fails high with the move two plies back, introducing a sort of "connection" between them. This is a sort of "plan" idea where the first move of the pair enables the second move to fail high. The benefit is that this introduces yet another pair of good moves that get ordered before history moves, and is therefore not subject to reduction. I have been unable to come up with a reference for this idea, but I believe I first saw it somewhere around the time Fruit showed up, I am thinking perhaps in the JICCA/JICGA. Any reference would be appreciated.

- A minor change to the way the PV and fail-hi/fail-low moves are displayed when pondering.
- Crafty now adds the ponder move to the front of the PV enclosed in parentheses so that it is always visible in console mode. The depths are reaching such extreme numbers the ponder move scrolls off the top of the screen when running in console mode or when "tail -f" is used to watch the log file while a game is in progress. This is a bit trickier than you might think since Crafty displays the game move numbers in the PV.
- The penalty for pawns on same color as bishop now only applies when there is one bishop.

Leading or Trailing Zeros

Crafty had always mapped square-index 0 to square 'a1', 7 to 'h1', and 63 to 'h8' respectively, but recently (crafty-20.6) reversed the bit-index versus square-index mapping from leading to trailing zero count based little-endian rank-file (LERF), using bitscan forward (LSB) to retrieve squares in **a1-h8** order, as most often used in CPW bitboard samples.

LSB

Trailing zero count aka bitscan forward for non empty sets as used in bitboard serialization for move generation and evaluation purposes, is implemented with x86-64 bsf instruction via intrinsic or inline assembly if available (there are also 32-bit x86 bsf versions), and a conditional 16-bit byte lookup approach otherwise - Windows 64, Linux 64 and lookup versions with preprocessor instructions for conditional compiles omitted [15]:

```
int LSB(BITBOARD arg1) {
```

```
  unsigned long index;
  if (_BitScanForward64(&index, arg1))
    return index;
  else
    return 64;
}

int static __inline__ LSB(long word)
{
  long dummy, dummy2;

asm("          bsfq    %1, %0     " "\n\t"
    "          jnz     1f         " "\n\t"
    "          movq    $64, %0    " "\n\t"
    "1:                           " "\n\t"
:   "=&r"(dummy), "=&r" (dummy2)
:   "1"((long) (word))
:   "cc");
  return (dummy);
}

unsigned char lsb[65536];

int LSB(BITBOARD arg1) {
  if ( arg1        & 65535) return (lsb[ arg1        & 65535]);
  if ((arg1 >> 16) & 65535) return (lsb[(arg1 >> 16) & 65535] + 16);
  if ((arg1 >> 32) & 65535) return (lsb[(arg1 >> 32) & 65535] + 32);
  return (lsb[arg1 >> 48] + 48);
}
```

Last One

Earlier Crafty versions prior to 20.6 had a leading zero count compliant, big-endian rank-file mapping. Left-bottom square (from White's point of view) 'a1' with square-index 0 was mapped to the leftmost, arithmetical most significant bit of an unsigned 64-bit integer with bit-index 63, while square 'h8' with square-index 63, was mapped to the rightmost, arithmetical least significant bit with bit-index 0. Bitscan forward and found index reversal was used in LastOne, to retrieve squares in **h8-a1** order [16]:

```
int LastOne(BITBOARD arg1)
{
  unsigned long index;
  if (_BitScanForward64(&index, arg1))
    return 63 - index;
```

```
  else
    return 64;
}
```

This one was found in 15.17 with MAC OS support [17]

```
int LastOne(register BITBOARD a)
{
  register unsigned long i;
  if (i = a & 0xffffffff)
    return(__cntlzw(i ^ (i - 1)) + 32);
  if (i = a >> 32)
    return(__cntlzw(i ^ (i - 1)));
  return(64);
}
```

