ネットワークプログラミングII

総合演習

402　　　　411

2017 年 7 月 18 日

1　概要

今回、ネットワークプログラミング II の総合演習として作成した作品は「六目並べ」である。

2　ソースコード

総合演習で作成したソースコードをリスト 1〜6 に示す。リスト 1 では、

リスト 1　sessionman.h

```c
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <netinet/in.h>

#define PORT    (in_port_t)50002
#define MAX_ATTENDANTS 5

extern void enter();
extern void sessionman_init(int num, int maxfd);
extern void sessionman_loop();
```

リスト 2 では、

リスト 2　sessionman.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>

#define MAX_ATTENDANTS 5
#define BUF_LEN        80

static char buf[BUF_LEN];
static fd_set mask;
static int width;
static int attendants;

typedef struct {
    int fd;
    // char name[16];
} ATTENDANT;

static ATTENDANT p[MAX_ATTENDANTS];

static void send_all(int i, int n);
static void ending();

void enter(int i, int fd)
{
    int len;
    static char *mesg = "Wait.\n";

    p[i].fd = fd;

    // Send "Wait." to player who is first entered room.
    if (i == 0) {
        write(fd, mesg, strlen(mesg));
    }
}

void sessionman_init(int num, int maxfd)
{
    int i;
    // static char *mesg = "Game Start.\n";
    char message[20];
    int rnd;

    srandom(time(NULL));
    rnd = random() % 2;

    attendants = num;

    width = maxfd + 1;
    FD_ZERO(&mask);
    FD_SET(0, &mask);
    for (i = 0; i < num; i++) {
        FD_SET(p[i].fd, &mask);
    }

    sprintf(message, ":%d Game Start.\n", rnd);
    write(p[0].fd, message, strlen(message));
    sprintf(message, ":%d Game Start.\n", 1 - rnd);
    write(p[1].fd, message, strlen(message));
}

void sessionman_loop()
{
    fd_set readOk;
    int i;

    while (1) {
        readOk = mask;
        select(width, (fd_set *)&readOk, NULL, NULL, NULL);

        // Is there are input from keyboard?
        if (FD_ISSET(0, &readOk)) {
            ending();
        }

        for (i = 0; i < attendants; i++) {
            if (FD_ISSET(p[i].fd, &readOk)) {
                int n;
                n = read(p[i].fd, buf, BUF_LEN);
                send_all(i, n);
            }
        }
    }
}

// Sub routine

```

```
static void ending()
{
    int i;
    for (i = 0; i < attendants; i++) {
        write(p[i].fd, "q", 1);
    }
    for (i = 0; i < attendants; i++) {
        close(p[i].fd);
    }
    exit(0);
}

static void send_all(int i, int n)
{
    int j;
    for (j = 0; j < attendants; j++) {
        write(p[j].fd, buf, n);
    }
}
```

リスト 3 では、

リスト 3　session.h

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <netinet/in.h>

#define PORT (in_port_t)50002
#define HOSTNAME_LENGTH 64

extern void session_init(int soc);
extern void session_loop();
```

