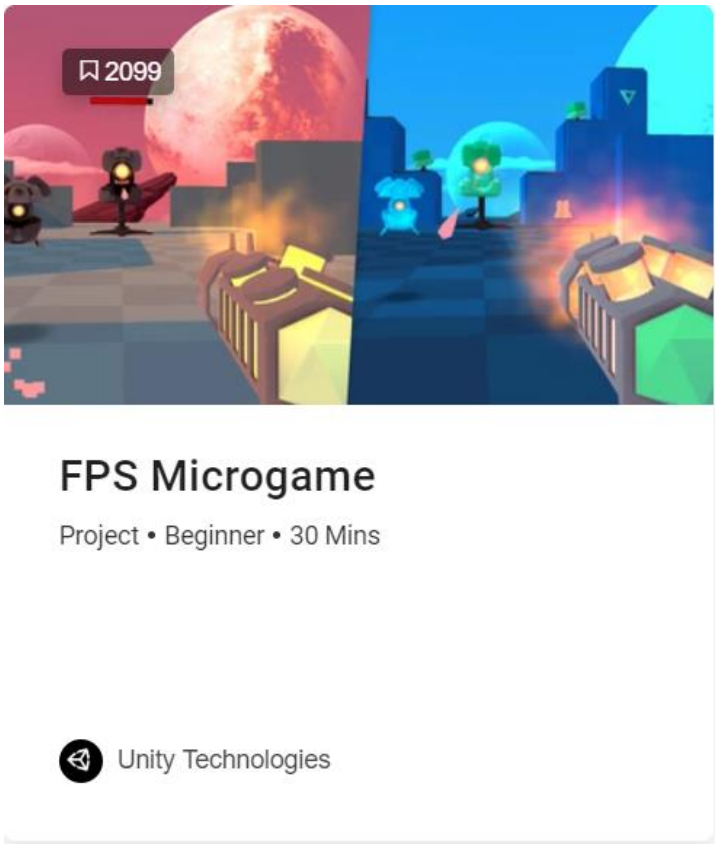
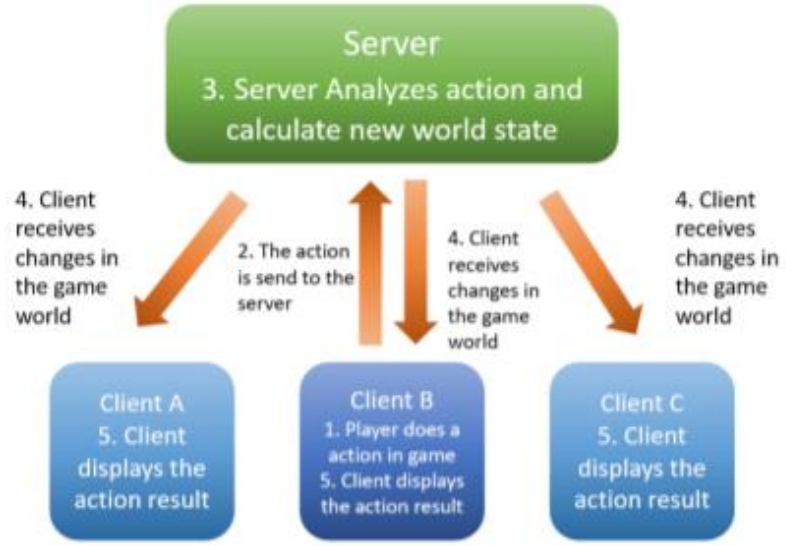
Final Project – Multiplayer Game B05901178 葉咸辰

