Home * Engines * **OliThink**

OliThink5 Java online [4]

OliThink,
an open source chess engine supporting the Chess Engine Communication Protocol written by Oliver Brausch with C and Java versions available, and binaries running under Windows, Linux and Mac OS [1]. The completely rewritten OliThink 5.x has a very fast move generator based on the framework of the Perft program OliPerft with a plain bitboard board representation without any piece-lists or board arrays [2]. OliThink's evaluation consists almost of material balance and mobility, plus a very simple pawn structure evaluation, rewarding passed pawns. OliThink 4.13 played the CCT6 in 2004, with four points out of nine games [3].

Table of Contents

Description

Search

Sliding Piece Attacks

C Source

Java Source

Selected Games

SEE

Bodo

See also

Forum Posts

1998 ...

2000 ...

2008

2009

2010

2011

2012

External Links

References

What links here?

Description

Search

OliThink's search relies on PVS without aspiration windows in its iterative deepening loop [5], along with a fixed sized transposition table. It further applies flexible null move pruning, late move reductions [6], IID, singular reply-, check- and passed pawn extensions [7]. Move ordering considers PV-moves stored in a triangular PV-Table, SEE, killer- and history heuristic.

Sliding Piece Attacks

OliThink pre 5 versions used rotated bitboards to determine sliding piece attacks. Since version 5, only the usual occupancy is used to map the masked line to an index, for files and diagonals by a north-fill multiplication and right shift as also applied in kindergarten bitboards [8], with the addition not only to lookup attack bitboards, but also X-ray attacks through the first blocking pieces (if any) of both ray-directions [9] . A pre-initialized array of 8 times 8K bitboards (512 Kbyte in total) is used for attacks on ranks, files, diagonals and anti-diagonals in its lower half, while the upper half holds appropriate x-ray attacks. Per line, a 13-bit index is composed of the 6-bit square index and a 7-bit occupancy key.

C Source

These are the relevant code snippets and data declarations of the attack and x-ray attack getter in the C source, initialization omitted [10]. Using addition instead of bitwise-or might take advantage of the x86 lea instruction, specially for the line-offsets:

```
static u64 rays[0x10000]; /* 8*64 = 512 KByte */
u64 bmask45[64];
u64 bmask135[64];

#define BOARD (colorb[0] | colorb[1])

#define RATT1(f)  rays[((f) << 7) | key000(BOARD, f)         ]
#define RATT2(f)  rays[((f) << 7) | key090(BOARD, f) | 0x2000]
#define BATT3(f)  rays[((f) << 7) | key045(BOARD, f) | 0x4000]
#define BATT4(f)  rays[((f) << 7) | key135(BOARD, f) | 0x6000]
#define RXRAY1(f) rays[((f) << 7) | key000(BOARD, f) | 0x8000]
#define RXRAY2(f) rays[((f) << 7) | key090(BOARD, f) | 0xA000]
#define BXRAY3(f) rays[((f) << 7) | key045(BOARD, f) | 0xC000]
#define BXRAY4(f) rays[((f) << 7) | key135(BOARD, f) | 0xE000]

int key000(u64 b, int f) {return (int) ((b >> (f & 56)) & 0x7E);}
int key090(u64 b, int f) {
   u64 _b = (b >> (f&7)) & 0x0101010101010101LL;
  _b = _b * 0x0080402010080400LL;
   return (int)(_b >> 57);
}
int keyDiag(u64 _b) {
   _b = _b * 0x0202020202020202LL;
   return (int)(_b >> 57);
}
int key045(u64 b, int f) {return keyDiag(b & bmask45[f]);}
int key135(u64 b, int f) {return keyDiag(b & bmask135[f]);}
```