Selected Games

WCCC 2004

WCCC 2004, round 9, Falcon - Crafty [18]

```
[Event "WCCC 2004"]
[Site "Ramat Gan, Israel"]
[Date "2004.07.11"]
[Round "9"]
[White "Falcon"]
[Black "Crafty"]
[Result "0-1"]

1.e4 e5 2.Nf3 Nc6 3.Bb5 Nf6 4.Nc3 Bd6 5.O-O O-
O 6.d3 h6 7.Be3 a6 8.Ba4 b5
9.Bb3 Na5 10.d4 exd4 11.Bxd4 Be7 12.e5 Ne8 13.Nd5 Nxb3 14.axb3 c5 15.B
e3
Bb7 16.b4 Bxd5 17.Qxd5 Nc7 18.Qb7 Rb8 19.Qe4 cxb4 20.Rfd1 Qc8 21.Nd4 R
e8
22.Nf5 Bf8 23.Qg4 Kh7 24.Bxh6 gxh6 25.Qh5 Kg8 26.Rd3 Re6 27.Rg3+ Rg6 2
8.Rxg6+
fxg6 29.Qxg6+ Kh8 30.Nxh6 Bxh6 31.Qxh6+ Kg8 32.Qg6+ Kf8 33.Rd1 Ne6 34.
Rd3
Ke7 35.Qf6+ Ke8 36.Rg3 Qc5 37.Rg8+ Nf8 38.Rg7 Rb6 39.Qf7+ Kd8  40.Rg8
Kc7
41.Qxf8 Qxf8 42.Rxf8 a5 43.Ra8 a4 44.g3 Rb8 45.Ra7+ Rb7 46.Ra5 Kb6 47.
```

```
Ra8
Ra7 48.Rb8+ Kc6 49.Rc8+ Kd5 50.e6 dxe6 51.Rd8+ Kc5 52.Rc8+ Kb6 53.Re8
a3
54.Rxe6+ Kc5 55.bxa3 0-1
```

WCCC 2006

WCCC 2006, round 8, Diep - Crafty [19]

```
[Event "WCCC 2006"]
[Site "Turin, Italy"]
[Date "2006.05.30"]
[Round "8"]
[White "Diep"]
[Black "Crafty"]
[Result "0-1"]

1.e4 e5 2.Nf3 d6 3.d4 exd4 4.Nxd4 Nf6 5.Nc3 Be7 6.Bf4 O-O 7.Qd2 c6 8.O-
O-O b5
9.f3 b4 10.Nce2 c5 11.Nb3 Nc6 12.Bxd6 c4 13.Nc5 Qa5 14.Bxe7 Nxe7 15.Qd
6 Nf5
16.exf5 Qxa2 17.g4 a5 18.Rd4 Re8 19.Kd1 Qxb2 20.Nc1 b3 21.cxb3 cxb3 22
.Bd3 a4
23.Re1 Rxe1+ 24.Kxe1 Qxc1+ 25.Ke2 Bb7 26.Qd8+ Ne8 27.Qe7 Bc6 28.Bc4 Nf
6 29.Bxf7+
Kh8 30.Nd3 b2 31.Rd8+ Rxd8 32.Qxd8+ Be8 33.Bxe8 Qc2+ 34.Ke3 Nd5+ 35.Kd
4 Qc3+
36.Ke4 Qc7 37.Qxc7 Nxc7 38.Nxb2 a3 39.Bf7 axb2 40.Ba2 Nb5 41.Kd3 Na3 4
2.g5 b1=B+
43.Bxb1 Nxb1 44.h4 Kg8 45.h5 Kf8 46.f4 Ke7 47.Kd4 Nd2 48.Ke3 Nc4+ 49.K
d4 Nd6
50.Ke5 Nf7+ 51.Kd5 h6 52.f6+ gxf6 53.g6 Nd8 54.f5 Nb7 55.Kc6 Na5+ 0-1
```

