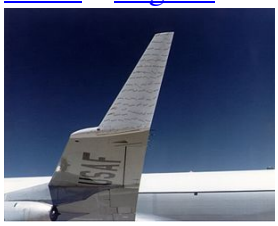
Home * Engines * **Winglet**



Winglet [5]

Winglet,
a didactic open source chess engines by Stef Luijten written in C++, licensed under the GNU General Public License. The development of Winglet was documented on the website tutorial *Winglet, Writing a Chess Program in 99 Steps*, started in 2011, now hosted by the Wayback Machine [1] . Winglet is intended as bitboard version of TSCP with WinBoard support [2], and is loosely derived from Wing, Stef Luijten's former private engine [3], in the meantime also open source [4].

Table of Contents

Description

Board Representation

Winglet applies a mixture of Kindergarten Bitboards and Magic Bitboards [6] to determine sliding piece attacks with 32 KiB precalculated lookup tables[64][64] each on ranks, files, diagonals and anti-diagonals, indexed by square and hashed line occupancy - the inner six bits multiplied by a magic factor and shifted right by the strange looking 57, while 58 is more natural to ensure a six bit index range, using a constant factor (b-File) for all squares of a diagonal or anti-diagonal, ...

```
U64 arrDiagonalAttacks[64][64]] /* requires initialization */

U64 diagonalKindergartenAttacks(U64 occ, enumSquare sq) {
   occ = (diagonalMaskEx[sq] & occ) * C64(0x0202020202020202) >> 58;
   return arrDiagonalAttacks[sq][occ];
}
```

... but "magic" Winglet factors are designed such that the most significant bit of the 64-bit product will always be clear, that is positive if interpreted as signed 64-bit integer. It seems, Winglet's occupied index calculations emulate Wing's rotated bitboard indices for same attack table layout:

```
/*                 Winglet's occupancy state             ==
 Wing's occupancy state */
(occ & MG_DIAGA8H1MASK[sq]) * MG_DIAGA8H1MAGIC[sq] >> 57 == (
occ045 >> DIAGA8H1_ATTACK_SHIFT[sq]) & 63
(occ & MG_DIAGA1H8MASK[sq]) * MG_DIAGA1H8MAGIC[sq] >> 57 == (
occ315 >> DIAGA1H8_ATTACK_SHIFT[sq]) & 63
(occ & MG_FILEMASK[sq])     * MG_FILEMAGIC[sq])    >> 57 == (
occ090 >> FILE_ATTACK_SHIFT[sq])     & 63
```

Search

- Iterative Deepening
- Alpha-Beta
- Principal Variation Search
- Quiescence Search
- MVV-LVA
- Static Exchange Evaluation
- Null Move Pruning

Evaluation

- Material Balance
- Piece-Square Tables
- Pawn Structure
  Passed Pawn
  Backward Pawn
  Doubled Pawn
  Isolated Pawn
- Bishop Pair
- Rook on Open File
- Rook behind Passers
- King Safety in Opening, Middlegame
  Pawn Shield
  King Tropism
- King Centralization in Endgame

See also

- Chesser by Syed Fahad [7]
- Godot by Ulysse Carion [8]
- Kenny by Kenshin Himura [9] [10]
- Vajolet by Marco Belli [11]
- Wing by Stef Luijten

Postings

- writing a chess engine in xx steps by wing, Winboard Forum, April 18, 2011
- Writing a chess program in xx steps by Stef Luijten, CCC, April 18, 2011
- Chess Programming/Concepts for Beginners by MoldyJacket, OpenChess Forum, April 18, 2011
- Winglet Chess Engine; Revived! by PortugalTheMan, Chess.com, Septermber 24, 2014 [12]

External Links

Chess Engine

- Index of /chess/engines/Jim Ablett/WINGLET by Jim Ablett, hosted by Kirill Kryukov
- PortugalTheMan/winglet · GitHub [13]
- winglet, writing a chess program in 99 steps by Stef Luijten, hosted by Abdullah Al-Ghaznawi [14]

Tutorial Archive

- Winglet, Writing a Chess Program in 99 Steps by Stef Luijten, hosted by the Wayback Machine
  01 Introduction - 05
  06 Reading user commands

07 Internal representation of the chess board - bitboards » Board Representation, Bitboards
08 Displaying the position » Chess Position
09 Reading a FEN string » Forsyth-Edwards Notation
10 Setting up the board manually
11 The move generator » Move Generation
12 Making the moves » Make Move
13 The evaluation function » Evaluation
14 Search » Search, Minimax, Alpha-Beta, PVS
15 Mate and draw detection » Checkmate, Stalemate
16 Repetition detection - Zobrist keys » Repetitions, Zobrist Keys
17 Iterative deepening and move ordering » Iterative Deepening, Move Ordering
18 Quiescence search and SEE » Quiescence Search, MVV-LVA, SEE
19 Null move pruning » Null Move Pruning
20 Time control and running test suites » Time Management
21 Connecting to Winboard » CECP, WinBoard

Misc

- Winglet from Wingtip device - Wikipedia
- Winglets and Sharklets | The Flying Engineer
- Boeing 737 Winglets
- Winglets | Airbus, a leading aircraft manufacturer

References

1. ^ Winglet, Writing a Chess Program in 99 Steps by Stef Luijten, hosted by the Wayback Machine
2. ^ Writing a chess program in xx steps by Stef Luijten, CCC, April 18, 2011
3. ^ Winglet, Writing a Chess Program in 99 Steps by Stef Luijten, hosted by the Wayback Machine
4. ^ Index of /chess/engines/Jim Ablett/WING by Jim Ablett, hosted by Kirill Kryukov
5. ^ Winglet with attached tufts of an KC-135A during NASA Winglet stu dy 1979. The tufts are needed to measure the airflow, Winglet from Wingtip device - Wikipedia, KC-135 EC79-11481: KC-135A in flight - closeup of winglet with attached tufts, August 20, 1979
6. ^ Writing a chess program in 99 steps - Move generation for sliding pieces by Stef Luijten, Wayback Machine
7. ^ Chesser: A Chess Engine derived from wingletx by Syed Fahad, CCC, December 24, 2014
8. ^ ucarion/godot · GitHub
9. ^ kenshinthebattosai/Kenny · GitHub
10. ^ New Winboard Engine 'Kenny' - JA builds available by Jim Ablett, CCC, January 08, 2013
11. ^ Re: where to start chess programming? by Marco Belli, CCC, June 22, 2014
12. ^ PortugalTheMan/winglet · GitHub
13. ^ Winglet Chess Engine; Revived! by PortugalTheMan, Chess.com, Septermber 24, 2014
14. ^ copy was taken in March 07, 2013, images and archives included

What links here?

Page	Date Edited
Abdullah Al-Ghaznawi	Apr 25, 2015
Engines	Mar 10, 2018
Godot	Apr 28, 2015
Hakkapeliitta	Apr 26, 2016
Kindergarten Bitboards	Aug 1, 2017
Little Wing	Oct 26, 2017
Marco Belli	Jan 6, 2018
Recommended Reading	Nov 17, 2017
Stef Luijten	Apr 26, 2015
Vajolet	Jan 25, 2018
Wing	Oct 26, 2017
Winglet	Apr 26, 2015

Up one Level