

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG



MÔN HỌC: LẬP TRÌNH MẠNG CĂN BẢN

BÁO CÁO ĐỒ ÁN

GAME BÀI XÌ DÁCH

Giảng viên hướng dẫn:

ĐẶNG LÊ BẢO CHƯƠNG

Nhóm sinh viên thực hiện:

Dương Bá Cần - 22520143

Nguyễn Duy Nhật Thành - 22521355

Lê Viết Dương - 22520300

THÀNH PHỐ HỒ CHÍ MINH, 2024

MỤC LỤC

PHẦN I: Định nghĩa.....	1
1. Tổng quan về bài Xi Dách	1
2. Luật chơi và các trường hợp đặc biệt	1
PHẦN II: Triển khai đề tài	2
1. Tính năng	2
2. Mô hình phân rã chức năng	3
3. Cấu trúc gói tin	4
4. Network Stack	5
5. Triển khai code phần Network	8
PHẦN III: Bảng phân công công việc.....	13
TÀI LIỆU THAM KHẢO.....	14

PHẦN I: Định nghĩa

1. Tổng quan về bài Xì Dách


Xì dách (hay Blackjack) xuất xứ từ nước Pháp với tên gọi Vingt-et-un và dần dần phổ biến ra toàn thế giới thông qua các hệ thống casino. Trong xì dách, giá trị của con át (xì) là vô cùng đặc biệt – nó có thể là 1, 10 hoặc 11.


Đây là một trò chơi dễ chơi và hấp dẫn người chơi với nhiều thử thách. Mục tiêu trò chơi là đánh bại “Nhà Cái” bằng các quyết định rút hay không những lá bài để có được tổng số điểm nhỏ hơn hoặc bằng 21 điểm, tránh vượt quá 21 điểm để tồn tại trong vòng chơi.

2. Luật chơi và các trường hợp đặc biệt

Tham gia trò chơi:

- Mỗi lần chơi được tối thiểu là 2 người chơi và tối đa là 4 người chơi.
- Người chơi nhập tên ingame và IP của Server để vào phòng chờ.
- Người chơi tham gia phòng chờ đầu tiên sẽ đảm nhận vai trò "Nhà cái" và đồng thời cũng có quyền bắt đầu trò chơi (Chia bài) khi phòng đã đủ người tham gia.

 **Chia bài:** Dùng bộ bài tiêu chuẩn 52 lá, nhà cái sẽ chia lần lượt mỗi người 2 lá. Các lá bài còn dư thì nhà cái cầm nhưng không được xem.

 **Tính điểm:** Sau khi chia bài, mỗi người chơi sẽ tùy vào điểm số của mình mà tùy chọn rút thêm bài hoặc dừng. Tổng điểm là điểm của tất cả các lá bài trên tay cộng lại. Nhà cái sẽ rút bài sau cùng.

Cách tính điểm như sau:

- Các lá bài: 2, 3, 4, 5, 6, 7, 8, 9, 10 thì số điểm tương ứng với con số.
- Các lá bài: J, Q, K thì mỗi lá được tính 10 điểm.
- Lá bài Át (A): Có thể tính linh hoạt là 1 điểm, 10 điểm hoặc 11 điểm.

Giai đoạn 1: Tính điểm 2 lá sau khi chia

Với 2 lá bài đầu tiên, nếu có bài sau đây sẽ dừng luôn để hưởng lợi:

- 2 lá Át (A,A) : Trường hợp Xì bàng

- 1 lá Át (A) và thêm một trong các lá 10, J, Q, K: Trường hợp Xi dách

Nếu nhà cái có bài như trên thì sẽ thắng hết nhà con (ngoại trừ nhà con nào có bài bằng hoặc cao hơn).

Giai đoạn 2: Bốc thêm bài

Sau khi tính điểm 2 lá, người chơi có quyền rút thêm bài hoặc dừng sao cho số điểm nằm trong khoảng từ 16 đến 21 điểm.

Nhà cái sẽ so bài với tất cả nhà con. Nếu nhà cái hơn điểm hoặc hơn trường hợp đặc biệt của một nhà con nào đó thì sẽ thắng nhà con đó. Ngược lại sẽ thua, còn nếu số điểm bằng nhau hoặc trường hợp đặc biệt giống nhau thì hòa.

