

```
1: #include "client_routine.h"
2:
3: int main(){
4:     char teamname[]={"teamC"};
5:     char ip_addr[80];
6:     int port;
7:     int teamcolor;
8:     char moveout[4];
9:     ConsoleBoard board;
10:    Reversi_AI_Random AI;
11:    int x_out;
12:    int y_out;
13:    int flagout;
14:    int flagin;
15:    Point mymove;
16:
17:    cout << "server IP address:";
18:    cin >> ip_addr;
19:    cout << "server port:";
20:    cin >> port;
21:
22:    if(set_socket(ip_addr, port)==false){
23:        cerr << "connection miss" << endl;
24:    }
25:
26:    printf("start setting game\n");
27:    if((teamcolor = setting_game(teamname)) == BLACK){
28:        printf("my color is BLACK!\n");
29:    }else{
30:        printf("my color is WHITE!\n");
31:    }
32:    printf("end initial setting\n");
33:
34:    while(true){
35:        cout << "Turn:" << board.getTurns() << endl;
36:        if((board.getTurns() != 0) || teamcolor == WHITE){
37:            flagin = board_update(board);
38:            cout << "Black Disk:" << board.countDisc(BLACK) << endl;
39:            cout << "White Disk:" << board.countDisc(WHITE) << endl;
40:            cout << "Empty:" << board.countDisc(EMPTY) << endl;
41:        }
42:        if((flagin & MTFLAG) != 0 ){
43:            cout << "receive MT" << endl;
44:
45:        }else if((flagin & ARFLAG) != 0 ){
46:            cout << "receive AR" << endl;
47:            board.changeCurrentColor();
48:
49:        }else if((flagin & ACFLAG) != 0 ){
50:            cout << "receive AC" << endl;
51:            AI.return_move(board, flagin, x_out, y_out, flagout);
52:
53:            Point ACdisk(x_out, y_out, flagout);
54:
55:            board.Reverse_disk(ACdisk, board.getCurrentColor());
56:            board.changeCurrentColor();
57:
58:            move_gen(moveout, x_out, y_out, flagout);
59:            cout << "move:" << moveout << " flag:" << flagout << endl;
60:
61:            send_func(moveout);
62:
```

```
63:    }else if((flagin & PSFLAG) != 0 ){
64:        cout << "receive PS" << endl;
65:        if(board.pass()==false){
66:            cerr << "miss Pass routine\n" << endl;
67:        }
68:    }else if((flagin & GFFLAG) != 0 ){
69:        board.print();
70:        cout << "game set!" << endl;
71:        break;
72:    }else{
73:        AI.return_move(board, flagin, x_out, y_out, flagout);
74:
75:        Point mymove(x_out, y_out, flagout);
76:
77:        if(board.move(mymove)==false){
78:            cerr << "board update(mymove) miss" << endl;
79:        }
80:
81:        board.print();
82:        cout << "Black Disk:" << board.countDisc(BLACK) << endl;
83:        cout << "White Disk:" << board.countDisc(WHITE) << endl;
84:        cout << "Empty:" << board.countDisc(EMPTY) << endl;
85:
86:        move_gen(moveout, x_out, y_out, flagout);
87:        cout << "move:" << moveout << " flag:" << flagout << endl;
88:
89:        send_func(moveout);
90:    }
91: }
92:
93: close_socket();
94: return 0;
95: }
```