

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



The [Red Queen](#) lecturing [Alice](#) ^[2]

RedQueen,

an [UCI](#) compliant [open source chess engine](#) under the [GNU General Public License](#), written in [C++](#) by [Ben-Hur Carlos Vieira Langoni Junior](#). Using minimal library dependencies as possible, it could easily be ported to various [operating systems](#) such as [Windows](#), [Linux](#), [Mac OS](#), and [Android](#). The name RedQueen was inspired by the [Red Queen character](#) in [Lewis Carroll's Through the Looking-Glass](#) novel ^[1].

Table of Contents

[Description](#)

[Bitboards](#)

[Search](#)

[Evaluation](#)

[Tournament Play](#)

[Selected Games](#)

[See also](#)

[Forum Posts](#)

[2010](#)

[2011](#)

[2012 ...](#)

[External Links](#)

[Chess Engine](#)

[Misc](#)[References](#)[What links here?](#)

Description

Bitboards

RedQueen is a [bitboard](#) engine and uses [Pradu Kannan's magic bitboards](#) ^[3] to determine [sliding piece attacks](#). [BitScans](#) are either implemented in [inline assembly](#) for [x86-64 processor instructions](#), or with [De Bruijn multiplication](#) and [double conversion](#) for forward and reverse scans respectively ^[4]:

```
const uint64_t debruijn64 = 0x07EDD5E59A4E28C2ULL;
const uint32_t index64[64] = {
    63,  0, 58,  1, 59, 47, 53,  2,
    60, 39, 48, 27, 54, 33, 42,  3,
    61, 51, 37, 40, 49, 18, 28, 20,
    55, 30, 34, 11, 43, 14, 22,  4,
    62, 57, 46, 52, 38, 26, 32, 41,
    50, 36, 17, 19, 29, 10, 13, 21,
    56, 45, 25, 31, 35, 16,  9, 12,
    44, 24, 15,  8, 23,  7,  6,  5
};

inline int bitScanForward(int* const index, const uint64_t mask) {
#ifdef __LP64__
    uint64_t ret;
    asm ("bsfq %[mask], %[ret]" : [ret] "=r" (ret) : [mask] "mr" (
mask));
    *index = int(ret);
#else
    *index = int(index64[((mask & -mask) * debruijn64) >> 58]);
#endif
    return mask?1:0;
}

inline int bitScanReverse(int* const index, const uint64_t mask) {
#ifdef __LP64__
    uint64_t ret;
    asm ("bsrq %[mask], %[ret]" : [ret] "=r" (ret) : [mask] "mr" (mask));
    *index = (unsigned int)ret;

```

```
#else
    union {
        double d;
        struct {
            unsigned int mantissal : 32;
            unsigned int mantissah : 20;
            unsigned int exponent : 11;
            unsigned int sign : 1;
        };
    } ud;
    ud.d = (double)(mask & ~(mask >> 32));
    *index = ud.exponent - 1023;
#endif
    return mask?1:0;
}
```

Similar holds for [population count](#) with [SSE4](#) instruction if available versus [SWAR-popcount](#), also using an optimized version for populations below 16, borrowed from [Stockfish](#).

Search

RedQueen applies a [parallel search](#) considering the [Young Brothers Wait Concept](#) using a pool of [threads](#) where a master spawns idle threads. The [search](#) is [PVS alpha-beta](#) with the [shared transposition table](#) inside an [iterative deepening](#) framework with [aspiration windows](#). [Selectivity](#) is applied by [check](#) and [PV extensions](#), [adaptive nullmove pruning](#), [razoring](#), [futility pruning](#), [reductions](#) at [PV](#)- and more aggressively at none [PV](#)-nodes. The [SEE swap algorithm](#) is used to prune bad [captures](#) in [quiescence search](#), as well in [move ordering](#), which is further improved by the obligatory [killer](#)- and [history heuristics](#).

Evaluation

The [evaluation](#) might be [lazy](#) and otherwise considers [tactical](#) and positional features as well as [material imbalances](#) along with a [pawn structure cache](#) and a [tapered eval](#) interpolating [middlegame](#) and [endgame scores](#) with [passed pawn](#) and [king safety](#) and most dominant terms beside material.

Tournament Play

Over the board, RedQueen played the [ICT 2010](#), [DOCCC 2010](#), [ICT 2011](#), [ICT 2012](#), and [PT 52 CSVN](#) tournaments in [Leiden](#), online the [ACCA 2010](#), [ACCA 2011](#), [CCT13](#), [CCT14](#), and the [WCRCC 2011](#).

Selected Games

[ACCA 2011](#), round 2, [RedQueen](#) - [Crafty](#) ^[5]

```
[Event "ACCA 2011"]
[Site "HGM's server"]
[Date "2011.11.12"]
[Round "2"]
[White "RedQueen"]
[Black "Crafty"]
[Result "1/2-1/2"]
```

```
1.c4 e6 2.Nf3 d5 3.d4 Nf6 4.Nc3 Be7 5.Bf4 O-
O 6.e3 Nbd7 7.c5 Nh5 8.Bd3 Nxf4
9.exf4 c6 10.Qc2 g6 11.h4 Nf6 12.a3 b6 13.b4 Qc7 14.g3 a5 15.Na4 b5 16
.Nb6
Ra7 17.O-
O axb4 18.axb4 Rxa1 19.Rxa1 Bb7 20.Qe2 Nd7 21.Nxd7 Qxd7 22.Ne5 Qc7
23.h5 Bf6 24.Qg4 Re8 25.Ra7 Kf8 26.hxg6 hxg6 27.Nxf7 Qxf7 28.Bxg6 Qg7
29.Rxb7
Qxb7 30.Bxe8 Kxe8 31.Qxe6+ Be7 32.f5 Qd7 33.Qg8+ Bf8 34.Qg6+ Kd8 35.Qf
6+ Be7
36.Qh8+ Qe8 37.Qe5 Kd7 38.Qe6+ Kd8 39.Qe5 Kd7 40.Qe6+ Kd8 41.Qe5 1/2-1
/2
```

