

目次

Mac、Linux で動作させるには？	1
ビルドする上で注意点	1
将棋神やねうら王を Mac で	1
将棋神やねうら王を Linux で	2
Mono のバグ	3
音声・画像素材について	3
思考エンジンについて	4
思考エンジンのコンパイル手順	4

Mac、Linux で動作させるには？

- MyShogi V1.32 ではコードの修正なしに Mono でコンパイルでき、Mac/Linux で動作するはずで
す。
- 参考記事 (古い記事も含まれています)
- MyShogi を Mac 環境で動かす
 - jnory さんの fork : <https://github.com/jnory/MyShogi>
 - AI 将棋ソフト『MyShogi』を MacBookPro でビルド&遊んでみた: <https://amado.hatenablog.com/entry/20180928>
 - AI 将棋ソフト『MyShogi』を MacBookPro でビルド&遊んでみた 2 (本家版):
<https://amado.hatenablog.com/entry/20181028/1540723499>
- MyShogi を Linux 環境で動かす
 - やねうら王 MyShogi を Linux でビルドしてみた: <http://hennohito.cocolog-nifty.com/blog/2018/06/myshogilin69.html>
 - Ubuntu で『将棋神 やねうら王』: <http://fxst24.blog.fc2.com/blog-entry-545.html>

ビルドする上で注意点

- Mono のコンパイラ (mcs) が現状 C#7.2 までしか対応しておらず、本プロジェクトでは C#7.3 をターゲットとしているので、そのままとコンパイルが通りません。
- MsBuild でビルドすると例外が出たときに行番号が表示されなくて困るのですが、mcs はまだ C#7.2 までしか対応していないのでビルドできません。
- mcs が C#7.3 まで対応したら、mcs でビルドするようにすべきだと思います。

将棋神やねうら王を Mac で

- Mono のコンパイラが動くようにする。(上の「AI 将棋ソフト『MyShogi』を MacBookPro でビルド&遊んでみた」を参考に)

- git clone > git clone https://github.com/yaneurao/MyShogi.git
- ビルド方法 > msbuild MyShogi.sln /p:Configuration=macOS
- 起動方法 > mono -arch=32 MyShogi/bin/macOS/MyShogi.exe
- (V1.30) 表示設定のフォント選択ダイアログがフリーズするらしいです。(Linux だとこの問題は起きない) Mac 用の Mono のバグではないかと…。
- (V1.32) libsoundio を用いて音が鳴るようになりました。サウンド再生のために別途以下のライブラリが必要です。https://github.com/jnory/MyShogiSoundPlayer/releases/
- (V1.32) 棋譜ウィンドウ、常に現在の局面が選択されている状態にしたいのですが、ListView の OwnerDraw、Mono でバグるのでいまのところ実現できません。
- (V1.32) 棋譜ウィンドウ、メインウィンドウに埋め込み状態からフロート状態に変更したときに再描画されません。
 - DockStyle の変更によって resize イベント起きないという Mono の bug です。回避が難しいので、そのあと棋譜ウィンドウをリサイズして再描画を促すか、再起動するかしてください。
 - 検討ウィンドウ、メインウィンドウに埋め込んでいると再描画されないことがあるようです。気になる人はフロート状態で使ってみてください。
- すべての機能を動作確認できているわけではありません。
- 参考)
- 『将棋神やねうら王』の Mac 対応の作業報告: http://yaneuraou.yaneu.com/2018/09/30/%E3%80%8E%E5%B0%86%

将棋神やねうら王を Linux で

- ビルド方法 > msbuild MyShogi.sln /p:Configuration=LINUX
- 起動方法 > MyShogi/bin/LINUX/MyShogi.exe
- その他、依存ライブラリ
- 棋譜のクリップボードへのコピー、クリップボードからのペーストには、xclip が /usr/bin/xclip にインストールされている必要があります。
- サウンド再生は、Mac 用と同じく、libsoundio を wrap した jnory さんのライブラリ (実行ファイル) に依存します。
 - https://github.com/jnory/MyShogiSoundPlayer/releases/tag/v0.1.2
 - このバージョンで Ubuntu18.04 で正常に音声再生されることを確認しました。
- 詳細は上の「将棋神やねうら王を Mac で」を参考にしてください。
- Ubuntu で『将棋神 やねうら王』: http://fxst24.blog.fc2.com/blog-entry-545.html
- (V1.32) この手順で Hyper-V + Ubuntu18.04 で動作することを確認しました。[2018/10/23 14:30]