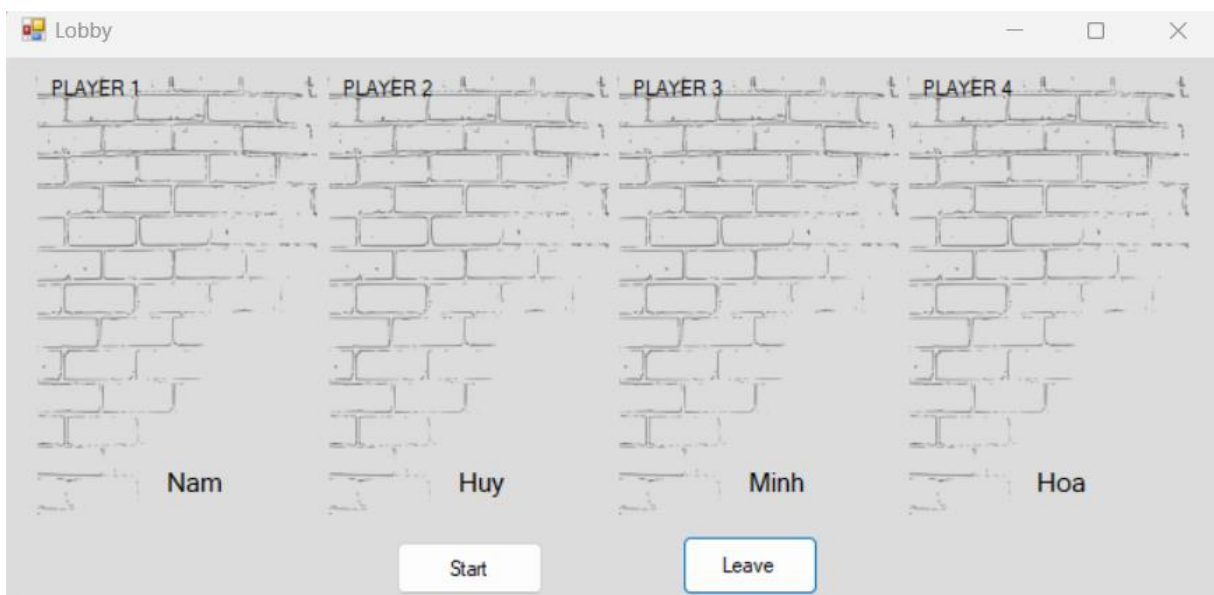
Các trường hợp đặc biệt:

- Xi bàng: bài có 2 lá Át (A,A)
- Xi Dách: bài có 1 lá Át (A) và 1 trong bốn là 10,J,Q,K
- Ngũ linh: 5 lá bài mà tổng điểm nhỏ hơn bằng 21
- Quắc: tổng điểm các lá bài lớn hơn 21 điểm

