BSC (b0n0 Simple Chess)

Nicolas ROBERT, Développeur de BSC
<nicolas.robert@tangentoide.net>

Mai 2008

Table des matières

Introduction 1
Tournoi Blitz rapide v3.0 - Partie en 5 minutes 2
Tournoi Blitz rapide v2 2
Résultats LCTII 4
Résultats BT2630 5
Téléchargement 6
Futures améliorations à apporter 6
Historique 6
3.2 du 23/04/2008 6
3.1b du 07/04/2008 6
3.0 du 30/03/2008 7
2.8 du 01/02/2003 7
2.72 7
2.71 7
2.7 7
2.6 7
2.5 8
2.4 8
2.3 8
2.2 8
2.1 8
2.0 8
1.9 8
1.8 9
1.7 9
1.5 9
1.2 9
1.1 9
Utilisation 10
Les fichiers 10
Le fichier bsc.rc 10
Le fichier bsc.opn 10
Le mode console 11
Le fichier d'apprentissage 13
Le mode xboard avec Winboard 13

Introduction

BSC est un programme d'échecs implanté de façon classique (en C) et de force moyenne. Ses premières versions sont fortement inspirées du programme *TSCP* (Tom Kerrigan Simple Chess Program). L'échiquier est représenté par deux tableaux de 64 cases, un déterminant la couleur de la pièce (NOIR, BLANC et VIDE), l'autre le type de pièce (ROI, DAME ... VIDE). Le déplacement des pièces est validé par des *mailbox*. La force du programme est estimée à environ 1900/2100 ELO sur un PII450 (100000 positions/sec analysées en moyenne). Selon mon analyse, le programme n'est pas brillant lors

de l'ouverture du jeu: en effet, les lignes de la bibliothèque d'ouverture ne sont pas les meilleurs et sont choisies aléatoirement (pas de score ou de statistiques liés à l'ouverture choisie) Cependant, il fonctionne correctement avec l'interface Arena et peu utiliser des bibliothèques d'ouvertures meilleurs intégrées dans Arena (ex. Frank Quisinsky Opening Book). En milieu de partie, le programme joue de façon correcte et peut trouver des combinaisons gagnantes intéressantes. Son jeu est globalement intéressant et convient au joueurs de club intermediaire. En fin de partie, il est assez fort et utilise les tables de fin partie 4 ou 5 pions.

Tournoi Blitz rapide v3.0 - Partie en 5 minutes

Les fonctions razoring et futility pruning désactivé (no ponder, 16MB Hash, 512kB Hash Pawn). Les tests sont réalisés sur un athlon XP 2400 avec Arena.

```
1   b0n0 Simple Chess 3.0  1700  1011011011101110101101111½011   21.0/30 +147
2   Tscp                   1700  0100100100010001010011½0000½100    9.0/30 -147
Estimation min: ~ 1800 Elo
```

Les fonctions razoring et futility pruning sont activées (no ponder, 16MB Hash, 512kB Hash Pawn)

```
-----------------b0n0 Simple Chess 3.0-----------------
b0n0 Simple Chess 3.0 - Tscp : 23,0/30 21-5-4 (=1011111100111=11111111110=1=0)
-----------------Tscp-----------------
Tscp - b0n0 Simple Chess 3.0 : 7,0/30 5-21-4 (=0100000011000=00000000001=0=1)  1

Estimation min: ~ 1860 Elo
```

Les fonctions razoring, futility pruning et nullmove sont activées (no ponder, 16MB Hash, 512kB Hash Pawn)

```
-----------------b0n0 Simple Chess 2.8-----------------
b0n0 Simple Chess 2.8 - b0n0 Simple Chess 3.2 : 23,0/66 17-37-12 (0=00=100001==0
-----------------b0n0 Simple Chess 3.2-----------------
b0n0 Simple Chess 3.2 - b0n0 Simple Chess 2.8 : 43,0/66 37-17-12 (1=11=011110==1


-----------------b0n0 Simple Chess 3.1-----------------
b0n0 Simple Chess 3.1 - Faile : 3,0/12 1-7-4 (0==000=001=0)  25%  -191
b0n0 Simple Chess 3.1 - Tscp  : 10,5/12 10-1-1 (111=11101111)  88%  +346
-----------------Faile-----------------
Faile - b0n0 Simple Chess 3.1 : 9,0/12 7-1-4 (1==111=110=1)  75%  +191
Faile - Tscp                  : 10,5/12 10-1-1 (11111101=111)  88%  +346
-----------------Tscp-----------------
Tscp - b0n0 Simple Chess 3.1  : 1,5/12 1-10-1 (000=00010000)  13%  -330
Tscp - Faile                  : 1,5/12 1-10-1 (00000010=000)  13%  -330

Estimation min: ~ 1750 Elo
Estimation max: ~ 2000 Elo
```

