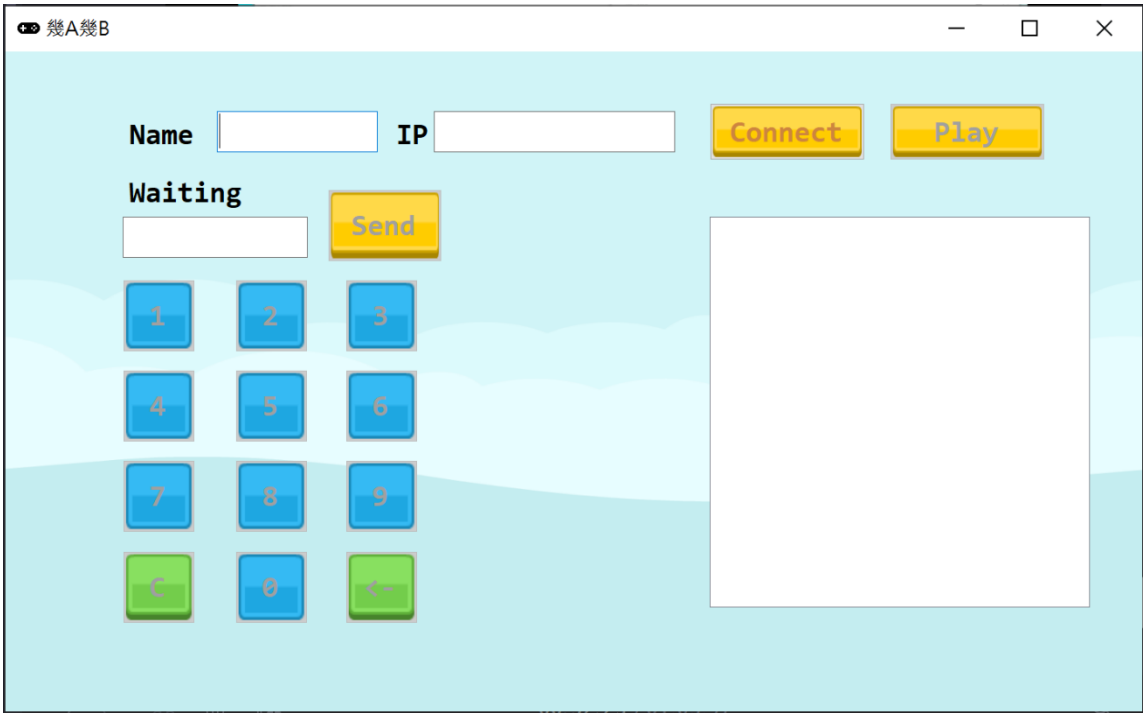
411085030 資工二 張毓旻

電腦網路期末程式作業

主題：1A2B（幾 A 幾 B）

一. 遊戲方法



1. 連線

   (1) 打開 Server，複製 IP 並按下 Server Start 按鈕

   (2) 打開 Client，輸入名字、貼上 IP 再按下 Connect 按鈕

   (3) 玩家至多 4 人，等到所有玩家皆已連線，任意一位玩家按下 Play 按鈕即可開始遊戲

2. 遊玩

   (1) 輸入

   按下方數字鍵便能輸入數字，按下 Send 送出數串

   * C 為清除鍵、<- 為刪除鍵（僅刪除一位數）
   * 若遇到以下狀況，按鈕輸入便會停止輸入
     - 輸入已輸入數串中的數字

- 已輸入數串為 4 位數

* 若送出數串長度小於 4，便會停止送出並跳出提示視窗

(2) 狀態

有 3 種狀態：Ask Ques、Answer、Waiting

- Ask Ques: 輸入數串為玩家的出題
- Answer: 輸入數串為玩家的答案
- Waiting: 尚未輪到玩家

每輪答題玩家各有 5 次機會，輪流答題，若答對便會取得 1 分，先得到 5 分者便會贏得這場比賽

* 側邊 ListBox 會顯示當前遊戲的答題狀態

二. Server Socket

```
public Server()
{
    InitializeComponent();
    Address = GetLocalIpAddress();
    Port = "8080";
    tb_IP.Text = Address + ":" + Port;       // 直接將 IP 寫入 Server 的顯示中

    clientID = 0;
    quesID = 0;
    ansID = 1;
    players = new List<ClientState>();

}

public string GetLocalIpAddress()     // 取得 IP 位置
{
    string ipAddress = string.Empty;
    IPHostEntry hostEntry = Dns.GetHostEntry(Dns.GetHostName());

    foreach (var address in hostEntry.AddressList)
    {
        if (address.AddressFamily == AddressFamily.InterNetwork)
        {
            return address.ToString();
        }
    }
    return ipAddress;
}

public void AcceptConnections()       // 新增 Client 連線
{
    TcpClient new_client;
    while (true)
    {
```

```
            new_client = tcpListener.AcceptTcpClient();
            try
            {
                if (new_client.Connected)
                {
                    players.Add(new ClientState(clientID, new_client));
                    Thread Listen_to_Client = new Thread(Listen);   // 為 Client 開各別的 Thread
                    Listen_to_Client.IsBackground = true;
                    Listen_to_Client.Start(new_client);
                    SendToClient(clientID, "ID" + clientID.ToString(), new_client);
                    ADD_TO_LIST("Client " + players[clientID].Name + " :" +
new_client.Client.RemoteEndPoint + " is joined");
                    clientID++;
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }
    }


    public void Listen(object obj)        // 監聽各 Client 傳來的訊息
    {
        TcpClient this_client = (TcpClient)obj;
        NetworkStream networkStream = this_client.GetStream();
        while (true)
        {
            if (networkStream.CanRead)
            {
                byte[] buffer = new byte[2048];
                int BytesReaded = networkStream.Read(buffer, 0, buffer.Length);
                if (BytesReaded > 0)
                {
                    string Message = Encoding.UTF8.GetString(buffer, 0, BytesReaded);
                    string Command = Message.Substring(0, 2);
```

```
                string iD = Message.Substring(2, 1);
                int ID = int.Parse(iD);
                ADD_TO_LIST("RCV- '" + Message + "' from Client " + iD);
                switch (Command)
                {
                    case "ID":
                        {
                            string name = Message.Substring(3);
                            players[ID].Name = name;
                            break;