Copyright

Many programmers did not grasp Crafty's Copyright statement, but apparently took remarks by Robert Hyatt like in his reply to Ren Wu, January 26, 1999, concerning code reuse and not reinventing the wheel as alibi for their chess programming [20] :

```
This is a basic tenet of software engineering called 'code reuse'. Why
should I pay you to write something from scratch and take a year, if
you can take something that exists and modify it to do the same thing
in a month? And then I don't have as much trouble debugging and
testing, since it is mostly already done...
```

that's not a bad side to this... Of course occasionally starting over is a good thing. But not starting from 'scratch'. IE if you don't know what has already been tried, you will re-invent the same bad wheels over and over and probably follow the same footsteps many before you did... software engineering wants to avoid that 'reinvention' problem...

Robert Hyatt further on the copyright problem [21] :
My primary requirement is that if something is done to crafty to make it 'better', then that 'something' must be as public as the original code was. Because many have contributed bits and pieces... Eugene, George, Steffen, Mark, SJE, and many others that are to numerous to mention. Seems unfair that they modify what I did, then they make their stuff public, and then someone else takes _all_ of this and purports it to be 'original'.

I suppose it has to do with 'national morals' or whatever, ie the software piracy problem in China, to name but one.

Crafty Clones

1. Bionic Impakt
2. Brause
3. Chinito
4. Fafis
5. Gunda-1
6. Kaissa (BY)
7. LaGrande
8. LaPetite
9. Patriot
10. Voyager

See also

- Cray Blitz
- Nalimov Tablebases
- Open Source Engines

Publications

- Robert Hyatt, Monroe Newborn (**1997**). *CRAFTY Goes Deep*. ICCA Journal, Vol. 20, No. 2
- Ernst A. Heinz (**2001**). *Modeling the "Go Deep" Behaviour of CRAFTY and DARK THOUGHT*. Advances in Computer Games 9 » Depth, DarkThought
- Levente Kocsis, Jos Uiterwijk, Eric Postma, Jaap van den Herik (**2002**). The Neural MoveMap Heuristic in Chess. CG 2002, pdf » Neural MoveMap Heuristic
- Albert Xin Jiang (**2003**). *Implementation of Multi-ProbCut in Chess*. CPSC 449 Thesis, pdf
- Jan Renze Steenhuisen (**2005**). *New Results in Deep-Search Behaviour*. ICGA Journal, Vol. 28, No. 4, pdf
- Matej Guid, Ivan Bratko (**2007**). *Factors affecting diminishing returns for searching deeper*. CGW 2007 » Crafty, Rybka, Shredder, Diminishing Returns
- Matej Guid, Ivan Bratko (**2007**). *Factors affecting diminishing returns for searching deeper*. ICGA Journal, Vol. 30, No. 2, pdf
- Matej Guid, Aritz Pérez, Ivan Bratko (**2007**). *How trustworthy is Crafty's analysis of world chess champions*? CGW 2007
- Matej Guid, Aritz Pérez, Ivan Bratko (**2008**). *How trustworthy is Crafty's analysis of world chess champions*? ICGA Journal, Vol. 31, No. 3, pdf
- Robert Hyatt (**2014**). *A Solution to Short PVs Caused by Exact Hash Matches*. ICGA Journal, Vol. 37, No. 3 » Transposition Table, Separate TT for the PV
- Monroe Newborn, Robert Hyatt (**2014**). *Computer Chess Endgame Play with Pawns: Then and Now*. ICGA Journal, Vol. 37, No. 4 » Peasant, Pawn Endgame
- Guy Haworth (**2015**). *Chess Endgame News*. ICGA Journal, Vol. 38, No. 1 » FinalGen

Forum Posts

1995 ...

