

連珠 (CUI - C Language)

S.Matoike

目次

0.1	連珠（五目並べ）	1
0.1.1	定数定義と関数プロトタイプ	1
0.1.2	主処理	2
0.1.3	盤面の表示	2
0.1.4	結果の表示	3
0.1.5	手の入力	3
0.1.6	勝敗の判定	4
	参考文献	5

0.1 連珠（五目並べ）

0.1.1 定数定義と関数プロトタイプ

ソースコード 1 定数定義：連珠（五目並べ）

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <sysexits.h>
4  //盤面の広さ設定
5  #define BOARD_SQUARE 20
6  //盤面の出力
7  void Board_Output(void);
8  //入力処理
9  void Game(void);
10 //盤面の調査
11 void Board_Scan(int x, int y);
12 //置いた場所を中心に並ぶ個数の調査
13 int Board_Scan_Sub( int x, int y, int move_x, int move_y );
14 //ゲーム終了処理
15 void Finish(void);
16 int board[BOARD_SQUARE][BOARD_SQUARE] = {{0}};
17 int player_number = 1;
```

0.1.2 主処理

ソースコード 2 主処理：連珠（五目並べ）

```
1  int main(void){
2      int i;
3      Board_Output();
4      printf("ゲームスタート!\n");
5      for( i = 0; i < (BOARD_SQUARE * BOARD_SQUARE); i++ ){
6          Game();
7          Board_Output();
8          if( player_number < 2 ) player_number++;
9          else player_number = 1;
10     }
11     return 0;
12 }
```

0.1.3 盤面の表示

ソースコード 3 盤面の出力：連珠（五目並べ）

```
1  //盤面の出力-----
2  void Board_Output(void){
3      int i, j;
4      printf(" ");
5      for( i = 0; i < BOARD_SQUARE; i++ ){
6          printf("%2d",i);
7      }
8      puts("");
9      for( i = 0; i < BOARD_SQUARE; i++ ){
10         printf("%2d",i);
11         for( j = 0; j < BOARD_SQUARE; j++ ){
12             switch( board[j][i] ){
13                 case 0:    printf("・"); break;
14                 case 1:    printf("○"); break;
15                 case 2:    printf("●"); break;
16             }
17         }
18         puts("");
19     }
20     puts("");
21 }
```

0.1.4 結果の表示

ソースコード 4 結果の表示：連珠（五目並べ）

```
1 //決着〜ゲーム終了-----
2 void Finish(void){
3     Board_Output();
4     printf("%dP (",player_number);
5     switch( player_number ){
6         case 0:     printf("・"); break;
7         case 1:     printf("○"); break;
8         case 2:     printf("●"); break;
9     }
10    printf(") の勝利です!\n");
11
12    exit(0);
13 }
```

0.1.5 手の入力

ソースコード 5 入力処理：連珠（五目並べ）

```
1 //入力処理-----
2 void Game(void){
3     int x, y;
4     printf("%dP (",player_number);
5     switch( player_number ){
6         case 0:     printf("・"); break;
7         case 1:     printf("○"); break;
8         case 2:     printf("●"); break;
9     }
10    printf(") のターンです.\n");
11    while(1){
12        while(1){
13            printf("置く場所を決めてください(x y) "); scanf("%d %d",&x,&y);
14            if( x >= 0 && x < BOARD_SQUARE && y >= 0 && y < BOARD_SQUARE ) break;
15            else printf("その場所には置けません\n");
16        }
17        if( board[x][y] == 0 ){
18            board[x][y] = player_number;
19            break;
20        } else printf("その場所には置けません\n");
21    }
22    Board_Scan(x, y);
23 }
```

0.1.6 勝敗の判定

ソースコード 6 盤面チェック：連珠（五目並べ）

```
1 //盤面の調査（5個並んだかの調査）-----
2 void Board_Scan( int x, int y ){
3     int n[4];                //8方向（直線4本分）に並んだ数
4     int move_x, move_y;
5     int i;
6     move_x = 1; move_y = 1;    //[\] 方向
7     n[0] = Board_Scan_Sub( x, y, move_x, move_y );
8     move_x = 0; move_y = 1;    //[|] 方向
9     n[1] = Board_Scan_Sub( x, y, move_x, move_y );
10    move_x = 1; move_y = 0;    //[―] 方向
11    n[2] = Board_Scan_Sub( x, y, move_x, move_y );
12    move_x = -1; move_y = 1;   //[/] 方向
13    n[3] = Board_Scan_Sub( x, y, move_x, move_y );
14    for( i = 0; i < 4; i++ ){
15        if(n[i] == 5) Finish();
16    }
17 }
18
19 int Board_Scan_Sub( int x, int y, int move_x, int move_y ){
20     int n = 1;                //置いた場所の1個分で初期化
21     int i;
22     for( i = 1; i < 5; i++ ){
23         if( board[ x+(move_x*i) ][ y+(move_y*i) ]==player_number ) n += 1;
24         else break;
25     }
26     for( i = 1; i < 5; i++ ){
27         if( board[ x+(-1*move_x*i) ][ y+(-1*move_y*i) ]==player_number ) n += 1;
28         else break;
29     }
30     return n;
31 }
```

参考文献

[1]