Mono のバグ

- Mono(Mac/Linux 共通)
- ListView の EnsureVisible で例外が出る。Visible=false のときにスクローバーの高さの計算を間違えるようだ。
 - Visible=false のときは EnsureVisible を呼び出さないようにして回避。
- GDI+ で alpha channel のある画像の転送でゴミが出る。(alpha = 0 の部分の画像の転送がおかしい)
 - 自前で転送するコードを書いて回避。
- Graphics.DrawImage() で転送元が半透明かつ、転送先が CreateBitmap() した Bitmap だと転送元の alpha が無視される
 - 転送先が 24bppRgb だとこの問題が出るようなので、Mono の時だけ転送先を 32bppArgb に変更して回避。
- Mono(Mac)
- ファイルダイアログを出すところでフリーズ。回避しようがないのであぼーん。
- Mono(Linux)
- ListViewEx で OwnerDraw にすると無限再帰になってメッセージの処理ができなくなる。これも Mono(Linux) のバグくさい。Mac ではどうだかわからない。
 - DrawSubItem のイベントで描画したときに画面が汚れた判定になっていて、再描画イベントがまた飛んでくるのがおかしい。
 - これのせいでメッセージを処理できなくなり、メニューなどが描画されない。
 - Mono のときだけ ListView の OwnerDraw で描画するのをやめる。
- メニューの「設定」の項目から右側の項目、文字フォントの変更が反映していない。
 - MenuStrip にぶらさげている ToolStripMenuItem の、ambient property になっている Font、親の Font が反映しないバグ。
 - 自前ですべてにフォントを設定することで回避。
- 棋譜ウィンドウをフロートに変更したときに再描画されない。
 - DockStyle の変更によって resize イベント起きないという Mono の bug のため。回避が難しいので、とりま放置。

音声・画像素材について

- 音声・画像素材など足りない素材は『将棋神やねうら王』のインストール先フォルダからコピーしてきてください。(そのうち整理して公開する予定です)

- フリーの画面素材: <https://github.com/jnory/MyShogi>
- フリーの音声素材: <https://github.com/matarillo/MyShogiSound> cf. <https://twitter.com/matarillo/status/1053131>
- その他、足りない素材は、『将棋神やねうら王』の Updater のなかに入っているかも。

思考エンジンについて

- 『将棋神やねうら王』の各思考エンジンを Mac/Linux 上でコンパイルする必要があります。(Windows 向けの実行ファイルをそのまま動かすことは出来ません)
- 『tanuki-2018』→ NNUE 型
- 『tanuki- SDT5』, 『Qhapaq 2018』『読み太 2018』→ KPPT 型
- 『やねうら王 2018』→ KKPT 型
- 『tanuki-詰将棋』→ MATE_ENGINE の 4 種類あります。

思考エンジンのコンパイル手順

- やねうら王の makefile を用いて clang でコンパイルします。

```
mingw32-make clean YANEURAOU_EDITION=YANEURAOU_2018_TNK_ENGINE
```

```
mingw32-make -j8 avx2 COMPILER=clang++ YANEURAOU_EDITION=YANEURAOU_2018_TNK_ENGINE
```

YaneuraOu-by-gcc.exe というファイルが出来ますので、これを MyShogi.exe 配下の以下のところにコピーします
 engine/tanuki2018/YaneuraOu2018NNUE_avx2.exe

