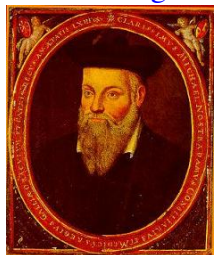


[Home](#) * [Engines](#) * **Prophet**



[Nostradamus](#) ^[5]

Prophet,

a [Chess Engine Communication Protocol](#) compliant [open source chess engine](#) under the [GNU General Public License](#), written by [James Swafford](#) in [C++](#) as a better [C](#). As successor of [Galahad](#), Prophet is a traditional [bitmap](#) based program. Prophet was used in experiments with [TD-Leaf](#) ^[1] when James Swafford was undergraduate at [East Carolina University](#) ^[2], and testbed for [parallel search](#) algorithms as topic of its author's Masters project ^[3]. In September 2017, Prophet3 was released as open source under the [MIT License](#), using [magic bitboards](#) ^[4].

Table of Contents

[Description](#)

[Bitboard Infrastructure](#)

[Rotated Bitboards](#)

[BitScan & PopCount](#)

[Search](#)

[Evaluation](#)

[Tournament Play](#)

[Selected Games](#)

[See also](#)

[Publications](#)

[Forum Posts](#)

[External Links](#)

[Chess Engine](#)

[Misc](#)

[References](#)

[What links here?](#)

Description

Bitboard Infrastructure

Rotated Bitboards

Prophet2 uses [rotated bitboards](#) with 1/2 MiB lookup tables, indexed by square and [8-bit line occupancy](#), to determine [sliding piece attacks](#), ignoring the possible fourfold reduction excluding the redundant [outer squares](#) ^[6].

```
Bitmap diag_alh8_attacks[64][256];
Bitmap diag_hla8_attacks[64][256];
Bitmap file_attacks[64][256];
Bitmap rank_attacks[64][256];

Bitmap GenBishopMap(Position* p,int sq) {
    Bitmap b=0;
    // find index for a1->h8 diagonal
    int ind = (p->allpieces45 >> diag_alh8_shift[sq]) &
diag_alh8_mask[sq];
    b = diag_alh8_attacks[sq][ind];
    // find index for h1->a8 diagonal
    ind = (p->allpieces315 >> diag_hla8_shift[sq]) & diag_hla8_mask[sq];
    b |= diag_hla8_attacks[sq][ind];
    return b;
}
```

BitScan & PopCount

Similar to [Amundsen](#), the [memory](#) bacchanal is attended by [population count](#), [bitscan forward](#) (GetLSBFast) and [reverse](#) (GetMSBFast) with three 16-bit indexed, 64K lookup tables of [double word](#) integers, another 3/4 MiB competing for the caches [\[7\]](#):

```
int  num_bits[65536];
int  lsb[65536];
int  msb[65536];

/*****
 *
 *
 * int GetLSBFast(Bitmap b)
 *
 *
 *
 * Returns the least significant bit in a bitmap (1-64), or 0 if no bit set.
 * Assumes lsb[] is initialized.  Much faster than GetLSBSlow().
 *
 *
 *****/

int GetLSBFast(Bitmap b) {
    int r = lsb[(int)(b & 65535)];
    if (r) return r;
    b >>= 16;
    r = lsb[(int)(b & 65535)];
    if (r) return r+16;
    b >>= 16;
    r = lsb[(int)(b & 65535)];
    if (r) return r+32;
    b >>= 16;
    r = lsb[(int)b];
    if (r) return r+48;
    return 0;
}
```

Search

Prophet's [search](#) is [alpha-beta PVS](#) with [transposition table](#) inside an [iterative deepening](#) framework with [fractional plies](#) and [aspiration windows](#) of $\pm 1/3$ pawn value. [Selectivity](#) is due to [adaptive null move pruning](#), [futility](#) and [extended futility pruning](#), and [fractional extensions](#) on [checks](#), [single replies](#), [passed](#)

[pawn advances](#) and [recaptures](#). Beside TT and keeping a [linked list](#) of [PVs on the Stack](#), [Move ordering](#) considers [killer](#) and [history heuristic](#), as well as [MVV-LVA](#) and [SEE](#) for [captures](#).

Evaluation

The [evaluation](#) in [centipawn resolution](#) determines a [score](#) based on [material balance](#) and [piece-square tables](#), and further considers various positional features, such as [development](#), [mobility](#), [king safety](#) and [pawn structure](#), and [trapped pieces](#).

Tournament Play

Prophet played five [CCT Tournaments](#), from [CCT7](#) in 2005, until [CCT11](#) in 2009, three [ACCA Americas' Computer Chess Championships](#), [ACCA 2006](#), [ACCA 2007](#) and [ACCA 2008](#), and the [WCRCC 2008](#) Second Annual ACCA World Computer Rapid Chess Championship.