Tournoi Blitz rapide v2

Pour comparer avec d'autres engines de sa catégorie sous winboard, voici le tableau final du tournoi blitz (1'): BSC (16 MB Hash, !mtdf, !null_move, !futility/razoring).

Exemple 1. Tournoi blitz 1 min, 84 jeux

```
------------------------------------------
   Engine      Score    1  2  3  4  5  6  7
------------------------------------------
1. BSC 2.0     10,0/12 ## 10 01 11 11 11 11
2. UFIM 1.43    8,5/12 01 ## 00 11 1= 11 11
3. TSCP 1.73    7,5/12 10 11 ## 00 10 =1 11
4. zephyr 0.61  6,5/12 00 00 11 ## 01 =1 11
5. UFIM 2.00    6,5/12 00 0= 01 10 ## 11 11
6. Adam 1.1     3,0/12 00 00 =0 =0 00 ## 11
7. List 4.61    0,0/12 00 00 00 00 00 00 ##
```

Exemple 2. Elo rating

```
Rating type: GLICKO
Initial rating: 2000
Initial RD: 350

-------------------------------------------------------------------------------
Place Engine                    Rating    RD Games Wins Draws Points Score %
-------------------------------------------------------------------------------
   1. BSC                         2318 134,1    12   10     0   10,0   83,33
   2. UFIM                        2100  96,6    24   14     2   15,0   62,50
   3. zephyr                      2068 130,9    12    6     1    6,5   54,17
   4. TSCP                        2057 128,1    12    7     1    7,5   62,50
   5. Adam                        1817 119,7    12    2     2    3,0   25,00
   6. List                        1569 167,2    12    0     0    0,0    0,00
```

Etant donnée les résultats, BSC est de la même catégorie d'engine qu'*UFIM*. Voici un tournoi blitz entre différentes version d'*UFIM* et BSC.

Exemple 3. Tournoi blitz 1 min, 60 jeux

```
--------------------------------------------------
   Engine    Score            1          2          3
--------------------------------------------------
1. BSC 2.0   15,5/20 ########## =1=111101= 11110111==
2. UFIM 1.43  8,0/20 =0=000010= ########## =011100011
3. UFIM 2.00  6,5/20 00001000== =100011100 ##########
```

Exemple 4. Elo Rating blitz (1min)

```
Rating type: GLICKO
Initial rating: 2000
Initial RD: 350

-------------------------------------------------------------------------------
Place Engine                      Rating    RD Games Wins Draws Points Score %
-------------------------------------------------------------------------------
   1. BSC                           2066 104,6    22   15     5   17,5   79,55
   2. UFIM                          1870  74,5    46   16     8   20,0   43,48
```

Résultats LCTII

Les résultats au test LCTII (niveau tournoi, 10 minutes par problème)