PHẦN II: Triển khai đề tài

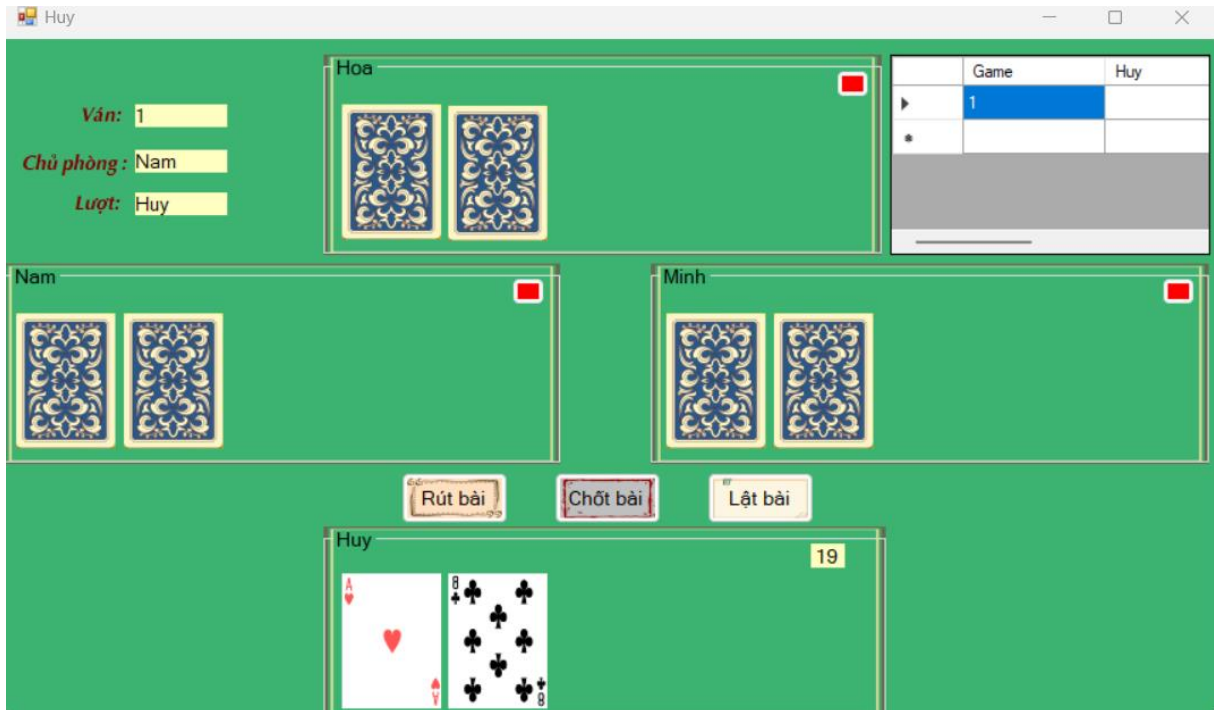
1. Tính năng

a) Thao tác với nút bấm:



Hình 1. Màn hình phòng chờ

🚩 Nút Start (Chia bài): Chỉ dành cho Host (cái)