- Re: Crafty version 8.7 by Robert Hyatt, rgcc, October 05, 1995
- Crafty V9.22 by Robert Hyatt, rgcc, April 05, 1996
- Crafty V9.26 by Robert Hyatt, rgcc, May 08, 1996
- Crafty's opening book - technical details by Robert Hyatt, rgcc, October 04, 1996
- Crafty 11.15 by Robert Hyatt, rgcc, January 30, 1997
- Repetitions in Crafty by Martin Borriss, rgcc, January 30, 1997 » Repetitions
- quiescence search by Andrew Tridgell, rgcc, April 16, 1997 » Quiescence Search, Check
- Crafty and fastest Cray: question to Bob by Jouni Uski, CCC, December 01, 1997
- Parallel Crafty by Robert Hyatt, CCC, March 19, 1998
- Current Crafty strength on SMP? by Charlton Harrison, rgcc, April 29, 1998
- crafty copyright problem by Robert Hyatt, rgcc, February 17, 1999 » Voyager
- Crafty Modifications MUST be made available to Robert Hyatt by KarinsDad, CCC, February 17, 1999
- Crafty implications ... by Chris Whittington, rgcc, March 4, 1999
- Do have the Crafty the Assembler written core? by Leonid, CCC, November 01, 1999

2000 ...

- Crafty internal iterative deepening by Tijs van Dam, CCC, January 26, 2000 » Internal Iterative

Deepening
- Crafty 17.13 - re-importing FEN files? by Wieland Belka, CCC, November 02, 2000
- Crafty 17.13 - using game archives by engines? by Wieland Belka, CCC, November 02, 2000
- razoring in crafty version 16.9, mid 1999 by Vincent Diepeveen, August 21, 2002
- Crafty SMP questions by Matthew Hull, CCC, August 05, 2003 » SMP
- Crafty 19.08 SE 2004 released ... by Michael Byrne, Winboard Forum, January 01, 2004 » Wb2UCI
- Re: Crafty 1910 SE vs Ruffian2 I hate this , but by Peter Skinner, CCC, February 04, 2004 » Ruffian
- is this a sign of broken smp by Michael Byrne, CCC, March 09, 2004 » SMP
- Question about Crafty and Bishop Pair by Renze Steenhuisen, CCC, July 13, 2004 » Bishop Pair

2005 ...

- Re: need help in compiling Crafty by Joshua Haglund, CCC, February 19, 2005
- Crafty and assembly code by Alain Zanchetta, CCC, May 08, 2005
- Re: BitBoard Tests Magic v Non-Rotated 32 Bits v 64 Bits by Robert Hyatt, CCC, August 25, 2007 » Magic Bitboards
- crafty-22.0 by Robert Hyatt, CCC, February 18, 2008
- Crafty questions by Pablo Vazquez, CCC, May 16, 2008
- Re: Lemming Poll by Robert Hyatt, CCC, September 22, 2008 » Tapered Eval, LearningLemming
- Crafty - no analysis output near mate? by cyberfish, December 03, 2008

2010 ...

- old crafty vs new crafty on new hardware by Robert Hyatt, CCC, September 11, 2010
- Crafty tests show that Software has advanced more by Don Dailey, CCC, September 12, 2010
- Final results - Crafty - hardware vs software by Robert Hyatt, CCC, September 13, 2010
- hardware doubling number for Crafty by Robert Hyatt, CCC, September 15, 2010

2011

- On Crafty... by Robert Hyatt, CCC, May 22, 2011
- Re: Robert - How did you came up with the name Crafty by Robert Hyatt, CCC, June 16, 2011
- Re: Still waiting on Ed by BB+, OpenChess Forum, July 07, 2011 » on 22.1 vs. 23.4. differences
- Re: Still waiting on Ed by Robert Hyatt, OpenChess Forum, July 07, 2011 » on 22.1 vs. 23.4. differences
- ICGA rule #2 / opening books / Diep-Crafty, Turino 2006 by Peter Berger, CCC, October 22, 2011 » Opening Book, WCCC 2006

2012

- Crafty source by Robert Hyatt, CCC, September 26, 2012

2013

- Crafty 23.6 released by Peter Skinner, CCC, June 20, 2013
- interesting SMP bug by Robert Hyatt, CCC, September 24, 2013 » Parallel Search
- Crafty 23.8 by Robert Hyatt, CCC, November 11, 2013

2014

- Crafty 24.0 by Robert Hyatt, CCC, May 26, 2014
- Crafty 24.1 by Daniel José Queraltó, CCC, October 01, 2014
- Threads test incl. Crafty 24.1 by Andreas Strangmüller, CCC, October 15, 2014 » Thread, Parallel Search

2015 ...