Tableau 1. LCTII Test

2.0	Found Time (sec)	Score (+Elo)	2.2	Found Time (sec)	Score (+Elo)	3.0 (Athlon XP 2400, 16MB Hash, 512KB HASHPAWN, no selective)	Found Time (sec)	Score (+Elo)	3.1b (Athlon XP 2400, 16MB Hash, 512KB HASHPAWN, no selective)	Found Time (sec)	3.2 (Athlon XP 2400, 16MB Hash, 512KB HASHPAWN, no selective)	Found Time (sec)
			POS1	379	10	POS1	32	20	POS1	45	POS1	61
POS2	32	20	POS2	0	30	POS2	0	30	POS2	0	POS2	0
			POS4	56	20	POS4	25	25	POS4	29	POS4	21
POS8	290	10										
POS9	542	5										
POS12	335	10										
POS13	0	30	POS13	0	30							
			CMB1	45	20	CMB1	4	30	CMB1	3	CMB1	3
CMB2	248	10	CMB2	1	30	CMB2	8	30	CMB2	11	CMB2	9
CMB3	12	25	CMB3	10	25	CMB3	6	30	CMB3	6	CMB3	5
CMB4	3	30	CMB4	2	30	CMB4	348	10	CMB4	-	CMB4	-
CMB5	27	25	CMB5	0	30	CMB5	8	30	CMB5	9	CMB5	5
			CMB6	341	10	CMB6	160	15	CMB6	113	CMB6	77
			CMB7	39	20	CMB7	41	20	CMB7	35	CMB7	26
			CMB8	166	15	CMB8	17	25	CMB8	15	CMB8	13
CMB9	518	5	CMB9	8	30	CMB9	1	30	CMB9	1	CMB9	0
			CMB10	120	15	CMB10	143	15	CMB10	105	CMB10	81
FIN1	8	30	FIN1	0	30	FIN1	2	30	FIN1	1	FIN1	1
FIN2	28	25	FIN2	0	30	FIN2	0	30	FIN2	0	FIN2	0
FIN3	4	30	FIN3	0	30	FIN3	131	15	FIN3	120	FIN3	136
FIN4	13	25	FIN4	4	30	FIN4	242	10	FIN4	59	FIN4	50
FIN6	93	15										

Les tests sur BSC 2.0 sont effectués sur un PII 450 Mhz, 16 MB Hash. Les tests sur BSC 2.2 sont effectués sur un P4 1.8Ghz, 16 MB Hash. Les autres positions ne sont pas résolues par le programme. D'après le test LCTII, il apparaît clairement que BSC n'est pas très positionnel comme programme. Au niveau combinatoire, il se tient correctement, sachant que le programme n'est pas trop rapide. Son point fort est les fins de partie.

BSC 2.0

- POS = 75
- CMB = 95
- FIN = 125

Total : 1900 + 75 + 95 + 125 = 2195 ELO

BSC 2.2

- POS = 60
- CMB = 230
- FIN = 120

Total : 1900 + 60 + 255 + 120 = 2335 ELO

BSC 3.0

- POS = 75
- CMB = 235
- FIN = 85

Total : 1900 + 75 + 235 + 85 = 2295 ELO

Résultats BT2630

Comparaison des heuristiques sur quelques postions de BT2630

Tableau 2. BT2630 Selective Options (Athlon XP 2400, 16MB Hash, 521KB HashPawn, 60sec)

BTXX	**3.0 (all false) Found Time in sec**	**3.2 (all false) Found Time in sec**	**3.0 (futility=true)**	**3.0 (razor=true)**	**3.0 (razor=true, futility=true)**	**3.0 (futility=true) (lmr=true)**	
BT2	6	8	7	7	6	6	9
BT3	22	19	23	22	23	18	9
BT4	7	6	X	7	X	7	7
BT5	2	2	1	2	1	2	2
BT6	1	1	4	1	4	1	0
BT7	2	2	2	1	1	2	2
BT8	0	0	0	0	0	0	0
BT9	0	0	0	0	0	0	0
BT10	19	12	18	18	17	15	30
BT11	2	2	2	2	2	2	9

Téléchargement

Cette version fonctionne en mode texte sous DOS (Cygwin Win32), avec *Winboard* [http://www.tim-mann.org/chess.html] ou encore *Arena* [http://www.playwitharena.com/].

Vous pouvez télécharger ce programme ici:

- *BSC v 3.2 23/04/2008 (Windows 32 Cygwin)* [bsc-win32.zip]
- *BSC v 3.2 23/04/2008 (Linux static)* [bsc-linux.tgz]

Vous pouvez télécharger une autre bibliothèque d'ouverture (8000 variations, 200000 coups, tiré de l'openbook de Fritz4): *BSC book Fritz4* [bscnewbook.zip]

Futures améliorations à apporter

Ci-dessous la liste des améliorations dans l'ordre prioritaire

- Amélioration de l'évaluation statique: clouage de pièces...
- Implanter le nouveau protocole UCI. Utilisation de WB2UCI pour le moment
- Mieux Gérer la représentation SAN (e4, Nxf3), plus de robustesse.
- Implanter les BitBoards pour accélérer le node rate.
- Amélioration de la bibliothèque d'ouverture: les lignes actuelles ne sont pas les meilleures! Construire une bibliothèque d'ouverture qui contient seulement les meilleurs lignes. (pas priorataire étant donné qu'une interface type Arena permet d'utiliser d'autre bibliotéques d'ouvertures et optimisées).
- Intégrer les ECO pour la bibliothèque d'ouverture. (intégré dans Arena donc est-ce vraiement utile ?)