リスト 3 では、

リスト 4　session.c

```
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <signal.h>
#include <ncurses.h>

#define BUF_LEN 80

#define INFO_WIN_WIDTH  40
#define INFO_WIN_HEIGHT 1

#define GOBAN_SCREEN_HEIGHT 20
#define GOBAN_SCREEN_WIDTH  40

static char goban_my_stone;
static char goban_peer_stone;

static char goban_plane[GOBAN_SCREEN_HEIGHT][GOBAN_SCREEN_WIDTH] = {
    ". . . . . . . . . . . . . . . . . . . . .",
    ". . . . . . . . . . . . . . . . . . . .",
    ". . . . . . . . . . . . . . . . . . . .",
    ". . . . . . . . . . . . . . . . . . . .",
    ". . . . . . . . . . . . . . . . . . . .",
    ". . . . . . . . . . . . . . . . . . . .",
    ". . . . . . . . . . . . . . . . . . . .",
    ". . . . . . . . . . . . . . . . . . . .",
    ". . . . . . . . . . . . . . . . . . . .",
    ". . . . . . . . . . . . . . . . . . . .",
    ". . . . . . . . . . . . . . . . . . . .",
    ". . . . . . . . . . . . . . . . . . . .",
    ". . . . . . . . . . . . . . . . . . . .",
    ". . . . . . . . . . . . . . . . . . . .",
    ". . . . . . . . . . . . . . . . . . . .",
    ". . . . . . . . . . . . . . . . . . . .",
    ". . . . . . . . . . . . . . . . . . . .",
    ". . . . . . . . . . . . . . . . . . . .",
    ". . . . . . . . . . . . . . . . . . . .",
    ". . . . . . . . . . . . . . . . . . . ."
};
static char goban_plane_orig[GOBAN_SCREEN_HEIGHT][GOBAN_SCREEN_WIDTH];

static WINDOW *win_info, *win_goban;
static WINDOW *frame_info, *frame_goban;

static char send_buf[BUF_LEN];
static char recv_buf[BUF_LEN];
static int session_soc;
static fd_set mask;
static int width;

static void init_goban();
static int is_my_turn(int, char);
static int put_stone(int, int, char);
static void die();
static int detect_rokumoku(char);

void session_init(int soc)
{
    int i;
    int x, y;
    session_soc = soc;
    width = soc + 1;
    FD_ZERO(&mask);
    FD_SET(0, &mask);
    FD_SET(soc, &mask);

    initscr();
    signal(SIGINT, die);

    win_info = newwin(INFO_WIN_HEIGHT, INFO_WIN_WIDTH, 22, 1);
    scrollok(win_info, FALSE);
    wmove(win_info, 0, 0);

    frame_goban = newwin(GOBAN_SCREEN_HEIGHT + 2, GOBAN_SCREEN_WIDTH + 2, 0, 0);
    win_goban = newwin(GOBAN_SCREEN_HEIGHT, GOBAN_SCREEN_WIDTH, 1, 1);
    box(frame_goban, '|', '-');
    scrollok(win_goban, FALSE);
    wmove(win_goban, 0, 0);

    cbreak();
    noecho();

    memcpy(goban_plane_orig, goban_plane, sizeof(goban_plane));
    init_goban();

    wrefresh(frame_info);
    wrefresh(win_info);
    wrefresh(frame_goban);
    wrefresh(win_goban);
}

void session_loop()
{
    int c;
    fd_set readOk;
    int i;
    int y, x;
    char message[BUF_LEN];
    int status;
    int is_game_loop   = 1;
    int is_game_finish = 0;
    int game_step = 0;

    while (1) {
        readOk = mask;
        select(width, (fd_set *)&readOk, NULL, NULL, NULL);

        if (FD_ISSET(0, &readOk)) {
            c = getchar();
            getyx(win_goban, y, x);
            switch (c) {
            case 'j':
                wmove(win_goban, y+1, x);
                break;
            case 'k':
                wmove(win_goban, y-1, x);
                break;
            case 'h':
                wmove(win_goban, y, x-2);
                break;
            case 'l':
                wmove(win_goban, y, x+2);
                break;
            case ' ':
                if (is_game_finish) break;
                if (!is_my_turn(game_step, goban_my_stone)) break;
                if (!put_stone(y, x, goban_my_stone)) break;

                sprintf(send_buf, "(%d,%d) %c\n", x, y, goban_my_stone);
                write(session_soc, send_buf, strlen(send_buf));

                break;
            case 'r':
            case 'c':
                sprintf(send_buf, "reset\n");
                write(session_soc, send_buf, strlen(send_buf));
                break;
            case 'q':
                sprintf(send_buf, "quit\n");
                write(session_soc, send_buf, strlen(send_buf));
                break;
            }
            wrefresh(win_info);
            wrefresh(win_goban);
        }

        if (FD_ISSET(session_soc, &readOk)) {
            status = read(session_soc, recv_buf, BUF_LEN);
            if (recv_buf[0] == ':') {
                // Game start!
                int id;
                sscanf(recv_buf, ":%d", &id);
                if (id == 0) {
                    goban_my_stone = 'x';
                    goban_peer_stone = 'o';
                    strcpy(message, "Wait.");
                } else {
                    goban_my_stone = 'o';
                    goban_peer_stone = 'x';
                    strcpy(message, "It's your turn!");
                }
                sprintf(recv_buf, "Game start! %s\n", message);
