

[Home](#) \* [Engines](#) \* [CPW-Engine](#) \* **Book**

This is the [opening book](#) module of the CPW-Engine. Its responsibilities are:

- reading opening book in the text format (e2e4 a7a6-d2d4 b7b5 a2a4 c8b7 b1d2, where "-" means that preceding move is not to be played)
- filtering out illegal moves from the text file and warning user about them
- choosing the opening move out of possible alternatives (either completely [randomly](#) or based on frequency)

Obviously, initializing a book takes some time (a couple of seconds in case of 15.000 moves) and really efficient method would be to read and write opening book in the internal format. However, even in this case the code presented here, after some modification, would be useful for feeding moves into the opening book.

See also [CPW-Engine\\_book\\_h](#)

```
#include "stdafx.h"
#include "book.h"

#define BOOK_SIZE 36000
#define EXCLUDED -1

U64 start_hash;
int book_index;
int book_move_count;

struct sBook {
    U64 hash;
    smove move;
    int freq;
};

sBook book[BOOK_SIZE];

enum eproto {
    PROTO_NOTHING,
    PROTO_XBOARD,
    PROTO_UCI
} extern mode;

void initBook() {
```

```
    book_index = 0;
    book_move_count = 0;

// if we want to have internal book even though the book file is missing,
    // we might add a couple of lines like that

    book_loadInternal();
    readBookFile();
    board_loadFromFen(STARTFEN);
}

void readBookFile() {
    FILE *cpw_book;
    char line[256];
    int line_no = 0;

    /* if the book file does not exist, then exit */

    if ( (cpw_book = fopen("cpw_book.txt", "r")) == NULL ) {
        printf(
"Cannot open cpw_book.txt, only small internal book will be used \n");
        return;
    }

    /* process book file line by line */

    while ( fgets(line, 250, cpw_book) ) {
        ++line_no;
        if(line[0] == ';') continue;
// don't process comment lines
        book_addline( line_no, line );
    }

    printf("Loaded %d book moves\n", book_move_count);
}

void book_addline( int line_no, char * movestring ) {
    char local_string[5];

    board_loadFromFen( STARTFEN );
    int limit = (int) strlen( movestring );

    for (int i = 0; i < limit / 5 + 1; i++) {
```

```
local_string[0] = movestring[5*i];
local_string[1] = movestring[5*i+1];
local_string[2] = movestring[5*i+2];
local_string[3] = movestring[5*i+3];
local_string[4] = movestring[5*i+4];

/*****
 * On finding an illegal move in a book file, *