選取遊戲：Unity FPS

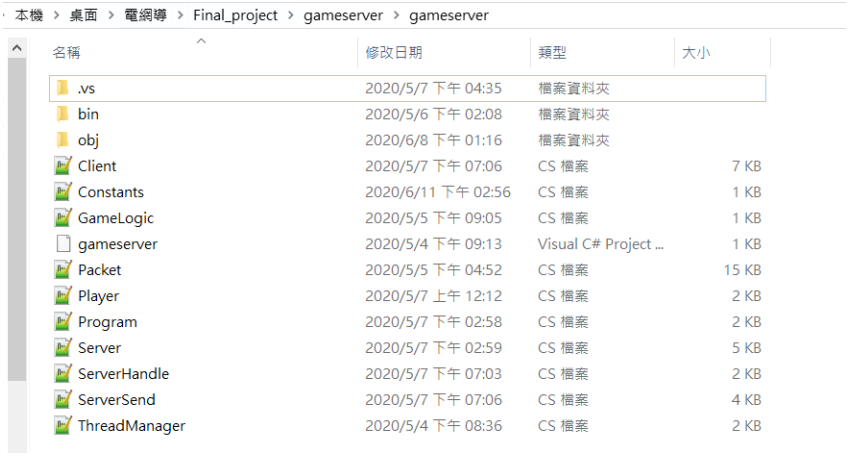


使用 Model：

- Model 1:



gameserver 的資料夾為 server



本機 › 桌面 › 電網導 › Final_project › gameserver › gameserver

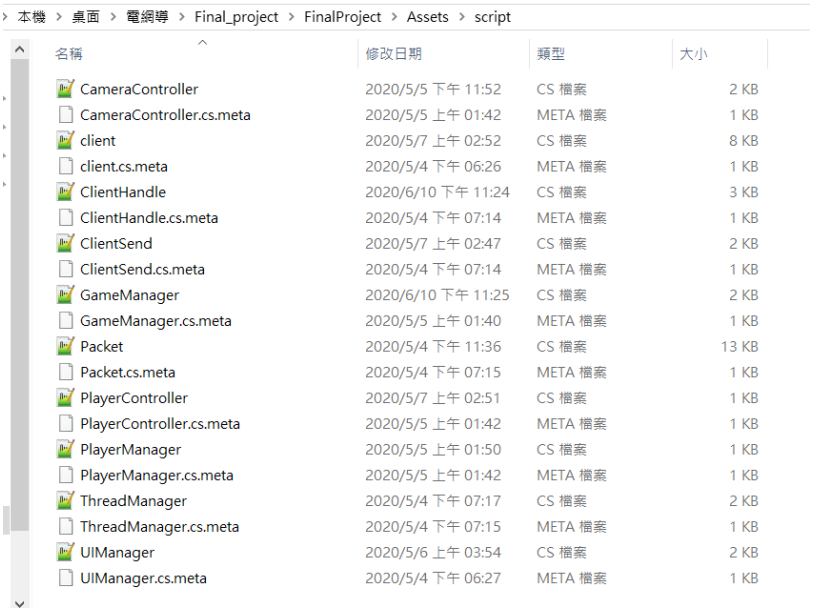
名稱	修改日期	類型	大小
.vs	2020/5/7 下午 04:35	檔案資料夾	
bin	2020/5/6 下午 02:08	檔案資料夾	
obj	2020/6/8 下午 01:16	檔案資料夾	
Client	2020/5/7 下午 07:06	CS 檔案	7 KB
Constants	2020/6/11 下午 02:56	CS 檔案	1 KB
GameLogic	2020/5/5 下午 09:05	CS 檔案	1 KB
gameserver	2020/5/4 下午 09:13	Visual C# Project ...	1 KB
Packet	2020/5/5 下午 04:52	CS 檔案	15 KB
Player	2020/5/7 上午 12:12	CS 檔案	2 KB
Program	2020/5/7 下午 02:58	CS 檔案	2 KB
Server	2020/5/7 下午 02:59	CS 檔案	5 KB
ServerHandle	2020/5/7 下午 07:03	CS 檔案	2 KB
ServerSend	2020/5/7 下午 07:06	CS 檔案	4 KB
ThreadManager	2020/5/4 下午 08:36	CS 檔案	2 KB

