

//-----//

Shogiban2 version 0.3

//-----//

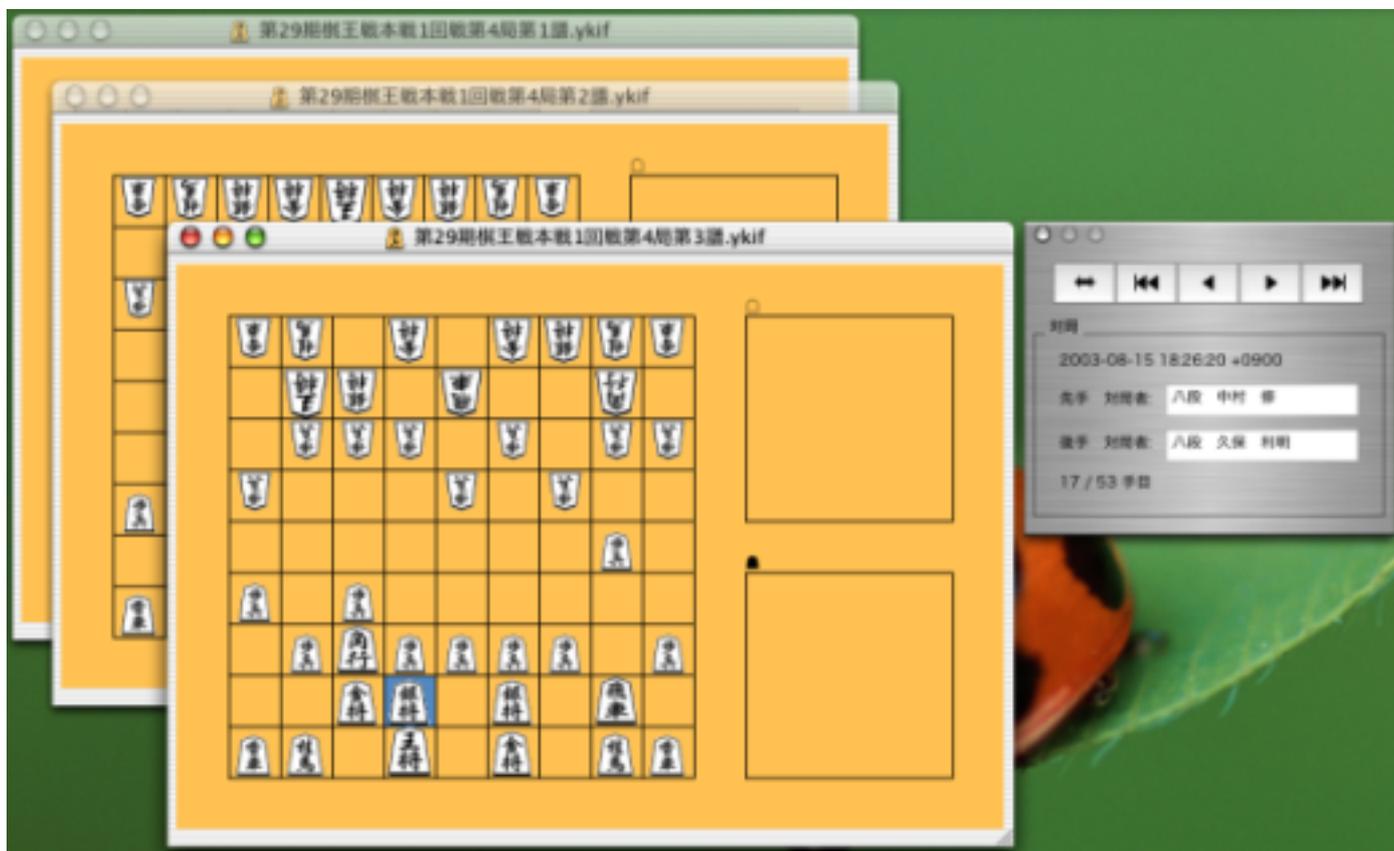


図 1) 実行画面

【インストール】

Shogiban2_03.zip を展開して作られる Shogiban2_03 フォルダを '/Users/あなたのアカウント/Applications' に移動してください。('あなたのアカウント' フォルダに 'Applications' フォルダがない場合は、新規に作成してください。)