                        }
                    case "PL":
                        {
                            for(int i = 0; i < players.Count; i++)
                            {
                                for(int j = 0; j < players.Count; j++)
                                {
                                    SendToClient(i, "AP" + players[j].Name, players[i].Socket);
                                }
                            }
                            NewRound(true);
                            break;
                        }
                    case "QU":
                        {
                            Ans = Message.Substring(3);
                            for (int i = 0; i < players.Count; i++)
                            {
                                SendToClient(i, "FQ" + ID, players[i].Socket);    // ID Finished
Questioning
                            }
                            NextAns(true);
                            break;
                        }
                    case "AN":
                        {
```

```
                    string temp = Message.Substring(3);
                    if(Check(temp, ID))
                    {
                        for(int i = 0; i < players.Count; i++)
                        {
                            SendToClient(i, "RE" + ID, players[i].Socket);  // Round End
(ID is correct)
                        }
                        NewRound();
                    }
                    else
                    {
                        for (int i = 0; i < players.Count; i++)
                        {
                            SendToClient(i, "WA" + ID, players[i].Socket);   // Wrong
Answewr
                        }
                        if (left > 0)
                            NextAns();
                        else
                        {
                            for (int i = 0; i < players.Count; i++)
                            {
                                SendToClient(i, "NO" + Ans, players[i].Socket);
                            }
                            NewRound();
                        }
                    }
                    break;
                }
            case "EN":
                {
                    quesID = -1;
                    break;
                }
            default:
```

```
                        MessageBox.Show("Wrong instruction");
                        break;
                }
            }
        }
    }

public void SendToClient(int Client_ID, string Mesaage, TcpClient client_socket)
{    // 將訊息傳送至 Client
    byte[] data = Encoding.UTF8.GetBytes(Mesaage);
    NetworkStream networkStream = client_socket.GetStream();
    try
    {
        if (networkStream.CanWrite)
        {
            networkStream.Write(data, 0, data.Length);
            ADD_TO_LIST("SND- '" + Mesaage + "' to Client " + Client_ID);
            Thread.Sleep(100);
        }
        else
            ADD_TO_LIST("Fail to sent to Client " + Client_ID);
    }
    catch (IOException ex)
    {
        ADD_TO_LIST(ex.Message);
    }
}
```