                werase(win_info);
                waddstr(win_info, recv_buf);
            }
            else if (recv_buf[0] == '(') {
                // Player put stone.
                char stone_char;
                sscanf(recv_buf, "(%d,%d) %c", &x, &y, &stone_char);
                put_stone(y, x, stone_char);
                game_step++;
                if ((status = is_my_turn(game_step, goban_my_stone)) > 0) {
                    sprintf(message, "It's your turn! (remains: %d)\n", status);
                } else {
                    sprintf(message, "%s\n", "Wait");
                }
                werase(win_info);
                waddstr(win_info, message);

                if (stone_char == goban_my_stone && detect_rokumoku(stone_char)) {
                    werase(win_info);
                    waddstr(win_info, "You win!");
                    is_game_finish = 1;
                }
                if (stone_char == goban_peer_stone && detect_rokumoku(stone_char)) {
                    werase(win_info);
                    waddstr(win_info, "You lose!");
                    is_game_finish = 1;
                }
            }
            else if (strstr(recv_buf, "reset") != NULL) {
                // Reset game.
                init_goban();
                game_step = 0;
                is_game_finish = 0;
                if (goban_my_stone == 'x') {
                    strcpy(message, "Wait.");
                } else {
                    strcpy(message, "It's your turn!");
                }
                sprintf(recv_buf, "Game start! %s\n", message);
                werase(win_info);
                waddstr(win_info, recv_buf);
            }
            else if (strstr(recv_buf, "quit") != NULL) {
                // Quit game.
                is_game_loop = 0;
            }
            else {
                // Received broadcast message.
                werase(win_info);
                waddstr(win_info, recv_buf);
            }

            wrefresh(win_info);
            wrefresh(win_goban);
        }

        if (is_game_loop == 0) break;
    }

    die();
}

static void init_goban()
{
    int x, y;
    memcpy(goban_plane, goban_plane_orig, sizeof(goban_plane_orig));

    wclear(win_goban);
    x = 0;
    for (y = 0; y < GOBAN_SCREEN_HEIGHT; y++) {
        wmove(win_goban, y, x);
        waddstr(win_goban, goban_plane[y]);
    }
    wmove(win_goban, GOBAN_SCREEN_HEIGHT/2, GOBAN_SCREEN_WIDTH/2);
}

// Return true if it's my turn.
// game_step: 0 1 2 3 4 5 6 7 8 9 10 ...
// stone:     o x x o o x x o o x x  ...
static int is_my_turn(int game_step, char stone_char)
{
    int mod;
    if (stone_char == 'o' && game_step == 0) return 1;
    if (stone_char == 'x' && game_step == 0) return 0;
    mod = (game_step - 1) % 4;
    if (stone_char == 'o' && mod == 2) return 2;
    if (stone_char == 'o' && mod == 3) return 1;
    if (stone_char == 'x' && mod == 0) return 2;
    if (stone_char == 'x' && mod == 1) return 1;
```

```c
    return 0;
}

static int put_stone(int y, int x, char stone_char)
{
    if (goban_plane[y][x] != '.') return 0;
    goban_plane[y][x] = stone_char;

    wmove(win_goban, y, x);
    waddch(win_goban, stone_char);
    wmove(win_goban, y, x);
    return 1;
}

static void die()
{
    endwin();
    close(session_soc);
    exit(0);
}

static int detect_rokumoku(char stone_char)
{
    int cnt = 0;
    int cnt2 = 0;
    int x, y;
    int k;
    for (y = 0; y < GOBAN_SCREEN_HEIGHT; y++) {
        cnt = 0;
        for (x = 0; x < GOBAN_SCREEN_WIDTH - 1; x += 2) {
            cnt = (goban_plane[y][x] == stone_char) ? (cnt + 1) : 0;
            if (cnt == 6) return 1;
        }
    }

    for (x = 0; x < GOBAN_SCREEN_WIDTH - 1; x += 2) {
        cnt = 0;
        for (y = 0; y < GOBAN_SCREEN_HEIGHT; y++) {
            cnt = (goban_plane[y][x] == stone_char) ? (cnt + 1) : 0;
            if (cnt == 6) return 1;
        }
    }

    for (y = 0; y < GOBAN_SCREEN_HEIGHT; y++) {
        for (x = 0; x < GOBAN_SCREEN_WIDTH - 1; x += 2) {
            cnt = 0;
            cnt2 = 0;
            for (k = 0; k < GOBAN_SCREEN_WIDTH / 2 - 1; k++) {
                if (!(y + k >= 0 && y + k < GOBAN_SCREEN_HEIGHT)) continue;
                if (!(y + k * 2 < GOBAN_SCREEN_WIDTH - 1)) continue;
                if (!(GOBAN_SCREEN_WIDTH - 2 - x - k * 2 >= 0)) continue;
                cnt  = (goban_plane[y + k][x + k * 2] == stone_char) ? (cnt + 1) :
                    0;
                cnt2 = (goban_plane[y + k][GOBAN_SCREEN_WIDTH - 2 - x - k * 2] ==
                    stone_char) ? (cnt2 + 1) : 0;
                if (cnt == 6) return 1;
                if (cnt2 == 6) return 1;
            }
        }
    }

    return 0;
}
```

