

1 Preface

I'm a software developer and occasional chess player and have been fascinated by the Toga¹ source code from the time I was pointed to it from some cc forum. I never had considered to spend weeks of my sparetime to create a decent playing engine by myself and add it to the hundreds of them already existing. But with Toga we got a readymade real top engine, playing well above 2800 Elo and beating most commercial engines, amongst them Fritz, which has been developed professionally for more than 15 years.

I expected such an engine code to contain quite some amount of sophisticated chess knowledge to achieve this strength (and found that it does indeed!).

Having a strong chess engine is one thing. But while these programs play at the level of the strongest human Grandmasters today, to the intermediate chess player they present nothing more than one single evaluation number (plus a line of moves considered best for both sides), the result from an averaging of countless subtle factors. You may trust this evaluation value or not (latest engine results against human Grandmasters hint that you **should** trust it), but, unless you aren't a Master yourself, you don't stand any chance to understand a top engine's evaluation of a position.

So I thought it would be an interesting experience to log every single detail from the engine's evaluation process, and make this information available during analysis to provide an average chess player with a better understanding of chess positions.

Since the logfiles still contain only numbers (albeit in great detail and layouted nicely in concise HTML tables) a user needs some background information to be able to read and understand them. I provide this knowledge throughout the oncoming 20 pages. After studying this paper, you'll know everything about the evaluation of this strong engine (I'll explain it in natural language without using any programming terms) and can manually evaluate a position in the same way Toga does it within one minute.

But you won't be able to repeat the evaluation 500000 times a second while trying out different lines. My solution actually does not fit very well for explaining tactics. For accessing this area you still have to play around while the engine is running and observe how the evaluation changes. But once you found a line with a high (or low) evaluation score, the logfile does explain where this value comes from.

I read that a commercial visualization tool for the Rybka engine will become available around summer (see www.uciengines.de/UCI-Engines/Rybka/vriv/vriv.html), and this is exactly what I'm longing for to get from this tool: an explanation about engine tactics, along with the positional analysis.

¹ With the term "Toga" I refer to the open source chess engine created as "Fruit 2.1" by Fabien Letouzey and enhanced as "Toga II" by Thomas Gaksch

2 Toga Logfile Creation

- The engine version “Toga LOG” creates one logfile in HTML format each time it completes one search depth for depth values defined by the new engine parameters “Toga Min Log Depth” (default = 11) and “Toga Max Log Depth” (default = 15)
- The logfiles are stored in the directory “C:\Toga LOG”, file names are built from the positions Fen and the search depth.
- You can choose between compact and verbose logging by the new engine parameter “Toga Verbose Logging” (default = on).
- Logging can be used in every engine mode like playing, analysis, engine match or auto annotation.
- Logging does not slow down the engine (anyway, my build seems to be a bit slower than the official release)
- Logging only works correctly for default engine parameters (except the new logging parameters “Toga Min Log Depth”, “Toga Max Log Depth” and “Toga Verbose Logging”).
- A tool “Toga Logviewer” exists to quickly open the logfile for a certain position

3 Logfile Content

The first table within each logfile contains header info: position, search depth, value and main line.

Position:	1r3rk1/3bb1pp/2p1p3/1p2Pp1Q/2pP4/1P4P1/q3NPBP/2RR2K1 w - - 1 1
Depth:	17
Value:	-74
Line:	b3c4 b5c4 e2c3 a2a3 h5e2 b8b4 c3b1 a3a4 b1c3 a4b3 e2e3 b3b2 c1b1 b2c2 b1c1 c2b3 c1b1 f5f4 b1b3 f4e3

Figure 1: Header Table

The next table shows the static evaluation of the current position:

Current Position	
Value:	-116
Phase:	16
Material:	-123
Position Table:	16
Pawns:	-21
Tempo:	18
Pattern:	0
Piece:	3
King:	-9
Passed Pawns:	0

Figure 2: Current Position

The evaluation value is different to the first table, because this is a static evaluation, while the final value was found after a search of 17 semimoves. But it's also interesting what the engine thinks of a position without searching.

The table in [Figure 2](#) was generated in compact logging mode. You can see that the evaluation value (orange) sums up from 8 building blocks (green). A negative value always means an advantage for Black, a positive value an advantage for White.

Then logging goes down the main line and prints the evaluation after each move until, in the last table, it gets to the final evaluation value (after 20 semimoves in this case, mainline length does not always reflect the search depth). Each table displays the moves played so far and the current position.

Line:	b3c4 b5c4 e2c3 a2a3 h5e2 b8b4 c3b1 a3a4 b1c3 a4b3 e2e3 b3b2 c1b1 b2c2 b1c1 c2b3 c1b1 f5f4 b1b3 f4e3	
Position:	5rk1/3bb1pp/2p1p3/4P3/1rpP4/1RN1p1P1/5PBP/3R2K1 w - - 0 1	
Value:	-74	
Phase:	50	
Material:	-130	(-120,-140)
Position Table:	8	(23,-6)
Pawns:	74	(48,100)
Tempo:	15	(20,10)
Pattern:	0	(0,0)
Piece:	9	(15,3)
King:	0	(0,0)
Passed Pawns:	-50	(-28,-73)

Figure 3: End of the Main Line

4 Verbose Logging

This section explains a verbose logfile based on the following position:



Figure 4: Sample Position

4.1 Game Phase

Evaluation of certain concepts changes for different phases of a chess game, while it evolves from opening to middlegame and endgame.