三.Client Socket

```
public void Send(string Message)      // 將訊息傳送至 Server
{
    byte[] data = Encoding.UTF8.GetBytes(Message);
    NetworkStream networkStream = client.GetStream();
    try
    {
        if (networkStream.CanWrite)
        {
            networkStream.Write(data, 0, data.Length);
            Thread.Sleep(50);
        }
        else
            ADD_TO_LIST("Fail to send to server ");
    }
    catch (IOException ex)
    {
        ADD_TO_LIST(ex.Message);
    }
}

public void Listen()     // 監聽來自 Server 的訊息
{
    NetworkStream networkStream = client.GetStream();
    while (true)
    {
        if (networkStream.CanRead)
        {
            byte[] buffer = new byte[2048];
            int BytesReaded = networkStream.Read(buffer, 0, buffer.Length);
            if (BytesReaded > 0)
            {
                string Message = Encoding.UTF8.GetString(buffer, 0, BytesReaded);
                string Command = Message.Substring(0, 2);
                switch (Command)
```

```
                {
                    case "ID":
                        {
                            myiD = Message.Substring(2);
                            myID = int.Parse(myiD);
                            ADD_TO_LIST("You are added into the game.");
                            ADD_TO_LIST("NO." + myiD);
                            break;
                        }
                    case "AP":
                        {
                            string tempName = Message.Substring(2);
                            AddPlayer(tempName);
                            ADD_TO_LIST("Player" + (players.Count - 1).ToString() + " " +
tempName + " is in the game.");
                            break;
                        }
                    case "YT":
                        {
                            string func = Message.Substring(2);
                            endl();
                            if (func == "Q")
                                Question();
                            else
                                Answer();
                            break;
                        }
                    case "NQ":
                        {
                            string iD = Message.Substring(2);
                            int ID = int.Parse(iD);
                            endl();
                            ADD_TO_LIST("It's " + players[ID].Name + " turn to ask the
question.");
                            break;
                        }
```

```
                case "NA":
                    {
                        string iD = Message.Substring(2);
                        int ID = int.Parse(iD);
                        endl();
                        ADD_TO_LIST("It's " + players[ID].Name + " turn to answer the
question.");
                        break;
                    }
                case "FQ":
                    {
                        string iD = Message.Substring(2, 1);
                        int ID = int.Parse(iD);
                        ADD_TO_LIST(players[ID].Name + " finished asking a question.");
                        break;
                    }
                case "FA":
                    {
                        string iD = Message.Substring(2, 1);
                        string result = Message.Substring(3);
                        int ID = int.Parse(iD);
                        ADD_TO_LIST(players[ID].Name + "'s answer is " + result + ".");
                        break;
                    }
                case "AB":
                    {
                        string result = Message.Substring(2);
                        ADD_TO_LIST("= " + result);
                        break;
                    }
                case "WA":
                    {
                        string iD = Message.Substring(2, 1);
                        int ID = int.Parse(iD);
                        ADD_TO_LIST(players[ID].Name + " got the wrong answer.");
                        break;
```

```
                    }
                case "RE":
                    {
                        string iD = Message.Substring(2, 1);
                        int ID = int.Parse(iD);
                        ADD_TO_LIST(players[ID].Name + " got the correct answer.");
                        players[ID].Ans++;
                        Refresh(ID);
                        break;
                    }
                case "NO":
                    {
                        string ans = Message.Substring(2);
                        ADD_TO_LIST("No one got the answer.");
                        ADD_TO_LIST("The Ans: " + ans);
                        break;
                    }
                case "EN":
                    {
                        StopAction(true);
                        break;
                    }
            }
        }
    }
}
```