- Grafty vs Crafty by Marcel van Kervinck, Rybka Forum, April 21, 2015 » Stockfish
- parallel speedup and assorted trivia by Robert Hyatt, CCC, June 05, 2015 » Parallel Search
- Crafty UCI version by Marek Soszynski, July 10, 2015 » UCI
- An interesting parallel search non-bug by Robert Hyatt, CCC, November 05, 2015 » Parallel Search
- Crafty 25.0 Release by Michael B, CCC, December 25, 2015
- Crafty 25.0 announcement by Robert Hyatt, CCC, December 28, 2015

2016

- crafty eval cache by Alvaro Cardoso, CCC, January 01, 2016 » Evaluation Hash Table
- NUMA 101 by Robert Hyatt, CCC, January 07, 2016 » NUMA
- Crafty 25.0.1 by Robert Hyatt, CCC, January 15, 2016
- Crafty's four hash tables by Louis Zulli, CCC, January 17, 2016 » Hash Table
- Crafty c questions by J. Wesley Cleveland, CCC, March 10, 2016 » C
- Crafty chess engine by Krzysztof Grzelak, CCC, March 20, 2016
- Crafty SMP measurement by Robert Hyatt, CCC, April 04, 2016 » Parallel Search
- Around Crafty dev. ... by Frank Quisinsky, CCC, August 31, 2016
  Re: Around Crafty dev. ... by Robert Hyatt, CCC, September 01, 2016
- Crafty 25.1 Release by Michael B, CCC, October 04, 2016
- Crafty v25.2 Release by Michael B, CCC, October 29, 2016

2017

- Crafty Play By Elo (Crafty v25.3) by Michael B, CCC, January 23, 2017 » Playing Strength

External Links

Chess Engine

- GitHub - MichaelB7/Crafty
- Crafty Chess managed by Tracy Riegle
- Index of /downloads/source
- Crafty from Wikipedia
- Crafty Documentation
- Crafty's ICGA Tournaments
- The chess games of Crafty (Computer) from chessgames.com
- Crafty from WBEC Ridderkerk
- How Far We've Come: 20 Years of Personal Computing by Blake Linton Wilfong, November 05, 2000 » ACM 1981, Sargon, Cray Blitz
- 186.crafty: SPEC CPU2000 Benchmark Description
- Crafty Versions at CCRL 40/40

Misc

- crafty - Wiktionary
- Craft from Wikipedia
- American craft from Wikipedia
- Arts and Crafts movement from Wikipedia
- American Craftsman from Wikipedia
- Guitar Craft from Wikipedia
- Robert Fripp and the League of Crafty Guitarists, YouTube Video

References

1. ^ Re: Robert - How did you came up with the name Crafty by Robert Hyatt, CCC, June 16, 2011
2. ^ Crafty from Wikipedia
3. ^ Re: Fastest Magic Move Bitboard Generator ready to use by Robert Hyatt, Winboard Forum, November 09, 2006
4. ^ Re: BitBoard Tests Magic v Non-Rotated 32 Bits v 64 Bits by Robert Hyatt, CCC, August 25, 2007
5. ^ Crafty Chess managed by Tracy Riegle
6. ^ Crafty 25.1 Release by Michael B, CCC, October 04, 2016
7. ^ Crafty Play By Elo (Crafty v25.3) by Michael B, CCC, January 23, 2017
8. ^ Two Girls Being Crafty: Owl Pin Cushion
9. ^ 2010 Fourth Annual ACCA World Computer Rapid Chess Championships - Participants
10. ^ CCT 15 Participants | CCT Events
11. ^ The 2010 Fifth Annual ACCA Americas' Computer Chess Championships - Results
12. ^ Crafty's ICGA Tournaments

