

うさぴょん2 PR文書

2016年4月21日

うさぴょんの育ての親

うさびょん

うさびょん - 201409262241

ファイル(E) 盤面(B) 表示(V) ヘルプ(H)

投 誰見 ?

後手番 うさびょんLv2 持ち駒:なし t:0m00s

9	8	7	6	5	4	3	2	1	
香	桂	銀	金	王	金	銀	桂	香	一
	飛						角		二
歩	歩	歩	歩	歩	歩	歩	歩	歩	三
									四
									五
									六
歩	歩	歩	歩	歩	歩	歩	歩	歩	七
	角						飛		八
香	桂	銀	金	王	金	銀	桂	香	九

先手番 あなた 持ち駒:なし t:0m00s



将棋ぶろぐらむ
うさびょん

先手:あなた
時間:0m32s 32s
後手:うさびょんLv2
時間:0m00s 0s

開始局面

OK

1/1

うさぴよんの歴史

- ◆ 1999年から『鈴木将棋』をベースに、開発を始める。(ただし、2015年選手権のうさぴよんには鈴木将棋ベースの部分はありません)
- ◆ 2001年3月に行われた第11回世界コンピュータ将棋選手権でデビュー。結果は55チーム中20位。
- ◆ 第25回世界コンピュータ将棋選手権で、一次予選11位で予選落ち。

で、実力はどのようなの？

- ◆ おそらく世界コンピュータ将棋選手権で一次予選は突破できるくらい → 突破できなかった(爆)
- ◆ 多分二次予選ではボコボコにされる
- ◆ 将棋倶楽部24でのRは1600~1800位と推測される

特徴 1

- ◆ ほぼ単純な $\alpha \beta$ 法
- ◆ おそらく最初(鈴木将棋を開発した時なので1999年)にマルチスレッド対応していたプログラム(並列化方法はYBWC)
- ◆ 並列化は8並列まで対応
- ◆ LateMoveReduction
- ◆ NullMoveは何故かかえって遅くなるので不採用 → 5月1日になってぎりぎり結局実装。効果の程は数%程度の高速化。

特徴 2

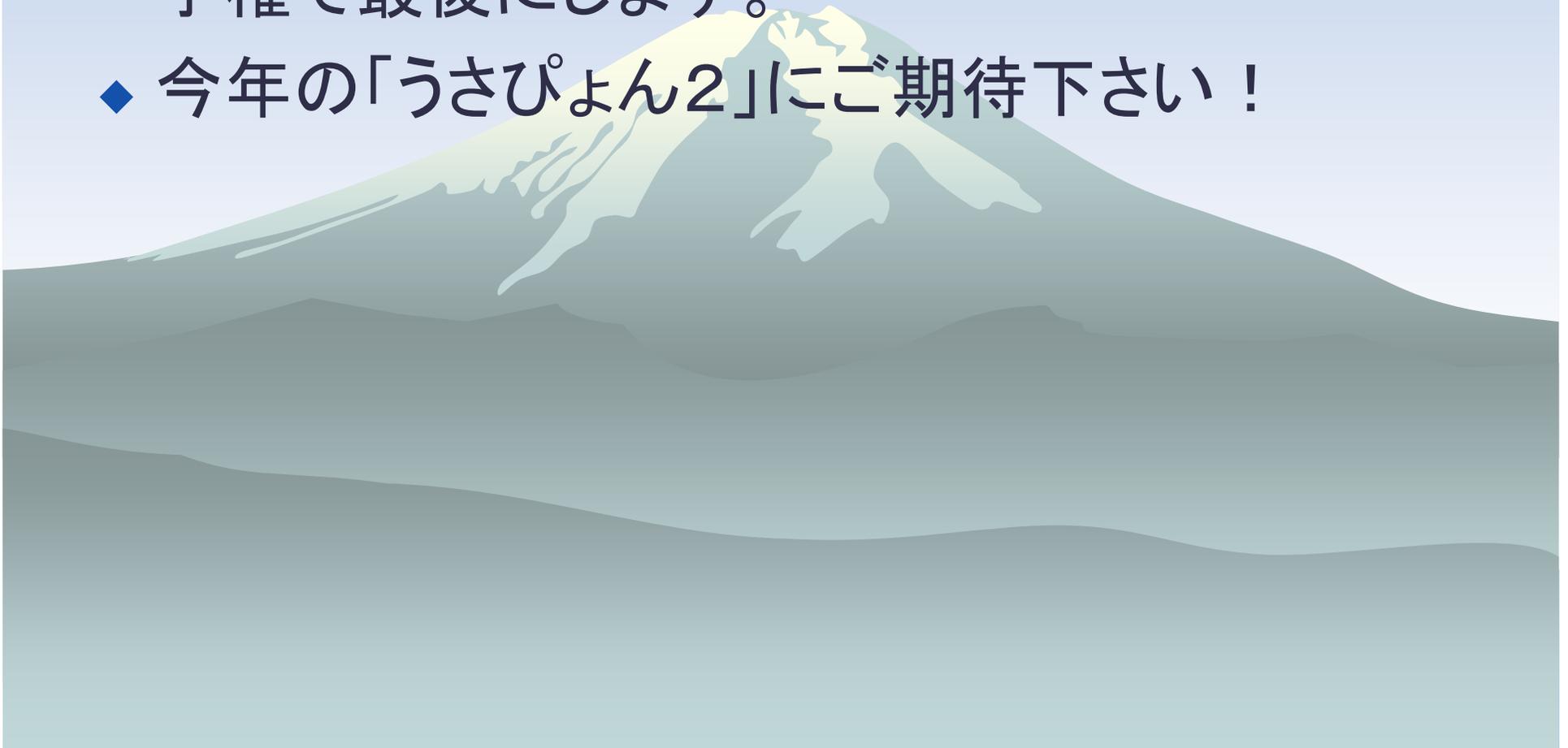
- ◆ 評価関数は手でチューニング
- ◆ 盤面「全体」は定跡から外れた局面でも「自陣のみ 9×5 枡」「自陣のみ 5×5 枡」一致していれば定跡として記憶された手順にボーナスを与えてその手を選びやすくしている
- ◆ 何故か強いプログラムに一発入れることが多い
- ◆ 何故か格下のプログラムに一発もらうことも多い