以下、各CPUのファイルをビルドする例です。

```
mingw32-make clean YANEURAOU_EDITION=YANEURAOU_2018_TNK_ENGINE
```

```
mingw32-make -j8 avx2 COMPILER=clang++ YANEURAOU_EDITION=YANEURAOU_2018_TNK_ENGINE
```

```
cp YaneuraOu-by-gcc.exe XXX/engine/tanuki2018/YaneuraOu2018NNUE_avx2.exe
```

```
mingw32-make clean YANEURAOU_EDITION=YANEURAOU_2018_TNK_ENGINE
```

```
mingw32-make -j8 sse42 COMPILER=clang++ YANEURAOU_EDITION=YANEURAOU_2018_TNK_ENGINE
```

```
cp YaneuraOu-by-gcc.exe XXX/engine/tanuki2018/YaneuraOu2018NNUE_sse42.exe
```

```
mingw32-make clean YANEURAOU_EDITION=YANEURAOU_2018_TNK_ENGINE
```

```
mingw32-make -j8 sse41 COMPILER=clang++ YANEURAOU_EDITION=YANEURAOU_2018_TNK_ENGINE
```

```
cp YaneuraOu-by-gcc.exe XXX/engine/tanuki2018/YaneuraOu2018NNUE_sse41.exe
```

```
mingw32-make clean YANEURAOU_EDITION=YANEURAOU_2018_TNK_ENGINE
```

```
mingw32-make -j8 sse2 COMPILER=clang++ YANEURAOU_EDITION=YANEURAOU_2018_TNK_ENGINE
```

```
cp YaneuraOu-by-gcc.exe XXX/engine/tanuki2018/YaneuraOu2018NNUE_sse2.exe
```

```
mingw32-make clean YANEURAOU_EDITION=YANEURAOU_2018_TNK_ENGINE
```

```
mingw32-make -j8 COMPILER=clang++ tournament YANEURAOU_EDITION=YANEURAOU_2018_TNK_ENGINE
```

```

cp YaneuraOu-by-gcc.exe XXX/engine/tanuki2018/YaneuraOu2018NNUE_tournament.exe

mingw32-make clean YANEURAOU_EDITION=YANEURAOU_2018_TNK_ENGINE
mingw32-make -j8 evallearn COMPILER=g++ YANEURAOU_EDITION=YANEURAOU_2018_TNK_ENGINE
cp YaneuraOu-by-gcc.exe XXX/engine/tanuki2018/YaneuraOu2018NNUE_learn_avx2.exe

mingw32-make clean YANEURAOU_EDITION=YANEURAOU_2018_TNK_ENGINE
mingw32-make -j8 evallearn-sse42 COMPILER=g++ YANEURAOU_EDITION=YANEURAOU_2018_TNK_ENGINE
cp YaneuraOu-by-gcc.exe XXX/engine/tanuki2018/YaneuraOu2018NNUE_learn_sse42.exe

// 32bit 環境向け

mingw32-make clean YANEURAOU_EDITION=YANEURAOU_2018_TNK_ENGINE
mingw32-make -j8 nosse COMPILER=clang++ YANEURAOU_EDITION=YANEURAOU_2018_TNK_ENGINE
cp YaneuraOu-by-gcc.exe XXX/engine/tanuki2018/YaneuraOu2018NNUE_nosse.exe

// KPPT 型、AVX2 用にコンパイルする例

mingw32-make clean YANEURAOU_EDITION=YANEURAOU_2018_OTAFUKU_ENGINE_KPPT
mingw32-make -j8 avx2 COMPILER=clang++ YANEURAOU_EDITION=YANEURAOU_2018_OTAFUKU_ENGINE_KPPT
cp YaneuraOu-by-gcc.exe XXX/engine/yomita2018/YaneuraOu2018KPPT_avx2.exe

// KKPT 型、AVX2 用にコンパイルする例

mingw32-make clean YANEURAOU_EDITION=YANEURAOU_2018_OTAFUKU_ENGINE_KPP_KKPT
mingw32-make -j8 avx2 COMPILER=clang++ YANEURAOU_EDITION=YANEURAOU_2018_OTAFUKU_ENGINE_KPP_KKPT
cp YaneuraOu-by-gcc.exe XXX/engine/yaneuraou2018/Yaneuraou2018_kpp_kkpt_avx2.exe

// tanuki-詰将棋エンジンを AVX2 用にコンパイルする例

mingw32-make clean YANEURAOU_EDITION=MATE_ENGINE
mingw32-make -j8 avx2 COMPILER=clang++ YANEURAOU_EDITION=MATE_ENGINE
cp YaneuraOu-by-gcc.exe XXX/engine/tanuki_mate/2018-Otafuku/tanuki_mate_avx2.exe

```