

Unity3D 具備遊戲大廳的連線機制，使用者可以透過在 Unity3D 提供的主服務器 (Master Server) 註冊一個可讓其他人連線進入的遊戲房間，而不需要輸入開啟遊戲房間者的 IP 位置或 GUID。

主服務器 (Master Server) 與第八章範例所提到的伺服器 (Server) 是不同的概念，主服務器 (Master Server) 的功能只為連線建立連線橋梁，遊戲的伺服器 (Server) 在主服務器 (Master Server) 上註冊遊戲，讓客戶端 (Client) 不必輸入額外的資訊即可連線上伺服器 (Server)，當伺服器 (Server) 與客戶端 (Client) 連上線後，主服務器 (Master Server) 就不會介入伺服器 (Server) 與客戶端 (Client)。

## 設定私人主服務器

主服務器 (Master Server) 是透過 `MasterServer` 類別運作，其實它是一個小型的伺服器，Unity3D 本身有提供預設的主服務器 (Master Server)，但是僅供測試用，如果是正式對外開放的遊戲，則必須自己架設主服務器 (Master Server)。架設方式可透過 Unity3D 官方網站提供的架設方式，或由本書所提供的執行程式。

所以使用主服務器 (Master Server) 的第一步驟就是選定官方的主服務器 (Master Server)，還是私人架設的主服務器 (Master Server)。

如果是選用私人的主服務器 (Master Server)，官方有提供下載，執行結果如圖 H-1，要連上私人主服務器必須在一開始設定 `MasterServer ipAddress` 與 `MasterServer.port`，如下所列的程式碼內容，如果沒設定則是使用官方預設的主服務器 (Master Server)。如果主服務器 (Master Server) 有問題，會顯示如圖 H-2 的錯誤，圖 H-2 為官方主服務器 (Master Server) 連線失敗的示意圖。

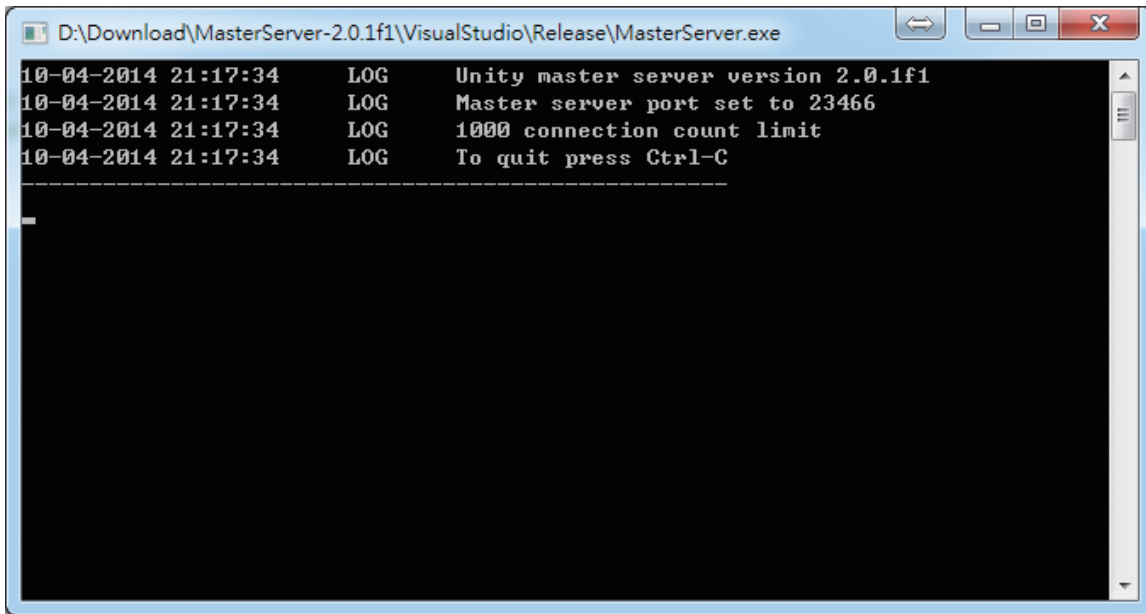


圖 H-1 私人主服務器 (Master Server)