【アンインストール】

'/Users/あなたのアカウント/Applications' の 'Shogiban2_03' フォルダ、
(中に含まれるすべてのファイル)

'/Users/あなたのアカウント/ライブラリ/Preferences' フォルダの jp.koyama.Shogiban2.plist
を削除してください。

【説明】

シンプルな将棋の棋譜管理ソフト。

変更点

2006/02/01(Wed)

< Shogiban2 version 0.2 からの変更点 >

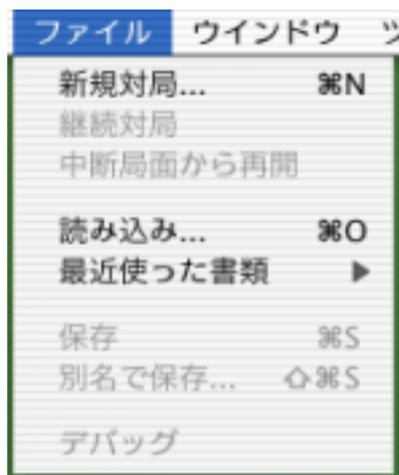
Universal Binary 化。

2004/11/26(Fri)

< Shogiban2 version 0.1 からの変更点 >

棋譜が入力状態のままウィンドウを閉じると保存するか確認するようにした。一手進める、一手戻すボタンを押したままにすると自動的に進むようにした。駒が成る場合に表示される駒の絵がぼけていたのをはっきり表示するように変更した。

メニュー



新規の棋譜を入力

棋譜を入力するにはメニュー 'ファイル' から '新規対局...' を選択してください。
対局者を入力した後（一度入力したデータは変更できません。）、駒を移動して入力してください。
間違えた箇所に駒を移動してしまった場合は、'一手取り消しボタン' を押してください。
'最初から入力ボタン' を押すと最初から入力することになります。

棋譜を入力し終えたなら、メニューから '別名で保存' を選んでください。
拡張子は ykif と自動的に付けられるので入力する必要はありません。
注意、入力した後必ず保存してください。保存しないと棋譜が再現できません。



棋譜を入力できる状態になるとウィンドウのクローズボタンに黒い点が表示されます。
(ダーティーフラグ [黒い点] がセットされます。)
[入力状態のままウィンドウを閉じると保存するか確認のダイアログが表示されます。](#)



継続対局

棋譜を読み込んだ後、駒を入力したい局面まで進めてそこで '継続対局' を選択してください。
その局面から入力することができます。
後は保存すると棋譜が再現できます。

中断局面から再開

棋譜を読み込んだ後、'中断局面から再開' を選択すると最後の局面まで駒が進み入力状態になります。

棋譜の再現

棋譜を読み込むと次の動作が可能です。
[一手戻す、一手進めるボタン](#)を押し続けると自動的に駒を進めます。



デバッグ

格納されている棋譜データをコンソール に表示します。
詳しくはソースを参照してください。

Shogiban2 version 0.1 からの移行

Finder で棋譜データをクリックし、Finder のメニュー 'ファイル' から '情報を見る' を選択してください。
下図のように表示されると思います。

'このアプリケーションで開く' の左横にある三角をクリックすると現れるポップアップボタンで Shogiban2 を選択してください。(Shogiban2 が表示されない場合は 'その他' から Shogiban2 を選んでください。)

その下の 'すべてを変更...' ボタンを押すとひとつひとつ設定する手間が省けます。

Shogiban2 version 0.1 を削除した後に棋譜データを開くと、自動的に Shogiban2 version 0.2 で開くかどうか聞いてくるので、そのままShogiban version 0.2 を選択してください。