Assests 資料夾為改動的部分

本機 › 桌面 › 電網導 › Final_project › FinalProject › Assets

名稱	修改日期	類型	大小
FPS	2020/6/8 下午 03:27	檔案資料夾	
Mirror	2020/6/10 下午 02:40	檔案資料夾	
ModAssets	2020/6/8 下午 03:24	檔案資料夾	
script	2020/6/8 下午 04:16	檔案資料夾	
ScriptTemplates	2020/6/10 下午 02:40	檔案資料夾	
TextMesh Pro	2020/6/8 下午 03:24	檔案資料夾	
FPS.meta	2020/4/24 上午 04:47	META 檔案	1 KB
IntroMenu	2020/6/10 下午 11:30	Unity scene file	84 KB
IntroMenu.unity.meta	2020/4/24 上午 04:47	META 檔案	1 KB
MainScene	2020/6/10 下午 11:28	Unity scene file	51 KB
MainScene.unity.meta	2020/6/10 下午 11:28	META 檔案	1 KB
Mirror.meta	2020/5/6 上午 01:38	META 檔案	1 KB
ModAssets.meta	2020/4/24 上午 04:47	META 檔案	1 KB
script.meta	2020/6/8 下午 06:34	META 檔案	1 KB
ScriptTemplates.meta	2020/5/6 上午 01:38	META 檔案	1 KB
TextMesh Pro.meta	2020/4/24 上午 04:47	META 檔案	1 KB