Toga has the simple concept of calculating 2 values in each evaluation block, one for the opening and one for the endgame. It calculates the game phase based on the number of pieces on the board and then takes the weighted average between opening and endgame value.

Phase = 24 - 4 * nr of Queens - 2 * nr of Rooks - nr of Bishops - nr of Knights

This results in a number from 0 in the opening to 24 in the endgame. I log this as a percentage value where 0 means all pieces are on the board, 100 that only Kings and Pawns are left.

For [Figure 2](#) above a phase value of 16 is displayed, meaning it's close to the opening, since only a Bishop and a Knight are missing, as you see in [Figure 4](#). At the end of the main line in [Figure 3](#) the phase increased to 50.

For each evaluation block I log the weighted value and then in brackets the opening and endgame values.

4.2 Material

Queen	(975,975)
Rook	(500,500)
Bishop	(325,325)
Bishop pair	(50,50)
Knight	(325,325)
Pawn	(70,90)

Material:	-123	(-120,-140)
N 325 (325,325)		
B -325 (-325,-325)	B pair -50 (-50,-50)	P -73 (-70,-90)

Figure 5: Material

For the position in [Figure 4](#) we see that White has a Knight opposed to a Bishop of Black. Black has the Bishop pair and is a Pawn up.

4.3 Position Table

For each piece there is a table with 64 entries, one for each square. Each pieces position is just looked up in the appropriate table, and the score is summed up. I append the position tables at the end of the document (see [Appendix](#)).

Position Table:	16	(21,-3)					
Kg1: 33 (50,-48)	Qh5: -1 (0,-8)	Rc1: 0 (0,0)	Rd1: 2 (3,0)	Bg2: -1 (0,-6)	Ne2: -9 (-10,-5)	Pieces: 24 (43,-67)	
f2: 0 (0,0)	h2: -12 (-15,0)	b3: -4 (-5,0)	g3: -4 (-5,0)	d4: 20 (25,0)	e5: 12 (15,0)	Pawns: 12 (15,0)	
Kg8: -33 (-50,48)	Qa2: 2 (0,16)	Rb8: 2 (3,0)	Rf8: 0 (0,0)	Bd7: 0 (0,0)	Be7: 0 (0,0)	Pieces: -28 (-47,64)	
c4: 0 (0,0)	b5: 4 (5,0)	fs: 0 (0,0)	c6: 0 (0,0)	e6: -12 (-15,0)	g7: 4 (5,0)	h7: 12 (15,0)	Pawns: 8 (10,0)

Figure 6: Position Table

For [Figure 4](#) we see that both Kings are on the best squares (+33, but this will change in the endgame to -48 to get the King into the center). Most valuable for White are the Pawns on d4 (+20) and e5 (+12), for Black e6 (+12, all positive scores for Black result in negative values), worst evaluated are the Pawns on h2 and h7 (-12 for the whole a- and h-file, that's for getting the engine capturing towards the center).

4.4 Pawns

For technical reasons Pawn evaluation in Toga is split into 2 blocks "Pawns" and "Passed Pawns", which I keep in the logfile. The "Pawns" block implements the concepts of **doubled**, **isolated**, **backward**, **open** and **candidate** Pawns.

4.4.1 Doubled

Only the more advanced pawn counts as doubled. In case of a triple pawn, the 2 foremost pawns count as doubled.

4.4.2 Isolated

No friendly pawn exists on the files to the left and right of the pawn.

4.4.3 Backward

The pawn is not isolated, but all pawns on the files to the left and right of the pawn are more advanced than the pawn. Even if this is true, the pawn still is not considered backward if one of the following rules applies:

1. a friendly pawn on a one of the adjacent files to the left and right is more advanced by only one rank and
 - 1.1 on the field in front of the pawn is no (friendly or hostile) pawn and
 - 1.2 there is no hostile pawn on the file to the left or right 1 or 2 ranks ahead
2. the pawn is still on the second rank, a friendly pawn on one of the adjacent files to the left and right is 2 ranks ahead and
 - 2.1 on the fields 1 and 2 ranks ahead of the pawn is no (friendly or hostile) pawn and
 - 2.2 there is no hostile pawn on the file to the left or right 1, 2 or 3 ranks ahead

Some examples:

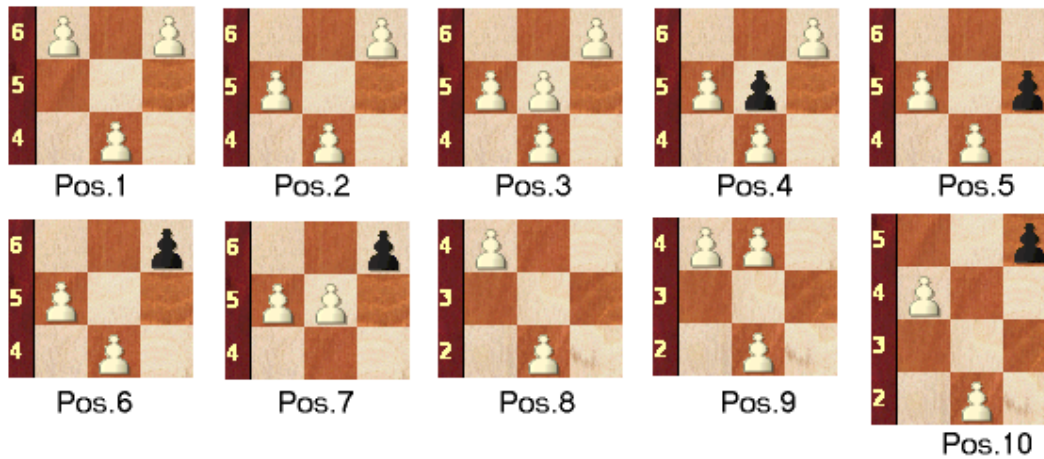


Figure 7: Backward Pawns

In Pos.1 b4 is backward. In Pos.2 b4 is not backward because of rule 1. In Pos.3 and Pos.4 b4 is backward because of rule 1.1. In Pos.5 and Pos.6 b4 is backward because of rule 1.2. In Pos.7 b4 is not backward. In Pos.8 b2 is not backward because of rule 2.1. In Pos.9 b2 is backward because of rule 2.1. In Pos.10 b2 is backward because of rule 2.2.

4.4.4 Open

There is no (friendly or hostile) pawn in front of the pawn.

4.4.5 Candidate

A pawn is considered a candidate if it is open and both of the following rules apply:

1. the number of hostile pawns on the adjacent files to the left and right in front of the pawn is not bigger than the number of friendly pawns on the adjacent files left and right behind or on equal row to the pawn
2. the number of hostile pawns currently attacking the pawn is not bigger than the number of friendly pawns defending the pawn

Some examples:

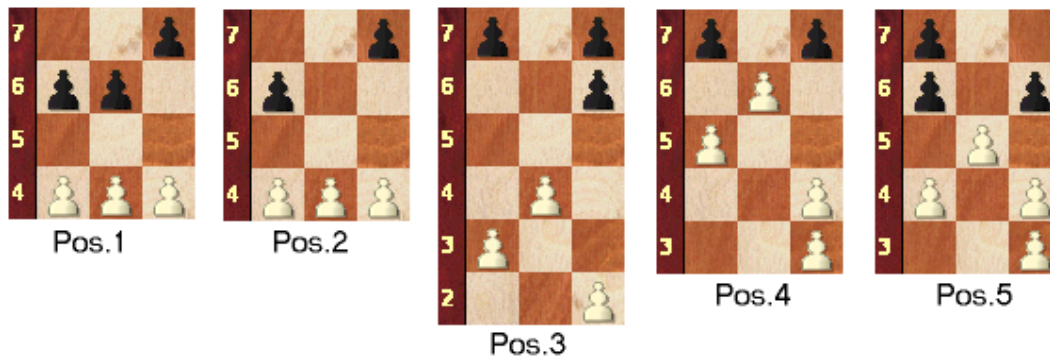


Figure 8: Candidate Pawns

In Pos.1 b4 is no candidate because it's not open. In Pos.2 b4 is a candidate. In Pos.3 b4 is no candidate because of rule 1. In Pos.4 b6 is no candidate because of rule 2 (but can get one by advancing one step). In Pos.5 b5 is a candidate.

4.4.6 Scoring

Doubled:	(10,20)
Isolated:	(10,20)
Isolated and Open:	(20,20)
Backward:	(8,10)
Backward and Open:	(16,10)
Candidate:	$(5 + 50 * \text{Bonus}(\text{rank}), 10 + 100 * \text{Bonus}(\text{rank}))$

Candidate scoring builds from a base value (5,10) and a value depending on advancement. There is a Bonus function which generates the following values depending on the rank:

Rank	Bonus
2	0
3	0
4	0.1
5	0.3
6	0.6
7	1

For [Figure 4](#) I get the following evaluation:

Pawns:	-21	(-20,-30)
b3: isolated: -11 (-10,-20)		
d4: backward and open: -14 (-16,-10)		
c4: doubled: 11 (10,20); candidate on rank 5: -23 (-5 - 50 * 0.3 (for rank 5) = -20,-10 - 100 * 0.3 (for rank 5) = -40)		
c6: backward 8: (8,10)		
e6: backward 8: (8,10)		

Figure 9: Pawns

4.5 Tempo

If it's White's move, (20, 10) is added, else (-20,-10). This factor obviously is there to achieve better comparison of lines with even/uneven depth. For manually comparing positions in the

logfile, however, it is confusing, so, to the right of the (orange) total value, I print the total value without this factor.

4.6 Pattern

A few simple concepts the engine cannot access by searching are implemented here (I show them for White, for Black they are mirrored. p/P means Black/White Pawn).

Trapped Bishop (-100,-100):

recognized in one of the following constellations:

- Ba7, pb6
- Bb8, pc7
- Bh7, pg6
- Bg8, pf7
- Ba6, pb5
- Bh6, pg5

Blocked Bishop (-50,-50):

- Bc1, Pd2, piece on d3
- Bf1, Pe2, piece on e3

Blocked Rook (-50,-50):

- Ra1 or Ra2 or Rb1, Kb1 or Kc1
- Rh1 or Rh2 or Rg1, Kf1 or Kg1

You will seldomly find values for this in the log file just because it's avoided by both sides. Still the engine needs it in the evaluation to see that it would be bad.

4.7 Piece

An extensive section accessing the concepts of **Mobility**, (Rook on) **Open File**, (Knight on) **Outpost**, (Rook, Queen on) **7th Rank**, (Queen's) **King Distance**.