図 2) 情報を見る

xshogi で保存したファイルの読み込み

xshogi (*1) で 'Save Game' を選択し、拡張子を xkif としてファイルに保存します。

後は Shogiban2 で開いてください。

gnushogi.xkif はコンピュータ同士の対戦結果です。

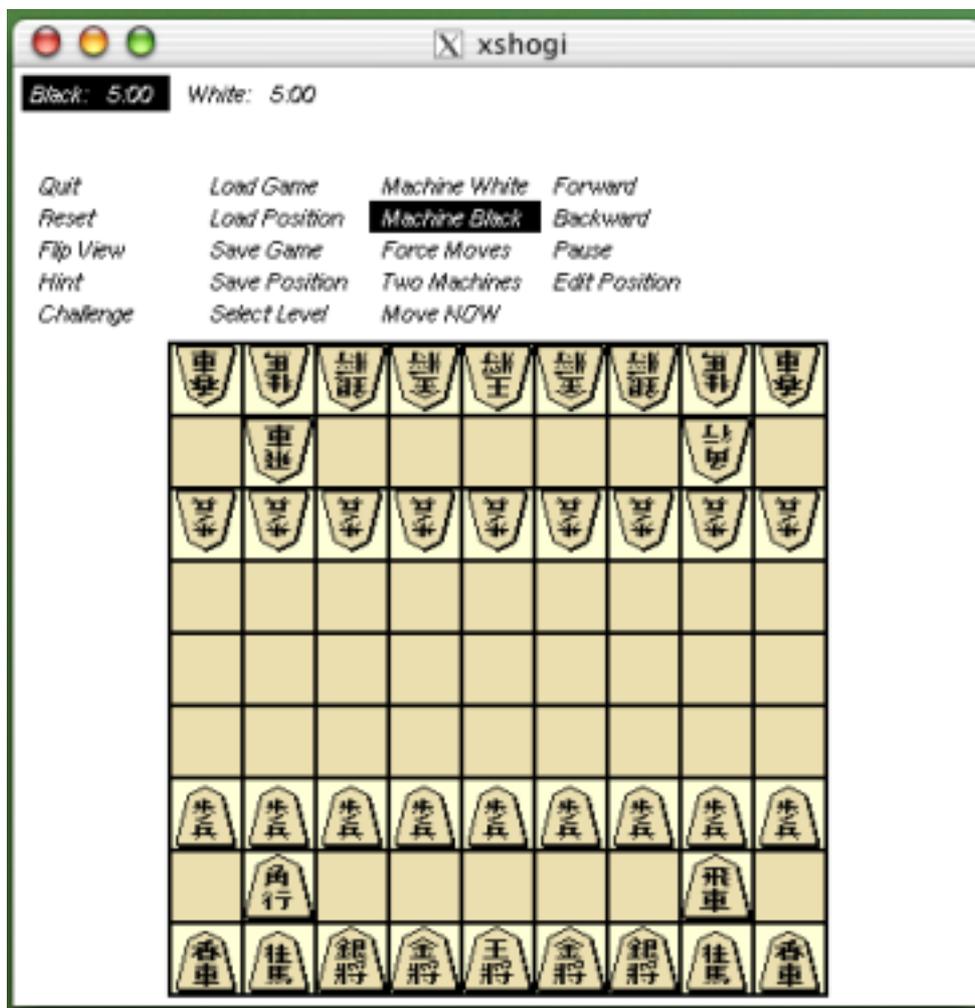


図 3) xshogi

(*1)xshogi とは X Window System で動作する X11 アプリケーションです。コンピュータと対戦できます。

Mac OS X で利用する方法を簡単に説明します。**1) xshogi をインストールする前の準備****1-1) X11.app のインストール**

ここに書かれている説明は Mac OS X 10.3 "Panther" 用ですが、Mac OS X "Tiger" でもほとんど同じだと思います。

Mac OS X version 10.3 "Panther" のカスタムインストールで X11 をインストールする必要があります。(すでに Panther がインストールされている場合は X11 以外のチェックをはずしてインストールしてください。)