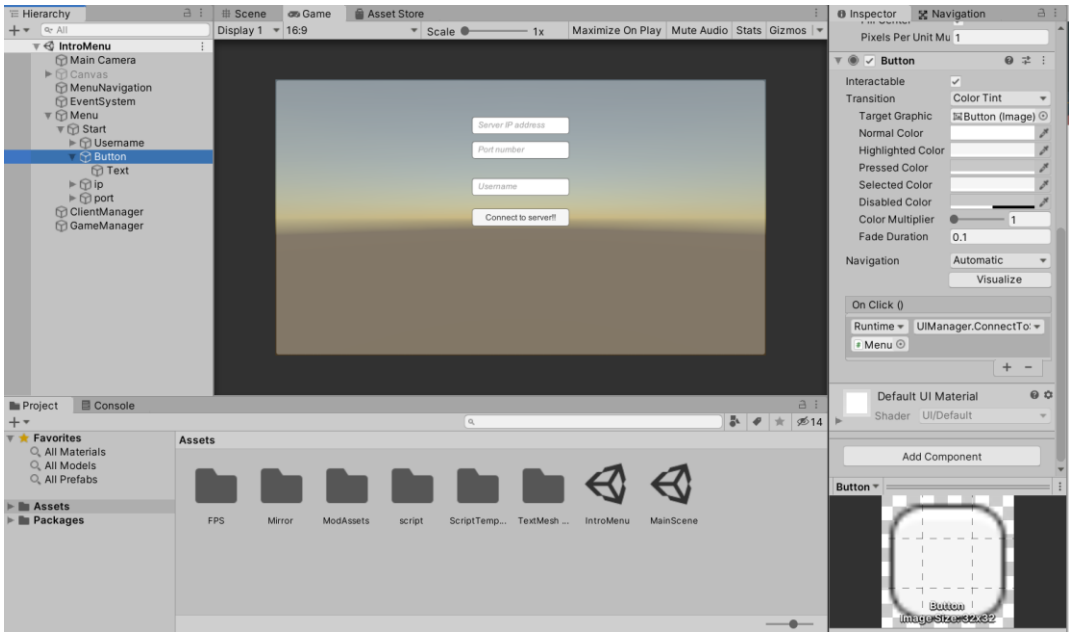
加了 script 去做 client 和 server 及遊戲的連線



本機 › 桌面 › 電網導 › Final_project › FinalProject › Assets › script

名稱	修改日期	類型	大小
CameraController	2020/5/5 下午 11:52	CS 檔案	2 KB
CameraController.cs.meta	2020/5/5 上午 01:42	META 檔案	1 KB
client	2020/5/7 上午 02:52	CS 檔案	8 KB
client.cs.meta	2020/5/4 下午 06:26	META 檔案	1 KB
ClientHandle	2020/6/10 下午 11:24	CS 檔案	3 KB
ClientHandle.cs.meta	2020/5/4 下午 07:14	META 檔案	1 KB
ClientSend	2020/5/7 上午 02:47	CS 檔案	2 KB
ClientSend.cs.meta	2020/5/4 下午 07:14	META 檔案	1 KB
GameManager	2020/6/10 下午 11:25	CS 檔案	2 KB
GameManager.cs.meta	2020/5/5 上午 01:40	META 檔案	1 KB
Packet	2020/5/4 下午 11:36	CS 檔案	13 KB
Packet.cs.meta	2020/5/4 下午 07:15	META 檔案	1 KB
PlayerController	2020/5/7 上午 02:51	CS 檔案	1 KB
PlayerController.cs.meta	2020/5/5 上午 01:42	META 檔案	1 KB
PlayerManager	2020/5/5 上午 01:50	CS 檔案	1 KB
PlayerManager.cs.meta	2020/5/5 上午 01:42	META 檔案	1 KB
ThreadManager	2020/5/4 下午 07:17	CS 檔案	2 KB
ThreadManager.cs.meta	2020/5/4 下午 07:15	META 檔案	1 KB
UIManager	2020/5/6 上午 03:54	CS 檔案	2 KB
UIManager.cs.meta	2020/5/4 下午 06:27	META 檔案	1 KB

TCP UDP 連線方式




```
    public void ConnectToServer()
    {
        startMenu.SetActive(false);
        usernameField.interactable = false;
        client.instance.ConnectToServer(IPaddress.text.ToString(), int.Parse(Port.text));
    }

    public void ConnectToServerUDP(int _local)
    {
        client.instance.udp.Connect(_local, IPaddress.text.ToString(), int.Parse(Port.text));
        Debug.Log("test for connect to server with UDP");
    }
}
```

```
    public void ConnectToServer(String ipaddress, int port)
    {
        InitializeClientData();

        isConnected = true;
        tcp.Connect(ipaddress, port);
    }

private void InitializeClientData()
{
    packetHandlers = new Dictionary<int, PacketHandler>()
    {
        { (int)ServerPackets.welcome, ClientHandle.Welcome },
        { (int)ServerPackets.spawnPlayer, ClientHandle.SpawnPlayer },
        { (int)ServerPackets.playerPosition, ClientHandle.PlayerPosition },
        { (int)ServerPackets.playerRotation, ClientHandle.PlayerRotation }
    };
    Debug.Log("Initialized packets.");
}
```

```
public static void PlayerRotation(Packet _packet)
{
    int _id = _packet.ReadInt();
    Quaternion _rotation = _packet.ReadQuaternion();
    try
    {
        GameManager.players[_id].transform.rotation = _rotation;
    }
    catch (KeyNotFoundException e)
    {
        Debug.Log($"{e}");
        // do nothing, since we have not spawn the client
    }
}
```