リスト**??**では、

リスト 5　server.c

```
#include <stdio.h>
#include <stdlib.h>
#include "sessionman.h"
#include "mylib.h"

int main(int argc, char const *argv[]) {
    int num;
    int soc;
    int maxfd;

    num = 2; // player count

    if ((soc = mserver_socket(PORT, num)) == -1) {
        fprintf(stderr, "cannot setup server\n");
        exit(1);
    }

    maxfd = mserver_maccept(soc, num, enter);

    sessionman_init(num, maxfd);

    sessionman_loop();

    return 0;
}
```

リスト**??**では、

リスト 6　client.c

```
#include "session.h"

int main(int argc, char const *argv[]) {
    int soc;
    char hostname[HOSTNAME_LENGTH];

    printf("Input sever's hostname: ");
    fgets(hostname, HOSTNAME_LENGTH, stdin);
    chop_newline(hostname, HOSTNAME_LENGTH);

    if ((soc = setup_client(hostname, PORT)) == -1) {
        exit(1);
    }

    session_init(soc);

    session_loop();

    return 0;
}
```

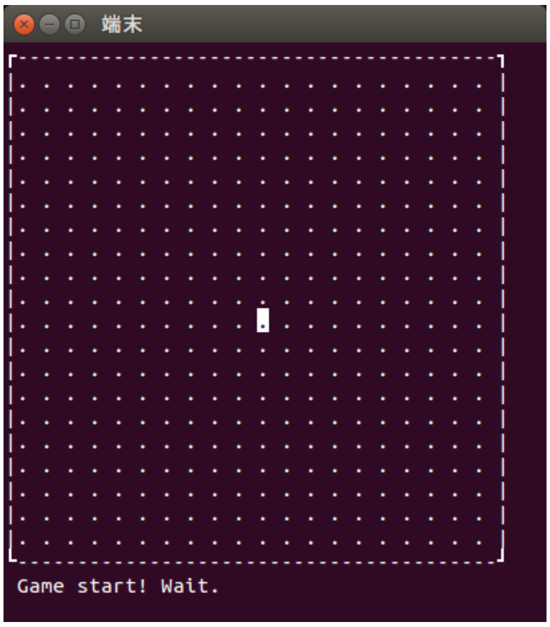
最後に、作成したプログラムをコンパイルする Makefile をリスト 7 に示す。