See also

- [Chess Pieces](#)
- [Comics](#)
- [Lewis Carroll characters](#)
- [Queen \(engine\)](#)

Forum Posts

2010

- [RedQueen 0.6](#) by [Ben-Hur Carlos Langoni](#), [CCC](#), August 21, 2010
- [Redqueen v0.7](#) by [Ben-Hur Carlos Langoni](#), [CCC](#), October 18, 2010
- [Redqueen v0.7.5 \(ACCA 2010\)](#) by [Ben-Hur Carlos Langoni](#), [CCC](#), November 08, 2010 » [ACCA 2010](#)
- [Redqueen v0.8](#) by [Ben-Hur Carlos Langoni](#), [CCC](#), December 01, 2010

- [Test position for average engines](#) by [Ben-Hur Carlos Langoni](#), [CCC](#), December 06, 2010 » [DOCCC 2010](#), [Fridolin](#)

2011

- [Redqueen v0.9 release](#) by [Ben-Hur Carlos Langoni](#), [CCC](#), January 31, 2011
- [Redqueen v0.9.1 release](#) by [Ben-Hur Carlos Langoni](#), [CCC](#), February 10, 2011
- [Redqueen v0.9.5 release](#) by [Ben-Hur Carlos Langoni](#), [CCC](#), February 21, 2011
- [RedQueen for Android](#) by [Aart Bik](#), [CCC](#), April 03, 2011
- [RedQueen v0.9.8 release](#) by [Ben-Hur Carlos Langoni](#), [CCC](#), April 27, 2011
- [2011 Fifth Annual ACCA WCRCC: RedQueen games](#) by [Ben-Hur Carlos Langoni](#), [CCC](#), July 23, 2011 » [WCRCC 2011](#)
- [RedQueen 1.0 beta release](#) by [Ben-Hur Carlos Langoni](#), [CCC](#), September 04, 2011
- [2011 6ht annual ACCA games \(RedQueen's games\)](#) by [Ben-Hur Carlos Langoni](#), [CCC](#), November 13, 2011 » [ACCA 2011](#)

2012 ...

- [RedQueen 1.1 released](#) by [Ben-Hur Carlos Langoni](#), [CCC](#), March 15, 2012
- [RedQueen 1.1.1 \(bugfix release\)](#) by [Ben-Hur Carlos Langoni](#), [CCC](#), March 15, 2012
- [RedQueen 1.1.1 on Mac OS X](#) by [Robert Purves](#), [CCC](#), March 16, 2012
- [Redqueen 1.1.1 64-bit - Is this a bug?](#) by [Gabor Szots](#), [CCC](#), March 26, 2012

External Links

Chess Engine

- [RedQueen Chess Engine website](#)
- [RedQueen | Chess Engine](#)
- [RedQueen 1.1.2 64-bit 2CPU](#) in [CCRL 40/4](#)
- [RedQueen 1.0.0 64-bit](#) in [CCRL 40/40](#)

Misc

- [Red Queen \(disambiguation\)](#) from Wikipedia
- [Red Queen \(Through the Looking-Glass\)](#) - Wikipedia
- [Red Queen's race](#) from Wikipedia
- [The Red Queen's Race](#) from Wikipedia ([Isaac Asimov](#))
- [Red Queen hypothesis](#) from Wikipedia
- [Red Queen \(comics\)](#) from Wikipedia
- [The Red Queen: Sex and the Evolution of Human Nature](#) - Wikipedia
- [Gryphon - Red Queen to Gryphon Three](#), 1974, [YouTube](#) Video

References

1. [^ RedQueen | Chess Engine](#)
2. [^](#) Illustration by [John Tenniel](#) of the [Red Queen](#) lecturing [Alice](#) for [Lewis Carroll's](#) "[Through The Looking Glass](#)", 1871, [Red Queen \(Through the Looking-Glass\) - Wikipedia](#)
3. [^ redqueen-1.1.2-windows.zip download](#), magicmoves.h, magicmoves.cpp
4. [^ redqueen-1.1.2-windows.zip download](#), bitboard.h
5. [^ 2011 6ht annual ACCA games \(RedQueen's games\)](#) by [Ben-Hur Carlos Langoni](#), [CCC](#), November 13, 2011

What links here?

Page	Date Edited
ACCA 2010	Jul 14, 2014
ACCA 2011	Jul 14, 2014
Ben-Hur Carlos Vieira Langoni Junior	Dec 17, 2013
BitScan	Sep 10, 2017
CCT13	Dec 6, 2013
CCT14	Aug 23, 2012
DOCCC 2010	Aug 15, 2015
Engines	Mar 10, 2018
Fridolin	Jun 27, 2017
Gavon	Apr 30, 2018
ICT 2010	Sep 14, 2015
ICT 2011	Mar 9, 2012
ICT 2012	Jun 11, 2012
PT 52	Nov 14, 2017
Queen (engine)	Nov 3, 2014
RedQueen	Nov 13, 2017
TCEC Season 5	Jun 2, 2014
TCEC Season 6	Dec 2, 2014
WCRCC 2011	Oct 21, 2014

[Up one Level](#)