#### 設定私人主服務器 (Master Server)

```
void Start() {
```

```
    MasterServer.ipAddress = "127.0.0.1";
```

```
    MasterServer.port = 23466;
```

```
}
```

Ⓐ 設定主服務器 (Master Server) 的 IP

Ⓑ 設定主服務器 (Master Server) 的 port

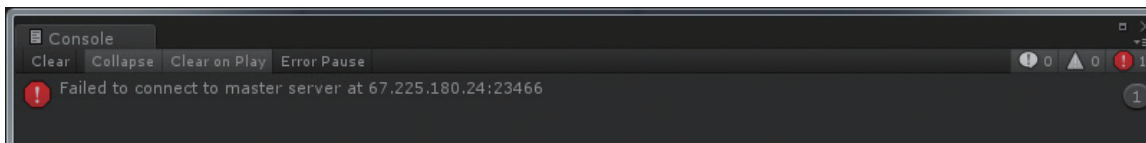


圖 H-2 主服務器 (Master Server) 連線失敗

## 透過主服務器建立伺服器

選擇好主服務器 (Master Server) 後，再來是與前面範例一樣，決定要當伺服器 (Server)，還是客戶端 (Client)，本範例是 P2P 連線機制，已選擇是伺服器 (Server)，其身分依然是一位玩家，不同於主從式網路 (Client Server) 的伺服器 (Server)。圖 H-3 為此範例的示意圖，此圖還尚未有人開啟伺服器 (Server)，圖 H-4 為完整的功能圖。

