TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BÁO CÁO BÀI TẬP LỚN

MÔN HỌC: LẬP TRÌNH MẠNG

Đề tài: Xây dựng ứng dụng Chơi cờ Caro

GIẢNG VIÊN HƯỚNG DẪN*: *ThS. Bành Thị Quỳnh Mai

***SINH VIÊN THỰC HIỆN*:** Nguyễn Viết Ngọc Quang – 20163315
Nguyễn Quang Anh - 20160152

Hà Nội, tháng 12 năm 2019

MỤC LỤC

Trang

I. Lời nói đầu

Hiện nay mạng Internet toàn cầu đã phát triển rất mạnh, đáp ứng rất tốt các nhu cầu về thông tin khoa hoa, giải trí, liên lạc, mua sắm,., của con người. Internet đã trở thành một phần không thể thiếu với cuộc sống hiện đại. Về bản chất, Internet là một hệ thống mạng, liên kết các máy tính trên toàn thế giới lại theo chuẩn chung của nó. Với phạm vi nhỏ hơn những cũng rất tương tự, có những mạng khác đó là WAN, LAN,…mỗi mô hình đều được ứng dụng rất nhiều tiện ích khác nhau.

Mô hình client-server là một mô hình nổi tiếng trong mạng máy tính,được áp dụng rất rộng rãi và là mô hình của mọi trang web hiện có. Ý tưởng của mô hình này là máy con (đóng vài trò là máy khách) gửi một yêu cầu (request) để máy chủ (đóng vai trò người cung ứng dịch vụ), máy chủ sẽ xử lý và trả kết quả về cho máy khách.

Với mục đích áp dụng mô hình Client – Server vào thực tế, nhóm em đã xây dựng ứng dụng game chơi cờ Caro. Chức năng mong muốn của chương trình là tạo một ứng dụng chơi game Caro và trò chuyện giữa các client trong cùng mạng.

Để thực hiện đề tài này, chúng em xin chân thành cảm ơn sự hướng dẫn, giúp đỡ của Th.S Bành Thị Quỳnh Mai – Giảng viên Viện Công nghệ Thông tin và Truyền thông, Trường Đại học Bách Khoa Hà Nội cùng với sự giúp đỡ của các bạn Việt Nhật K61.

II. Ý tưởng

Xây dựng ứng dụng: Game chơi cờ Caro.
Từ yêu cầu của đề bài, ta phân tích để tìm ra mục tiêu cụ thể cần đạt được:

- Thiết kế mô hình client – server.
- Thuật toán.
- Phương án hiển thị giao diện cho người dùng.
- Tính năng thuận tiện.

III. Cơ sở lý thuyết

- Ngôn ngữ lập trình C: Là ngôn ngữ lập trình phổ biến, phù hợp với xây dựng hệ thống, tốc độ xử lý nhanh.
- Thư viện GTK+: Bộ công cụ tạo giao diện người dùng.
- Mô hình client – server:

Mô hình mạng client server là mô hình nổi tiếng trong hệ thống mạng máy tính, và chúng được áp dụng rộng rãi trên hầu hết các trang web hiện có. Với mô hình máy, các máy tính con được đóng vai trò như một máy khách, chúng làm nhiệm vụ gửi yêu cầu đến các máy chủ. Để máy chủ xử lý yêu cầu và trả kết quả về cho máy khách đó.