4.7.1 Mobility

This counts the possible moves for each piece without captures. Scoring:

Knight: (moves - 4) * 4

Bishop: (moves - 6) * 5

Rook: (moves - 7) * (2 in the opening, 4 in the endgame)

For the Queen, Mobility calculation was implemented the same way originally but than replaced by King Distance evaluation.

4.7.2 Open File

Applies for Rooks only. There is a distinction between a semiopen file (no friendly but hostile pawn) and a full open file (no pawns at all). Additional bonus is given if the Rook is on the same (semiopen or open) file as the hostile King or an adjacent file to the hostile King.

Scoring:

Rook on closed file: (-10,-10)

Rook on semiopen file: (0,0)

Rook on semiopen file adjacent to hostile King: (10,0)

Rook on semiopen file same as hostile King: (20,0)
Rook on open file: (10,10)
Rook on open file adjacent to hostile King: (20,10)
Rook on open file same as hostile King: (30,10)

4.7.3 Outpost

Valid only for Knights in following conditions:

- the Knight is protected by at least one friendly pawn (it's not checked if it can get chased or exchanged)
- the Knight's square has a positive value in the KnightOutpostMatrix

The KnightOutpostMatrix for White (for Black it's mirrored) is defined as

8	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
6	0	0	4	5	5	4	0	0
5	0	2	5	₁₀	₁₀	5	2	0
4	0	2	5	₁₀	₁₀	5	2	0
3	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
	A	B	C	D	E	F	G	H

Scoring: **KnightOutpostMatrix[square]**

If the Knight is protected by 2 Pawns it scores twice. In this case I write the 2nd outpost score into the "7th Rank" column which is otherwise not used for the Knight.

4.7.4 7th Rank:

Applies for Rook and Queen in one (or both) of the following conditions:

- Rook or Queen is on the 7th rank and least one hostile pawn is on the 7th rank
- Rook or Queen is on the 7th rank and the hostile King is on th 8th rank

Scoring:

Rook on 7th: (20,40)
Queen on 7th: (10,20)

4.7.5 King Distance:

Applies for the Queen only and evaluates distance in ranks plus distance in files to the hostile King.

Scoring:

Queen's King Distance: 10 – distance ranks – distance files

For [Figure 4](#) I get the following evaluation:

Piece	Mobility	Open Line	Outpost	King Distance	7th Rank
Qh5				6	
Rc1	-4 (-4,-8)	0 (0,0)			
Rd1	-7 (-6,-12)	-10 (-10,-10)			
Bg2	5 (5,5)				
Ne2	-8 (-8,-8)				
Qa2				-3	-11 (-10,-20)
Rb8	2 (2,4)	10 (10,10)			
Rf8	4 (4,8)	10 (10,10)			
Bd7	20 (20,20)				
Be7	-10 (-10,-10)				

Figure 10: Piece Evaluation

4.8 King

This evaluates the concepts of **Pawn Shelter**, **Pawn Storm** and **Piece Attack**. It only adds to the opening evaluation, and is calculated only if the opponent has the Queen and at least one other piece.

4.8.1 Pawn Shelter

For each of the 3 friendly pawns on the King's file and the 2 adjacent files it calculates a penalty based on the advance of the pawn. It calculates the distance of the Pawn to the 8th rank and applies the formula $36 - (\text{distance to } 8^{\text{th}})^2$ as a penalty value. For the 2nd rank this results in $36 - 6^2 = 0$, for the 3rd in $36 - 5^2 = 11$, etc. If there is no Pawn on a row, it subtracts a penalty of 36. The penalty for the King's file is doubled. If no penalty results from this calculation it gives a penalty of 11 because of the weak backrank.

Score (fpap = friendly pawn advance penalty = $-36 + (\text{distance to } 8^{\text{th}})^2$):

Pawn Shelter: 2 * fpap for the King's file + fpap for each of the adjacent files
-11 if this results in 0

4.8.2 Pawn Storm

For this concept it examines the hostile Pawn advances on the 3 files in front of the King. For each Pawn on the 4th rank it counts -10, for each on the 5th it counts -30, for each hostile Pawn on the 6th rank it counts -60. Hostile pawns on the 7th rank are not counted. All 3 Files have the same weight (no doubling here on the King's file).

Score (hpap = hostile pawn advance penalty = -10 for 4th, -30 for 5th, -60 for 6th rank):

Pawn Storm: hpap for the King's file + hpap for each of the 2 adjacent files to the left and right

If the King retained the right to castle, it calculates the same penalty for the King's destination square after castling (if castling is possible for both sides, it takes the destination square with the smaller penalty) and then calculates the average from the King's current squares penalty and the castling destination squares penalty as the penalty score.

4.8.3 Piece Attack

A Knight is considered to attack the hostile King if it can reach a field adjacent to the king within one move.

A Bishop, Rook or Queen is considered to attack the hostile King if,

1. on a board with only the hostile King and the piece on it, it could reach a square adjacent to the King within one move, and,

- on the real board, without capturing, can reach a square lying within 2 squares (=2 King moves) distance from the King within one move.

A King attack by a piece weighs by the pieces quality: Knight = 1, Bishop = 1, Rook = 2, Queen = 4. These values are summed up for all attacking pieces.