Historique

3.2 du 23/04/2008

Last stable release

- Stabilisation de la 3.1b
- Improve search extension
- nullmove, razoring and futility disable in endgame (no short path to win with tablebase)
- Fix bug in exit with learning (segmentation fault)

3.1b du 07/04/2008

BSC v 3.1b 07/04/2008 (Windows 32 Cygwin) [bsc31b-win32.zip]

- Support for tablebases Scorpio (EGBB) 4 or 5 MEN

- Support Learning
- Support for analyzing in winboard/arena (.)
- Support FEN castle
- Reduce check extension: successive check or capture check

3.0 du 30/03/2008

BSC v 3.0 30/03/2003 [bsc30-win32.zip]

BSC v 3.2 23/04/2008 (Linux static) [bsc30-linux.tgz]

- New evaluation
- Late Move reduction implantation
- New futility prunning at pre-pre and pre frontier node
- Fix null move with 2 ply reduce (3 before)
- New depth extend heuristic in alpha/beta
- Fix bug in aspiration search with draw (no max depth)

2.8 du 01/02/2003

BSC v 2.8 01/02/2003 [bsc28-win32.zip]

- Fix bug in null move search
- New function (faster) for draw repetition moves
- Add winboard adaptater bsc.eng for Chessbase

2.72

- Fix bug in alpha-beta (stop search when no move in PV with score=INF)

2.71

- Fix bug in alpha-beta (stop search when no move in PV with score=INF)

2.7

BSC 2.7 du 14/12/2002 [bsc27-win32.zip]

- Fix parameters in extension search, pawn hash for pawn struct, logo and wb2uci in package
- Méthode implantée : Pawn hashing

2.6

BSC v 2.6 du 02/12/2002 [bsc26-win32.zip]

- Fix parameters in evaluation, extension search, null move

2.5

BSC v 2.5 du 06/11/2002 [bsc25-win32.zip]

- Fix bugs in evaluation

2.4

BSC v 2.4 02/11/2002 [bsc24-win32.zip]

- Fix bugs in evaluation

2.3

BSC v 2.3 24/10/2002 [bsc23-win32.zip]

- Fix bugs in evaluation (pawn position) , Fix bug in check control

2.2

2.2 du 23/10/2002

- Fix bugs in evaluation, fix bug in aspiration search (PV not follow), optimisation du mode blitz, optimisation de code

2.1

2.1 du 18/10/2002

- Heuristique de permuation de coups, Nouveau time control, Nouvelle évaluation statique (fix bug ? for castle move), Recapture extension, Nouvelle implentation des futility (fix bug in mat balance), Fractionnal Depth extension, nouvelle implantation de la fonction *is_attacked*
- Méthodes implantées : Fractionnal Depth, Recapture extension

2.0

2.0 du 20/07/2002

- 12/10/2002: Fix bug in time control
- Mat Proof-search (recherche de mat + rapide), futility+razoring fix bug, heuristic extension pour la recherche de sacrifice de pieces majeures, extension pour les promotions, nouvelle fonction d'évaluation avec mobilitée, mode blitz
- Méthodes implantées : + proof search, , blitz search (faster algorithm): NegaMax simplifié pour recherche plus rapide!

1.9

1.9 du 05/06/2002

- Nouvelle Implantation des Hashtables

- Utilisation Futility Prunning dans le fichier de configuration
- Fixe bug dans root_search, posant des problèmes à mtdf
- nouveau fichier pour la bibliothèque d'ouverture, basé sur l'openbook de Fritz4, refixe les limites de la bibliothèque d'ouverture (5000 lignes à 10000 lignes)
- Gestion de la notation SAN en mode console
- Méthodes implantées actuellement: NegaMax (null window), aspiration search, root_search, mtdf, transposition table, iterative deepening, move ordering, PV search, Killer heuristic (2 best), History heuristic, null move heuristic, futility heuristic, quiesce search, some selective extensions, ponder move, opening book, fifty move