インストールに成功するとフォルダ '/アプリケーション/ユーティリティ/' に X11 という名前のアプリケーションがあるはずですが。(Finder で確認してください。)



X11

1-2) Mac OS X Xcode Tools のインストール

ここに書かれている説明は Mac OS X 10.3 "Panther" 用ですが、Mac OS X "Tiger" でもほとんど同じだと思います。

Mac OS X version 10.3 "Panther" を購入すると Mac OS X Xcode Tools という CDか、Install Disc の中に Xcode Tools という名のフォルダがあると思います。(Mac OS X 10.3 "Panther" がインストールされている機種では フォルダ '/アプリケーション/Installers/Xcode Tools/' にあります。) その中の Developer.mpkg をダブルクリックしてインストールしてください。ただし、簡易インストールでは X11 SDK がインストールされないなので、カスタマイズを選択して X11 SDK にチェックを入れてください。下図はボタンがアップグレードになっていますが、一度もインストールされていないとインストールと表示されていると思います。インストールボタンを押してインストールしてください。(また X11 SDK だけインストールしたい場合は、X11 SDK だけにチェックを入れてインストールしてください。)



図 4) Mac OS X Xcode Tools

インストールに成功すると '/Developer' という名前のフォルダができていると思います。



Developer

最新の Mac OS X Xcode Tools は ADC で入手できます。

<http://developer.apple.com/ja/>

ADCへの加入は無料で入会できます。ADCに6ヶ月アクセスしていないと停止されるので注意してください。

2) xshogi のインストール

xshogi のインストールには ソースをダウンロードして自分で make する場合と Fink を利用する方法があります。Fink を利用するほうが簡単ですが、自分で make するのも悪くないと思います。あと基本的なことは Unix の本を買って勉強してください。そんなに難しいことではありません。

2-1) Fink を使ったインストール

ここに書かれている説明は Fink-0.7.1 を利用していますが、Mac OS X のそれぞれのバージョンに対応する Fink をダウンロードしてください。

Fink [<http://fink.sourceforge.net/index.php?phpLang=ja>] から [ダウンロード \[http://fink.sourceforge.net/download/index.php?phpLang=ja \]](http://fink.sourceforge.net/download/index.php?phpLang=ja) をクリックして解説してある通り、インストールしてください。簡単に説明すると

Fink 0.7.1 バイナリインストーラをクリック、ダウンロード先が表示されるので好きな場所をクリック、Fink-0.7.1-Installer.dmg のダウンロードが開始される。ダウンロードが終了するとダウンロードされたファイルをダブルクリック、マウントされたイメージを開いて Fink 0.7.1 Installer.pkg をダブルクリック、インストールが始まります。最後にシェルの設定ファイルを書き換えるかどうか聞かれるので Yes を選択する。



図 5) Fink

2-1-1) すべてのパッケージをインストールできるようにする

これは必ずしも必要ではありません。省略したい場合は、2-1-2) に進んでください。

すべてのパッケージをインストールできるようにするため、[Fink FAQ \[http://fink.sourceforge.net/faq/usage-fink.php#unstable \]](http://fink.sourceforge.net/faq/usage-fink.php#unstable) の質問 5.8にある通り、unstable を使えるようにする必要があります。

/sw/etc/fink.cnf の内容に修正を行う必要があります。

ただし、通常のエディタでは書き込めないかもしれません。ターミナル上で `sudo emacs` を使った例を記しておきます。

ターミナル [/アプリケーション/ユーティリティ/ターミナル] を起動し `cd /sw/etc` でカレントディレクトリを移動し `ls -alF` で `fink.conf` というファイルがあることを確認してください。その後 `sudo emacs` で `emacs` を起動します。それぞれのコマンドの後に `return` キーを押すのを忘れないでください。後、`sudo emacs` の後には `Password` を聞かれるので `管理者アカウントのパスワード`を入力してください。