 * report line number and move sequence.      *
 *****/

if ( com_ismove( local_string ) ) {
    if ( !book_add( local_string ) ) {
        printf("book file error in line %d: ", line_no);
        printf(movestring);
        printf("\n");
    };
}
else
    return;
}

int book_add( char * movestring ) {

/*****
 * Don't record anything if a move has been already recorded. *
 * Make a move, because a new one is likely to be a couple   *
 * of steps further down the path, and exit.                  *
 *****/

if ( book_present( movestring ) ) {
    move_make( strToMove( movestring ) );
    return 1;
}

smove curr_move = strToMove( movestring );

if ( move_isLegal( curr_move ) ) {

    book[book_index].hash = b.hash;
    book[book_index].move = curr_move;
    ++book_move_count;

    if (movestring[4] == '-')
        book[book_index].freq = EXCLUDED;
}
```

```
        else
            book[book_index].freq = 1;

        ++book_index;

        move_make( strToMove( movestring ) );
        return 1;
    }
    else
        return 0;
}

int book_present( char * movestring ) {
    U64 curr_hash = b.hash;
    smove curr_move = strToMove( movestring );

    for (int i = 0; i < book_index; i++) {

        if ( book[i].hash == curr_hash &&
            book[i].move.from == curr_move.from &&
            book[i].move.to   == curr_move.to ) {

            /* see whether a move has been marked as excluded */
            if ( movestring[4] == '-' )
                book[i].freq = EXCLUDED;
            else {
                if ( book[i].freq > EXCLUDED )
                    ++book[i].freq;
            }

            return 1;
        }

    }

    return 0;
}

int book_getMaxFreq() {
    int result = 0;

    for (int i = 0; i < BOOK_SIZE; i++) {
        if ( book[i].hash == b.hash ) {
            if ( book[i].freq > result )
                result = book[i].freq;
        }
    }
}
```

```
    }

    return result;
}

int getBookMove( int book_type ) {
    int result = 0;
    int priority = EXCLUDED;
    int temp;
    smove book_move;
    char move_string[6];
    int max_freq = book_getMaxFreq();

    if ( book_type == BOOK_NONE ) return 0;

    srand( gettime() );

    for (int i = 0; i < BOOK_SIZE; i++) {
        if ( book[i].hash == b.hash ) {

            /*****
            * Debug code, showing possible choices as well as      *
            * move frequencies (console mode only)                  *
            *****/

            if ( mode == PROTO_NOTHING ) {