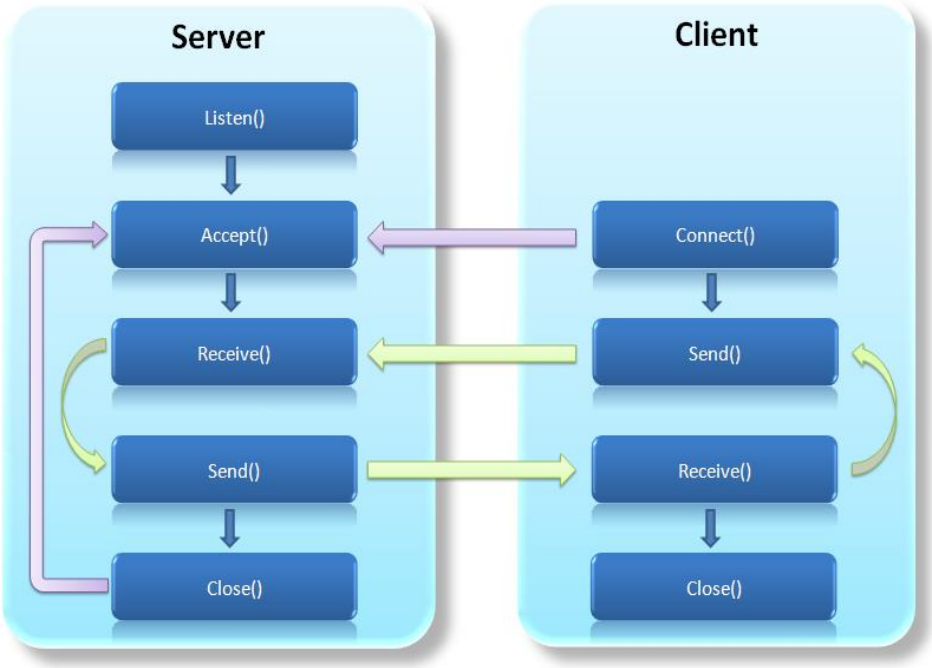
Mô hình client/server như sau: Client/Server là mô hình tổng quát nhất, trên thực tế thì một server có thể được nối tới nhiều server khác nhằm làm việc hiệu quả và nhanh hơn. Khi nhận được một yêu cầu từ client, server này có thể gửi tiếp yêu cầu vừa nhận được cho server khác ví dụ như database server vì bản thân nó không thể xử lý yêu cầu này được. Máy chủ server có thể thi hành cácnhiệm vụ đơn giản hoặc phức tạp. Thực tế trong các ứng dụng của mô hình client/server, các chức năng hoạt động chính là sự kết hợp giữa client và server với sự chia sẻ tài nguyên, dữ liệu trên cả hai máy.

Vai trò của client: Trong mô hình client/server, client được coi như là người sử dụng các dịch vụ trên mạng do một hoặc nhiều máy chủ cung cấp và server được coi như là người cung cấp dịch vụ để trả lời các yêu cầu của các clients. Điều quan trọng là phải hiểu được vai trò hoạt động của nó trong một mô hình cụ thể, một máy client trong mô hình này lại có thể là server trong một mô hình khác.

TCP (Transmission Control Protocol)

Giao thức TCP (Transmission Control Protocol) là giao thức kết nối (connection-oriented), nó đòi hỏi thiết lập kết nối trước khi bắt đầu gửi dữ liệu và kết thúc kết nối khi việc gửi dữ liệu hoàn tất theo đúng thứ tự: thiết lập kết nối, truyền dữ liệu và kết thúc kết nối

IP: IP là một địa chỉ của một máy tính trên mạng, dựa vào địa chỉ IP giao thức ICP có thể truyền dữ liệu chính xác từ một máy này qua máy kia thông qua hệ thống mạng. Ở trên mạng, một máy tính sẽ có một địa chỉ IP khác nhau, từ địa chỉ IP có thể biết được máy nào trên mạng và ngược lại.



Không như giao thức UDP - giao thức có thể lập tức gửi gói tin mà không cần thiết lập kết nối, TCP đòi hỏi thiết lập kết nối trước khi bắt đầu gửi dữ liệu và kết thúc kết nối khi việc gửi dữ liệu hoàn tất.

IV. Phân tích chức năng

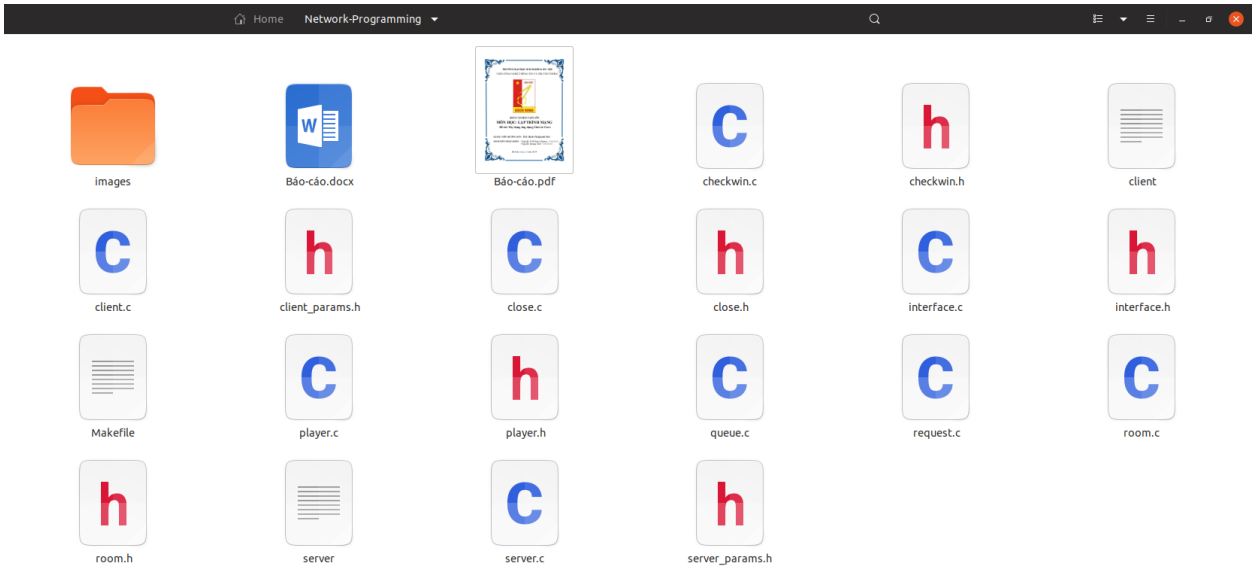
1. Client:

- Connect tới server sử dụng giao thức TCP.
- Gửi dữ liệu tới server:
    - Tên mà người dùng nhập: name.
    - Tin nhắn tới tất cả người dùng khác.
    - Yêu cầu tham gia trò chơi.
    - Đánh dấu một vị trí trên bàn cờ.
    - Tin nhắn tới người đang chơi cùng.
    - Yêu cầu thoát khỏi màn hình đang chơi.
    - Yêu cầu thoát trò chơi.
- Nhận dữ liệu từ server:
    - Bảng tin nhắn từ tất cả người chơi.
    - Tên của người chơi sau khi tạo thành công.
    - Danh sách tất cả các phòng chơi cùng với trạng thái của mỗi phòng.
    - Dữ liệu của bàn cờ mà người chơi đang chơi.
    - Bảng tin nhắn của người đang chơi cùng.

2. Server:

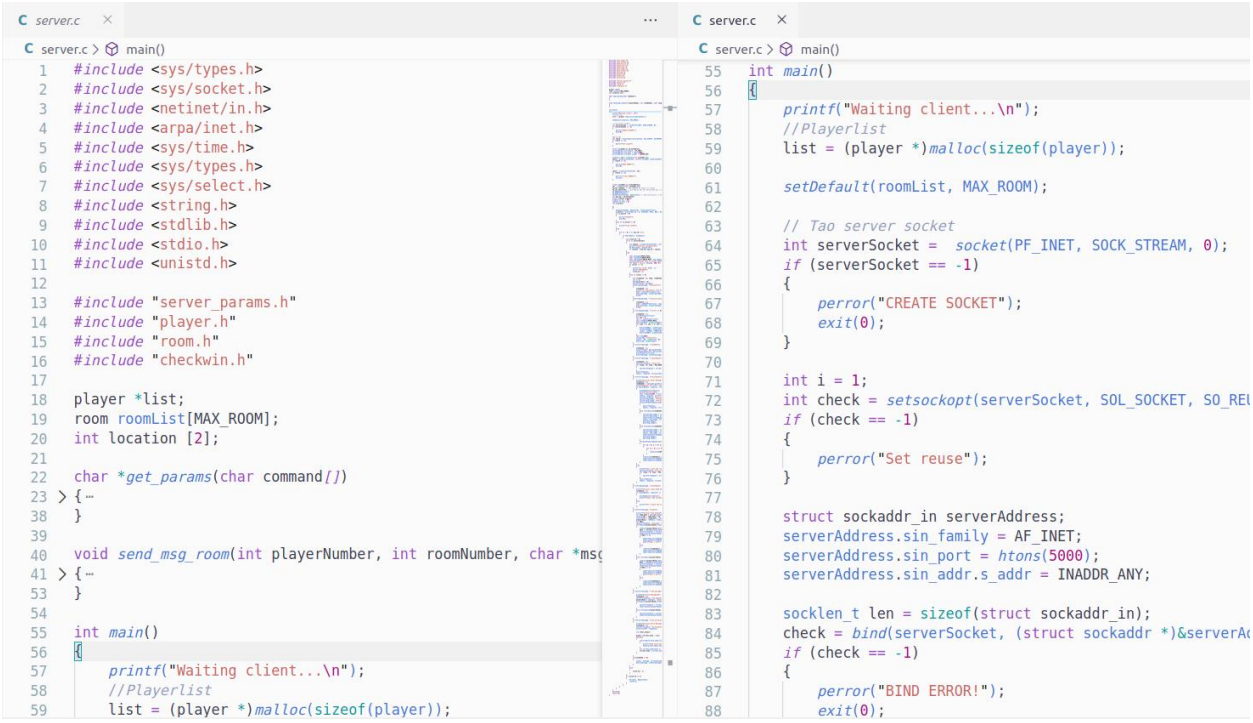
- Phục vụ đa kết nối từ nhiều client sử dụng giao thức TCP.
- Gửi và nhận các dữ liệu tới client (dữ liệu tương tự như phía client).
- Quản lý việc ánh xạ username và IP, port tương ứng của client (tuỳ các phiên đăng nhập khác nhau mà IP và port của người dùng sẽ khác nhau).
- Quản lý trạng thái đang trong trò chơi của người dùng.