---[ターミナル]-----

```
Yoshiaki-Koyama:~ yoshiaki$ cd /sw/etc
Yoshiaki-Koyama:/sw/etc yoshiaki$ ls -alF
total 8
drwxr-xr-x 12 root  admin 408  7 Jan 18:52 ./
:
drwxr-xr-x  5 root  admin 170  5 Dec 16:16 apt/
drwxr-xr-x  4 root  admin 136  5 Dec 16:16 dpkg/
-rw-r--r--  1 root  admin 822  5 Dec 16:33 fink.conf
drwxr-xr-x  3 root  admin 102  5 Dec 17:27 glib-2.0/
:
Yoshiaki-Koyama:/sw/etc yoshiaki$ sudo emacs
```

`C-x d` (`C-x d` は、`control` キーを押したまま `x` キーを押し、次に `d` キー を押すことを指します。) で `Dired (directory)`: と表示されるので `return` キーを押し、カーソルキーで `fink.conf` に移動し `return` キーを押してください。

後は、ファイル `/sw/etc/fink.conf` の上部に移動し、ファイルの `Trees:` の行に `unstable/main` と `unstable/crypto` を追加します。(下記 `fink.conf` の緑色の部分)

保存は `C-x C-s` です。(`C-x C-s` は、`control` キーを押したまま `x` キーを押し、次に `Control` キーを押したまま `s` キー を押すことを指します。)

---[fink.conf]-----

```
# Fink configuration, initially created by bootstrap.pl
Basepath: /sw
RootMethod: sudo
Trees: local/main stable/main stable/crypto unstable/main unstable/crypto
:
```

`emacs` を終了するには `C-x C-c` と入力してください。(`C-x C-c` は、`control` キーを押したまま `x` キーを押し、次に `Control` キーを押したまま `c` キー を押すことを指します。)

この後、ターミナルを起動し、 `fink selfupdate; fink index` を実行します。(緑色の部分を入力してください。)

---[ターミナル]-----

```
Yoshiaki-Koyama:~ yoshiaki$ fink selfupdate; fink index
```

と入力し `return` キーを押す。

2-1-2)

ターミナル [/アプリケーション/ユーティリティ/ターミナル] を起動し、 `fink scanpackages; fink index` を実行します。(緑色の部分を入力してください。)

```
---[ ターミナル ]-----
Yoshiaki-Koyama:~ yoshiaki$ fink scanpackages; fink index
```

と入力し `retun` キーを押す。
これで Fink のインストールは終了です。

2-1-3) xshogi のインストール

次に xshogi のインストールです。

ターミナルで `fink list` を実行します。表示されているリストの中に xshogi があることを確認します。(リストの中に xshogi がない場合は 2-1-1) の設定をおこなってください。)

```
---[ ターミナル ]-----
Yoshiaki-Koyama:~ yoshiaki$ fink list
:
xsane          0.89-12    Graphical frontend for sane
xscorch        0.1.16-10 Scorched Earth - "the mother of all games"
xscrabble      0901-3    Scrabble game for X
xscreensaver   4.18-1    Extremely diverse screen saver for X11
xserver        [virtual package]
xshogi         1.3-5     Japanese version of chess (under X11)
```

```
Yoshiaki-Koyama:~ yoshiaki$ fink install xshogi
```

`fink install xshogi` でインストールを開始し、`make` が実行されます。プロンプト `$` が表示され、入力待ち状態になるとインストール終了です。