                convert_0x88_a( book[i].move.from, move_string );
                convert_0x88_a( book[i].move.to   , move_string+2 );
                printf( move_string );

                if ( book[i].freq == EXCLUDED )
                    printf("?");
                else
                    printf("(%d)", book[i].freq);

                printf(" ");
            }

            /*****
            * Get the value used to decide which move to play. *
            * If book is meant to be broad, it will be a random *
            * value, distributed evenly between moves that are *
            * not excluded. When using narrow book, frequency *
            * of a move will be taken into account.             *
            *****/
        }
    }
}
```

```
    if ( book_type == BOOK_NARROW )
        temp = ( abs( rand() ) / max_freq ) * ( book[i].freq );
    else
        temp = abs( rand() );

    /*****
    * If new move is backed up by a greater number,
    * it replaces the old one.
    *****/

    if ( ( temp > priority ) &&
        ( book[i].freq != EXCLUDED )
        ) {
        priority = temp;
        book_move = book[i].move;
        result = 1;
    }
}

printf("\n");
if (result) com_sendmove( book_move );

return result;
}

void book_loadInternal() {

    // Four knights

    book_addline(0,
    "e2e4 e7e5 b1c3 g8f6 g1f3 b8c6 f1b5 f8b4 e1g1 e8g8 d2d3 d7d6 c1g5 b4c3
    b2c3 d8e7 f1e1 c6d8 d3d4 d8e6");

    // Italian

    book_addline(0,
    "e2e4 e7e5 g1f3 b8c6 f1c4 f8c5 c2c3 g8f6 d2d4 e5d4 c3d4 c5b4 c1d2 b4d2
    b1d2 d7d5");
    book_addline(0,
    "e2e4 e7e5 g1f3 b8c6 f1c4 f8c5 c2c3 g8f6 d2d3 d7d6 b2b4 c5b6 a2a4 a7a5
    b4b5 c6e7");
    book_addline(0, "e2e4 e7e5 g1f3 b8c6 f1c4 f8c5 d2d3 g8f6 c2c3");

    // Two knights
```

```
    book_addline(0,
    "e2e4 e7e5 g1f3 b8c6 f1c4 g8f6 f3g5 d7d5 e4d5 c6a5 c4b5 c7c6 d5c6 b7c6
    b5e2 h7h6 g5f3 e5e4 f3e5 f8d6 f2f4 e4f3 e5f3 e8g8 d2d4 c6c5");
    book_addline(0,
    "e2e4 e7e5 g1f3 b8c6 f1c4 g8f6 d2d4 e5d4 e1g1 f6e4 f1e1 d7d5 c4d5 d8d5
    b1c3 d5h5 c3e4 c8e6");
    book_addline(0,
    "e2e4 e7e5 g1f3 b8c6 f1c4 g8f6 d2d3 f8e7 e1g1 e8g8 c2c3 d7d6 c4b3");

    // Scotch

    book_addline(0,
    "e2e4 e7e5 g1f3 b8c6 d2d4 e5d4 f3d4 g8f6 d4c6 b7c6 e4e5 d8e7 d1e2 f6d5
    c2c4 c8a6 g2g3 g7g6 b2b3 f8g7 c1b2 e8g8 f1g2 a8e8 e1g1 ");
    book_addline(0,
    "e2e4 e7e5 g1f3 b8c6 d2d4 e5d4 f3d4 f8c5 c1e3 d8f6 c2c3 g8e7 g2g3 e8g8
    f1g2");

    // Ruy Lopez