1.8

1.8 du début 2002

- aspiration search, futility pruning
- Méthodes implantées : aspiration search, futiliy

1.7

1.7 du 24/09/2001

- killer move, hashtables affinées
- Méthodes implantées : killer move

1.5

1.5 du 07/05/2001

- mtdf, null move heuristic, hashtables affinées
- Méthodes implantées : mtdf, null move heuristic

1.2

1.2 du 18/02/2001

- NegaMax with null-win, root_search, hashtables affinées, fichier de configuration « bsc.rc »
- Méthodes implantées : NegaMax with null-win, root_search

1.1

1.1 du 16/02/2001

- Implantation des hashtables, de la bibliothèque d'ouverture, du pondering, du check-node en partant de l'implantation de TSCP
- Méthodes implantées : simple NegaMax, transposition table, iterative deepening, move ordering, PV search, History heuristic, quiesce search, some selective extensions, ponder move, opening book, fifty move

Utilisation

Les fichiers

La distribution BSC comprend les fichiers suivants:

- bsc.exe : l'executable
- bsc.rc : le fichier de configuration de BSC
- bsc.init : le fichier de configuration de BSC pour Winboard
- bsc.lrn : le fichier d'apprentissage de BSC
- cygwin1.dll : la librairie dynamique Cygwin 32
- bsc.opn : le fichier de la bibliothèque d'ouverture de BSC (3558 lignes)
- bsc.bmp : logo de BSC
- bsc.pdf : ce document au format PDF
- lct2.epd, bt2630.epd et wac.epd : deux fichiers de tests de positions (Win At Chess, BT2630 et LCT2)
- wb2uci.eng, wb2uci.exe : adaptateur Winboard vers UCI

Le fichier bsc.rc

Exemple 5. exemple de configuration

```
HASH=16 (les tables de transpositions sont activées, 16 Mo, 0 pour ne pas avoir
HPAWN=512 (512 Ko pour le hashing sur la structure de pions)
CHECKNODES=100000 (on vérifie que le temps n'est pas dépassé toutes les 300000 
NULLMOVE=1 (null-move heuristic activée)
MTDF=0 (mtdf désactivé)
ABWINDOW=5 (alpha/beta aspiration search ou mtdf window de 0.05 point matériel,
FUTILITY=1 (futility activé: pre et pre-pre frontier node)
RAZOR=1 (razoring activé: pre-pre-pre frontier node)
LMR=0 (Late Move Reduction désactivé)
EGBB=1 (Tablebase activée)
EGBBCACHE=4 (Cache de Tablebase)
EGBBPATH=c:/egbb/ (Chemin des tablebases Scorpio egbb)
5MEN=0 (utilise 5MEN => 400 Mo of RAM)
LEARN=0 (apprentissage desactivé )
RESIGN_VALUE=-9000 (valeur de résignation)
```

Les options de reduction de recherche Futility prunning, Razoring et Nullmove sont activées par défault. L'autre option selective de recherche LMR peut améliorer considérablement le niveau du programme ou l'analyse de certaines positions. Elle est cependant désactivée par défaut dans la distribution actuelle. Pour mieux comprendre les différents options que supporte BSC et que vous pouvez activer, lisez les articles suivants (en Anglais uniquement): *Chess Tree Search* [search.html] *Chess Tree Search Theory* [theory.html] *MTDF* [mtdf.html]

Le fichier bsc.opn

C'est le fichier de la bibliothèque d'ouverture de BSC. La structure de ce fichier est simple et contient les différentes lignes que peut jouer BSC lors de la phase d'ouverture du jeu. Vous pouvez ajouter

ou modifié les lignes d'ouvertures à l'aide d'un éditeur de texte. Le choix d'une ligne se fait de façon aléatoire.

Exemple 6. bsc.opn

```
d2d4g8f6c2c4g7g6g1f3f8g7g2g3e8g8f1g2c7c6b1c3d7d5c4d5c6d5f3e5e7e6e1g1b8c6e5c6b7c
d2d4g8f6c2c4g7g6g1f3f8g7g2g3e8g8f1g2c7c6b1c3d7d5c4d5c6d5f3e5e7e6e1g1f6d7e5f3b8c
```

Le mode console

Après avoir lancer bsc.exe, vous pouvez jouer aux échecs avec BSC en mode console en tapant les commandes au clavier.