4) に進んでください。

3) 自分で make する場合

GNU's Not Unix! [<http://www.gnu.org/home.ja.html>] から 'フリーソフトウェア一覧' を選択し、Games、gnushogi [<http://directory.fsf.org/games/gnushogi.html>] と選び Source tarball [<http://ftp.gnu.org/pub/gnu/gnushogi/gnushogi-1.3.tar.gz>] をダウンロードしてください。
gnushogi-1.3.tar.gz の中に xshogi も含まれています。後、xshogi の内部で gnushogi を呼び出しています。

後は ターミナル [/アプリケーション/ユーティリティ/ターミナル] で次のコマンド(緑色の部分)を実行してください。(詳しくは 展開してできる INSTALL というファイルを参照してください。)

ホームに `gnushogi-1.3.tar.gz` がある場合。

```
---[ ターミナル ]-----
Yoshiaki-Koyama:~ yoshiaki$ gnutar xzvf gnushogi-1.3.tar.gz
Yoshiaki-Koyama:~ yoshiaki$ cd gnushogi-1.3
Yoshiaki-Koyama:~/gnushogi-1.3 yoshiaki$ ./configure
Yoshiaki-Koyama:~/gnushogi-1.3 yoshiaki$ make
Yoshiaki-Koyama:~/gnushogi-1.3 yoshiaki$ sudo make install
```

私は 'NetInfo マネージャ' [/アプリケーション/ユーティリティ/NetInfo マネージャ] で メニュー 'セキュリティ/認証' を選んでから 'セキュリティ/ルートユーザを有効' を選択し、'セキュリティ/ルートパスワードの変更' でパスワードを設定しています。そのため、sudo make install ではなく

```
---[ ターミナル ]-----  
Yoshiaki-Koyama:~/gnushogi-1.3 yoshiaki$ su  
Yoshiaki-Koyama:/Users/yoshiaki/gnushogi-1.3 root# make install  
Yoshiaki-Koyama:/Users/yoshiaki/gnushogi-1.3 root# exit  
Yoshiaki-Koyama:~/gnushogi-1.3 yoshiaki$ cd ..  
Yoshiaki-Koyama:~ yoshiaki$
```

としました。

次に、ホームに .bashrc とファイルを作り、設定する必要があります。
通常のエディタでは .bashrc ファイルを編集できないので、ターミナル上で emacs を使った例を記しておきます。

ターミナル [/アプリケーション/ユーティリティ/ターミナル] を起動し emacs を起動する。

```
---[ ターミナル ]-----  
Yoshiaki-Koyama:~ yoshiaki$ emacs
```

C-x C-f (C-x C-f は、control キーを押したまま x キーを押し、次に Control キーを押したまま f キー を押すことを指します。) で Find file: と表示されるので .bashrc と入力し return キーを押してください。

後は .bashrc の中身に下の緑色の部分を入力してください。

```
---[ .bashrc ]-----  
export PATH=$PATH:/usr/local/bin:/Developer/Tools
```

保存は C-x C-s です。(C-x C-s は、control キーを押したまま x キーを押し、次に Control キーを押したまま s キー を押すことを指します。)

emacs を終了するには C-x C-c と入力してください。(C-x C-c は、control キーを押したまま x キーを押し、次に Control キーを押したまま c キー を押すことを指します。)

できれば ホーム/.profileも同じ内容にしてください。

下記のように、フォルダ /usr/local/bin にファイル gnushogi、xshogi があればインストール成功です。(フォルダ /usr/local/bin は Finder では見えなくなっています。拙作 YKExplorer2 を使えば簡単に表示でき、Finder で表示させることもできます。)

```
---[ ターミナル ]-----  
Yoshiaki-Koyama:~ yoshiaki$ ls -alF /usr/local/bin  
total 552  
drwxr-xr-x  4 root  wheel   136 13 Oct 22:17 ./  
drwxr-xr-x  6 root  wheel   204 13 Oct 22:17 ../  
-rwxr-xr-x  1 root  wheel 155524 13 Oct 22:17 gnushogi*  
-rwxr-xr-x  1 root  wheel 126248 13 Oct 22:17 xshogi*
```