V. Mã nguồn:



1. Server:

- **server.c:** Server thực hiện nhận, xử lí và gửi dữ liệu.



```c
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <sys/time.h>
#include <sys/types.h>
#include <sys/select.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>

#include "server_params.h"
#include "player.h"
#include "room.h"
#include "checkwin.h"

player *list;
room roomList[MAX_ROOM];
int location [2];

char *get_params(char command[])
{ ...
}

void send_msg_room(int playerNumber, int roomNumber, char *msg
{ ...
}

int main()
{
    printf("Waiting client...\n");
    //Playerlist
    list = (player *)malloc(sizeof(player));

    setDefault(roomList, MAX_ROOM);

    // Tao server socket
    int serverSocket =  socket(PF_INET, SOCK_STREAM, 0);
    if (serverSocket == -1)
    {
        perror("CREATE SOCKET");
        exit(0);
    }

    int i = 1;
    int check = setsockopt(serverSocket, SOL_SOCKET, SO_REU
    if (check == -1)
    {
        perror("Set reuse");
    }

    struct sockaddr_in serverAddress;
    serverAddress.sin_family = AF_INET;
    serverAddress.sin_port = htons(5000);
    serverAddress.sin_addr.s_addr = INADDR_ANY;

    socklen_t len = sizeof(struct sockaddr_in);
    check = bind(serverSocket, (struct sockaddr *)&serverAd
    if (check == -1)
    {
        perror("BIND ERROR!");
        exit(0);
```

- **player.h:** Khai báo struct player. Các hàm liên quan đến player như thêm player, cài tên player, thông tin player…

player.h

player.h > addNewPlayer(player *, char *, int, char *, int)

```c
typedef struct player
{
    int number;
    char ip_addr[20];
    int port;
    char name[20];
    struct player *next;
    struct player *prev;
} player;

void printPlayerList(player *list);
player *addNewPlayer(player *list, char *ip_addr, int port, char *name, int number);
void setPlayerName(player *list, char *name, int i);
int countPlayer(player *list);
char *playerInfo(player *list, int index);
player *playerDisconnect(player *list, int number);
char *getPlayerName(player *list, int number);
```

- **room.h:** Khai báo struct room. Các hàm liên quan đến room như thêm room, rời room…

room.h

room.h > Room > Board

```c
#define MAX_ROOM 10

typedef struct Room
{
    int id;
    int Player1;
    int Player2;
    int inGame;
    int turn;
    char Board[10][10];
} room;

int enterRoom(int player, room roomList[], int roomNumber);
int leaveRoom(int player, room roomList[]);
int inRoom(int player, room roomList[]);
void printRoomList(room roomList[]);
void setDefault(room roomList[], int max);
void changeTurn(room roomList[], int roomNumber);
int countPlayerInRoom(room roomList[], int roomNumber);
```

- **checkwin.h:** Kiểm tra player đã thắng chưa sau mỗi nước cờ.

checkwin.h

```c
#define N 10

int checkWin(int x, int y, char board[N][N], char flagWin);
```

2. Client:

- **client.c:** Client thực hiện nhận, xử lí và gửi dữ liệu.