Exemple 7. Mode console

```
BSC: b0n0 Simple Chess
----------------------
version 3.0, 30/03/2008
Copyright (C) 2008, ROBERT Nicolas <nicolas.robert@tangentoide.net>

help displays a list of commands.

Opening book found => bsc.opn
3558 lines in opening book
16383 ko for HASH
512 ko for PAWN HASH
hash=16MB, hashpawn=512KB, check_nodes=300000n, null move=false, mtdf=false, al
ha/beta window=5cp, fut=true, razor=true, Late Move Reduction=false

0>bsc# help
help
on - computer plays for the side to move
off - computer stops playing
st n - search for n seconds per move
sd n - search n ply per move (>=1)
undo - takes back a move
new - starts a new game
d - display the board
ml - display the legal moves
fen - setup a fen position
mat x - brut search for a mat in x ply
test file.epd n_sec nb_pos - test EPD file
hard - enable pondering
easy - disable pondering
book - disable book
bye - exit the program
xboard - switch to XBoard mode
Enter moves in coordinate notation, e.g., e2e4, e7e8Q
0>bsc#
```

Notation d'origine: Pour déplacer une pièce, il faut entrer case_départcase_arrivée: e2e4. La promotion s'écrit a7a8Q (Q pour reine, R pour tour, N pour cavalier et B pour fou). Les notations (0-0 et 0-0-0) pour le roque ne sont pas supportées. Pour roquer, la notation sera par exemple e1g1 (0-0 pour les blancs). Les commandes disponibles en mode console sont:

Extension San: vous pouvez entrez vos coups en utilisant la notation SAN: O-O, Nf3, e4, fxe6, Qb1xc4, Nbd2, a8=Q, fxe8=R. N'insertion du + pour signaler un échec en fin de notation est déconseillé.

- on : bsc prend la main et joue
- off : mode deux joueurs
- st n: fixe la durée en secondes pour jouer
- sd n: fixe la profondeur de recherche pour jouer
- new: commence une nouvelle partie
- undo: retour en arrière
- d: affiche l'échiquier
- ml: affiche la listes des coups valides
- fen fen_position: entre une position au format fen
- mat x: recherche un mat en force brute (en coups, soit 2x demi-coups). Les echécs sont étendus ce qui permet de trouver des mat en x+n coups.

```
3q1rk1/p4pp1/2pb3p/3p4/6Pr/1PNQ4/P1PB1PP1/4RRK1 b - - bm Bh2+; id "WAC.009";
0>bsc# mat 2
Searching 4 half ply
info depth 4 currmove h4g4 currmovenumber 1/38 nodes 1 time 0
info depth 4 currmove d8b8 currmovenumber 2/38 nodes 2960 time 0
info depth 4 currmove d8a8 currmovenumber 3/38 nodes 6223 time 0
info depth 4 currmove d8e8 currmovenumber 4/38 nodes 8275 time 15
info depth 4 currmove d8c7 currmovenumber 5/38 nodes 11264 time 15
info depth 4 currmove d8b6 currmovenumber 6/38 nodes 14895 time 15
info depth 4 currmove d8a5 currmovenumber 7/38 nodes 17901 time 31
info depth 4 currmove d8d7 currmovenumber 8/38 nodes 20524 time 31
info depth 4 currmove d8e7 currmovenumber 9/38 nodes 24010 time 31
info depth 4 currmove d8f6 currmovenumber 10/38 nodes 27546 time 31
info depth 4 currmove d8g5 currmovenumber 11/38 nodes 31216 time 31
info depth 4 currmove f8e8 currmovenumber 12/38 nodes 34575 time 31
info depth 4 currmove g8h8 currmovenumber 13/38 nodes 37733 time 46
info depth 4 currmove a7a6 currmovenumber 15/38 nodes 40172 time 46
info depth 4 currmove a7a5 currmovenumber 16/38 nodes 42423 time 46
info depth 4 currmove f7f6 currmovenumber 17/38 nodes 44587 time 46
info depth 4 currmove f7f5 currmovenumber 18/38 nodes 46949 time 46
info depth 4 currmove g7g6 currmovenumber 19/38 nodes 49502 time 46
info depth 4 currmove g7g5 currmovenumber 20/38 nodes 51822 time 46
info depth 4 currmove c6c5 currmovenumber 21/38 nodes 53451 time 62
info depth 4 currmove d6c7 currmovenumber 22/38 nodes 55566 time 62
info depth 4 currmove d6b8 currmovenumber 23/38 nodes 58129 time 62
info depth 4 currmove d6e7 currmovenumber 24/38 nodes 60843 time 62
info depth 4 currmove d6c5 currmovenumber 25/38 nodes 63475 time 62
info depth 4 currmove d6b4 currmovenumber 26/38 nodes 66377 time 62
info depth 4 currmove d6a3 currmovenumber 27/38 nodes 68953 time 62
info depth 4 currmove d6e5 currmovenumber 28/38 nodes 71757 time 78
info depth 4 currmove d6f4 currmovenumber 29/38 nodes 74079 time 78
info depth 4 currmove d6g3 currmovenumber 30/38 nodes 76043 time 78
info depth 4 currmove d6h2 currmovenumber 31/38 nodes 78330 time 78
Found! (88257 nodes,time=0 sec)
d6h2 g1h1 h2g3 h1g1 h4h1 g1h1 d8h4 h1g1 h4h2 #
info depth 4 currmove h6h5 currmovenumber 32/38 nodes 88257 time 93
info depth 4 currmove d5d4 currmovenumber 33/38 nodes 90607 time 93
info depth 4 currmove h4h5 currmovenumber 34/38 nodes 92641 time 93
```

```
info depth 4 currmove d8c8 currmovenumber 35/38 nodes 94512 time 93
info depth 4 currmove h4h3 currmovenumber 36/38 nodes 97862 time 93
info depth 4 currmove h4h2 currmovenumber 37/38 nodes 99925 time 93
info depth 4 currmove h4h1 currmovenumber 38/38 nodes 101973 time 93
Found (105698 nodes examined,time=0 sec)
0>bsc#
```