Java Source

In Java, the code looks quite similar, embedded inside the class OliThink [11], using the unsigned right shift operator (`>>>`) instead the arithmetical one (`>>`) inside the keyxxx routines would safe the post-masking with `0x7f`:

```
final static long[] rays = new long[0x10000];
final static long[] bmask45 = new long[64];
final static long[] bmask135 = new long[64];

static long BOARD() { return (colorb[0] | colorb[1]); }

static long RATT1(int f)  {return rays[((f) << 7) | key000(BOARD()
, f)         ];}
static long RATT2(int f)  {return rays[((f) << 7) | key090(BOARD()
, f) | 0x2000];}
static long BATT3(int f)  {return rays[((f) << 7) | key045(BOARD()
, f) | 0x4000];}
static long BATT4(int f)  {return rays[((f) << 7) | key135(BOARD()
, f) | 0x6000];}
static long RXRAY1(int f) {return rays[((f) << 7) | key000(BOARD()
, f) | 0x8000];}
static long RXRAY2(int f) {return rays[((f) << 7) | key090(BOARD()
, f) | 0xA000];}
static long BXRAY3(int f) {return rays[((f) << 7) | key045(BOARD()
, f) | 0xC000];}
static long BXRAY4(int f) {return rays[((f) << 7) | key135(BOARD()
, f) | 0xE000];}

static int key000(long b, int f) {return (int) ((b >> (f & 56)) &
 0x7E);}
static int key090(long b, int f) {
   long _b = (b >> (f&7)) & 0x0101010101010101L;
   _b = _b * 0x0080402010080400L;
   return (int)((_b >> 57) & 0x7F);
}
static int keyDiag(long _b) {
   _b = _b * 0x0202020202020202L;
   return (int)((_b >> 57) & 0x7F);
}
static int key045(long b, int f) {return keyDiag(b & bmask45[f]);}
static int key135(long b, int f) {return keyDiag(b & bmask135[f]);}
```

Selected Games

SEE

CCT6, SEE - OliThink 4.13 [12]

```
[Event "CCT6"]
```

```
[Site "Internet Chess Club"]
[Date "2004.01.31"]
[Round "3"]
[White "SEE"]
[Black "OliThink 4.13"]
[Result "0-1"]

1.Nf3 d5 2.d4 e6 3.Bd2 c5 4.e3 Nc6 5.Bb5 Bd7 6.O-
O Qb6 7.Nc3 cxd4 8.Nxd4 Nxd4 9.Bxd7+
Kxd7 10.exd4 Qxd4 11.Qe2 Nf6 12.Be3 Qb4 13.a3 Qa5 14.b4 Qa6 15.Qxa6 bx
a6 16.Bd4 Rc8
17.Ra2 a5 18.b5 Bc5 19.Ne2 Bxd4 20.Nxd4 a4 21.Rb2 Rc3 22.Ra1 Rb8 23.Nc
6 Rc8 24.Ne5+
Ke8 25.Raa2 Ne4 26.Nc6 a6 27.Na7 R8c7 28.b6 Rb7 29.Rb4 Nd6 30.g3 Nc4 3
1.Nc6 Kd7 32.Nd4
Nxb6 33.Ne2 Rc5 34.Rab2 Kc6 35.Kg2 e5 36.c3 f6 37.Rb1 a5 38.R4b2 g5 39
.f3 h5 40.g4 h4
41.Kf2 Rb8 42.Ke1 Kc7 43.Kf2 Nd7 44.Rxb8 Nxb8 45.Ke3 Nd7 46.Kf2 Nb6 47
.Ke3 Nc4+ 48.Kd3
Nxa3 49.Ra1 Nc4 50.Rb1 a3 51.Nc1 Nb2+ 52.Kc2 d4 53.Na2 d3+ 54.Kd2 Rd5
55.c4 Nxc4+
56.Ke1 d2+ 57.Ke2 Nb2 58.Rd1 Nxd1 0-1
```