特徴 3

- ◆ 局面を「進める」関数は持っているが、局面を「戻す」関数は持っていない
- ◆ なので、スタックに局面を積んでいる
- ◆ 何故「戻す」関数がないかというと、「進める」関数が異常に複雑なため
- ◆ 「進める」関数では、駒を動かしたことにより盤面全体で「交換値」を再計算している
- ◆ 指し手の「仮評価」も一手進めてみて求めている
- ◆ 指し手は「仮評価」の結果が高かった順に探索
- ◆ そんなこんなで「前向き枝刈」しているのにあまり先読みできない異常に遅いプログラム

第26回コンピュータ将棋選手権 に向けて

- ◆ 「うさびょん」としての開発・参加は今年の選手権で最後にします。
- ◆ 今年の「うさびょん2」にご期待下さい！



...というわけで、うさびょん2

- ◆ クラスタ構成
- ◆ なのはminiを一部改造したマスター部→「Puella α 」と「大合神くじらちゃん」を参考にしています。⇒自力で0から実装することにしました。「なのはmini」も「Puella α 」も「大合神くじらちゃん」も関係しません。(2016/4/21)
- ◆ なのはminiをStockFish7ベースに書き換えたスレーブ部(lazy SMPを採用)
- ◆ 評価バイナリはAperyの電王戦トーナメント版を採用。そのため、Originalのなのはminiとは少しだけ評価関数が変わっている。
- ◆ 2台できちんと並列化効果があることが確認出来たら、最大4台位まで台数を増やすかも。(現時点では3台を予定、並列化効果が上がらなければもしかすると1台だけでの出場になるかも...)

本当は...

- ◆ 評価関数は自前で学習して用意するつもりでしたが、Aperyの評価バイナリ・評価関数より優秀なものを短期間で作れる気がしないので諦めましたw
- ◆ 探索部はスレーブも「なのはmini」をそのまま使うつもりでしたが、4コアだけを使用したものと、8コアフルに使用したもののfloodgate上でのRの差が70くらいしかない&8コアだとスケールしない(らしい)という情報を得て、StockFish7ベースへ書き換えを行っています。

というわけで、

- ◆ 今までと違い、
 - ❖ 自前で実装する部分が少ない
 - ❖ 他人の書いた長いソースコードを読むのは、過去のライブラリ制度の下で、ライブラリ審査委員としてBonanzaを読んだ時以来
 - ❖ Chessのソースコードを真剣に読むというのが初体験に近い
- ◆ そのため、実装する部分は減っているはずなのに、「むしろ難しいなー」とか思っております。

最後に

- ◆ 将棋所に変わるGUIを自前で実装したいなあ、「うさびょん(マスコット)」も表示したいし。でも、そんな時間ないんだらうなあ…。(やっぱり無理！(2016/4/21))

- ◆ 今年には全力全開で (AMD製CPU使用の中での)
1位を取りに行きます！