- test fichier.epd temps_réflexion_secondes nombres_positions: test un fichier EPD de positions. Le résultat de la recherche est inscrit dans le fichier bsc.log
- hard, easy: réfléchi ou non lorsque c'est votre tour
- book: désactive l'utilisation de la bibliothèque d'ouverture
- xboard: switch en mode xboard pour Winboard
- bye: quitte le programme

Les commandes en mode xboard ne sont pas expliquées. Elles sont conformes au protocole décrit par xboard.

Le fichier d'apprentissage

En activant l'option d'apprentissage, la ligne *Learn file => bsc.lrn open (5000 entries)* apparaît au démarrage du moteur d'échecs. La distribution est initialisée avec un fichier d'apprentissage à 5000 entrées. Le fichier d'apprentissage peut-être réinitialisé par la commande **clear_learn_file**.

```
0>bsc# clear_learn_file
Cleaning Learn File for 5000 entries...Done!
Learn file => bsc.lrn open (5000 entries)
```

Un nouveau fichier d'apprentissage peut également être crée par la commande **create_learn_file n** où n est le nombre d'entrée dans le fichier d'apprentissage.

```
0>bsc# create_learn_file 10000
Creating Learn File for 10000 entries...Done!
```

Le mode xboard avec Winboard

Winboard est une interface graphique permettant d'installer différents moteurs d'échecs. BSC supporte le protocole xboard.