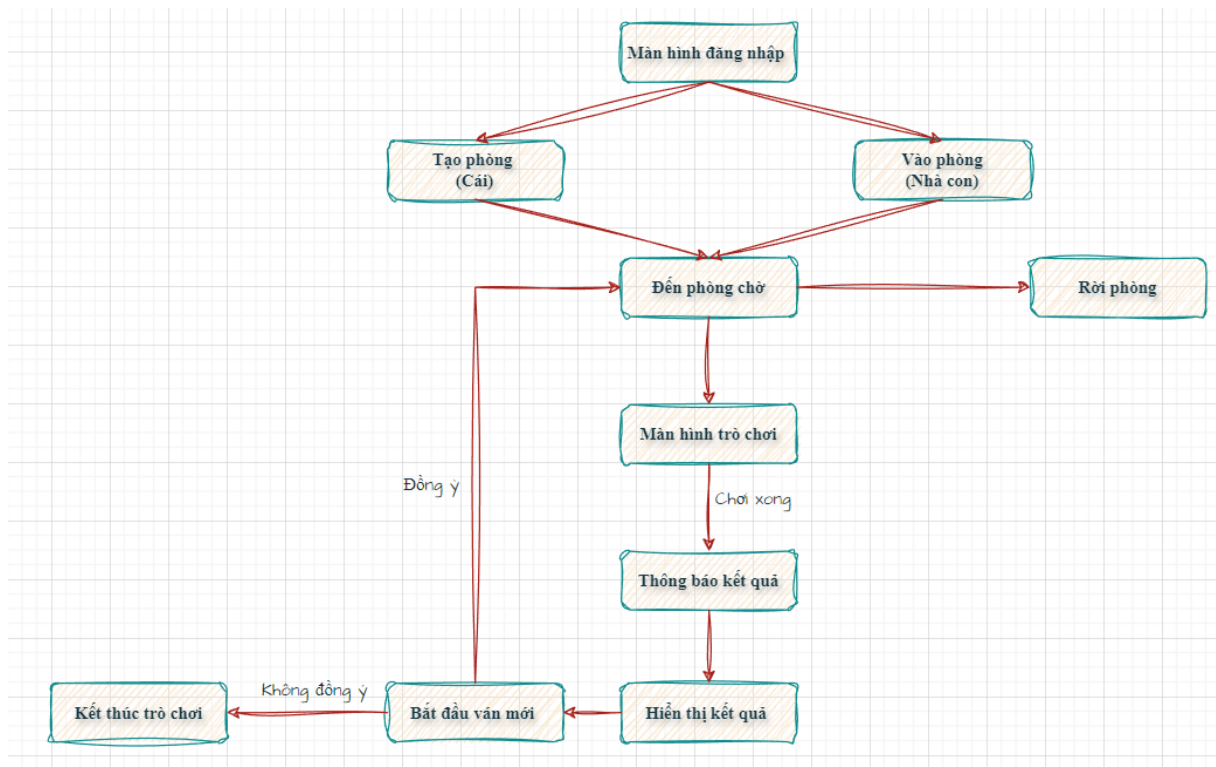
Hình 2. Màn hình phòng chơi

- 🚩 Nút “Rút bài”: dùng để rút 1 lá bài ngẫu nhiên từ bộ bài. Nhà cái và nhà con có tổng điểm lớn hơn 21 không được rút bài.
- 🚩 Nút “Chốt bài”: chỉ dành cho nhà con, dùng để qua lượt khi đã rút bài xong. Nhà con có tổng điểm nhỏ hơn 16 không được chốt bài.
- 🚩 Nút “Lật bài”: dùng để mở bài chính mình cho mọi người xem. Mục đích là hiện thực hóa nhất có thể bài xì dách trong thực tế vào chương trình. Khi người chơi rút bài đến mức có khả năng chiến thắng thì có thể lật trước khi cái xét bài.
- 🚩 Nút “Xét bài”: được thiết kế ở góc mỗi ô thông tin người chơi. Dành riêng cho cái để xét bài nhà con mà họ muốn.

b) Các tính năng khác:

- Tạo phòng, vào phòng
- Có cơ chế tự động mở bài và qua lượt khi có người chơi Xì bàng, Xì Dách. Nếu người đó là nhà cái thì lượt chơi chuyển thành của nhà cái
- Bắt đầu lại một ván mới
- Có DataGridView hiển thị kết quả mỗi ván (Thắng/Thua/Hòa)

2. Mô hình phân rã chức năng



Hình 3. Mô hình phân rã chức năng

3. Cấu trúc gói tin

a) Client đến Server:

Control Message: CONNECT/ DISCONNECT/ DISCONNECTGameRoom/ Start/ DrawCard/ Show/ Block/ ReStart/ Check	Length
Sender: Client	Receiver: Server
Body	