The score for King attack from pieces is then calculated as

Piece King Attack: $20 * \text{sum of attacking values} * \text{KingAttackWeight}(\text{number of attackers})$

KingAttackWeight is a function defined as

Nr of Attackers	Weight
1	0
2	0.5
3	0.75
4	0.88
5	0.94
6	0.97
7	0.99

So, a King attack by only one piece does not count at all.

For [Figure 4](#) I get the following evaluation:

King:	-9	(-12,0)
White Pawn Shelter: -22		
Black Pawn Storm: -10		
Black Pawn Shelter: 20		
White Pawn Storm: 0		
King Attack by Qa2 -80		
Attack to white King $80 * 0.00$ (for 1 pieces) = 0		
King Attack by Qh5 80		
Attack to black King $80 * 0.00$ (for 1 pieces) = 0		

Figure 11: King Attack

White has an advanced g-Pawn, resulting in $-(36 - 5^2) * 2$ (for the King's file) = -22. The advanced Black f-Pawn adds -10 (for 4th rank), so the result is -32.

Black has an advanced f-Pawn which gives a penalty of $(36 - 4^2) = 20$.

The King attack by the Black Queen is a bit strange since the Queen can only go to d2 which is 3 squares away from the King. But for some reason (I'm not sure if it's a bug²) the blocking square counts. So it sees the Black Queen can go to e2 (it does not consider capturing since it produces the same result with a Black piece on e2) which is 2 squares away from the White King, and considers this as an attack. Anyway, since KingAttackWeight for 1 piece is 0, no attack gets counted.

² Changing the do{ }while into a while{ } in attack.cpp, line 487 would result in expected behavior

The White Queen clearly attacks the Black King according to the definition, but the single piece attack also does not count.

2 more samples:

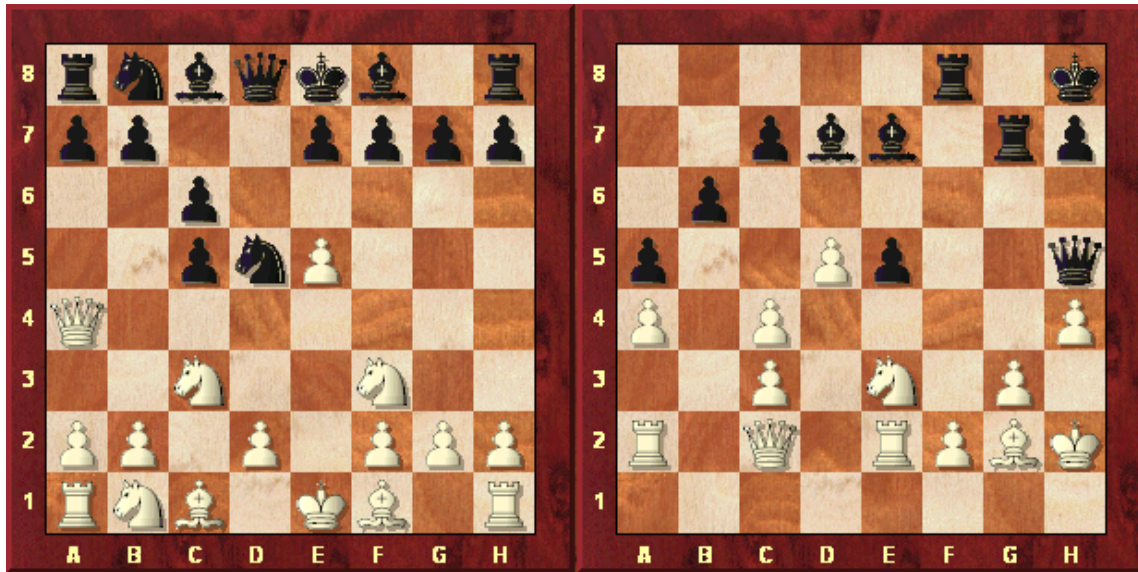


Figure 12: Sample Positions

Left Position:

White Pawns:

The Pawn in front of the White King is advanced to the 5th rank resulting in $-(36 - 3^2) = -27 * 2$ (for King's file) = -54. No Black Pawn has advanced. White retained castling rights for both directions. The value for c1 is -36 (White Pawn is missing on that file) * 2 - 10 (Black Pawn has advanced to 4th rank) = -82. The value for g1 is -11 (no weaknesses except backrank). It takes the average between -54 and -11 = -32.

Black Pawns:

The d-Pawn is missing, -36, White's e-Pawn has advanced to 5th rank, -30, = -66 for King on e8. For King on c8 it's -36 (missing d-Pawn) - 11 * 2 (advanced c-Pawn), no Pawn Storm, = -47. For King on g8 it's -11. Average between -66 and -11 = -38.

After castling kingside both sides would have only -11, so in conjunction with the Position Table values both sides have a strong desire to castle.

Piece Attacks:

The only piece attack comes from the White Queen, which is blocked at c6, within 2 squares from the Black King. The single piece attack does not count.

King score in [Figure 12](#) (left): $-32 + 38 = 6$

Right Position:

White Pawns:

The Pawn in front of the White King is advanced to the 4th rank resulting in $-(36 - 4^2) = -20 * 2$ (for the King's file) = -40. The Pawn to the left of the White King is advanced to the 3th rank, $-(36 - 5^2) = -11$. No black pawn has advanced, resulting in -51 for White's Pawn evaluation.