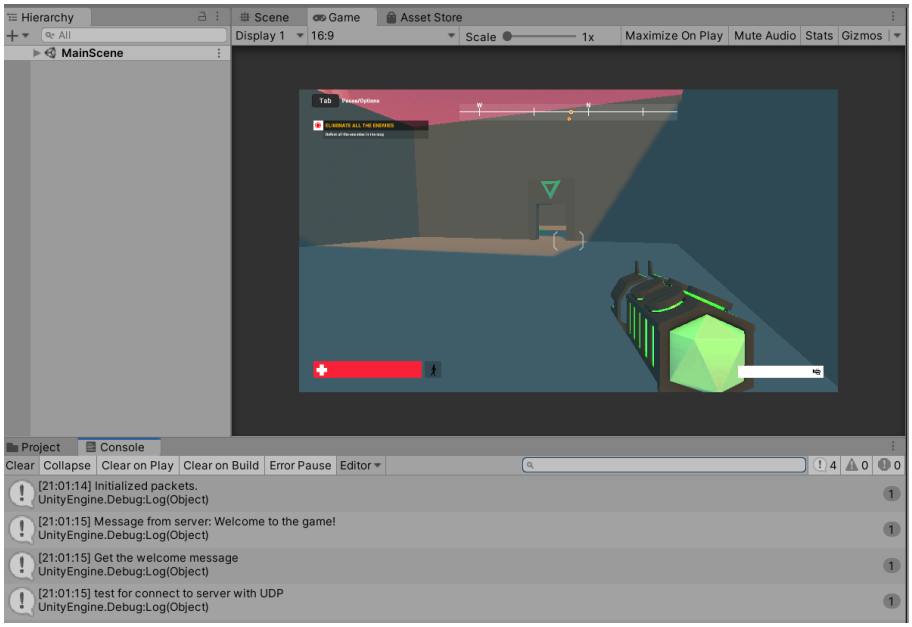
```
public static void Welcome(Packet _packet)
{
    string _msg = _packet.ReadString();
    int _myId = _packet.ReadInt();

    if(_msg == null){
        return;
    }
    else{
        SceneManager.LoadScene("MainScene");
    }
    Debug.Log($"Message from server: {_msg}");
    client.instance.id = _myId;
    ClientSend.WelcomeReceived();
    Debug.Log("Get the welcome message");
    UIManager.instance.ConnectToServerUDP(((IPEndPoint)client.instance.tcp.socket.Client.LocalEndPoint).Port);
}

public static void SpawnPlayer(Packet _packet)
{
    int _id = _packet.ReadInt();
    string _username = _packet.ReadString();
    Vector3 _position = _packet.ReadVector3();
    Quaternion _rotation = _packet.ReadQuaternion();

    GameManager.instance.SpawnPlayer(_id, _username, _position, _rotation);
}

public static void PlayerPosition(Packet _packet)
{
    int _id = _packet.ReadInt();
    Vector3 _position = _packet.ReadVector3();
    if ((_id != client.instance.id))
    {
        Debug.Log($"UDP Message from server: {_id} and new position x:{ _position.x}");
        try
        {
            GameManager.players[_id].transform.position = _position;
        }
        catch (KeyNotFoundException e)
        {
            // do nothing, since we have not spawn the client
            Debug.Log($"{e}");
        }
    }
}
```

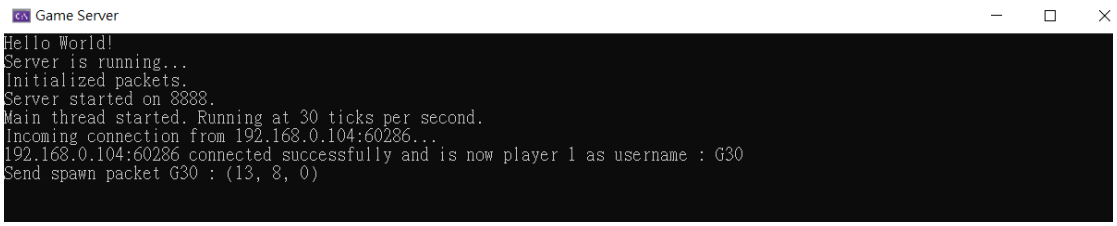
```
public class TCP
{
    public TcpClient socket;

    private NetworkStream stream;
    private Packet receivedData;
    private byte[] receiveBuffer;


    public void Connect(String ipaddress, int port)
    {
        socket = new TcpClient
        {
            ReceiveBufferSize = dataBufferSize,
            SendBufferSize = dataBufferSize
        };

        receiveBuffer = new byte[dataBufferSize];
        socket.BeginConnect(IPAddress.Parse(ipaddress), port, ConnectCallback, socket);
    }
```

```
public class UDP
{
    public UdpClient socket;
    public IPEndPoint endPoint;


    public void Connect(int _localPort, String ipaddress, int port)
    {

        endPoint = new IPEndPoint(IPAddress.Parse(ipaddress), port);
        socket = new UdpClient(_localPort);

        socket.Connect(endPoint);
        socket.BeginReceive(ReceiveCallback, null);

        using (Packet _packet = new Packet())
        {
            SendData(_packet);
        }
    }
```