13. ^ Re: Hardware vs Software - test result by Robert Hyatt, CCC, December 03, 2008
14. ^ Crafty 25.0 Release by Michael B, CCC, December 25, 2015
15. ^ Index of /downloads/source crafty-23.5.zip, inline64.h, boolean.c
16. ^ Index of /downloads/source, crafty-20.5.zip, boolean.c
17. ^ Index of /downloads/source, crafty-15.17.zip, boolean.c
18. ^ Ramat-Gan 2004 - Chess - Round 9 - Game 3 (ICGA Tournaments)
19. ^ Turin 2006 - Chess - Round 8 - Game 5 (ICGA Tournaments)
20. ^ Re: Bionic v Crafty - a possible solution by Robert Hyatt, CCC, January 26, 1999
21. ^ Re: crafty copyright problem by Robert Hyatt, rgcc, February 17, 1999

What links here?

Page	Date Edited
ACCA 2006	Jul 14, 2014
ACCA 2007	Jul 14, 2014
ACCA 2008	Jul 14, 2014
ACCA 2009	Jul 14, 2014
ACCA 2010	Jul 14, 2014
ACCA 2011	Jul 14, 2014
ACCA 2012	Jul 14, 2014
Alain Zanchetta	Feb 20, 2018
Albert Xin Jiang	Jun 22, 2015
Albrecht Heeffer	Sep 6, 2016
Alexander Naumov	Sep 19, 2014
Alvaro Cardoso	Jan 12, 2018
Amyan	Dec 10, 2013
Andreas Strangmüller	May 15, 2017
Andrew Tridgell	Aug 3, 2015
APHID	Jun 26, 2017
Arasan	Apr 8, 2018
Aritz Pérez	Jan 2, 2017
Artem Petakov	Jan 14, 2017
Aspiration Windows	Nov 1, 2017
AtlanChess	Sep 15, 2016
Bad bishop	Jun 22, 2015
Bionic	May 27, 2013
Bionic Impakt	Sep 6, 2014
Bishop Pair	Sep 10, 2017
Bishops of Opposite Colors	Aug 22, 2012
BitScan	Sep 10, 2017
Book Learning	Jan 1, 2017
Bookbuilder	Feb 4, 2018
Bookup	Feb 4, 2018
Bratko-Kopec Test	Jun 1, 2015
Brause	Aug 30, 2017

Page	Date Edited
C	Feb 19, 2018
CCT Tournaments	Jan 29, 2017
CCT1	Dec 30, 2012
CCT10	May 5, 2013
CCT11	Feb 17, 2015
CCT12	Jan 28, 2018
CCT13	Dec 6, 2013
CCT14	Aug 23, 2012
CCT15	Oct 21, 2014
CCT16	May 13, 2014
CCT2	Sep 7, 2012
CCT3	Feb 14, 2013
CCT4	Apr 22, 2013
CCT5	Feb 22, 2013
CCT6	May 29, 2014
CCT7	Dec 16, 2017
CCT8	Apr 6, 2013
CCT9	Aug 25, 2013
Centipawns	Aug 9, 2016
CGW 2007	Jan 2, 2017
Check	Feb 1, 2018
Check Extensions	Apr 19, 2018
Chess Assistant	Jul 16, 2017
Chess Wizard	Dec 22, 2014
Chezzz	Jan 20, 2013
Chinito	Jan 7, 2016
Cluster Toga	Jan 7, 2016
Copy-Make	May 23, 2017
Crafty	Jan 28, 2018
Cray Blitz	Dec 25, 2017
Cutechess-cli	Jan 6, 2018
DanChess	Jun 17, 2012
DEC Alpha	Aug 15, 2015
Deep Sjeng	Jan 7, 2016
Dennis Sceviour	Feb 14, 2018
Depth	Feb 25, 2018
DOCCC 2001	Aug 15, 2015
Dreamer	Aug 5, 2015
Edwards' Tablebases	Sep 26, 2016
Eigenmann Endgame Test	Jun 1, 2017
Endgame Tablebases	Mar 6, 2018
Engine releases	Apr 23, 2018
Engines	Mar 10, 2018
Equinox	Dec 12, 2016
Ethereal	Mar 20, 2018

