Table of Contents

Description

When communicating with the GUI, the Engine is supposed to use the algebraic game notation. The code on this page is used to translate the positions in the engine native form to the GUI friendly representation and back again.

The algebraic game notation first has 2 characters representing the square the piece comes from (eg.: "e2"). Then 2 characters to indicate the square the piece moves to (eg.: "e4"). And finally, in case of a promotion, the piece the pawn is promoted to (q | r | b | n). So a possible starting move for white could be "e2e4". And a possible promotion move could be "a7a8n".

Code

int convert_0x88_a(sq, a)

```
int convert_0x88_a(int sq, char * a) {
    a[0] = COL(sq) + 'a';
    a[1] = ROW(sq) + '1';
    a[2] = 0;
    return 0;
}
```

int convert_a_0x88(a)

```
int convert_a_0x88(char * a) {
    int sq = 0;
    sq = a[0] - 'a';
    sq += (a[1] - '1') * 16;

    return sq;
}
```

int algebraic_moves(a)

```
int algebraic_moves(char * a) {

    smoves m;

    while ((a[0] >= 'a') && (a[0] <= 'h')) {

        m.from = convert_a_0x88(a);
        m.to = convert_a_0x88(a+2);

        m.piece_from = b.pieces[m.from];
        m.piece_to = b.pieces[m.from];
        m.piece_cap = b.pieces[m.to];

        m.flags = 0;
        m.castle = 0;
        m.ep = 0;
        m.ply = 0;
        m.score = 0;

        switch (a[4]) {
        case 'q': m.piece_to = PIECE_QUEEN; a++; break;
        case 'r': m.piece_to = PIECE_ROOK; a++; break;
        case 'b': m.piece_to = PIECE_BISHOP; a++; break;
        case 'n': m.piece_to = PIECE_KNIGHT; a++; break;
        }

        //castling
        if ((m.piece_from == PIECE_KING) &&
            ((m.from == E1 && (m.to == G1 || m.to == C1)) ||
            (m.from == E8 && (m.to == G8 || m.to == C8)))) {
                m.flags = MFLAG_CASTLE;
        }
```

```
        /*ep
            if the moving-
piece is a Pawn, the square it moves to is empty and
            it was a diagonal move it has to be an en-passant capture.
        */
        if ((m.piece_from == PIECE_PAWN) &&
            (m.piece_cap == PIECE_EMPTY) &&
            ((abs(m.from-m.to)==15)||(abs(m.from-m.to)==17))) {
                m.flags = MFLAG_EPCAPTURE;
        }

        move_make(m);

        a += 4;

        while (a[0]==' ') a++;

    }

    return 0;
}
```

What links here?

Page	Date Edited
Algebraic Chess Notation	Sep 25, 2017
CPW-Engine	Dec 31, 2014
CPW-Engine_algebraic	May 15, 2011
Entering Moves	Sep 7, 2017

Up one Level