    book_addline(0,
    "e2e4 e7e5 g1f3 b8c6 f1b5 a7a6 b5a4 g8f6 e1g1 f8e7 f1e1 b7b5 a4b3 d7d6
    c2c3 e8g8 h2h3 c6a5 b3c2 c7c5 d2d4 d8c7");
    book_addline(0,
    "e2e4 e7e5 g1f3 b8c6 f1b5 f8c5 c2c3 g8f6 e1g1 e8g8 d2d4 c5b6 f1e1 d7d6
    h2h3 c6e7 b1d2");
    book_addline(0,
    "e2e4 e7e5 g1f3 b8c6 f1b5 g8f6 e1g1 f8c5 f3e5 c6e5 d2d4 c7c6 d4e5 f6e4
    b5d3 d7d5 e5d6 e4f6 f1e1");
    book_addline(0,
    "e2e4 e7e5 g1f3 b8c6 f1b5 g8f6 e1g1 f6e4 d2d4 f8e7 d1e2 e4d6 b5c6 b7c6
    d4e5 d6b7 b1c3 e8g8 f3d4");
    book_addline(0,
    "e2e4 e7e5 g1f3 b8c6 f1b5 a7a6 b5a4 g8f6 e1g1 f6e4 d2d4 b7b5 a4b3 d7d5
    d4e5 c8e6 c2c3 e4c5 b3c2 e6g4");
    book_addline(0,
    "e2e4 e7e5 g1f3 b8c6 f1b5 a7a6 b5a4 g8f6 e1g1 f8e7 f1e1 b7b5 a4b3 d7d6
    c2c3 e8g8 h2h3 c8b7 d2d4 f8e8 b1d2 e7f8 a2a3 h7h6 b3c2 c6b8 b2b4 b8d7
    c1b2 g7g6");

    // Petroff

    book_addline(0,
    "e2e4 e7e5 g1f3 g8f6 f3e5 d7d6 e5f3 f6e4 d2d4 d6d5 f1d3 b8c6 e1g1 c8g4
    c2c4 e4f6");
```

```
    book_addline(0,
    "e2e4 e7e5 g1f3 g8f6 f3e5 d7d6 e5f3 f6e4 d2d4 d6d5 f1d3 b8c6 e1g1 f8e7
    c2c4 c6b4");
    book_addline(0,
    "e2e4 e7e5 g1f3 g8f6 f3e5 d7d6 e5f3 f6e4 b1c3 e4c3 d2c3");
    book_addline(0,
    "e2e4 e7e5 g1f3 g8f6 d2d4 f6e4 f1d3 d7d5 f3e5 b8d7 e5d7 c8d7 e1g1 f8d6
    c2c4 c7c6 b1c3 e4c3 b2c3");

    // Sicilian

    book_addline(0,"e2e4 c7c5 c2c3 d7d5 e4d5 d8d5 d2d4 e7e6");
    book_addline(0,
    "e2e4 c7c5 c2c3 g8f6 e4e5 f6d5 d2d4 c5d4 g1f3 e7e6 c3d4 b7b6 b1c3 d5c3
    b2c3 d8c7");
    book_addline(0,
    "e2e4 c7c5 g1f3 d7d6 d2d4 c5d4 f3d4 g8f6 b1c3 a7a6 f1e2 e7e5 d4b3 f8e7
    e1g1 e8g8 a2a4 b7b6");
    book_addline(0,
    "e2e4 c7c5 b1c3 b8c6 g2g3 g7g6 f1g2 f8g7 d2d3 e7e6 c1e3 d7d6 g1e2 c6d4
    d1d2");
    book_addline(0,
    "e2e4 c7c5 g1f3 d7d6 d2d4 c5d4 f3d4 g8f6 b1c3 a7a6 c1g5 e7e6 d1d2 f8e7
    e1c1 e8g8");
    book_addline(0,
    "e2e4 c7c5 g1f3 d7d6 d2d4 c5d4 f3d4 g8f6 b1c3 a7a6 g2g3 e7e5 d4e2 b7b5
    f1g2 c8b7 e1g1 b8d7");
    book_addline(0,
    "e2e4 c7c5 g1f3 b8c6 d2d4 c5d4 f3d4 g7g6 b1c3 f8g7 c1e3 g8f6 f1c4 e8g8
    ");
    book_addline(0,
    "e2e4 c7c5 g1f3 b8c6 d2d4 c5d4 f3d4 g8f6 b1c3 d7d6 f1e2 e7e5 d4b3 f8e7
    e1g1 e8g8 c1e3 c8e6");
    book_addline(0,
    "e2e4 c7c5 g1f3 b8c6 d2d4 c5d4 f3d4 g8f6 b1c3 d7d6 f1e2 g7g6 c1e3 f8g7
    e1g1 e8g8 d4b3 c8e6");

    // French

    book_addline(0,
    "e2e4 e7e6 d2d4 d7d5 e4e5 c7c5 c2c3 b8c6 g1f3 d8b6 a2a3 c5c4");
    book_addline(0,
    "e2e4 e7e6 d2d4 d7d5 b1c3 g8f6 c1g5 f8e7 e4e5 f6d7 g5e7 d8e7 f2f4 e8g8