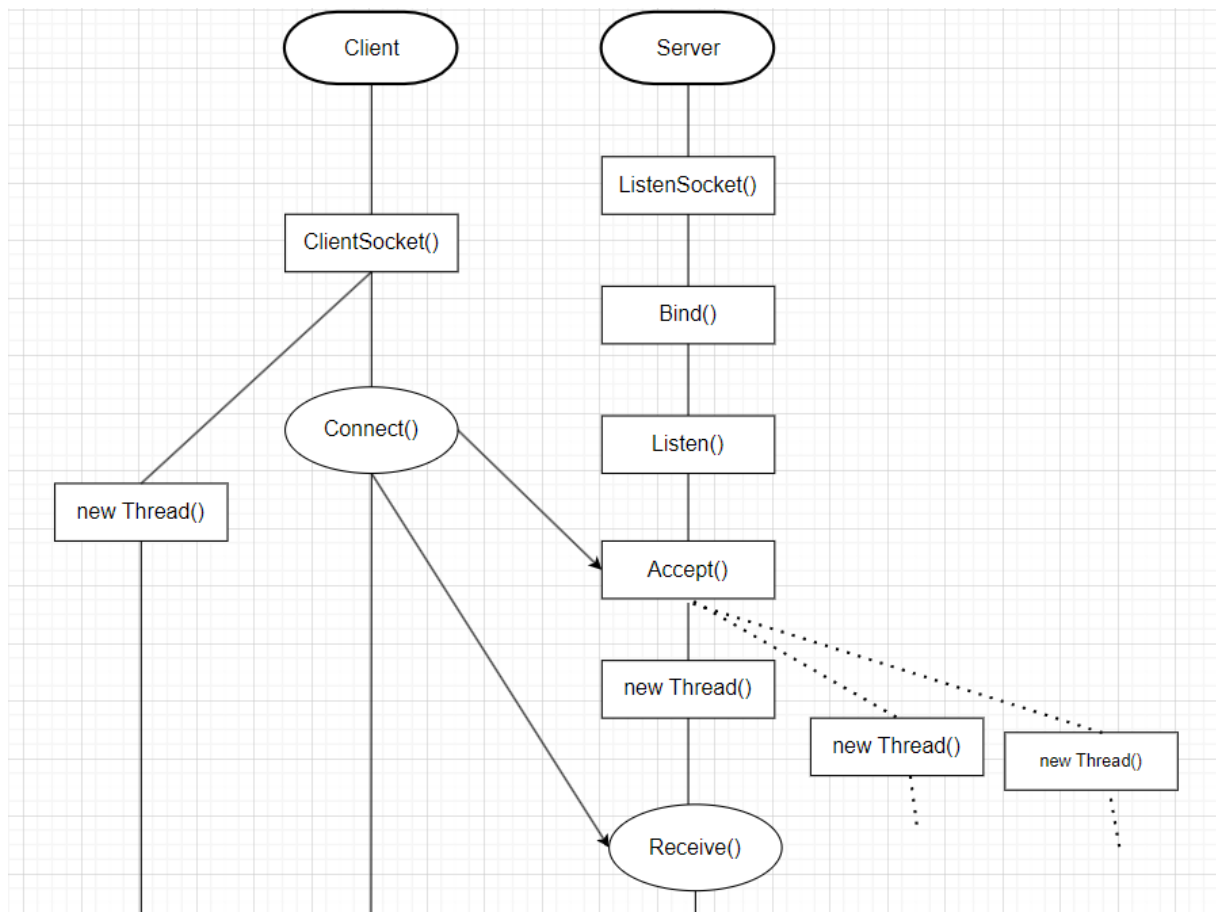
Packet	
Player Name	Turn (Check)

b) Server đến Client :

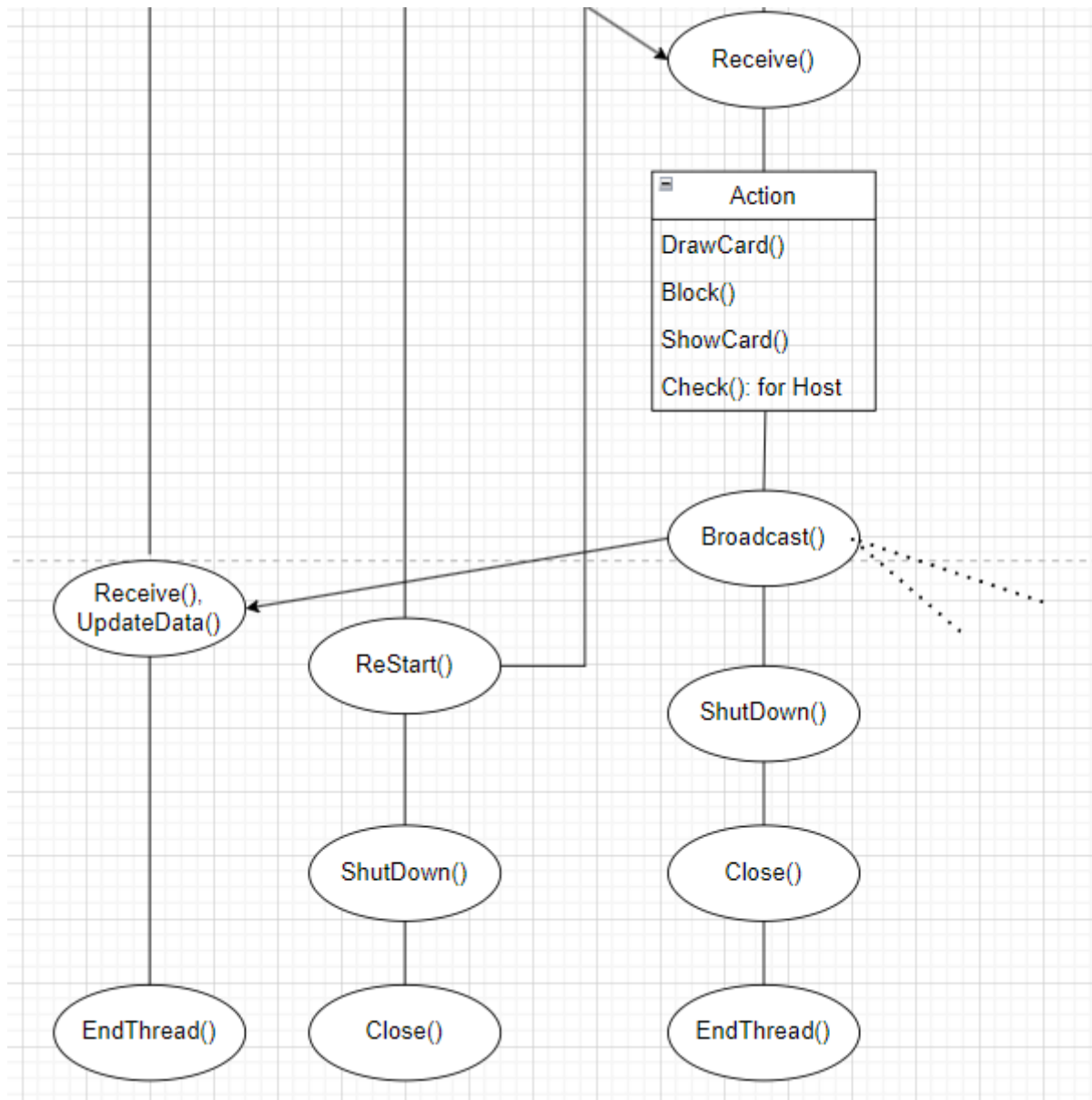
Control Message: LOBBYINFO/ INIT/ OTHERINFO/ OTHERINFORestart/ SETUP/ TURN/ SetResult/ NextTurn/ UpdateCard/ UpdateCardForOtherPlayer/ ShowBy/ ShowSuccess/ ReStart/ ReSetup/ Result/ ResultOther/ CloseAll	Length
Sender: Server	Receiver: Client
Body	