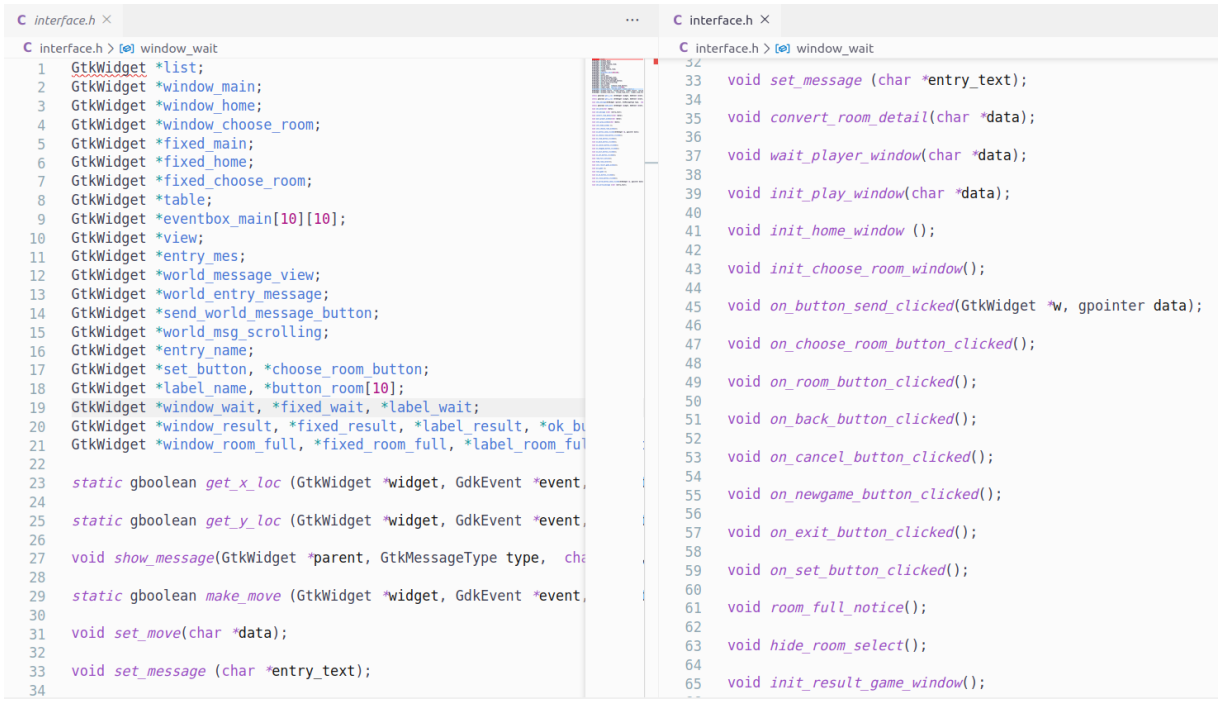
client.c

```c
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#include <signal.h>
#include <sys/select.h>
#include <sys/time.h>
#include <netdb.h>

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <gtk/gtk.h>

#include "client_params.h"
#include "interface.h"
#include "request.c"
#include "queue.c"

flag_turn = 1;

char *get_data(char command[])
{…
}
```

client.c

```c
void recv_msg()
{…
}

gboolean timer_exe(gpointer p)
{…
}


int main (int argc, char *argv[])
{
    responses = createQueue();

    if (!g_thread_supported ())
    {
        g_thread_init(NULL);
    }
    // initialize GDK thread support
    gdk_threads_init();
    gdk_threads_enter();
    g_timeout_add(100, (GSourceFunc)timer_exe, NULL);

    gtk_init (&argc, &argv);


    clientSocket = socket(PF_INET, SOCK_STREAM, 0);
```