Selected Games

[CCT11](#), round 2, [Prophet](#) - [NoonianChess](#) ^[8]

```
[Event "CCT11"]
[Site "Internet Chess Club"]
[Date "2009.03.21"]
[Round "2"]
[White "Prophet"]
[Black "NoonianChess"]
[Result "1-0"]
```

```
1.c4 g6 2.Nc3 Bg7 3.g3 c5 4.Bg2 Nc6 5.Nf3 d6 6.O-
O Nf6 7.d4 cxd4 8.Nxd4
Bd7 9.Nc2 O-
O 10.b3 a6 11.Bb2 Ng4 12.h3 Nh6 13.Rb1 Bf5 14.e4 Be6 15.Nd5
f6 16.a4 Re8 17.Qd2 Bf7 18.Bc3 g5 19.a5 Rb8 20.Nb6 Bh5 21.Kh1 e5 22.Ne
3
Qe7 23.f4 gxf4 24.gxf4 Qd8 25.Ned5 Kh8 26.Bf3 Bxf3+ 27.Rxf3 f5 28.exf5
Nxf5 29.fxe5 Nh4 30.Rf7 Nxe5 31.Bxe5 Rxe5 32.Nd7 Qe8 33.Nxe5 dxe5 34.R
bf1
Qe6 35.Qe3 Ng6 36.R1f6 Qc8 37.Rd6 e4 38.Rdd7 Ne5 39.Rxg7 Qxd7 40.Rxd7
```

Nxd7

41.Qd4+ Kg8 42.Ne7+ Kf7 43.Qxd7 Rf8 44.Nf5+ Kf6 45.Qe7+ Kxf5 46.Qxf8+ Kg5
47.Qf1 e3 48.c5 e2 49.Qxe2 Kf4 50.Qe7 h6 51.Qxb7 h5 52.Kg2 Ke3 53.Qd5 Ke2
54.c6 Ke3 55.c7 Kf4 56.Kh1 Ke3 57.c8=Q Kf2 58.Qd4+ Ke2 59.Qxa6+ Ke1
60.Qe6+ Kf1 61.Qdf6# 1-0

See also

- [chess4j](#)
- [Galahad](#)
- [Metaphysics](#)
- [Tristram](#)

Publications

- [James Swafford](#) (2002). *Optimizing Parameter Learning using Temporal Differences*. [AAAI-02](#), Student Abstracts, [pdf](#)
- [James Swafford](#) (2008). *A Survey of Parallel Search Algorithms over Alpha-Beta Search Trees using Symmetric Multiprocessor Machines*. Masters Project, [East Carolina University](#), advisor [Ronnie Smith](#), [pdf](#)

Forum Posts

- [It's Prophet](#) by [James Swafford](#), [CCC](#), February 04, 2005 » [CCT7](#)
- [Re: Speedup with bitboards on 64-bit CPUs](#) by [James Swafford](#), [CCC](#), April 28, 2007
- [pthread weirdness](#) by [James Swafford](#), [CCC](#), May 29, 2007
- [Prophet ACCA PanAm 2008 tournament notes](#) by [James Swafford](#), [CCC](#), November 10, 2008 » [ACCA 2008](#)
- [prophet3 released with source code](#) by [James Swafford](#), [CCC](#), September 30, 2017

External Links

Chess Engine

- [prophet | James Swafford](#)
- [GitHub - jswoff/prophet3: a C based chess engine](#)
- [Prophet Chess by James Swafford](#), [Wayback Machine](#)
- [James Swafford » Computer Chess](#)
- [Prophet 2.0 64-bit](#) in [CCRL 40/40](#)

Misc

- [Prophet from Wikipedia](#)
- [Prophet \(disambiguation\) from Wikipedia](#)
- [Prophecy from Wikipedia](#)
- [Prophecy \(disambiguation\) from Wikipedia](#)
- [Nostradamus: The Last Prophecy from Wikipedia](#)
- [Jazz is Dead - Estimated Prophet](#), [Fox Theatre](#), [Boulder, Colorado](#), April 13, 1999, [YouTube](#) Video lineup: [Jimmy Herring](#), [T Lavitz](#), [Alphonso Johnson](#), [Billy Cobham](#)

References

1. [^ James Swafford \(2002\)](#). *Optimizing Parameter Learning using Temporal Differences*. [AAAI-02](#), Student Abstracts, [pdf](#)
2. [^ prophet | James Swafford](#)
3. [^ James Swafford \(2008\)](#). *A Survey of Parallel Search Algorithms over Alpha-Beta Search Trees using Symmetric Multiprocessor Machines*. Masters Project, [East Carolina University](#), advisor [Ronnie Smith](#), [pdf](#)
4. [^ prophet3 released with source code](#) by [James Swafford](#), [CCC](#), September 30, 2017
5. [^](#) The Portrait of [Michel de Nostredame](#) (Nostradamus), a French Renaissance Medicine & Astrologer, painted by his son [César de Nostredame](#) (1553-1630?) about 1614 A.D. 16cm x 18cm. (cf. Jean Boyer, “Deux peintres oubliés du XVIe siècle: Etienne Martellange et César de Nostredame”, Bulletin de la Société de l'Histoire de l'art français, Année 1971 (1972), pp.13-20)
6. [^ prophet-20b1-ja.zip](#), [globals.cpp](#), [bitmaps.cpp](#)
7. [^ prophet-20b1-ja.zip](#), [globals.cpp](#), [bitmaps.cpp](#)
8. [^ CCT11 results and games](#)

What links here?

Page	Date Edited
ACCA 2006	Jul 14, 2014
ACCA 2007	Jul 14, 2014
ACCA 2008	Jul 14, 2014
BitScan	Sep 10, 2017
CCT10	May 5, 2013
CCT11	Feb 17, 2015
CCT7	Dec 16, 2017
CCT8	Apr 6, 2013
CCT9	Aug 25, 2013
chess4j	Jan 1, 2018
Engine releases	Apr 23, 2018
Engines	Mar 10, 2018
Galahad	Jan 8, 2015
James Swafford	Jan 1, 2018
Prophet	Sep 30, 2017
WCRCC 2008	Nov 21, 2016

[Up one level](#)