

五目並べプログラム

対戦型五目並べのプログラムを作成するための枠組とサンプル

5目並べのルールは、下記を参照すること。（禁じ手の説明がある）

[五目ルール2](#)←-ここをクリック

ファイル構成

本枠組み（ディレクトリ）には、下記ファイルが含まれている。

gomoku.c

五目並べのメイン。
勝ち負け判定、禁じ手判定、盤の表示などの処理がある。

gomoku.h 各定義など。

gomoku_ai.c 五目並べAI 下記関数のみで構成される。

calcPutPos関数：コンピュータの出す手を計算する。
inputPutPos（石を置く場所を入力する関数）から呼び出される。

引数

盤の状態とコンピュータが白石か黒石かが入力、位置のx座標、y座標が出力。

戻り値

正常終了で0

```
int calcPutPos(int board[][BOARD_SIZE], int com, int *pos_x, int *pos_y)
```

Makefile

五目並べプログラム用Makefile

gomoku.c

対戦ゲームの枠組み。プログラム内にコメントで説明を入れてあるので参照すること。このファイルは変更してはならない。

大域変数

- check

有効にする禁じ手を設定する。（各回ごとに違う設定にする予定）

- 1:三三のみ
- 2:四四のみ
- 3:三三と四四
- 4:長連のみ
- 5:三三と長連
- 6:四四と長連
- 7:全部

関数

int main()

メイン関数

引数

無

戻り値

正常 0

変数

```
int board[BOARD_SIZE][BOARD_SIZE];  
    盤面 (board[y座標][x座標の順])  
int which_turn;  
    手番 (どちらが石を打つかを表す変数)  
int com;  
    コンピュータの石の色を表す変数
```

以下の関数を勝敗が決まるまで、繰り返し呼び出す。

- int inputPutPos(int board[][BOARD_SIZE], int which, int com)

碁石の置く場所の入力処理を行う関数。

引数

int board[][BOARD_SIZE] 盤面
int which 手番（白の番か黒の番か）を保持する
int com コンピュータの石の色を保持する

戻り値

正常 0

コンピュータの番の時は、**calcPutPos** 関数を呼び、人の番のときは、入力を促すメッセージを出力し、入力を待つ。入力された値が、枠外かすでに石が置かれている場所の場合は再入力をさせる。

- void changeTurn(int *which_turn)

手番の交代処理を行う関数

引数

int which_turn 手番を保持する

戻り値

なし

- int checkOutPos(int x, int y)

枠外か否かを判定する関数

引数

int x x座標
int y y座標

戻り値

枠内 1
枠外 0

- int gameInit(int board[][BOARD_SIZE], int *which_turn, int *com)

ゲーム情報を初期化する関数

引数

int board[][BOARD_SIZE] 盤面
int which 手番（白の番か黒の番か）を保持する
int com コンピュータの石の色を保持する

戻り値

正常 0

- void boardInit(int board[][BOARD_SIZE])

盤面の初期化をする関数

引数

int board[][BOARD_SIZE] 盤面

戻り値

正常 0

- void boardPrint(int board[][BOARD_SIZE])

盤面を表示する関数

引数

int board[][BOARD_SIZE] 盤面

戻り値

なし

- int gameEndProcess(int board[][BOARD_SIZE])

ゲームの終了判定処理を行う関数

check 変数の値に応じて、禁じ手、勝敗の判定を行う。詳細は、プログラム内に記載しているので参照してほしい。

引数

int board[][BOARD_SIZE] 盤面

戻り値

勝負がついたとき 1

それ以外 0

- int lenCheck(int board[][BOARD_SIZE], int x, int y)

5連チェック

座標(x,y)が5連になっているかをチェックする

引数

int board[][BOARD_SIZE] 盤面

int x x座標

int y y座標

戻り値

5連になっている 1
5連になっていない 0

- int isSan(int board[][BOARD_SIZE], int x, int y)

3連チェック（禁じ手チェック用）

引数

int board[][BOARD_SIZE] 盤面
int x x座標
int y y座標

戻り値

3連になっている 1
3連になっていない 0

- int isShi(int board[][BOARD_SIZE], int x, int y)

4連チェック（禁じ手チェック用）

引数

int board[][BOARD_SIZE] 盤面
int x x座標
int y y座標

戻り値

4連になっている 1
4連になっていない 0

- int isRoku(int board[][BOARD_SIZE], int x, int y)

6連チェック（禁じ手チェック用）

引数

int board[][BOARD_SIZE] 盤面
int x x座標
int y y座標

戻り値

6連になっている 1
6連になっていない 0

gomoku_ai.c

コンピュータの出す手を考える関数。この中身を皆さんが作成する。**関数名、関数が受け取る引数とその変数型、関数の戻り値の変数型変えてはならない。** サンプルとして、前に置いた場所の近くにランダムに置くものを作成している。プログラムにコメントを入れてあるので、参考してほしい。

```
int calcPutPos(int board[][BOARD_SIZE], int com, int *pos_x, int *pos_y)
```

定義（gomoku.hで定義されている）

```
#define BOARD_SIZE 15 // 盤面サイズ 15 * 15
#define STONE_SPACE 0 // 盤面にある石 なし
#define STONE_BLACK 1 // 盤面にある石 黒
#define STONE_WHITE 2 // 盤面にある石 白
```

引数

```
int board[][BOARD_SIZE]
    盤面（BOARD_SIZE*BOARD_SIZE） board[y座標][x座標]
    各要素は、STONE_SPACE, STONE_BLACK, STONE_WHITE の値をとる
    x座標, y座標は、左上が（0, 0）である。

int com
    コンピュータの石の色を表す変数
    STONE_BLACK, STONE_WHITEのどちらかを取る。

int *pos_x, *pos_y
    コンピュータの出す手を返す引数（ポインタ）
    pos_x は、x座標を入れるポインタ, pos_y は、y座標を入れるポインタ。
```

戻り値

正常 0