目次

開発者向けマニュアル

- 操作マニュアルは、オンラインマニュアルのほうをご覧ください。ここではそれ以外について説明します。

デバッグ機能の使い方

- 思考エンジンとは USI プロトコルというプロトコルでやりとりをしています。USI プロトコルについては、ググってください。
- このやりとりしている内容をリアルタイムに閲覧ことが出来ます。
- 1. メニューの「ウィンドウ」→「デバッグ用のログ」→「デバッグウィンドウの表示」を選びます。
  - ショートカットキーとしては Ctrl + D (Debug の D) が用意されています。
  - そうするとデバッグウィンドウが出てきて、やりとりしている内容が表示されます。
  - デバッグウィンドウの一番下にフィルターを指定する欄があり、正規表現文字列で、フィルター条件を書くことが出来ます。指定した正規表現文字列にマッチする行だけが表示されるようになります。
  - 思考エンジンとのやりとりは、「1>」とある左の数字は pipe id です。複数の思考エンジンを同時に動かしていると、それぞれここの数値が変わります。
  -「>」とあるのは、エンジン側から GUI が受信したの意味です。「<」ならば、GUI からエンジン側に送信したの意味です。
- 上述の「デバッグウィンドウ」に表示されている文字列をファイルに書き出すことが出来ます。
- 1. メニューの「ウィンドウ」→「デバッグ用のログ」→「ファイルへのロギング開始」を選びます。
  - そうすると MyShogi.exe のあるフォルダにログ・ファイルが書き出されます。(ファイル名は現在時刻から生成される名前になります。)
- 2. 終了したい時は、メニューの「ウィンドウ」→「デバッグ用のログ」→「ファイルへのロギング終了」を選びます。
- デバッグウインドウ上で複数行を選択して、Ctrl+C でクリップボードにコピーすることが出来ます。

エンジン定義ファイルの書き方

- 1) MyShogi.exe の存在するフォルダ配下にある engine/ というフォルダのなかにフォルダを作成してそこに思考エンジンを配置します。
- 2) 思考エンジンのフォルダのなかに “EngineDefine.xml” というファイル名のファイルを配置します。
- このフォーマットについては MyShogi.Model.Shogi.EngineDefine にある EngineDefine クラスを参考にしてください。(このクラスを serializer で書き出しているだけなので…)
- 3) 思考エンジン本体は、
- Yaneuraou2018_kpp_kkpt_avx2.exe
- Yaneuraou2018_kpp_kkpt_sse42.exe
- Yaneuraou2018_kpp_kkpt_sse41.exe
- Yaneuraou2018_kpp_kkpt_sse2.exe
- Yaneuraou2018_kpp_kkpt_nosse.exe のようなファイル名にします。(詳しくは EngineDefine クラスのところの説明を参考にすること)
- 4) 評価関数ファイルは、MyShogi.exe の存在するフォルダ配下の eval/に配置したいので、例えば、やねうら王であれば、“engine_options.txt” を用いて、このファイルに > option name EvalDir type string default ../../eval/yaneuraou2018_kppkkpt > option name EvalShare type check default true > option name BookDir type string default ../../book のように書いて、EvalDir などのデフォルト値を変更しておくと使い勝手が良いと思います。

Mac、Linux 向けのビルド情報

MyShogi は Windows 向けに開発されていますが、機種依存するコードは最小限となっているので、他の環境でも頑張れば動くようです。

- Mac、Linux で動作させるには