    d1d2 c7c5 g1f3 b8c6 e1c1 c5c4");
    book_addline(0,
    "e2e4 e7e6 d2d4 d7d5 b1c3 d5e4 c3e4 b8d7 g1f3 g8f6 e4f6 d7f6 f1d3 b7b6
```

```
d1e2 c8b7? c1g5 f8e7");
    book_addline(0,
    "e2e4 e7e6 d2d4 d7d5 b1c3 f8b4 e4e5 g8e7 a2a3 b4c3 b2c3 c7c5 g1f3 b8c6
    a3a4 d8a5 d1d2 c8d7");
    book_addline(0,
    "e2e4 e7e6 d2d4 d7d5 b1c3 f8b4 e4e5 g8e7 a2a3 b4c3 b2c3 c7c5 a3a4 b8c6
    g1f3 d8a5 c1d2 c8d7");
    book_addline(0,
    "e2e4 e7e6 d2d4 d7d5 b1c3 f8b4 e4e5 c7c5 a2a3 b4c3 b2c3 g8e7 d1g4 d8c7
    g4g7 h8g8 g7h7 c5d4 g1e2 b8c6 f2f4 c8d7");
    book_addline(0,
    "e2e4 e7e6 d2d4 d7d5 b1d2 c7c5 e4d5 e6d5 g1f3 b8c6 f1b5 f8d6 d4c5 d6c5
    e1g1 g8e7");
    book_addline(0,
    "e2e4 e7e6 d2d4 d7d5 b1d2 c7c5 g1f3 g8f6 e4d5 e6d5 f1b5 c8d7 b5d7 b8d7
    e1g1 f8e7");
    book_addline(0,
    "e2e4 e7e6 d2d4 d7d5 b1d2 g8f6 e4e5 f6d7 f1d3 c7c5 c2c3 b8c6 g1e2 c5d4
    c3d4 f7f6 e5f6 d7f6 e1g1 f8d6");

    // Caro-Kann

    book_addline(0,
    "e2e4 c7c6 d2d4 d7d5 b1c3 d5e4 c3e4 b8d7 g1f3 g8f6 e4f6 d7f6 f3e5");
    book_addline(0,
    "e2e4 c7c6 b1c3 d7d5 d2d4 d5e4 c3e4 b8d7 f1c4 g8f6 e4g5 e7e6 d1e2 d7b6
    ");
    book_addline(0,
    "e2e4 c7c6 b1c3 d7d5 d2d4 d5e4 c3e4 b8d7 e4g5 g8f6 f1d3 e7e6");
    book_addline(0,
    "e2e4 c7c6 d2d4 d7d5 b1c3 d5e4 c3e4 c8f5 e4g3 f5g6 h2h4 h7h6 g1f3 g8f6
    -f3e5 g6h7 f1d3 b8d7 d3h7");
    book_addline(0,
    "e2e4 c7c6 d2d4 d7d5 b1c3 d5e4 c3e4 c8f5 e4g3 f5g6 h2h4 h7h6 g1f3 b8d7
    h4h5 g6h7 f1d3 h7d3 d1d3 g8f6 c1d2 e7e6 e1c1");
    book_addline(0,
    "e2e4 c7c6 d2d4 d7d5 b1c3 d5e4 c3e4 c8f5 e4g3 f5g6 h2h4 h7h6 g1f3 b8d7
    h4h5 g6h7 f1d3 h7d3 d1d3 e7e6 c1d2 g8f6 e1c1");
    book_addline(0,
    "e2e4 c7c6 d2d4 d7d5 b1c3 d5e4 c3e4 c8f5 e4g3 f5g6 g1f3 b8d7 h2h4 h7h6
    f1d3 g6d3 d1d3 e7e6 c1d2 g8f6 e1c1");
    book_addline(0, "e2e4 c7c6 d2d4 d7d5 b1d2 d5e4 d2e4");
    book_addline(0,
    "e2e4 c7c6 d2d4 d7d5 e4d5 c6d5 c2c4 g8f6 b1c3 e7e6 g1f3");
    book_addline(0,
    "e2e4 c7c6 d2d4 d7d5 e4d5 c6d5 c2c4 g8f6 b1c3 b8c6 c1g5 e7e6 c4c5 f8e7
```

```
f1b5 e8g8 g1f3 f6e4");
    book_addline(0,
    "e2e4 c7c6 d2d4 d7d5 e4e5 c8f5 f1d3 f5d3 d1d3 e7e6 b1c3 d8b6");
    book_addline(0,
    "e2e4 c7c6 b1c3 d7d5 g1f3 c8g4 h2h3 g4f3 d1f3 e7e6 d2d4 g8f6 f1d3 d5e4
    c3e4 d8d4 c2c3 d4d8");

    // Pirc and modern

    book_addline(0,
    "d2d4 d7d6 e2e4 g8f6 b1c3 g7g6 f1c4 c7c6 d1e2 f8g7 g1f3 e8g8 c1g5 b7b5
    c4d3 d8c7");
    book_addline(0,
    "e2e4 d7d6 d2d4 g8f6 b1c3 g7g6 c1g5 f8g7 d1d2 b8d7 e1c1 e7e5 d4e5 d6e5