Packet								
Player Name	Turn	Count Card	Show Card (0/1)	ID game	Host (0/1)	List Card	Total Point	Special Case

4. Network Stack



Hình 4. Khởi tạo kết nối



Hình 6. Quá trình thực hiện các hành động cho đến khi kết thúc

5. Triển khai code phần Network

a) Server:

Thiết lập một server socket TCP:

```

static void Main(string[] args)
{
    try
    {
        serverSocket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
        IPEndPoint serverEP = new IPEndPoint(IPAddress.Parse("127.0.0.1"), 8989);
        serverSocket.Bind(serverEP);
        serverSocket.Listen(4);
        Console.WriteLine("[ Waiting for connection from players ... ]");
        while (true)
        {
            client = serverSocket.Accept();
            Console.WriteLine(">> Connection from " + client.RemoteEndPoint);
            clientThread = new Thread(() => readingClientSocket(client));
            clientThread.Start();
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine("Kết nối lỗi: " + ex.ToString());
        Console.Read();
    }
}

```

- Tạo một đối tượng Socket.
- Thiết lập điểm cuối IP (IPEndPoint): gồm địa chỉ IP và Port tùy chỉnh.
- Gán và lắng nghe trên điểm cuối.
- Chờ đợi và chấp nhận kết nối từ Client (tối đa 4 kết nối): Với mỗi kết nối mới, chương trình sẽ tạo một Thread để xử lý việc giao tiếp với client thông qua phương thức `readingClientSocket()`.

Phương thức `readingClientSocket()`:

```

public static void readingClientSocket(Socket client)
{
    Player p = new Player();
    p.playerSocket = client;
    connectedPlayers.Add(p);
    byte[] buffer = new byte[1024];
    while (p.playerSocket.Connected)
    {
        if (p.playerSocket.Available > 0)
        {
            string msg = "";

            while (p.playerSocket.Available > 0)
            {
                int bRead = p.playerSocket.Receive(buffer);
                msg += Encoding.UTF8.GetString(buffer, 0, bRead);
            }

            Console.WriteLine(p.playerSocket.RemoteEndPoint + ": " + msg);
            AnalyzingMessage(msg, p);
        }
    }
}

```

- Tạo một đối tượng Player mới: Gán socket của client (đã kết nối) vào thuộc tính `playerSocket` của đối tượng Player. Thêm đối tượng Player này vào danh sách `connectedPlayers` để quản lý tất cả các player đã kết nối.