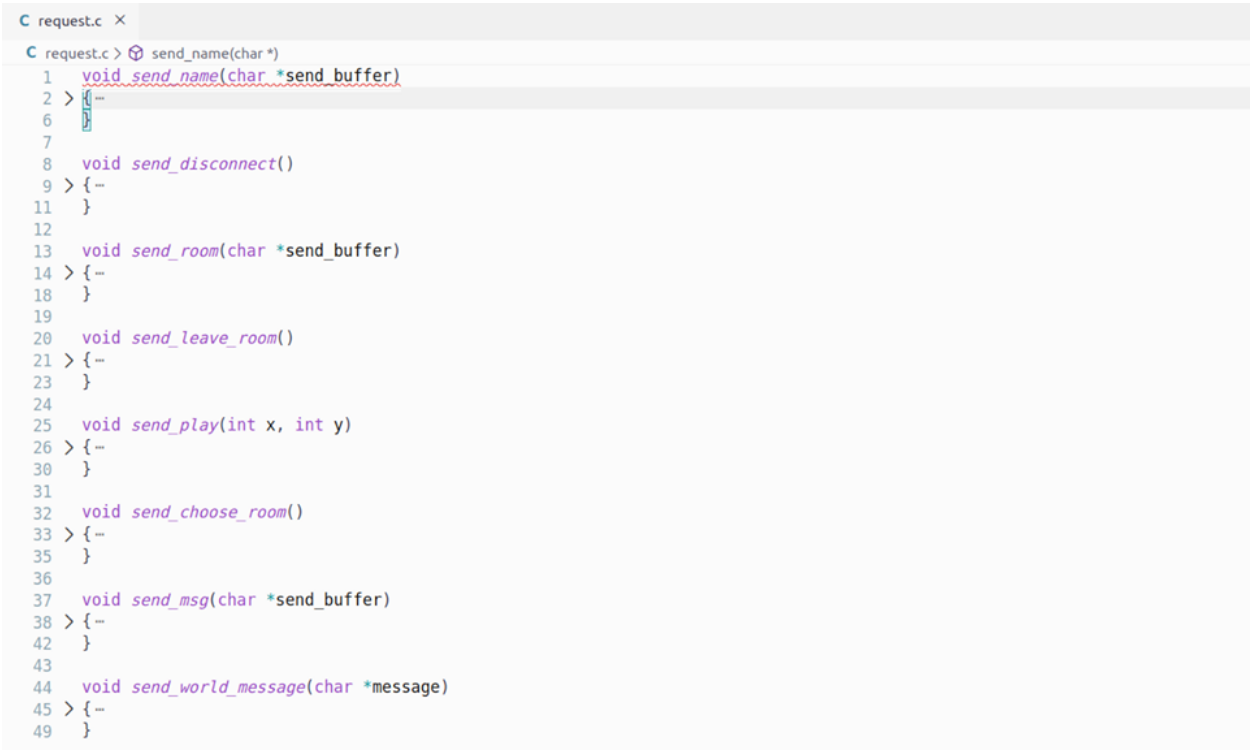
- **interface.h:** Sử dụng GTK thiết kế giao diện của client.



```
GtkWidget *list;
GtkWidget *window_main;
GtkWidget *window_home;
GtkWidget *window_choose_room;
GtkWidget *fixed_main;
GtkWidget *fixed_home;
GtkWidget *fixed_choose_room;
GtkWidget *table;
GtkWidget *eventbox_main[10][10];
GtkWidget *view;
GtkWidget *entry_mes;
GtkWidget *world_message_view;
GtkWidget *world_entry_message;
GtkWidget *send_world_message_button;
GtkWidget *world_msg_scrolling;
GtkWidget *entry_name;
GtkWidget *set_button, *choose_room_button;
GtkWidget *label_name, *button_room[10];
GtkWidget *window_wait, *fixed_wait, *label_wait;
GtkWidget *window_result, *fixed_result, *label_result, *ok_bu
GtkWidget *window_room_full, *fixed_room_full, *label_room_ful

static gboolean get_x_loc (GtkWidget *widget, GdkEvent *event,

static gboolean get_y_loc (GtkWidget *widget, GdkEvent *event,

void show_message(GtkWidget *parent, GtkMessageType type,  cha

static gboolean make_move (GtkWidget *widget, GdkEvent *event,

void set_move(char *data);

void set_message (char *entry_text);
```

```
void set_message (char *entry_text);

void convert_room_detail(char *data);

void wait_player_window(char *data);

void init_play_window(char *data);

void init_home_window ();

void init_choose_room_window();

void on_button_send_clicked(GtkWidget *w, gpointer data);

void on_choose_room_button_clicked();

void on_room_button_clicked();

void on_back_button_clicked();

void on_cancel_button_clicked();

void on_newgame_button_clicked();

void on_exit_button_clicked();

void on_set_button_clicked();

void room_full_notice();

void hide_room_select();

void init_result_game_window();
```

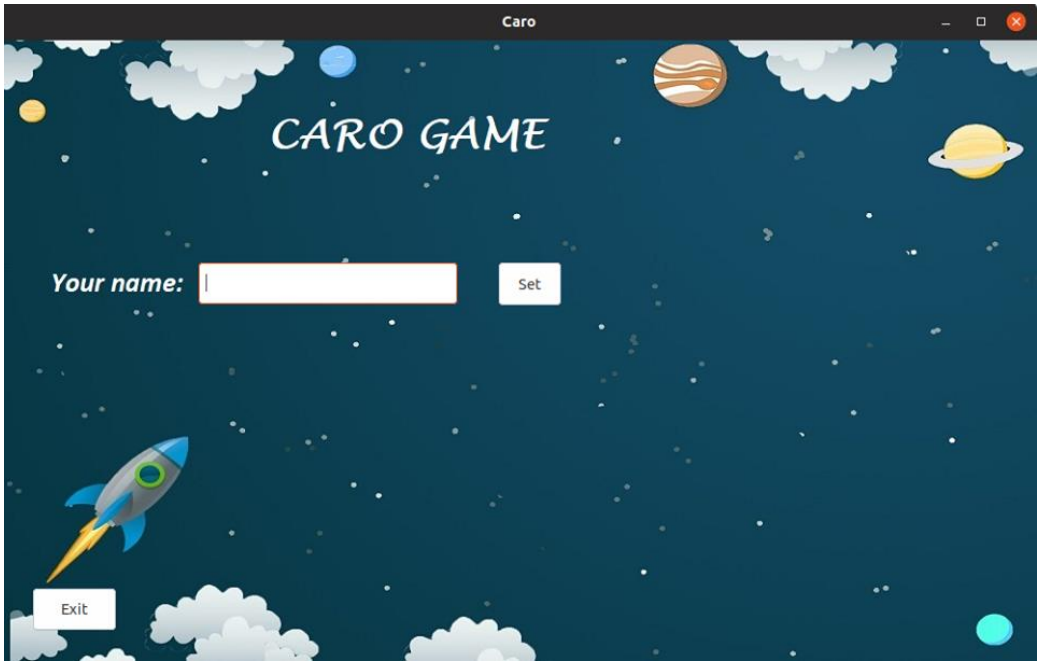
- **request.c:** Các hàm gửi dữ liệu từ client đến server.