Page	Date Edited
Eugen	Jan 7, 2016
Eugenio Castillo Jiménez	Jul 23, 2017
Evaluation Hash Table	Dec 4, 2016
Evaluation of Pieces	Jan 8, 2018
Fafis	May 17, 2016
Fernando Villegas	Aug 15, 2016
FinalGen	Aug 14, 2016
Firefly	Oct 23, 2017
Fortress (Engine)	Oct 19, 2017
Frank Phillips	Sep 26, 2016
Frédéric Louguet	Aug 4, 2013
Gabriele Müller	Jan 22, 2012
Gavon	Apr 30, 2018
Gazebo	May 21, 2017
Ghost	Feb 2, 2018
GridChess	Jul 13, 2016
GridGinkGo	Jul 13, 2016
GridProtector	Jul 13, 2016
Guido Schimmels	Apr 23, 2013
Gunda-1	Jan 8, 2016
Guy Haworth	Nov 20, 2017
Hakkapeliitta	Apr 26, 2016
Hans Secelle	Nov 2, 2012
Hardware	Jan 20, 2018
Harun Taner	May 17, 2015
Hash Table	Jan 1, 2018
Hidden Passed Pawn	Nov 11, 2017
Historical Examples	Jan 16, 2017
Hossa	Jan 7, 2016
Houdini	Apr 14, 2018
ICGA Journal	Dec 21, 2017
InterChess	Jan 8, 2016
Internal Iterative Deepening	Feb 5, 2018
IsiChess	Jan 7, 2016
Itanium	Aug 29, 2015
Ivan Bratko	Aug 16, 2017
Jan Renze Steenhuisen	Sep 10, 2017
John Merlino	Feb 26, 2015
Jouni Uski	Sep 28, 2017
Junior	Feb 27, 2018
Kaissa (BY)	May 16, 2016
King of Kings	Sep 4, 2013
King Safety	Feb 14, 2018
Knowledge	Jul 22, 2017
LaGrande	Jan 22, 2012

Page	Date Edited
LaPetite	Jan 22, 2012
Leila	May 8, 2017
Leonid Liberman	May 23, 2015
List (Program)	Jan 7, 2016
Logging	Jan 20, 2018
Louis Zulli	Feb 18, 2015
Magic Bitboards	Apr 13, 2018
Marcel van Kervinck	Dec 6, 2017
Martin Thoresen	Apr 29, 2015
Match Statistics	Mar 31, 2018
Matej Guid	Jan 6, 2018
Mathematician	Apr 9, 2018
Matthew Lai	Dec 6, 2017
MBChess	Sep 29, 2012
Michael Byrne	Jun 23, 2017
Michel Langeveld	Feb 4, 2018
Monroe Newborn	Dec 23, 2017
Movei	Jan 7, 2016
Nalimov Tablebases	Dec 23, 2016
Naum	Nov 23, 2017
Neural MoveMap Heuristic	Sep 24, 2015
Novag Best-Publication Awards to Include	Jan 2, 2014
Null Move Pruning	Dec 2, 2017
NUMA	Mar 29, 2017
OliThink	May 19, 2017
Open Source Engines	Jul 14, 2015
Opponent Model Search	Aug 14, 2017
Outside passed pawn	Jun 7, 2012
Pablo Vazquez	Apr 11, 2014
Parallel Search	Dec 30, 2017
Patriot	Sep 27, 2016
Pawn Endgame	Oct 11, 2017
Pawn Race	Dec 16, 2017
Peasant	May 21, 2015
Pepito	Nov 27, 2014
Perft	Sep 26, 2017
Perft Results	Feb 10, 2018
Persistent Hash Table	Dec 31, 2017
Peter Berger	Feb 29, 2016
Peter Skinner	Oct 30, 2016
Playing Strength	Mar 31, 2018
Plisk	Dec 31, 2014
Point Value by Regression Analysis	Aug 26, 2017
PowerPC	Oct 6, 2017
Principal Variation Search	Oct 22, 2017