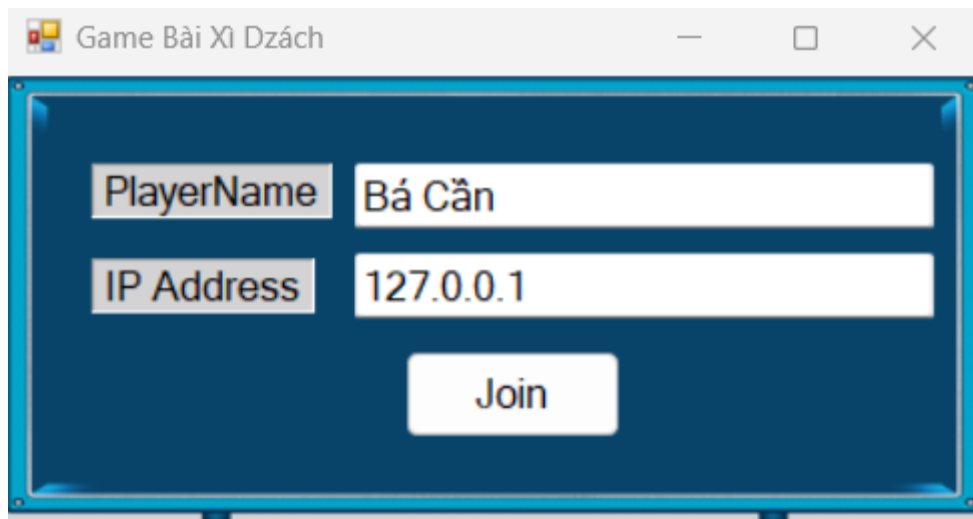
- Khi playerSocket vẫn kết nối và dữ liệu sẵn sàng để đọc, đọc dữ liệu từ socket vào buffer sau đó chuyển đổi dữ liệu từ buffer thành chuỗi msg (string).
- Gọi phương thức `AnalyzingMessage()` để phân tích và xử lý thông điệp yêu cầu từ Client.

Phương thức `AnalyzingMessage()`:

```
public static void AnalyzingMessage(string msg, Player p)
{
    string[] arrPayload = msg.Split(';');
    switch (arrPayload[0])
    {
        case "CONNECT":
        case "DISCONNECT":
        case "DISCONNECTGameRoom":
        case "Start":
        case "DrawCard":
        case "Show":
        case "Block":
        case "ReStart":
        case "Check":
        default:
            break;
    }
}
```

- Xử lý các thông điệp nhận được từ Client và thực hiện các hành động tương ứng như kết nối, ngắt kết nối, bắt đầu trò chơi, rút bài, hiển thị bài, chặn lượt, khởi động lại trò chơi và kiểm tra kết quả.
- Các thông điệp được phân tích và tách thành các phần tử, sau đó các hành động cụ thể được thực hiện dựa trên loại thông điệp và các tham số đi kèm.
- Mỗi loại hành động đều có các bước xử lý cụ thể, từ việc cập nhật thông tin người chơi, gửi thông điệp đến các người chơi khác, đến việc xử lý logic trò chơi.

b) Client:



Bắt sự kiện nút Join trong WelcomeForm:

```
private void btnJoin_Click(object sender, EventArgs e)
{
    if (!string.IsNullOrEmpty(txtPlayerName.Text))
    {
        IPEndPoint serverEP = new IPEndPoint(IPAddress.Parse(txtIpAddress.Text), 8989);
        publicFunction.Connect(serverEP);
        player = new Player();
        player.PlayerName = txtPlayerName.Text;
        publicFunction.thisPlayer = player;
        lobby = new Lobby();
        lobby.publicFunction = publicFunction;
        publicFunction.Send("CONNECT;" + txtPlayerName.Text);
        lobby.FormClosed += new FormClosedEventHandler(lobby_FormClosed);
        this.Hide();
        lobby.Show();
    }
    else
    {
        MessageBox.Show("Vui lòng nhập tên người chơi!");
    }
}
```