4) xshogi の実行

フォルダ '/アプリケーション/ユーティリティ/' にある X11 を起動します。xterm というウィンドウが開きますから、そこで

```
Yoshiaki-Koyama:~ yoshiaki$ xshogi &
```

と入力し return キーを押すと xshogi が起動します。終了するには Quit を押してください。

Quit を選択しないでウィンドウを閉じると gnushogi というプロセスが終了しない場合があります。その時は アクティビティモニタ [/アプリケーション/ユーティリティ/アクティビティモニタ] で終了させてください。

これだと不便なので X11 のメニュー 'Applications/Customize' を選択し、'Add Item' ボタンを押して、

Name : xshogi

Command xshogi

と設定してください。設定を終了したら、'Done' ボタンを押してください。

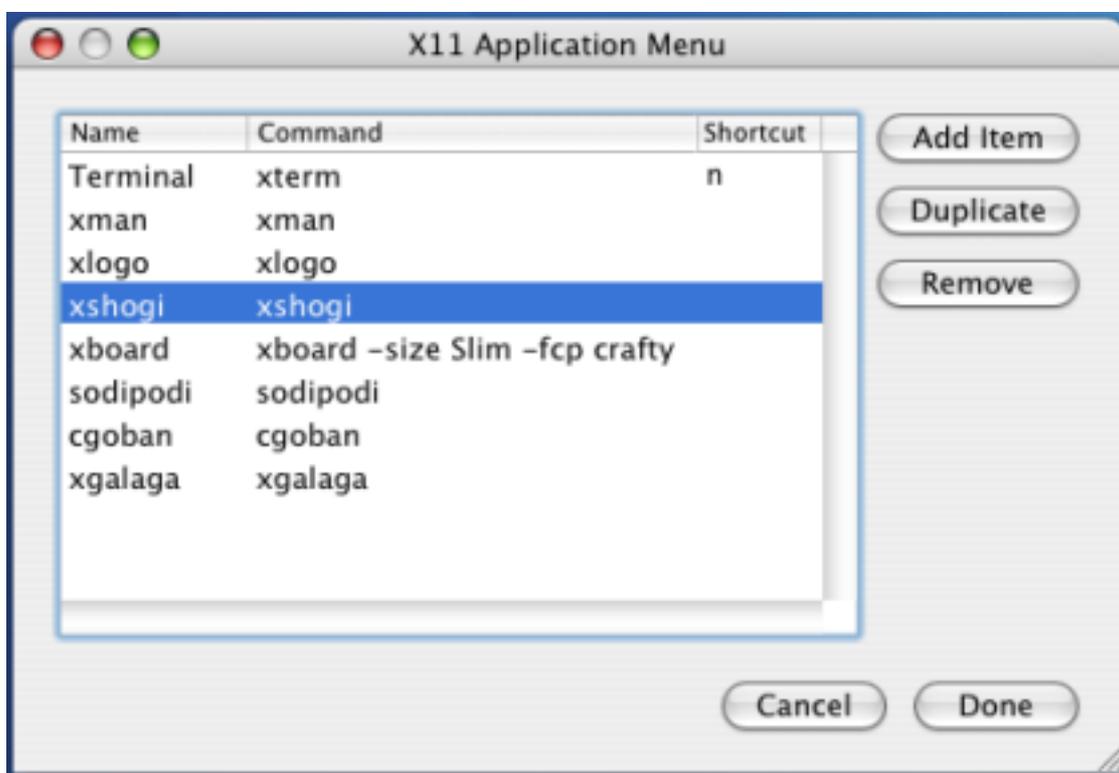


図 6) X11

X11 のメニュー 'Application/xshogi' を選択すれば、xshogi が実行されます。

【著作権】

フリーウェア、含まれているソースの著作権は私に。変更して公開する場合は、私にメールを下さい。

将棋駒データ

<http://www.inv.co.jp/~yoshio/DB/index.html>

【保証】

著者は本ソフトの使用で起こった障害に対しての責任、保証義務を一切負わないものとします。

【動作環境】

動作確認は、Mac OS X バージョン 10.4.4 ビルド 8G32でおこなっています。

【独り言】

2006/02/01(Wed)