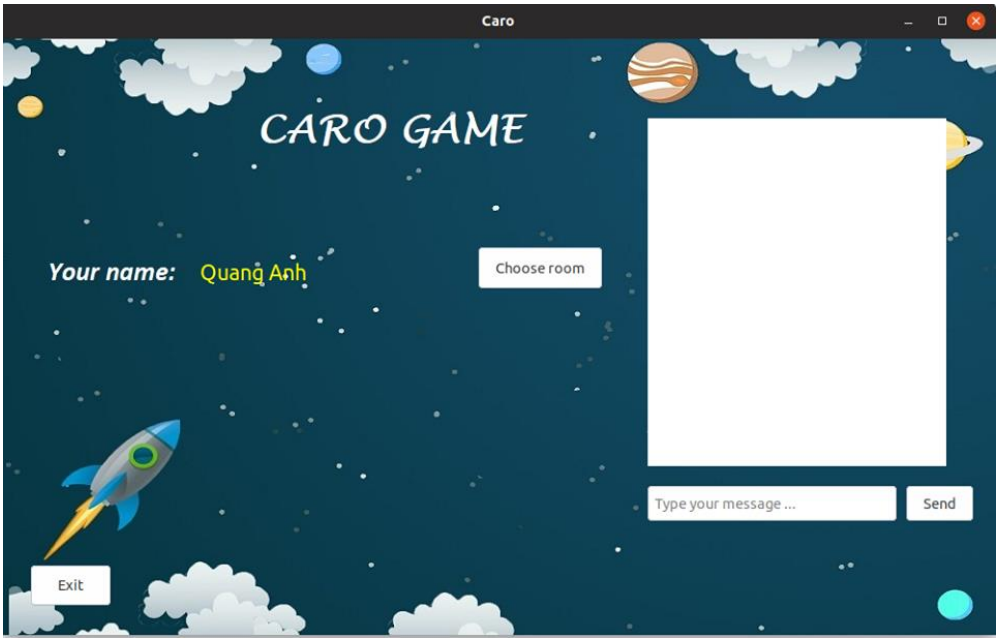
```
void send_name(char *send_buffer)
{…
}

void send_disconnect()
{…
}

void send_room(char *send_buffer)
{…
}

void send_leave_room()
{…
}

void send_play(int x, int y)
{…
}

void send_choose_room()
{…
}

void send_msg(char *send_buffer)
{…
}

void send_world_message(char *message)
{…
}
```