Bodo

CCT6, Bodo - OliThink 4.13 [13]

```
[Event "CCT6"]
[Site "Internet Chess Club"]
[Date "2004.02.01"]
[Round "9"]
[White "Bodo"]
[Black "OliThink 4.13"]
[Result "1-0"]

1.Nf3 d5 2.g3 g6 3.Bg2 Bg7 4.d4 Nf6 5.Ne5 c6 6.O-
O Nbd7 7.c4 Ne4 8.cxd5 Nxe5 9.dxe5
cxd5 10.Qa4+ Bd7 11.Qb4 Bxe5 12.Qxb7 Nf6 13.Nc3 e6 14.e4 Bxc3 15.bxc3
Nxe4 16.Bxe4 dxe4
17.Bh6 f5 18.Rfd1 Rc8 19.Qxa7 Rg8 20.Rab1 Rc7 21.Qd4 Qc8 22.Qf6 g5 23.
Rb8 Qxb8 24.Bxg5
Rxg5 25.Qh8+ Ke7 26.Qxb8 Rc8 27.Qd6+ Kf6  1-0
```

See also

- Thought

Forum Posts

1998 ...

- OliThink 2.2.1 released by Ralph Jörg Hellmig, CCC, December 07, 1998
- Olithink, test it! by Carlos E.A. Drake, Winboard Forum, September 24, 1999

2000 ...

- OliThink bug? 1 0 Winboard tournament by Aloisio Ponti Lopes, CCC, July 02, 2000
- Olithink 3.0.1 by Martin G, Winboard Forum, March 26, 2001
- Olithink by Dann Corbit, Winboard Forum, March 28, 2001
- Question about details of hashing (olithink) by Michel Langeveld, CCC, January 22, 2004
- Looking for a last moment operator for Olithink 4.1.3 for CCT-6 by Dann Corbit, CCC, January 26, 2004 » CCT6
- Re: Watch Olithink by Oliver Brausch, CCC, January 27, 2004
- Olithink @ CCT6 by Peter Skinner, CCC, February 01, 2004
- OliThink@CCT6 - Programmers View by Oliver Brausch, CCC, February 03, 2004
- OliThink 5 by Oliver Brausch, Winboard Forum, October 29, 2004
- OliThink 5 by Oliver Brausch, rgcc, October 30, 2004
- Re: Question about SEE (Static exchange evaluation) by Oliver Brausch, CCC, December 18, 2007 » X-ray Attacks
- OliPerft with divide Option as Pre Version for OliThink 5 by Oliver Brausch, CCC, December 27, 2007 » Perft

2008

- Problem with Transposition Table and Repitition-Draw by Oliver Brausch, CCC, January 11, 2008 » Transposition Table, Repetitions
- OliThink 5.0.4 - GNU Chess 5.0.7 Bullet by Oliver Brausch, CCC, January 13, 2008
- OliThink 5.0.8 released by Oliver Brausch, CCC, January 25, 2008
- The limits of "Just-mobility-evaluation" by Oliver Brausch, CCC, January 29, 2008
- OliThink 5.0.9 released by Oliver Brausch, CCC, February 06, 2008
- OliThink 5.1.0 released by Oliver Brausch, CCC, March 19, 2008
- OliThink 5.1.1 released by Oliver Brausch, CCC, March 21, 2008
- OliThink 5.1.2 released by Oliver Brausch, CCC, April 03, 2008
- OliThink couldn't win K+B+P vs K against Toga by Oliver Brausch, CCC, April 05, 2008 » Toga
- OliThink 5.1.3 released by Oliver Brausch, CCC, April 11, 2008
- OliThink 5.1.4 released by Oliver Brausch, CCC, October 16, 2008
- Tournament with OliThink, Crafty and Glaurung. OliThink 5.1. by Oliver Brausch, CCC, October 19, 2008 » Crafty, Glaurung
- OliThink 5.1.6 released by Oliver Brausch, CCC, October 19, 2008

- OliThink 5.1.7 released by Oliver Brausch, CCC, October 27, 2008
- Olithink 5.1.8 released because of better ChessDM vs Crafty by Oliver Brausch, CCC, October 29, 2008
- Kindergarten Bitboard Approach by Gerd Isenberg by Edsel Apostol, CCC, November 05, 2008 » Kindergarten Bitboards