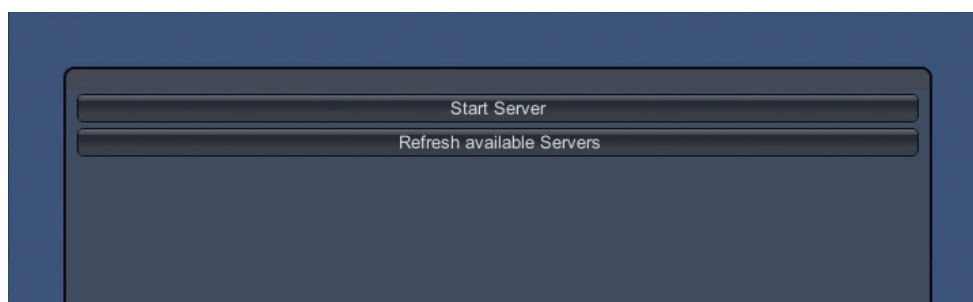


圖 H-3 主服務器 (Master Server) 範例示意圖

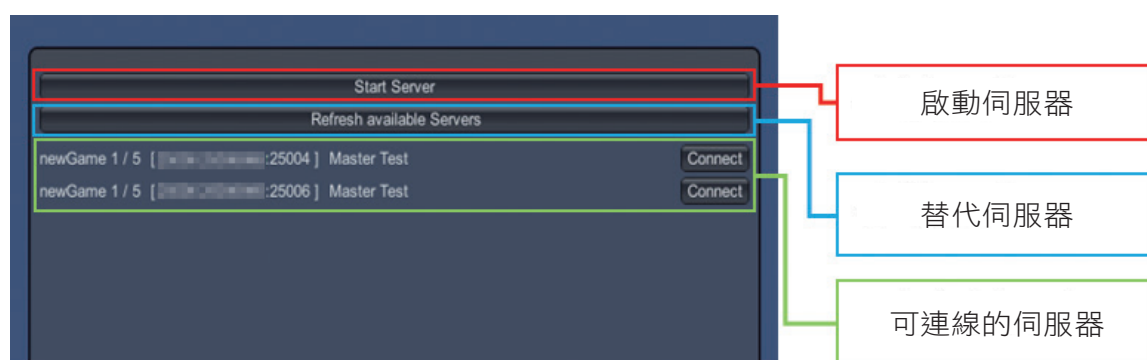


圖 H-4 主服務器功能示意圖

在按下啟動伺服器按鈕，除了進行如第八章範例的初始化伺服器 (Server) `Network.InitializeServer()` 外，還必須向主服務器 (Master Server) 註冊，才能讓客戶端 (Client) 看到。註冊方法使用的是透過 `MasterServer.RegisterHost()`，註冊後主服務器 (Master Server) 就會有這些資訊。

**MasterServer.RegisterHost()**

```
MasterServer.RegisterHost(string gameType, string gameName, string commit);
```

參數	型態	說明
gameType	string	遊戲型態名稱
gameName	string	遊戲名稱
commit	string	遊戲簡介

表 H-1 註冊主服務器 (Master Server)

伺服器 (Server) 開啟後，就開始等待客戶端 (Client) 連接，如圖 H-5 所示，其實就是一個遊戲房間，等待其他玩家的加入。

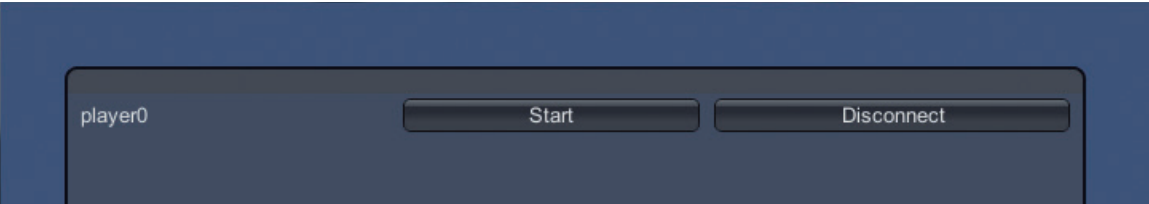


圖 H-5 遊戲房間截圖

### 客戶端連上主服務器上的伺服器

再來就是客戶端 (Client) 如何取得主服務器 (Master Server) 已經開好的伺服器 (Server) 資料，主要是透過 `MasterServer.RequestHostList()` 取得想要的遊戲型態，再由 `MasterServer.PollHostList()` 取得所有資料，如下所示。`MasterServer.PollHostList()` 取得的資料型態為 `HostData` 的陣列，`HostData` 資料結構如表 H-3。

`MasterServer.PollHostList()`

```
MasterServer.PollHostList(string gameType);
```

參數	型態	說明
gameType	string	遊戲型態名稱

表 H-2 `MasterServer.PollHostList()` 取得資料

取得主服務器 (Master Server) 資料

```
void Awake() {  
    MasterServer.ClearHostList();  
    MasterServer.RequestHostList("GameType");  
}  
  
void Update() {  
    HostData[] hostData = MasterServer.PollHostList();  
    int i = 0;  
    foreach (HostData element in hostData) {  
        ...  
    }  
}
```

A 清空舊的資料

B 取得想要的遊戲型態 (名為 GameType 的遊戲型態)

C 取得詳細資料

D 處理取得的 HostData 資料

變數	說明
HostData.comment	註冊時所輸入的遊戲簡介。
HostData.connectedPlayers	目前此伺服器的總連線數量 (客戶端數量)。
HostData.playerLimit	此伺服器的可連線數量上限 (客戶端數量上限)。
HostData.gameName	註冊時所輸入的遊戲名稱。
HostData.gameType	註冊時所輸入的遊戲型態。
HostData.useNat	伺服器連線方式是否有使用網路位址轉換穿透 (NAT punchthrough)。
HostData.guid	網路位址轉換穿透 (NAT punchthrough) 所需的 GUID。
HostData.ip	伺服器的 IP。
HostData.port	伺服器的通訊埠 (Port)。
HostData.passwordProtected	是否由密碼保護。