リスト 7　Makefile

```
MYLIBDIR = mylib
MYLIB    = $(MYLIBDIR)/mylib.a
OBJS1    = server.o sessionman.o
OBJS2    = client.o session.o
CFLAGS   = -I$(MYLIBDIR)

all: bin bin/s bin/c

bin:
  mkdir $@

bin/s: $(OBJS1)
  $(CC) -o $@ $^ $(MYLIB) -lncurses

bin/c: $(OBJS2)
  $(CC) -o $@ $^ $(MYLIB) -lncurses

server.o: sessionman.h
client.o: session.h

clean:
  $(RM) bin/s bin/c $(OBJS1) $(OBJS2) *~
```

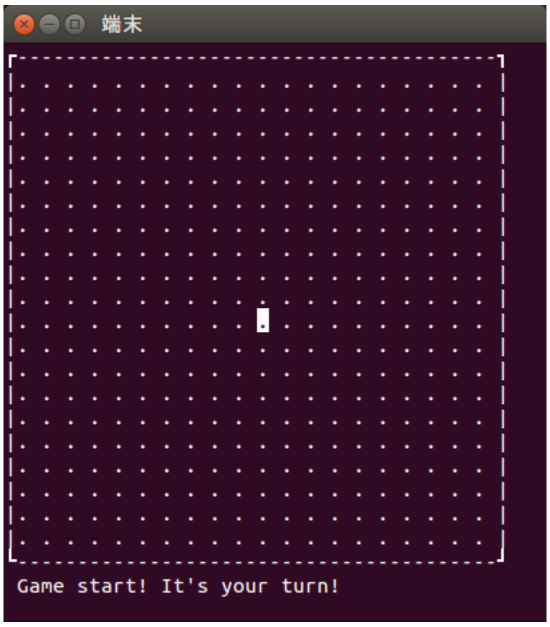
3　実行結果

リスト 7 の Makefile を使って make すると、bin ディレクトリの下に s と c という実行ファイルができる。s と c はそれぞれサーバ用とクライアント用を表しているので、まず bin/s を実行してから bin/c を実行する。今回は「六目並べ」で二人対戦なので、2 つ画面を用意してそれぞれで bin/c を実行した結果を図 1 と 2 に示す。