2009

- Bug found in OliThink 5.1.9 => Corrected code (5.2.0) only by Oliver Brausch, CCC, September 18, 2009
- OliThink 5.2.1 Java by Oliver Brausch, CCC, September 25, 2009
- OliThink5 Java can be played in browser with self written GUI by Oliver Brausch, CCC, November 25, 2009
- OliThink GUI in Java... Complete source posted by Oliver Brausch, CCC, November 25, 2009 » GUI, Java
- OliThink 5.2.2 released with 48MB Hashsize by Oliver Brausch, CCC, December 05, 2009
- OliThink 5.2.3 released by Oliver Brausch, CCC, December 16, 2009
- OliThink 5.2.4 released by Oliver Brausch, CCC, December 22, 2009
- OliThink 5.2.5 released by Oliver Brausch, CCC, December 25, 2009
- OliThink 5.2.6 released introducing LMR by Oliver Brausch, CCC, December 31, 2009

2010

- OliThink 5.2.7 released by Oliver Brausch, CCC, January 05, 2010
- Problem with exploding tree because of extensions by Oliver Brausch, CCC, January 05, 2010 » Search Explosion, Extensions
- OliThink 5.2.9 released by Oliver Brausch, CCC, January 09, 2010
- OliThink 5.3.0 released by Oliver Brausch, CCC, January 25, 2010
- OliThink 5.3.0 Java performance by Oliver Brausch, CCC, July 18, 2010

2011

- Olithink by colin, CCC, May 22, 2011

2012

- Open Source Blitz Rating List: Olithink 5.3.0 by Lucas Braesch, CCC, January 21, 2012
- OliThink 5.3.1 released (Win, Mac, Linux and Java) by Oliver Brausch, CCC, February 28, 2012
- Open Source Bullet: Olithink 5.3.1 by Lucas Braesch, CCC, February 29, 2012
- OliThink 5.3.2 released (Source, Windows, Mac and Linux) by Oliver Brausch, CCC, March 02, 2012
- Open Source Bullet: Olithink 5.3.2, Diablo 1.4 by Lucas Braesch, CCC, March 05, 2012

External Links

- Chess Engine OliThink by Oliver Brausch
- OliThink 5.3.0 in CCRL 40/40
- OliThink 5.3.2 64-bit in CCRL 40/40
- Download OliThink Free for Mac OS X by Softpedia
  OliThink Screenshots

References

1. ^ Chess Engine OliThink by Oliver Brausch
2. ^ OliPerft with divide Option as Pre Version for OliThink 5 by Oliver Brausch, CCC, December 27, 2007
3. ^ OliThink@CCT6 - Programmers View by Oliver Brausch, CCC, February 03, 2004
4. ^ Chess Engine OliThink by Oliver Brausch
5. ^ OliThink 5.1.2 released by Oliver Brausch, CCC, April 03, 2008
6. ^ OliThink 5.2.6 released introducing LMR by Oliver Brausch, CCC, December 31, 2009
7. ^ OliThink 5.1.1 released by Oliver Brausch, CCC, March 21, 2008
8. ^ Kindergarten Bitboard Approach by Gerd Isenberg by Edsel Apostol, CCC, November 05, 2008
9. ^ Re: Question about SEE (Static exchange evaluation) by Oliver Brausch, CCC, December 18, 2007
10. ^ Chess Engine OliThink - OliThink Source Code Vers. 5.3.2 Native - olithink.c (slightly edited)
11. ^ Chess Engine OliThink - OliThink Source Code Vers. 5.3.2 Java - olithink.java (slightly edited)
12. ^ CCT6 hosted by Volker Richey
13. ^ CCT6 hosted by Volker Richey

What links here?

Page	Date Edited
CCT6	May 29, 2014
Engines	Mar 10, 2018
GUI	Mar 16, 2018
Java	Feb 25, 2018
Kindergarten Bitboards	Aug 1, 2017
Mobility	Jan 17, 2018
Occupancy of any Line	Sep 16, 2016
OliThink	May 19, 2017
Oliver Brausch	May 19, 2017
Perft	Sep 26, 2017
Plisk	Dec 31, 2014
Sliding Piece Attacks	May 27, 2016
X-ray Attacks (Bitboards)	Mar 31, 2015

Up one level