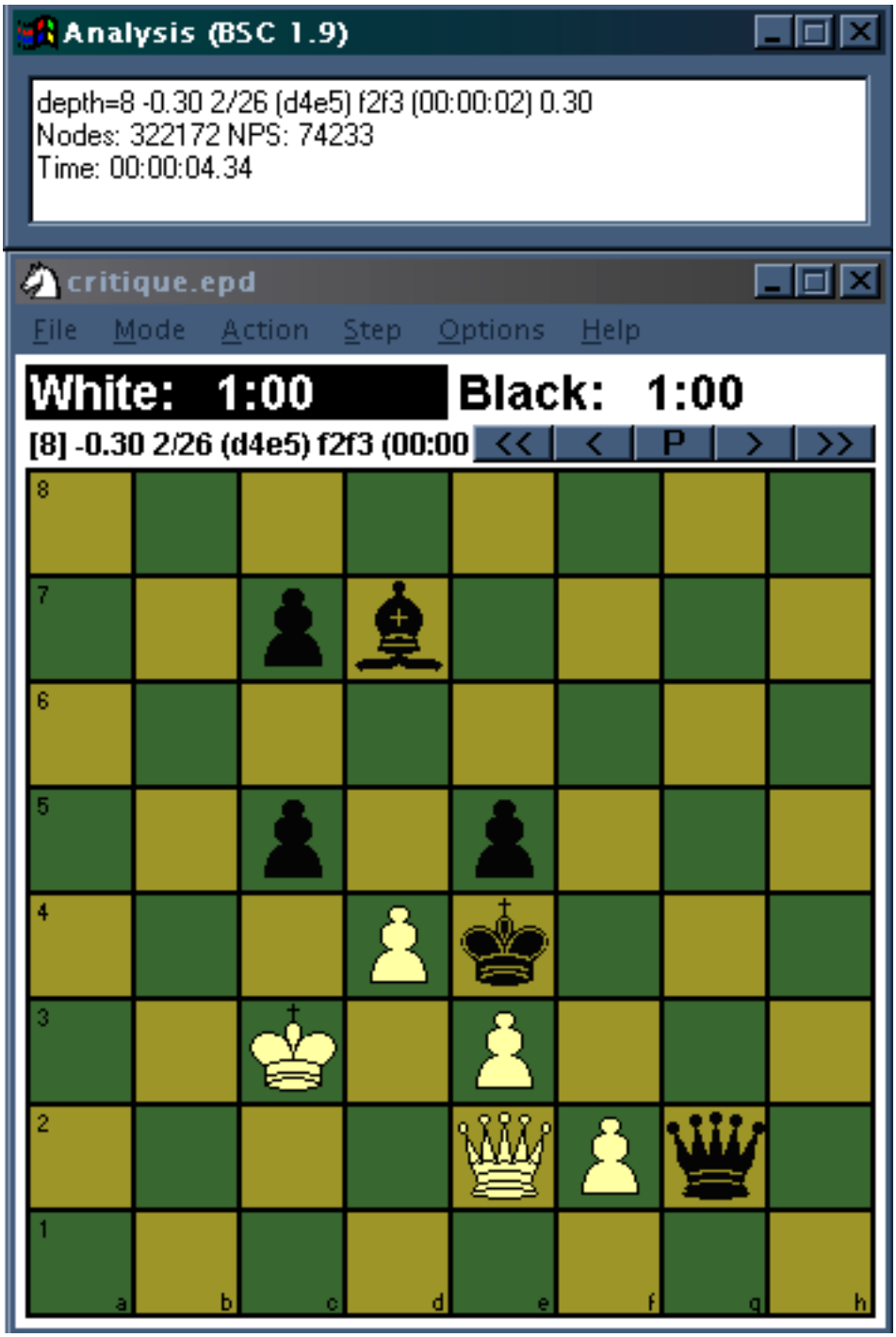
Pour installer BSC avec Winboard, éditer le fichier winboard.ini et ajouter les lignes « bsc /fd="C:\progra~1\winboard\bsc" » et « bsc /sd="C:\progra~1\winboard\bsc" » dans les sections firstChessProgramNames et secondChessProgramNames.

Exemple 8. winboard.ini

```
/firstChessProgramNames={bsc  /fd="C:\progra~1\winboard\bsc"
"GNUChess -T 16000"
"wcrafty-18.10.exe hash 16000000 xboard" /fd="C:\progra~1\winboard\crafty"
Comet_b37 /fd="C:\progra~1\winboard\comet"
}
/secondChessProgramNames={bsc /sd="C:\progra~1\winboard\bsc"
"GNUChess -T 16000"
"wcrafty-18.10.exe hash 16000000 xboard" /sd="C:\progra~1\winboard\crafty"
Comet_b37 /sd="C:\progra~1\winboard\comet"
}
```

BSC supporte le mode analyse sous Winboard.

Figure 1. Mode analyse sous Winboard



Dans la fenêtre d'analyse, vous voyez apparaître : la profondeur de recherche, le score de la recherche, l'analyse du coup N sur les coups possible, la meilleur variation, le temps mis pour trouver le meilleur coup, le pourcentage de réussite de recherche dans les tables de transposition, le nombre de positions analysées, le nombre de positions analysées par seconde.