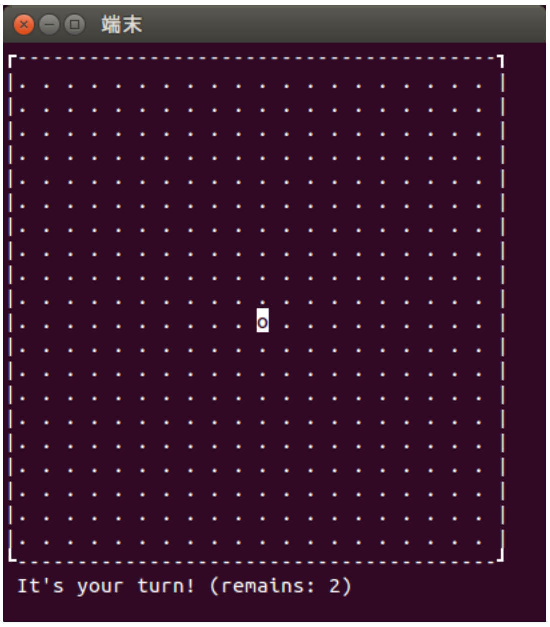
```
Game start! Wait.
```

図 1　Caption



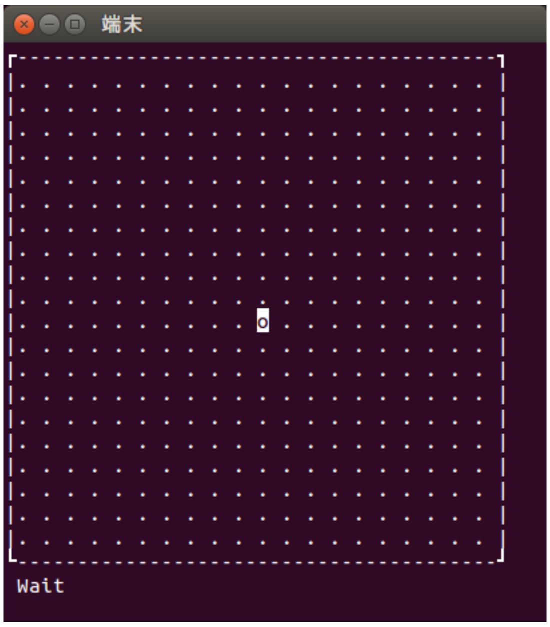
```
Game start! It's your turn!
```

図 2　Caption



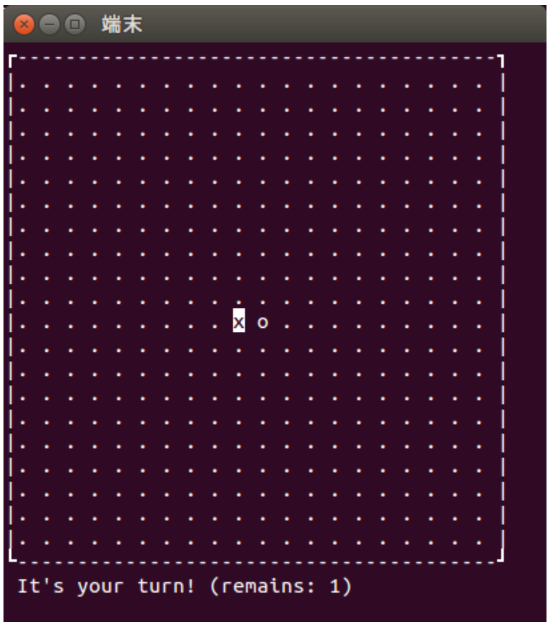
```
It's your turn! (remains: 2)
```

図 3　Caption



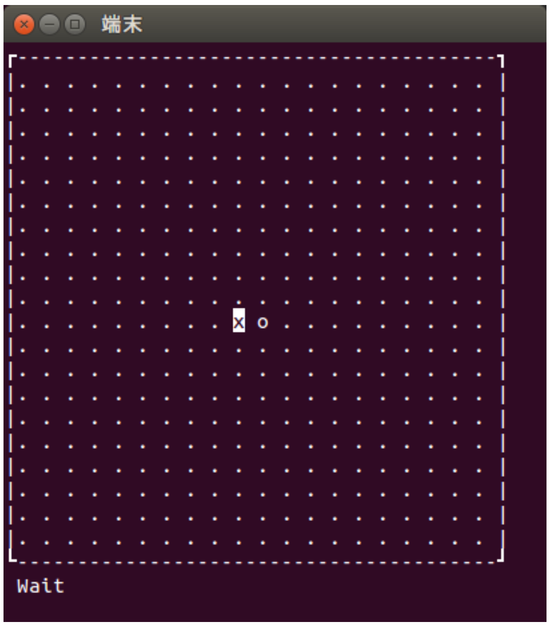
```
Wait
```

図 4　Caption



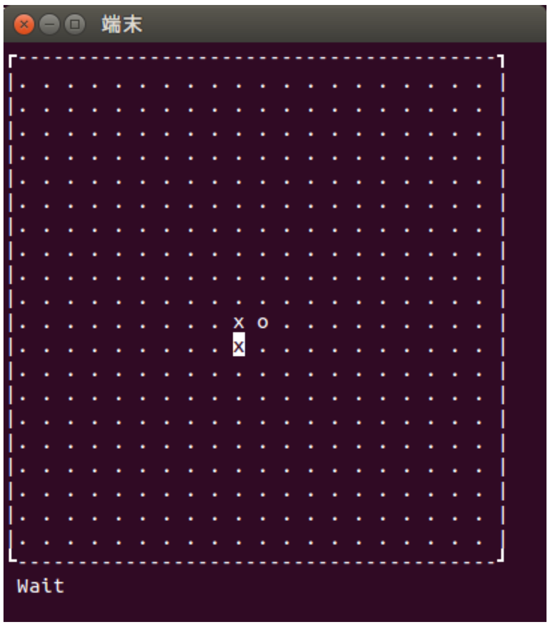
```
┌----------------------------------------┐
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . x o . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
└----------------------------------------┘
It's your turn! (remains: 1)
```