表 H-3 HostData 資料介紹

透過處理 HostData.useNat 方式如下所示，來決定連線方式，不需要讓客戶端 (Client) 輸入 IP 及 GUID。

#### HostData.useNat 判斷連線類型

```
HostData[] data = MasterServer.PollHostList();
foreach(var element in data){
    if(element.connectedPlayers<element.playerLimit){
        if (GUILayout.Button("Connect")){
            if (element.useNat){
                Network.Connect(element.guid);
            }else{
                Network.Connect(element.ip, element.port);
            }
        }
    }
}
```

Ⓐ 目前連線數 element.connectedPlayers 小於最大連線數 element.playerLimit 才可進行連線

Ⓑ 透過網路位址轉換穿透 (NAT punchthrough)

Ⓒ 透過伺服器 IP 與通訊埠 (Port) 連線

當有客戶端 (Client) 連上伺服器 (Server) 後，就好比有玩家加入了遊戲房，此時，開起遊戲房間的伺服器 (Server) 畫面將會如圖 H-6 所示，開啟者視同房間的主人，可以決定遊戲何時開始，當然也能決定斷線關閉房間。



圖 H-6 遊戲房主啟動畫面

客戶端 (Client)，也就是其他玩家，連上伺服器 (Server) 後，相當於加入了這間遊戲房間，身分是客人，所以無法主導遊戲的開始，但是，玩家有權利離開此房間，需要提供客戶端 (Client) 選擇斷線的功能，如圖 H-7 所示。

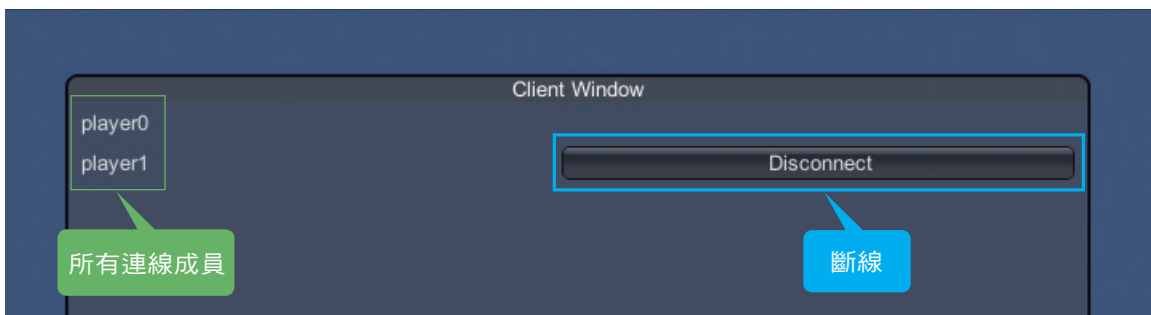


圖 H-7 遊戲房客啟動畫面

## 遊戲房間的連線管理

伺服器 (Server)，也就是遊戲房間的主人，有權力決定遊戲開始，當然也要負起管理房間裡所有房客連線。有玩家加入房間，也就是有客戶端 (Client) 連線進入伺服器 (Server) 時，必須告知遊戲房內的所有玩家，如圖 H-8；當有玩家離開遊戲房間，也就是客戶端 (Client) 斷線離開時，伺服器 (Server) 必須負責告訴其他連線中的客戶端 (Client)，還有哪些玩家還留在此遊戲房間中，如圖 H-9，才能在各玩家畫面顯示正確的連線玩家資訊。依照此邏輯撰寫即可完成此範例。

