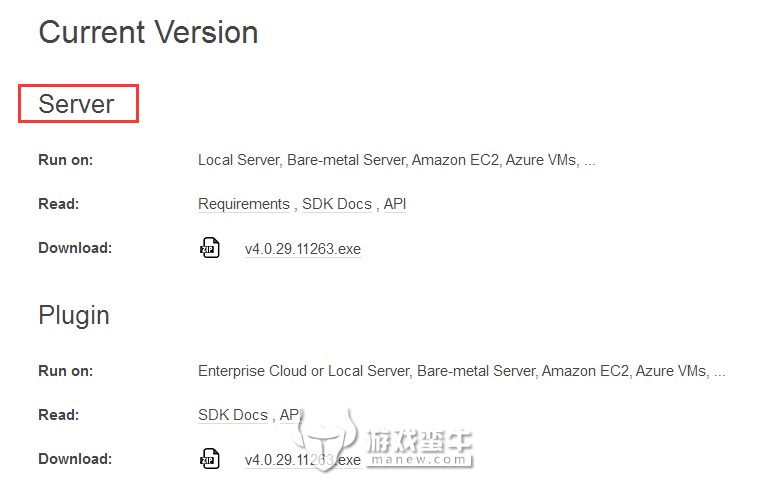
**Unity3D利用Photon实现实时联网对战（一）Photon Sever介绍**

Photon 是一个实时多人游戏开发框架,它是快速、精益和灵活的。   
Photon服务器功能强大，能够满足一般的中小型网络游戏的需求，而且开发效率高，所以就称为了大多数开发者的首选。由于网上关于Photon联网的介绍内容比较少，现在小義就跟大家分享一下关于Photon的内容。

首先Photon的官网地址： https://www.photonengine.com/zh-CN/Photon

这是Photon Sever SDK下载地址 https://www.photonengine.com/zh-CN/OnPremise/Download   
Photon Sever SDK提供了一个集成的服务器，在开发联网游戏的时候，就不需要我们自己来架设服务器啦！下载第一个版本的Sever，就是普通的Photon Sever SDK，其可以运行在本地和云上等。



这是免费版本的PUN SDK下载地址： https://www.assetstore.unity3d.com/en/#!/content/1786

利用PUN开发联网游戏的优点有：   
（1）、不需要自己搭建服务器，运行在Photon Cloud上；   
（2）、可以支持多种Unity导出平台，比如：Web、PC、Android和iOS等；   
（3）、Photon所有的特性都是实时的。

那下面就来了解一下Photon Sever SDK，因为它是我们开发联网游戏的服务器，所以首先需要设置好服务器。   
将下载好的Photon Sever SDK解压，之后打开Photon-OnPremise-Server-SDK\_v4文件夹，其下有几个文件夹需要重点解释一下。

Photon Sever的目录结构

1、deploy目录 :这是Photon Sever的启动目录，这里需要重点介绍一下。   
   1.bin\_tools目录，其下存放的是Photon Sever的一些工具，比如7zip是压缩和解压的工具，因为不是重要内容这里就不展开介绍啦。

   2.bin\_Win32/bin\_Win64目录，这里面是Photon Sever的启动目录，如果本地PC的操作系统是64位的就打bin\_Win64下的PhotonControl可执行文件，如果本地PC的操作系统是32位的那就打开 bin\_Win32下的 PhotonControl可执行文件。

2、lib目录 存放的是一下库文件，比如说Photon3.dll文件等。

3、doc目录 存放的是一些关于Photon Sever的介绍和关于Photon Sever开发的API。

启动Photon Sever   
打开PhotonControl可执行文件，单机小图标，之后选择LoadBalancing(My Cloud)，启动应用程序即可。   
其他选项：   
1、Photon Control  查看当前Photon Sever的版本号。   
2、License Info  License文件信息，可以显示最多连接人数等。   
3、Get Your Free License 下载免费的License，使最大连接数到达100个，如果是默认的话，最大连接数只有20个。   
4、Game Sever IP Config 可以查看当前服务器的IP地址，也可以设置IP。   
5、Open Logs  查看打印日志信息，可以看到连接服务器是否成功等信息。   
6、Exit Photon Control 退出Photon Sever服务器。