Black Pawns:

No Pawn is right to the Black King, resulting in 36. White's h-Pawn has advanced to rank 4, counting 10, so Black's Pawn evaluation is 46.

Piece Attacks:

The White King is attacked by Qh5 (-80), Rg7 (-40), Bd7 (-20) and Be7³ (-20), but not by Rf8 which cannot reach a square adjacent to the White King. The score is $-160 * 0.88$ (KingAttackWeight for 4 attacking pieces) = -140.

The Black King is attacked by Qc2, but one attacker does not count.

King score in **Figure 12** (right): $-51 + 46 - 140 = -145$

Figure 12 (right) is a position from an interesting game Rybka – Toga. Toga had sacrificed 2 pawns for a strong King attack early in the game, but at some point couldn't make progress. Rybka failed to find a way to simplify either. Both sides had been shifting pieces forth and back for more than 50 moves, the game looked completely drawish. Static evaluation by Toga is -71, after the expected move **74.Qe4** even -93. At a depth of 17, Rybka evaluates the position with +72, Toga sees +36.

But suddenly, in the same position she had been holding accurately for such a long time, Rybka played the tactical blunder **74. Re1??**, after which Toga finally got it's attack going with **74... Bxh4! 75. Kgl Rxx3! 76. Qd1 Bg4 77. Qd3 Rgf3 78. d6 Bxf2+ 79. Rxf2 Rxf2**. Both engines take exactly 4 s to find Bxh4 after Re1, but in the move before Rybka takes 18 s to abandon Re1, although it sees Bxh4 already after 11 s. Toga just never considers to play Re1.

You normally would expect that engines, after having played around in a constant position for some time, have the positions stored in hash memory and do not blunder even when getting low on time. The point here was that Rybka recently had played one single pawn move, **h2-h4**, obviously to bypass the 50 move rule, since she felt for winning, thereby

1. providing Toga an important attack mark
2. rendering her cash useless, in a situation where
3. she was running low on time

Analysis shows that, even after **75... Rxx3**, it is not necessary for White to give up the exchange. According to Toga best line is **76. Qe2 Rxx2 77. Nxx2 Bxf2 78. Qxf2 Rxf2 79. Rxf2** where White has 2 Rooks for Queen and Pawn.

I sometimes read that Rybka plays human like. Indeed here it seems that Toga's unexpected strike had put her down and she stopped fighting.

So, in this game, I would say, Rybka beat herself ;-).

4.9 Passed Pawns

A pawn is **passed** if it is open and on the 2 adjacent files to the left and the right of the pawn no hostile pawn is ahead.

³ for Be7, it sees the adjacent square for rule 1 at the imaginary position i3 which might be a bug according to the authors intention. But within this algorithm nothing is really correct and nothing is really wrong, so it doesn't matter much. In this position it's actually good to count Be7 as an attacker, and it should even count Rf8.

Scoring is similar to candidate (a passed pawn does not count as a candidate, passed status is the next level to promotion, so to say). For opening score, it is

Passed Pawn Opening: $10 + 60 * \text{Bonus}(\text{rank})$

For endgame score, 3 more concepts are added:

1. **distance of the Kings** in moves to the field in front of the pawn; evaluation is $20 * \text{distance of hostile King} - 5 * \text{distance of friendly King}$
2. a passed Pawn is considered **free** if the square in front of it is empty and it can securely advance one step without getting lost; score = 60
3. a passed Pawn is **unstoppable** if the opponent has no pieces (just King and Pawns) and
 - a. no friendly piece is in the way to queening and
 - b. the hostile King cannot reach the Pawn before queening; score = 800
4. a passed Pawn is **unstoppable** also if the opponent has no pieces (just King and Pawns) and
 - a. the friendly King is on an adjacent square to the queening square and
 - b. the friendly King is on an adjacent square to the Pawn; score = 800

Samples:

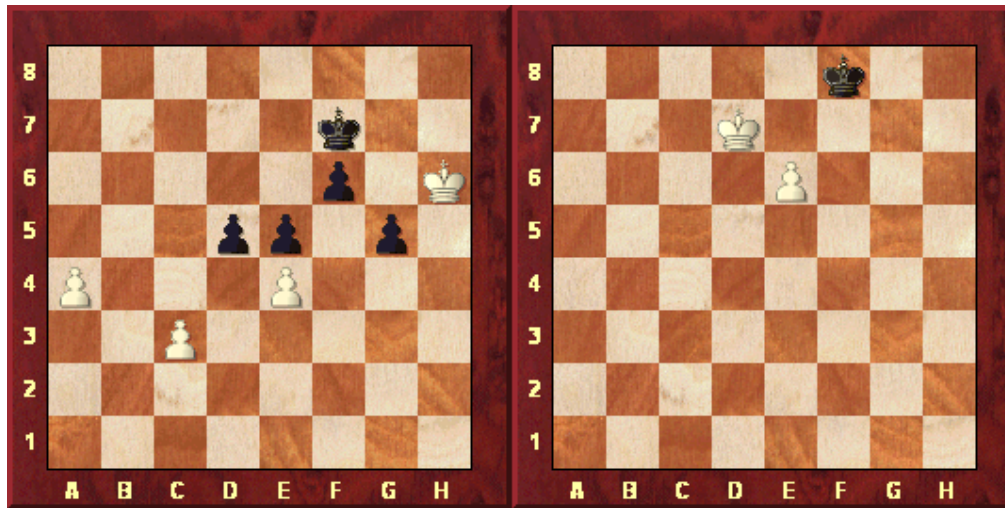


Figure 13: Sample Positions

If in **Figure 13** (left) if it's White's move, a4 is unstoppable, g5 is free. If it's Black's move, g5 is unstoppable, a4 is free (rules 2 and 3). In the right position e6 is unstoppable in any case (rule 4).