Demo

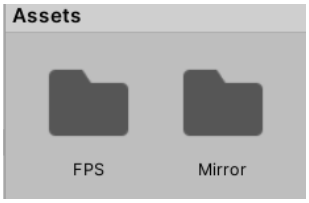
```
using System;
using System.Collections.Generic;
using System.Text;

namespace gameserver
{
    class Constants
    {
        public const string serverIPaddress = "192.168.0.104";
        public const int serverPort = 8888;
        public const int MaxPlayerNumber = 30;
        public const int TICKS_PER_SEC = 30;
        public const int MS_PER_TICK = 1000 / TICKS_PER_SEC;
    }
}
```

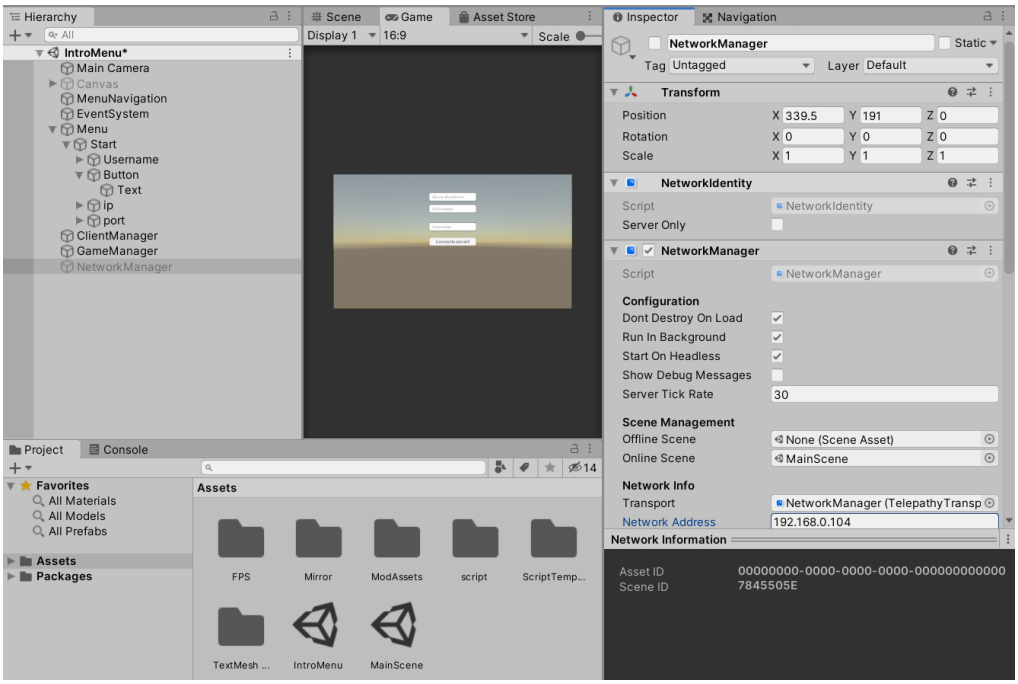




到此為我成功完成的部分

其他嘗試：Mirror






```
using Mirror;

public class NetworkGamePlayer : NetworkBehaviour
{
    [SyncVar]
    private string displayName = "Loading...";

    private NetworkManager room;
    private NetworkManager Room
    {
        get
        {
            if (room != null) { return room; }
            return room = NetworkManager.singleton as NetworkManager;
        }
    }

    public override void OnStartClient()
    {
        DontDestroyOnLoad(gameObject);

        Room.GamePlayers.Add(this);
    }

    public override void OnNetworkDestroy()
    {
        Room.GamePlayers.Remove(this);
    }

    [Server]
    public void SetDisplayName(string displayName)
    {
        this.displayName = displayName;
    }
}
```

```
using Mirror;
using System;
using System.Collections.Generic;
using System.Linq;
using UnityEngine;
using UnityEngine.SceneManagement;


public class NetworkManager : NetworkManager
{
    [SerializeField] private int minPlayers = 2;
    [Scene] [SerializeField] private string menuScene = string.Empty;

    [Header("Maps")]
    [SerializeField] private int numberOfRounds = 1;
    [SerializeField] private MapSet mapSet = null;

    [Header("Room")]
    [SerializeField] private NetworkRoomPlayer roomPlayerPrefab = null;

    [Header("Game")]
    [SerializeField] private NetworkGamePlayer gamePlayerPrefab = null;
    [SerializeField] private GameObject playerSpawnSystem = null;
    [SerializeField] private GameObject roundSystem = null;

    private MapHandler mapHandler;

    public static event Action OnClientConnected;
    public static event Action OnClientDisconnected;