図 5　Caption



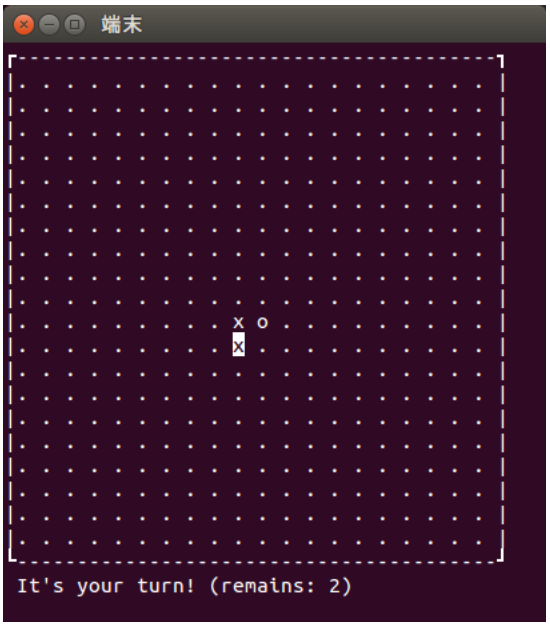
```
┌----------------------------------------┐
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . x o . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
└----------------------------------------┘
Wait
```

図 6　Caption



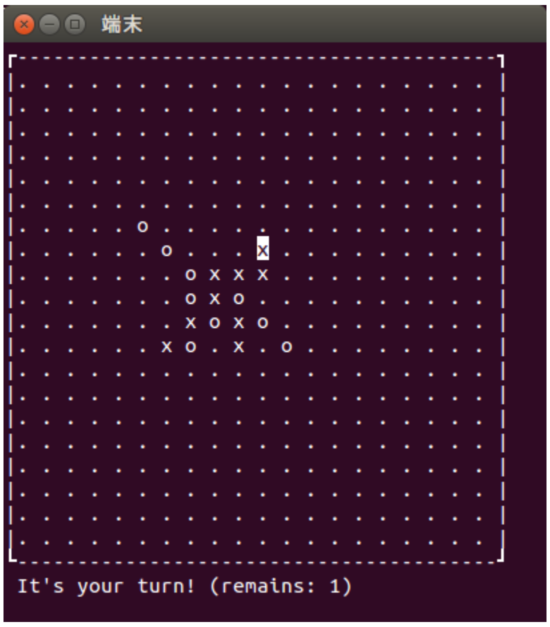
```
┌----------------------------------------┐
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . x o . . . . . . . . . |
|. . . . . . . . . x . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
└----------------------------------------┘
Wait
```

図 7　Caption



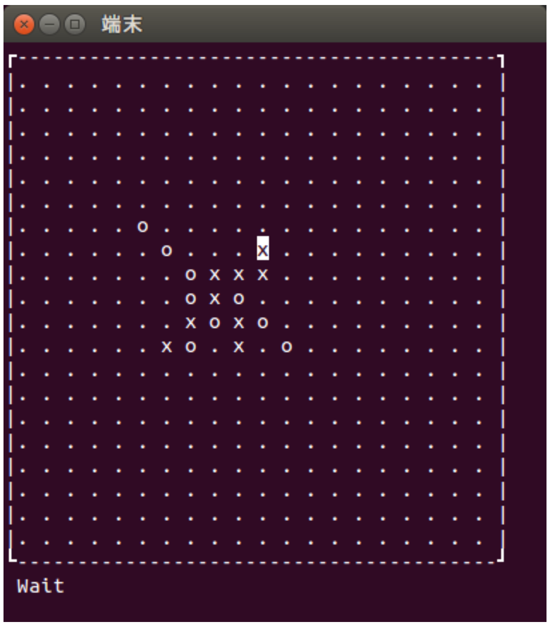
```
┌----------------------------------------┐
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . x o . . . . . . . . . |
|. . . . . . . . . x . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
|. . . . . . . . . . . . . . . . . . . . |
└----------------------------------------┘
It's your turn! (remains: 2)
```