    g1f3 h7h6 g5h4 g6g5 h4g3 d8e7");

    // QGA

    book_addline(0,
    "d2d4 d7d5 c2c4 d5c4 g1f3 g8f6 e2e3 e7e6 f1c4 c7c5 e1g1 a7a6 d1e2 b7b5
    c4d3 c5d4 e3d4 b8c6");

    // QGD

    book_addline(0,
    "d2d4 d7d5 c2c4 e7e6 b1c3 f8e7 g1f3 g8f6 c4d5 e6d5");
    book_addline(0,
    "c2c4 e7e6 d2d4 d7d5 b1c3 c7c5 c4d5 e6d5 g1f3 b8c6 g2g3 g8f6 f1g2 f8e7
    e1g1 e8g8");
    book_addline(0,
    "c2c4 e7e6 b1c3 d7d5 d2d4 g8f6 c1g5 f8e7 e2e3 e8g8 g1f3 b8d7 a1c1 c7c6
    ");
    book_addline(0,
    "d2d4 d7d5 c2c4 e7e6 b1c3 g8f6 c1g5 b8d7 c4d5 e6d5 e2e3 c7c6 f1d3 f8e7
    d1c2 e8g8 g1e2 f8e8");
    book_addline(0,
    "d2d4 d7d5 c2c4 e7e6 b1c3 g8f6 c1g5 b8d7 e2e3 c7c6 g1f3 d8a5 f3d2 f8b4
    d1c2 e8g8 g5h4 c6c5");
    book_addline(0,
    "d2d4 d7d5 c2c4 e7e6 b1c3 g8f6 g1f3 f8e7 c4d5 e6d5 c1g5 e8g8");

    // Slav

    book_addline(0,
    "d2d4 d7d5 c2c4 c7c6 b1c3 g8f6 g1f3 d5c4 a2a4 c8f5 f3e5 e7e6 f2f3 f8b4
    c1g5 h7h6 g5f6 d8f6 e2e4 f5h7");
```

```
    book_addline(0,
"d2d4 d7d5 c2c4 c7c6 g1f3 g8f6 b1c3 e7e6 e2e3 b8d7 f1d3 f8d6");

    // Catalan

    book_addline(0,
"d2d4 e7e6 c2c4 d7d5 g2g3 g8f6 g1f3 f8e7 f1g2 e8g8 e1g1 f6d7 d1c2 c7c6
b1d2 b7b6 e2e4 c8b7");

    // Nimzo-Indian

    book_addline(0,
"d2d4 g8f6 c2c4 e7e6 b1c3 f8b4 d1c2 c7c5 d4c5 e8g8 a2a3 b4c5 g1f3 b7b6
");
    book_addline(0,
"d2d4 g8f6 c2c4 e7e6 b1c3 f8b4 d1c2 e8g8 a2a3 b4c3 c2c3 b7b6 c1g5 c8b7
");
    book_addline(0,
"d2d4 g8f6 c2c4 e7e6 b1c3 f8b4 g1f3 b7b6 g2g3 c8b7 f1g2");
    book_addline(0,
"d2d4 g8f6 c2c4 e7e6 b1c3 f8b4 a2a3 b4c3 b2c3 e8g8 f2f3 d7d5 c4d5 e6d5
e2e3 c8f5 g1e2 b8d7 e2g3 f5g6");
    book_addline(0,
"d2d4 g8f6 c2c4 e7e6 b1c3 f8b4 c1d2 e8g8 e2e3 d7d5 g1f3 c7c5 a2a3 b4c3
d2c3 f6e4 a1c1 e4c3 c1c3 c5d4");
    book_addline(0,
"d2d4 g8f6 c2c4 e7e6 b1c3 f8b4 e2e3 e8g8 f1d3 d7d5 g1f3 c7c5 e1g1 b8c6
a2a3 b4c3 b2c3 d5c4 d3c4 d8c7");
    book_addline(0,
"d2d4 g8f6 c2c4 e7e6 b1c3 f8b4 d1c2 d7d5 a2a3 b4c3 c2c3 b8c6 g1f3 f6e4
c3b3 c6a5 b3a4 c7c6");

    // Queen's Indian

    book_addline(0,
"d2d4 g8f6 c2c4 e7e6 g1f3 b7b6 g2g3 c8b7 f1g2 f8e7 e1g1 e8g8 b1c3 f6e4
d1c2 e4c3 c2c3");
    book_addline(0,
"d2d4 g8f6 c2c4 e7e6 g1f3 b7b6 e2e3 c8b7 f1d3 f8e7 b1c3 d7d5 e1g1 e8g8
d1e2 b8d7");

    // King's Indian

    book_addline(0,
"d2d4 g8f6 c2c4 g7g6 b1c3 f8g7 e2e4 d7d6 f2f3 e8g8 c1e3 e7e5 d4d5 f6h5
d1d2 f7f5 e1c1 b8d7");
```

```
    book_addline(0,
"d2d4 g8f6 c2c4 g7g6 b1c3 f8g7 e2e4 d7d6 g1f3 e8g8 f1e2 e7e5 d4d5 a7a5
");