Endgame score is

Passed Pawns Endgame:

$20 + (120 + \text{Kings distances score} + \text{free score} + \text{unstoppable score}) * \text{Bonus}(\text{rank})$

I show the Passed Pawns evaluation in one more sample position:



Figure 14: Sample Position

Passed Pawns:	-50	(-28,-73)
c4 passed Pawn rank 5 -50 (-28,-73)		
Opening: $-10 - 60 * 0.3$ (for rank 5) = -28		
Endgame: $-20 - (120 + 80 (20 * \text{distance white King}) - 25 (5 * \text{distance black King})) * 0.3$ (for rank 5) = -73		

Figure 15: Passed Pawns

In **Figure 14** the White King is 4 moves away from the square c3 in front of the Passed Pawn at c4 ($20 * 4 = 80$), while the Black King is 5 moves away from that square ($5 * 5 = 25$).

4.10 Draw Evaluation

Toga implements draw evaluation for move repetition and the 50 moves rule, which may overwrite the evaluation value with 0. I do not log this. If in a logfile the total is 0 and does not match with the subvalues, the engine found a draw.

5 Toga Logviewer

The “Toga Logviewer” is a tool for quick access to logfiles from the CB Gui.

If you want to view an existing logfile for a position displayed in the Gui, call “copy position” in the Gui and then press the “Load/Paste” button in the Logviewer to see the logfile. If logfiles for different depths are available, use the dropdown to switch between them.

Even more interesting is “live” analysis while the engine is running. In that case you don’t have to copy the position. If you press the button and the viewer does not find a valid Fen in the clipboard, it just displays the latest logfile written by the engine (matching the selected depth).



Figure 16: Toga Logviewer

Still more automatism provides the “auto” option, where the tool loads each logfile automatically after creation.

It may be disturbing when the viewer loads the logfile for the next depth while you are reading. Therefore, I prefer using some fixed depth which the engine reaches immediately (e.g. 9, which is the same value I enter into the “Toga Min Log Depth” engine parameter) so the viewer follows the navigation in the Gui smoothly. When you press the “Load/Paste” button in this mode, you deliberately update to the highest available depth.

The last feature is “top” which tells the viewer to stay on top of the Gui.

Btw, when I copied positions back from the viewer into the Gui (by selecting the Fen string with the mouse and pressing Ctrl-c), I sometimes got an “Illegal Position” error from the Gui. I found that I had copied a “Tab” character before the Fen accidentally which was not accepted. I could avoid this by selecting the text (dragging the mouse) from beginning to end instead as from end to beginning in the viewer.

The Logviewer requires the (free) .NET 2.0 Runtime.

6 Appendix

The next two pages show the Position Tables for all pieces.

White Pawn Opening

-15 -5 0 5 5 0 -5 -15
-15 -5 0 5 5 0 -5 -15
-15 -5 0 5 5 0 -5 -15
-15 -5 0 15 15 0 -5 -15
-15 -5 0 25 25 0 -5 -15
-15 -5 0 15 15 0 -5 -15
-15 -5 0 5 5 0 -5 -15
-15 -5 0 5 5 0 -5 -15

White Pawn Endgame

0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0

White Knight Opening

-135 -25 -15 -10 -10 -15 -25 -135
-20 -10 0 5 5 0 -10 -20
-5 5 15 20 20 15 5 -5
-5 5 15 20 20 15 5 -5
-10 0 10 15 15 10 0 -10
-20 -10 0 5 5 0 -10 -20
-35 -25 -15 -10 -10 -15 -25 -35
-50 -40 -30 -25 -25 -30 -40 -50

White Knight Endgame

-40 -30 -20 -15 -15 -20 -30 -40
-30 -20 -10 -5 -5 -10 -20 -30
-20 -10 0 5 5 0 -10 -20
-15 -5 5 10 10 5 -5 -15
-15 -5 5 10 10 5 -5 -15
-20 -10 0 5 5 0 -10 -20
-30 -20 -10 -5 -5 -10 -20 -30
-40 -30 -20 -15 -15 -20 -30 -40

White Bishop Opening

-8 -8 -6 -4 -4 -6 -8 -8
-8 0 -2 0 0 -2 0 -8
-6 -2 4 2 2 4 -2 -6
-4 0 2 8 8 2 0 -4
-4 0 2 8 8 2 0 -4
-6 -2 4 2 2 4 -2 -6
-8 0 -2 0 0 -2 0 -8
-18 -18 -16 -14 -14 -16 -18 -18

White Bishop Endgame

-18 -12 -9 -6 -6 -9 -12 -18
-12 -6 -3 0 0 -3 -6 -12
-9 -3 0 3 3 0 -3 -9
-6 0 3 6 6 3 0 -6
-6 0 3 6 6 3 0 -6
-9 -3 0 3 3 0 -3 -9
-12 -6 -3 0 0 -3 -6 -12
-18 -12 -9 -6 -6 -9 -12 -18