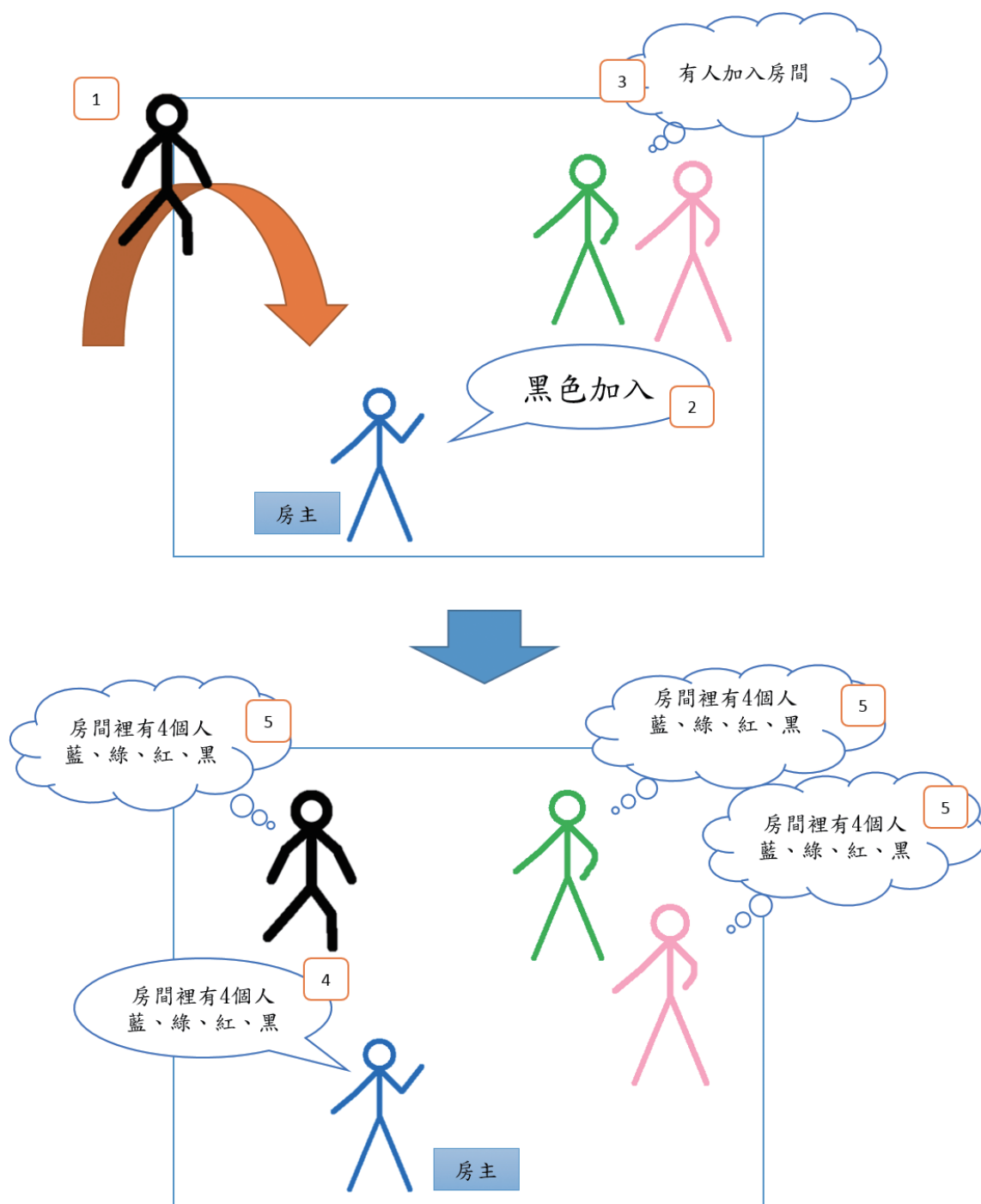


圖 H-8 玩家加入遊戲房間示意圖

