

5 五将棋をプレイするゲーム AI の強化

MI/CS 実験第二 3a
第 2 回

November 29, 2023

サンプルプログラム

こちらのリンクからソースコードを入手してください

- https://drive.google.com/drive/folders/1YzUONqkFZ5Dd6AWI7lrG-dmzHrroVX6C?usp=share_link

コンパイル方法 (Windows)

Visual Studio 推奨

1. <https://visualstudio.microsoft.com/ja/downloads/> から Visual Studio をダウンロード
2. mics2_3a.sln を開く
3. ソリューション構成を「Release」, ソリューションプラットフォームを「x64」とする
4. ビルド実行. 実行ファイルは build 以下に生成される

コンパイル方法 (Mac・Ubuntu など)

g++と make コマンドを実行できるようにする． GCC のバージョンは 8.1 以降とする

```
$ cd mics2_3a/source # 移動  
$ make COMPILER=g++ -j  
$ ./minishogi-by-gcc # 実行
```

コンパイル方法 (CED 端末)

CED 端末を使用する場合, COMPILER=g++8 とする

```
$ cd mics2_3a/source # 移動  
$ make COMPILER=g++8 -j  
$ ./minishogi-by-gcc # 実行
```

サンプルプログラムについて

- コンピュータ将棋エンジンやねうら王¹をベースとした5五将棋エンジン
- 開発言語：C++

ファイルの説明

source/

bitboard.h/.cpp	ビットボード
config.h	構成, 設定
evaluate.h/.cpp	評価関数
misc.h/.cpp	バージョン情報や乱数生成など
movegen.cpp	指し手生成
position.h/.cpp	局面の更新処理
search.h/.cpp	探索部
types.h/.cpp	各種構造体の定義
usi.h/.cpp	USI プロトコルによる入出力
extra/	
bitop.h	ビット操作
macros.h	マクロ
rp_cmd.cpp	ランダムプレイヤーによるテスト
user_test.cpp	ユーザーのテスト用

コマンドの実行

エンジンを実行するとコマンドの入力を受け付ける
各コマンドに対する処理は `usi.cpp` に記述されている

コマンドの例

<code>usi</code>	エンジン情報
<code>isready</code>	局面の初期化
<code>d</code>	盤面の表示
<code>compiler</code>	コンパイラ情報
<code>mated</code>	現局面が詰んでいるか

user コマンド

user コマンドを実行すると, extra/user_test.cpp にある関数 user_test を呼び出す

```
void user_test(Position& pos, std::istream& is)
{
}
```

```
void user_test(Position& pos, std::istream& is)
{
    std::cout << "Hello world" << std::endl;
}
```

標準出力では printf 関数を使うこともできるが、C++では **std::cout** を使う方法がある。user コマンドを実行すると "Hello world" が出力される

```
#include "../position.h" // 追加
void user_test(Position& pos, std::istream& is)
{
    std::cout << pos << std::endl;
}
```

isready コマンドを実行後 (局面の初期化が必要) に user コマンドを実行すると現局面が出力される

```
void user_test(Position& pos, std::istream& is)
{
    int x;
    is >> x;
    std::cout << x << std::endl;
}
```

コマンドに引数をとることができる. "user 5"と入力すると「5」が出力される

主な列挙型

型名	意味	例
Color	手番	BLACK (先手), WHITE (後手)
File	筋	FILE_1 (1 筋), FILE_2 (2 筋)
Rank	段	RANK_3 (3 段目), RANK_4 (4 段目)
Square	升目	SQ_11 (右上), SQ_55 (左下)
Value	評価値	VALUE_MATE (詰み)
Piece	駒	B_PAWN (先手の歩), W_SILVER (先手の銀)
Move	指し手	MOVE_RESIGN (投了)

これらはすべて `types.h` に定義されており, `std::cout` で標準出力できる.

主な構造体

型名	役割
Bitboard	ビットボード
Position	局面
StateInfo	局面情報
ExtMove	指し手と評価値がペアになった構造体
LimitsType	持ち時間設定など

参考文献

- C++入門 (かいてい.net)
- cpprefjp - C++日本語リファレンス
- やねうら王オープンソースプロジェクト (やねうら王開発者による解説記事)
- SFEN 文字列について (Qiita)