図 8　Caption



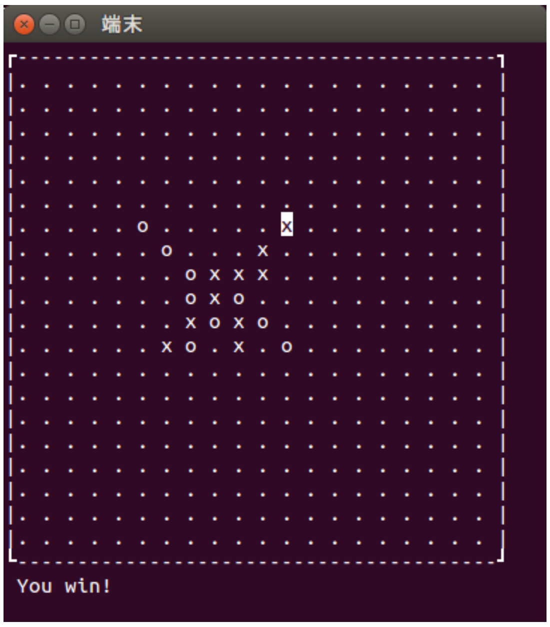
```
┌-----------------------------------------┐
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . o . . . . . . . . . . . . . . |
| . . . . . . o . . . x . . . . . . . . . |
| . . . . . . . o x x x . . . . . . . . . |
| . . . . . . . o x o . . . . . . . . . . |
| . . . . . . . x o x o . . . . . . . . . |
| . . . . . . x o . x . o . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
└-----------------------------------------┘
It's your turn! (remains: 1)
```

図 9　Caption



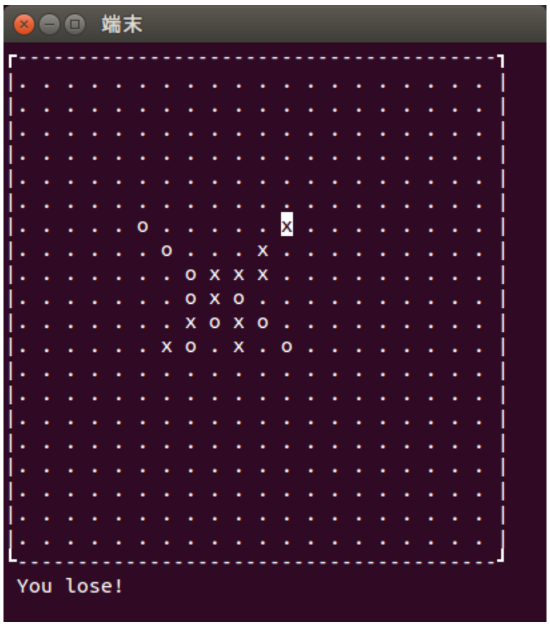
```
┌-----------------------------------------┐
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . o . . . . . . . . . . . . . . |
| . . . . . . o . . . x . . . . . . . . . |
| . . . . . . . o x x x . . . . . . . . . |
| . . . . . . . o x o . . . . . . . . . . |
| . . . . . . . x o x o . . . . . . . . . |
| . . . . . . x o . x . o . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
└-----------------------------------------┘
Wait
```

図 10　Caption



```
┌-----------------------------------------┐
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . o . . . . . x . . . . . . . . |
| . . . . . . o . . . x . . . . . . . . . |
| . . . . . . . o x x x . . . . . . . . . |
| . . . . . . . o x o . . . . . . . . . . |
| . . . . . . . x o x o . . . . . . . . . |
| . . . . . . x o . x . o . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
└-----------------------------------------┘
You win!
```

図 11　Caption



```
┌-----------------------------------------┐
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . o . . . . . x . . . . . . . . |
| . . . . . . o . . . x . . . . . . . . . |
| . . . . . . . o x x x . . . . . . . . . |
| . . . . . . . o x o . . . . . . . . . . |
| . . . . . . . x o x o . . . . . . . . . |
| . . . . . . x o . x . o . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
| . . . . . . . . . . . . . . . . . . . . |
└-----------------------------------------┘
You lose!
```

図 12　Caption

4　考察