Gimp.app [<http://gimp-app.sourceforge.net/>] を参考に X11 ベースのアプリケーション xshogi をダブルクリックするだけで起動する xshogi.app を作成してみました。(私の環境以外で動作するか確認はしていません。)



図 7) xshogi.app

Fink を使ったり、ソースからビルドする必要がありません。

私の技術力不足から無駄な部分も多いので公開するかどうか悩んでいます。

私のホームページを '将棋' というキーワード訪れる人も多いので需要はあると思うのですが！

【開発環境】

Xcode バージョン 2.2

コンポーネントのバージョン

Xcode IDE: 655.0

Xcode Core: 658.0

ToolSupport: 651.0

Interface Builder Version 2.5.3 (443)

The X Window System X11 1.0.1 - XFree86 4.3.0

Gimp.app

EGWORD 14

【参考文献】

[Apple Developer Documentation](http://developer.apple.com/documentation/)

<http://developer.apple.com/documentation/>

[Cocoa Documentation](http://developer.apple.com/documentation/Cocoa/Cocoa.html)

<http://developer.apple.com/documentation/Cocoa/Cocoa.html>

[Cocoa API Reference](http://developer.apple.com/referencelibrary/API_Fundamentals/Cocoa-api-date.html)

http://developer.apple.com/referencelibrary/API_Fundamentals/Cocoa-api-date.html

[Application Kit Reference for Objective-C](http://developer.apple.com/documentation/Cocoa/Reference/ApplicationKit/ObjC_classic/index.html)

http://developer.apple.com/documentation/Cocoa/Reference/ApplicationKit/ObjC_classic/index.html

[Foundation Reference for Objective-C](http://developer.apple.com/documentation/Cocoa/Reference/Foundation/ObjC_classic/index.html)

http://developer.apple.com/documentation/Cocoa/Reference/Foundation/ObjC_classic/index.html

Universal Binary Programming Guidelines

http://developer.apple.com/documentation/MacOSX/Conceptual/universal_binary/index.html

Universal Binary Programming Guidelines - 日本語翻訳

http://developer.apple.com/ja/documentation/MacOSX/Conceptual/universal_binary/index.html

書籍名 : MacOS X プログラミング入門 Objective-C

出版社名 : 広文社

著者名 : 荻原 剛志 著

価格 : 4500円

書籍名 : Mac OS X Cocoa プログラミング

出版社名 : 株式会社 ピアソン・エデュケーション

著者名 : アーロン・ヒロガス 著 村上 雅章 訳

価格 : 3700円

書籍名 : Happy Macintosh Developing Time—はじめてさわる、はじめてわかる!!Xcodeに完全対応、Web Kitも解説

出版社名 : ビーエヌエヌ新社

著者名 : 木下 誠 著

価格 : 3000円

書籍名 : MacOS X プログラミング入門 Cocoa はやっぱり！

出版社名 : 広文社

著者名 : 鶴藺賢吾 著

価格 : 3800円

書籍名 : はじめての Mac OS X Cocoa プログラミング

出版社名 : ローカス

著者名 : 中村 正弘 著

価格 : 4800円

書籍名 : Java 言語で学ぶ デザインパターン入門

出版社名 : ソフトバンク パブリッシング

著者名 : 結城 浩 著

価格 : 3800円

【メール】

御意見、感想はこちらまで。

小山 嘉昭

KOYAMA Yoshiaki

koyama-@za2.so-net.ne.jp

yos_koyama@mac.com

Yoshiaki's HomePage

<http://koyama-cool.ne.jp/>

<http://www001.upp.so-net.ne.jp/koyama/>