VI. Thiết kế giao diện

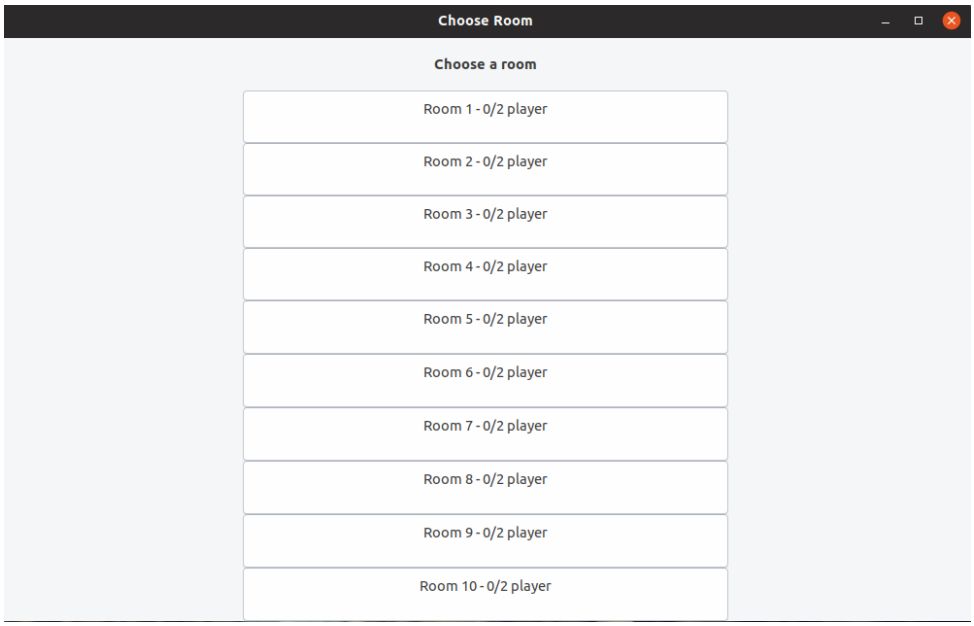
1. Màn hình chính



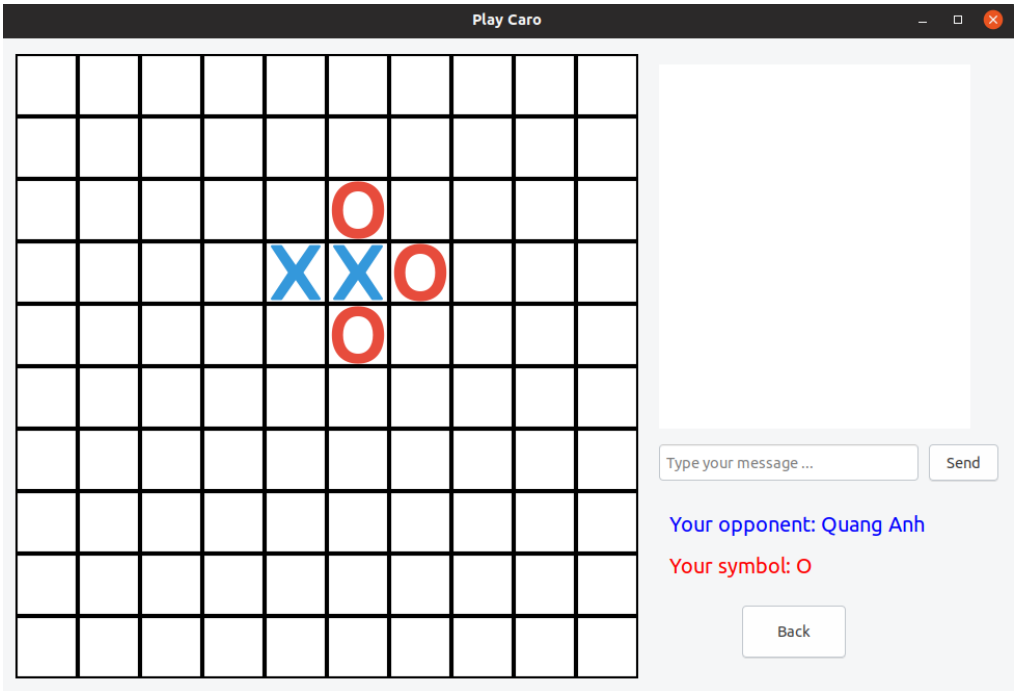
2. Màn hình khi tạo tên thành công



3. Màn hình chọn phòng



4. Màn hình chơi game



VII. Kết luận

1. Kết quả đạt được

Đề tài "Ứng dụng Chơi cờ Caro" đã thực hiện được những nội dung:

- Tìm hiểu được các thức lập trình, các giao thức TCP/IP.
- Tìm hiểu được các thức hoạt động của một chương trình chơi game trực tuyến và chat trực tuyến đơn giản.
- Xây dựng được một chương trình chat đơn giản cài đặt trên mạng với một số tính năng như: gửi tin nhắn…

2. Hướng phát triển

Về cơ bản các nghiên cứu đã đạt được những yêu cầu đã đặt ra. Tuy nhiên nếu có điều kiện, chúng em sẽ cố gắng phát triển thêm một số chức năng khác như:

- Người dùng có thể đăng ký và sử dụng một tài khoản để chơi game.
- Cho phép lưu lại lịch sử người chơi.

VIII. Tài liệu tham khảo

- Slide bài giảng môn Lập trình mạng của thầy Bùi Trọng Tùng, viện Công nghệ Thông tin và Truyền thông, Đại học Bách Khoa Hà Nội.
- Sự hướng dẫn, giảng dạy học phần Thực hành Lập trình mạng của cô Bành Thị Quỳnh Mai, viện Công nghệ Thông tin và Truyền thông, Đại học Bách Khoa Hà Nội.