White Rook Opening

-6 -3 0 3 3 0 -3 -6
-6 -3 0 3 3 0 -3 -6
-6 -3 0 3 3 0 -3 -6
-6 -3 0 3 3 0 -3 -6
-6 -3 0 3 3 0 -3 -6
-6 -3 0 3 3 0 -3 -6
-6 -3 0 3 3 0 -3 -6
-6 -3 0 3 3 0 -3 -6

White Rook Endgame

0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0

White Queen Opening

0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
-5 -5 -5 -5 -5 -5 -5 -5

White Queen Endgame

-24 -16 -12 -8 -8 -12 -16 -24
-16 -8 -4 0 0 -4 -8 -16
-12 -4 0 4 4 0 -4 -12
-8 0 4 8 8 4 0 -8
-8 0 4 8 8 4 0 -8
-12 -4 0 4 4 0 -4 -12
-16 -8 -4 0 0 -4 -8 -16
-24 -16 -12 -8 -8 -12 -16 -24

White King Opening

-40 -30 -50 -70 -70 -50 -30 -40
-30 -20 -40 -60 -60 -40 -20 -30
-20 -10 -30 -50 -50 -30 -10 -20
-10 0 -20 -40 -40 -20 0 -10
0 10 -10 -30 -30 -10 10 0
10 20 0 -20 -20 0 20 10
30 40 20 0 0 20 40 30
40 50 30 10 10 30 50 40

White King Endgame

-72 -48 -36 -24 -24 -36 -48 -72
-48 -24 -12 0 0 -12 -24 -48
-36 -12 0 12 12 0 -12 -36
-24 0 12 24 24 12 0 -24
-24 0 12 24 24 12 0 -24
-36 -12 0 12 12 0 -12 -36
-48 -24 -12 0 0 -12 -24 -48
-72 -48 -36 -24 -24 -36 -48 -72

Black Pawn Opening

15 5 0 -5 -5 0 5 15
15 5 0 -5 -5 0 5 15
15 5 0 -15 -15 0 5 15
15 5 0 -25 -25 0 5 15
15 5 0 -15 -15 0 5 15
15 5 0 -5 -5 0 5 15
15 5 0 -5 -5 0 5 15
15 5 0 -5 -5 0 5 15

Black Pawn Endgame

0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0

Black Knight Opening

50 40 30 25 25 30 40 50
35 25 15 10 10 15 25 35
20 10 0 -5 -5 0 10 20
10 0 -10 -15 -15 -10 0 10
5 -5 -15 -20 -20 -15 -5 5
5 -5 -15 -20 -20 -15 -5 5
20 10 0 -5 -5 0 10 20
135 25 15 10 10 15 25 135

Black Knight Endgame

40 30 20 15 15 20 30 40
30 20 10 5 5 10 20 30
20 10 0 -5 -5 0 10 20
15 5 -5 -10 -10 -5 5 15
15 5 -5 -10 -10 -5 5 15
20 10 0 -5 -5 0 10 20
30 20 10 5 5 10 20 30
40 30 20 15 15 20 30 40

Black Bishop Opening

18 18 16 14 14 16 18 18
8 0 2 0 0 2 0 8
6 2 -4 -2 -2 -4 2 6
4 0 -2 -8 -8 -2 0 4
4 0 -2 -8 -8 -2 0 4
6 2 -4 -2 -2 -4 2 6
8 0 2 0 0 2 0 8
8 8 6 4 4 6 8 8

Black Bishop Endgame

40 30 20 15 15 20 30 40
30 20 10 5 5 10 20 30
20 10 0 -5 -5 0 10 20
15 5 -5 -10 -10 -5 5 15
15 5 -5 -10 -10 -5 5 15
20 10 0 -5 -5 0 10 20
30 20 10 5 5 10 20 30
40 30 20 15 15 20 30 40

Black Rook Opening

6 3 0 -3 -3 0 3 6
6 3 0 -3 -3 0 3 6
6 3 0 -3 -3 0 3 6
6 3 0 -3 -3 0 3 6
6 3 0 -3 -3 0 3 6
6 3 0 -3 -3 0 3 6
6 3 0 -3 -3 0 3 6
6 3 0 -3 -3 0 3 6

Black Rook Endgame

0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0

Black Queen Opening

5 5 5 5 5 5 5 5
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0

Black Queen Endgame

24 16 12 8 8 12 16 24
16 8 4 0 0 4 8 16
12 4 0 -4 -4 0 4 12
8 0 -4 -8 -8 -4 0 8
8 0 -4 -8 -8 -4 0 8
12 4 0 -4 -4 0 4 12
16 8 4 0 0 4 8 16
24 16 12 8 8 12 16 24

Black King Opening

-40 -50 -30 -10 -10 -30 -50 -40
-30 -40 -20 0 0 -20 -40 -30
-10 -20 0 20 20 0 -20 -10
0 -10 10 30 30 10 -10 0
10 0 20 40 40 20 0 10
20 10 30 50 50 30 10 20
30 20 40 60 60 40 20 30
40 30 50 70 70 50 30 40

Black King Endgame

72 48 36 24 24 36 48 72
48 24 12 0 0 12 24 48
36 12 0 -12 -12 0 12 36
24 0 -12 -24 -24 -12 0 24
24 0 -12 -24 -24 -12 0 24
36 12 0 -12 -12 0 12 36
48 24 12 0 0 12 24 48
72 48 36 24 24 36 48 72