- Tạo một đối tượng IPEndPoint để lưu trữ địa chỉ IP và cổng kết nối của Server.
- Sử dụng phương thức **Connect()** từ đối tượng publicFunction để kết nối đến Server.
- Tạo và thiết lập người chơi, tạo phòng chờ (lobby).
- Sử dụng phương thức **Send()** từ đối tượng publicFunction để gửi thông điệp “CONNECT ; Tên người chơi” đến Server.

Phương thức Send():

```
public void Send(string strInput)
{
    byte[] msg = Encoding.UTF8.GetBytes(strInput);
    clientSocket.Send(msg);
}
```

Phương thức Connect():

```
public void Connect(IPEndPoint serverEP)
{
    try
    {
        clientSocket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
        clientSocket.Connect(serverEP);
        recvThread = new Thread(() => Receive());
        recvThread.Start();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.ToString());
    }
}
```

- Tạo một socket, kết nối socket đến điểm cuối mạng serverEP (đại diện cho máy chủ mà ứng dụng sẽ kết nối tới).
- Tạo một luồng mới (recvThread) để chạy phương thức **Receive()** trong nền. Điều này giúp ứng dụng có thể nhận dữ liệu từ máy chủ một cách không đồng bộ mà không làm gián đoạn luồng chính của ứng dụng.

Phương thức Receive():

```
public void Receive()
{
    while (clientSocket.Connected)
    {
        if (clientSocket.Available > 0)
        {
            string msg = "";
            byte[] buffer = new byte[1024];
            while (clientSocket.Available > 0)
            {
                int bRead = clientSocket.Receive(buffer);
                msg = Encoding.UTF8.GetString(buffer, 0, bRead);
            }
            AnalyzingReturnMessage(msg);
        }
    }
}
```

- Khi clientSocket vẫn kết nối và dữ liệu sẵn sàng để đọc, đọc dữ liệu từ socket vào buffer sau đó chuyển đổi dữ liệu từ buffer thành chuỗi msg (string).
- Gọi phương thức **AnalyzingReturnMessage()** để phân tích và xử lý thông điệp phản hồi từ Server.

Phương thức AnalyzingReturnMessage():

```
public void AnalyzingReturnMessage(string msg)
{
    string[] arrPayload = msg.Split(';');

    switch (arrPayload[0])
    {
        case "LOBBYINFO":...
        case "INIT":
            ...
            break;
        case "OTHERINFO":...
        case "OTHERINFORestart":...
        case "SETUP":...
        case "TURN":...
        case "SetResult":...
        case "NextTurn":...
        case "UpdateCard":...
        case "UpdateCardForOtherPlayer":...
        case "ShowBy":...
        case "ShowSuccess":...
        case "ReStart":...
        case "ReSetup":...
        case "Result":...
        case "ResultOther":...
        case "CloseAll":...
        default:
            break;
    }
}
```

Tương tự như phương thức `AnalyzingMessage()`, cũng xử lý các thông điệp nhận được từ Server, phân tích và thực hiện các hành động tương ứng dựa trên nội dung của thông điệp để thể hiện lên màn hình trò chơi và thực hiện logic trò chơi.

PHẦN III: Bảng phân công công việc

Họ và Tên	Nội dung công việc	Đóng góp trong đồ án
Dương Bá Cần	Thiết kế mô hình hoạt động, lập trình socket, logic game, thuyết trình, network stack	50%
Nguyễn Duy Nhật Thành	Làm slide , giao diện các Form, thuyết trình, cấu trúc gói tin, hiện thực luật chơi vào chương trình	25%
Lê Viết Dương	Tổng hợp file báo cáo, giao diện các Form, thuyết trình, mô hình phân rã chức năng	25%

TÀI LIỆU THAM KHẢO

Project Game bài UNO qua LAN – Thực hiện bởi nhóm sinh viên Vũ Quốc An, Đặng Đại Minh Châu, Trần Gia Huy – Khóa K20 – Trường Đại học CNTT.