    book_addline(0,
"d2d4 g8f6 c2c4 g7g6 g2g3 f8g7 f1g2 e8g8 b1c3 d7d6 g1f3 b8d7 e1g1 e7e5
e2e4 c7c6 h2h3 d8b6");
    book_addline(0,
"d2d4 g8f6 c2c4 g7g6 b1c3 f8g7 e2e4 d7d6 f2f4 c7c5 g1f3 e8g8 d4d5 e7e6
f1d3 e6d5 c4d5 d8b6");
    book_addline(0,
"d2d4 g8f6 c2c4 g7g6 b1c3 f8g7 e2e4 d7d6 g1f3 e8g8 f1e2 e7e5 e1g1 b8c6
d4d5 c6e7 f3e1 f6e8 f2f3 f7f5");
    book_addline(0,
"d2d4 g8f6 c2c4 g7g6 b1c3 f8g7 g1f3 e8g8 c1f4 d7d6 h2h3 b8d7 e2e3 c7c6
");

    // Grunfeld

    book_addline(0,"d2d4 g8f6 c2c4 g7g6 b1c3 d7d5 c1f4 f8g7");
    book_addline(0,
"d2d4 g8f6 c2c4 g7g6 b1c3 d7d5 c4d5 f6d5 e2e4 d5c3 b2c3 c7c5 f1c4 f8g7
gle2 e8g8 e1g1 c5d4 c3d4 b8c6");
    book_addline(0,
"d2d4 g8f6 c2c4 g7g6 b1c3 d7d5 g1f3 f8g7 d1b3 d5c4 b3c4 e8g8 e2e4 c8g4
cle3 f6d7 e1c1 b8c6");

    // Benoni

    book_addline(0,
"d2d4 g8f6 c2c4 c7c5 d4d5 e7e6 b1c3 e6d5 c4d5 d7d6 e2e4 g7g6 f1d3 f8g7
gle2 e8g8 e1g1 a7a6 a2a4 d8c7");

    // Dutch

    book_addline(0,
"d2d4 f7f5 g2g3 e7e6 f1g2 g8f6 g1f3 f8e7 e1g1 e8g8 c2c4 d7d6 b1c3 d8e8
d1c2 e8h5 b2b3 b8c6 cla3 a7a5");

    // Queen's Pawn

    book_addline(0,
"d2d4 d7d5 g1f3 g8f6 c1f4 c7c5 e2e3 b8c6 c2c3 d8b6 d1c1 c8f5 d4c5 b6c5
b1d2 a8c8 f3d4 c6d4 e3d4 c5b6");

    // English
```

```
    book_addline(0,
"c2c4 e7e5 b1c3 g8f6 g1f3 b8c6 e2e4 f8b4 d2d3 d7d6 f1e2 e8g8 e1g1 b4c3
b2c3 d8e7");
    book_addline(0,
"c2c4 e7e5 b1c3 g8f6 g1f3 b8c6 g2g3 d7d5 c4d5 f6d5 f1g2 d5b6 e1g1 f8e7
d2d3 e8g8 c1e3 f7f5");

    book_addline(0,
"c2c4 g8f6 b1c3 d7d5 c4d5 f6d5 e2e4 d5f4 f1c4 c8e6 c4e6 f7e6");
    book_addline(0,
"c2c4 g8f6 b1c3 e7e5 g1f3 b8c6 g2g3 f8c5 f1g2 d7d6 e1g1 e8g8 d2d3 h7h6
");
    book_addline(0,
"c2c4 g8f6 b1c3 e7e5 g1f3 b8c6 g2g3 f8b4 f1g2 e8g8 e1g1 e5e4 f3e1 b4c3
d2c3 h7h6 e1c2 b7b6");

    book_addline(0,
"c2c4 c7c5 g1f3 b8c6 b1c3 g8f6 g2g3 g7g6 f1g2 f8g7 e1g1 e8g8 d2d4 c5d4
f3d4 c6d4 d1d4 d7d6 d4d3");

    // Reti
    book_addline(0,
"g1f3 d7d5 g2g3 g8f6 f1g2 g7g6 e1g1 f8g7 d2d3 e8g8 b1d2 b8c6 e2e4 e7e5
c2c3 a7a5 f1e1 d5e4 d3e4");
    book_addline(0,
"g1f3 d7d5 c2c4 e7e6 g2g3 g8f6 f1g2 f8e7 e1g1 e8g8 b2b3 c7c5 c4d5 f6d5
c1b2 b8c6 d2d4 b7b6 b1c3 d5c3");
    book_addline(0,
"g1f3 d7d5 c2c4 d5c4 e2e3 c7c5 f1c4 e7e6 e1g1 g8f6 b2b3 b8c6 c1b2 a7a6
a2a4 f8e7");
}
```

## What links here?

Page

[CPW-Engine](#)

[CPW-Engine\\_book](#)

[CPW-Engine\\_book\\_h](#)

[Opening Book](#)

[Pseudorandom number generator](#)

Date Edited

Dec 31, 2014

May 8, 2017

Sep 27, 2008

Feb 26, 2018

May 11, 2017

[Up one Level](#)