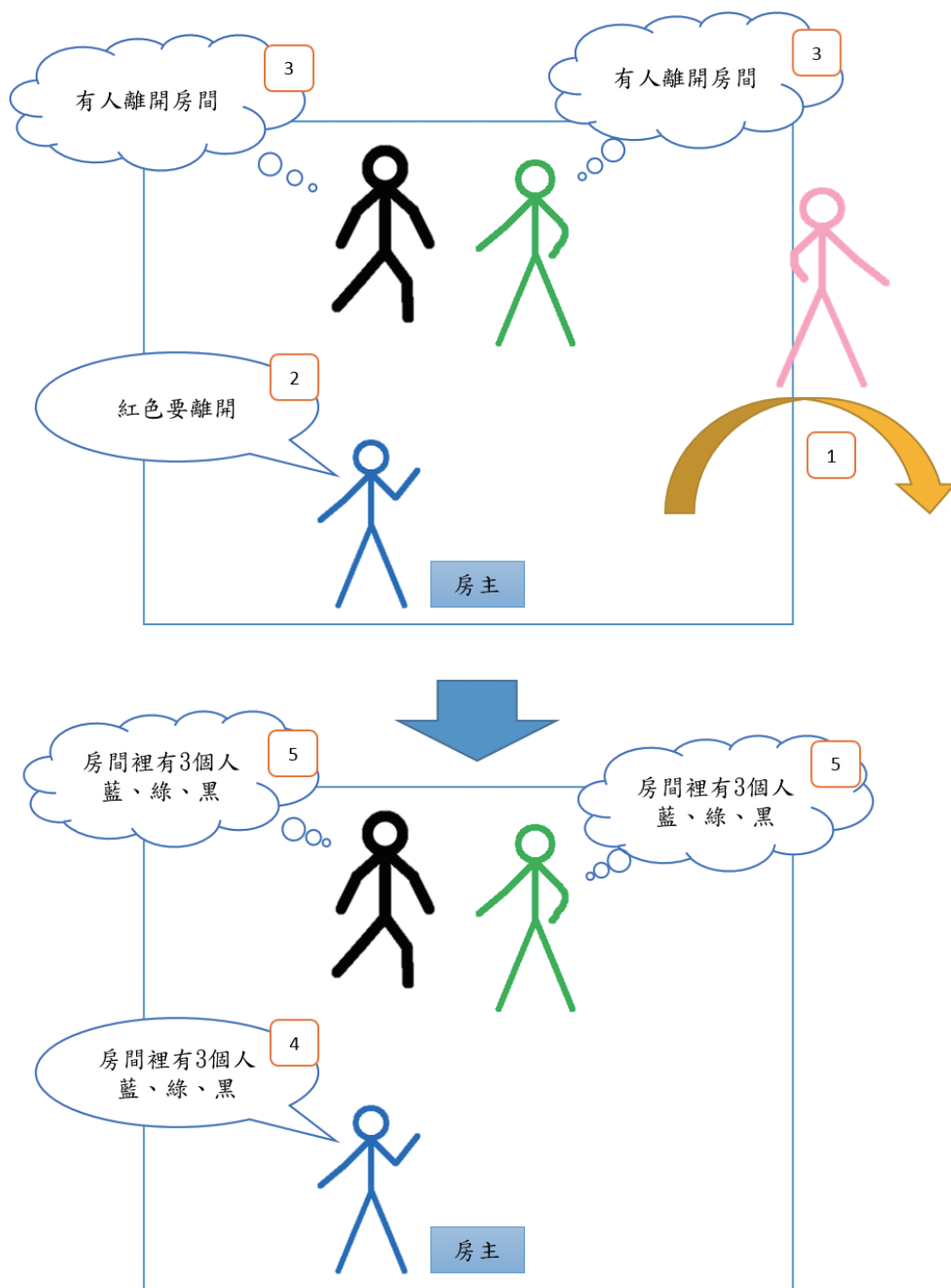


圖 H-9 玩家離開遊戲房間示意圖