Page	Date Edited
ProbCut	Mar 25, 2016
Protector	Oct 12, 2017
Protocols	Jan 20, 2018
PVS and aspiration	Oct 8, 2014
PyChess	Dec 28, 2017
Quantifying Evaluation features	Jun 6, 2012
Quiescence Search	Aug 19, 2017
Rafael Peña	Nov 9, 2011
Razoring	Oct 5, 2015
RedQueen	Nov 13, 2017
Relative History Heuristic	Jun 8, 2015
Rémi Coulom	Feb 5, 2018
Ren Wu	Jul 23, 2017
Repetitions	Jan 16, 2018
Robert Houdart	Dec 8, 2017
Robert Hyatt	Dec 25, 2017
Rudolf Huber	Jan 7, 2016
Rybka	Mar 27, 2017
Rybka Controversy	Jan 18, 2018
Sargon	Dec 25, 2017
Searcher	Sep 26, 2016
SEE - The Swap Algorithm	Jun 5, 2017
Severi Salminen	May 7, 2015
Shredder	Jan 21, 2018
Sloppy	Sep 17, 2016
SMP	Dec 26, 2017
Software	Jul 28, 2017
Steffen A. Jakob	Jan 7, 2016
Stockfish	Apr 7, 2018
Strategic Test Suite	Dec 19, 2016
Suryadi Harmanto	Mar 21, 2015
Tapered Eval	Jan 9, 2018
TCEC Season 5	Jun 2, 2014
TCEC Season 6	Dec 2, 2014
TCEC Season 7	Jan 23, 2015
TCEC Season 8	Nov 30, 2015
Thread	Apr 22, 2018
Tijs van Dam	Feb 27, 2017
Tony Hedlund	Jun 19, 2017
UCI	Mar 4, 2018
University of Alabama at Birmingham	Jul 14, 2014
Unmake Move	Jun 10, 2017
Vladimir Yelin	May 16, 2016
Voyager	Aug 12, 2012
Wb2UCI	Feb 1, 2018

Page	Date Edited
WCCC 1999	Jul 13, 2017
WCCC 2004	Jan 28, 2018
WCCC 2005	Dec 27, 2016
WCCC 2006	Jan 28, 2018
WCRCC 2007	Nov 21, 2016
WCRCC 2008	Nov 21, 2016
WCRCC 2009	Jul 14, 2014
WCRCC 2010	Jul 14, 2014
WCRCC 2011	Oct 21, 2014
WCRCC 2012	Jul 14, 2014
WCRCC 2014	Jul 27, 2014
WCRCC 2015	Jul 19, 2015
WCRCC 2016	Jun 27, 2017
Weak Pawns	Dec 19, 2013
Who's Who	Sep 6, 2017
Wieland Belka	Jul 13, 2013
Win at Chess	Sep 25, 2017
Wing	Oct 26, 2017
WMCCC 1996	Sep 15, 2016
WMCCC 1997	Apr 4, 2017
WMCCC 2000	Feb 27, 2018
WMCCC 2001	Jul 25, 2017
World Computer Chess Championship	Mar 6, 2018
Xiphos	Apr 22, 2018
Zappa	Oct 24, 2017
Đorđe Vidanović	Feb 20, 2017

Up one level