    public static event Action<NetworkConnection> OnServerReadied;
    public static event Action OnServerStopped;

    public List<NetworkRoomPlayer> RoomPlayers { get; } = new List<NetworkRoomPlayer>();
    public List<NetworkGamePlayer> GamePlayers { get; } = new List<NetworkGamePlayer>();

    public override void OnStartServer() => spawnPrefabs = Resources.LoadAll<GameObject>("SpawnablePrefabs").ToList();

    public override void OnStartClient()
    {
        var spawnablePrefabs = Resources.LoadAll<GameObject>("SpawnablePrefabs");

        foreach (var prefab in spawnablePrefabs)
```

```
using Mirror;
using TMPro;
using UnityEngine;
using UnityEngine.UI;


public class NetworkRoomPlayer : NetworkBehaviour
{
    [Header("UI")]
    [SerializeField] private GameObject lobbyUI = null;
    [SerializeField] private TMP_Text[] playerNameTexts = new TMP_Text[4];
    [SerializeField] private TMP_Text[] playerReadyTexts = new TMP_Text[4];
    [SerializeField] private Button startGameButton = null;

    [SyncVar(hook = nameof(HandleDisplayNameChanged))]
    public string DisplayName = "Loading...";
    [SyncVar(hook = nameof(HandleReadyStatusChanged))]
    public bool IsReady = false;

    private bool isLeader;
    public bool IsLeader
    {
        set
        {
            isLeader = value;
            startGameButton.gameObject.SetActive(value);
        }
    }

    private NetworkManager room;
    private NetworkManager Room
    {
        get
        {
            if (room != null) { return room; }
            return room = NetworkManager.singleton as NetworkManager;
        }
    }

    public override void OnStartAuthority()
    {
        CmdSetDisplayName(PlayerNameInput.DisplayName);
```

但最後沒有成功...

Work Distribution：全部皆我自己完成

References

[1] https://learn.unity.com/projects
[2] https://www.youtube.com/watch?v=5LhA4Tk_uvI&list=PLS6sInD7ThM1aUDj81ZrF4b4lpvejB2uB
[3] https://github.com/DapperDino/Mirror-Multiplayer-Tutorials
[4] https://gamedevacademy.org/how-to-create-a-multiplayer-game-in-unity/
[5] https://docs.unity3d.com/Manual/UNet.html