**Unity3D利用Photon实现实时联网对战（二）PUN SDK介绍**

1、配置Photon Server服务器

（1）、配置Photon Server的服务器IP地址   
在IP地址的列表中找到本机的IP地址，确定即可，这里一定要设置成本机的IP地址，否则客户端会连接不上服务器，查找本机IP地址的方法这里就不介绍啦，自行度娘一下。

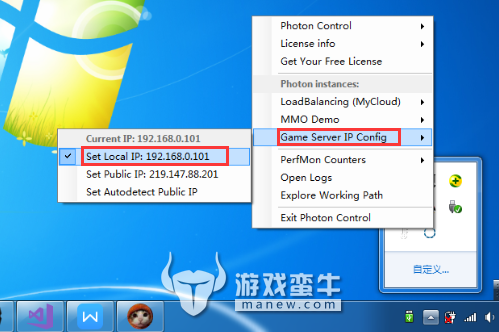


图 2-1 配置Photon Server服务器

（2）、启动Photon Server服务器

启动服务器，开启应用，等到Photon Server的图标颜色由灰变亮就说明服务器启动成功。

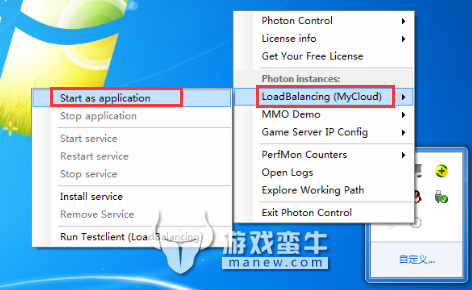
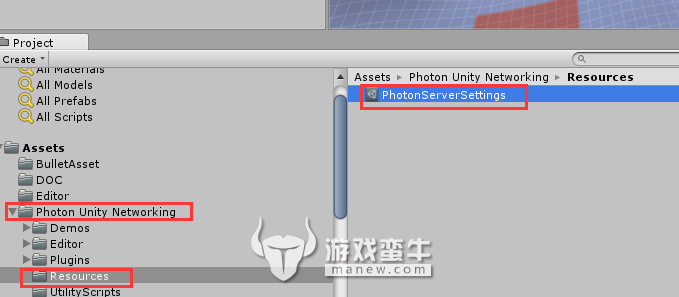


图 2-2 配置Photon Server服务器

2、配置客户端   
将PUN SDK导入Unity中后，找到Photon Unity Networking–>Resources下的PhotonServerSettings配置文件。如图所示：

   
（1）、Hosting   
设置服务器的类型，其中有Not Set（不设置）、Photon Cloud（Photon云服务器）、Self Hosted（自己作为主机）、Offline Mode（离线模式）和Best Region（最佳区域）。   
在这里我们选择Self Hosted，就是将自己的机器作为服务器来使用。

（2）、Server Address   
设置服务器的IP地址。因为联网的游戏需要服务器做同步和更新，所以客户端必须连接服务器才可以，而通过IP地址就可以连接服务器。   
在Server Address项中， 将当做服务器机器的IP地址填入该项内，这里我们输入本机的IP地址。

（3）、Server Port   
设置服务器的端口号，虽然说上面我们通过IP地址连接了服务器，但是服务器上有好多应用，还必须让服务器知道是需要对接哪个应用程序，这里对接不同的应用程序是通过端口号来实现的。

一般设置该端口号从范围从0到65535，而端口号又分为知名端口号和动态端口号。

知名端口顾名思义就是大众所熟知的端口号，范围从0到1023，这些端口号一般固定分配给一些服务。比如21端口分配给FTP服务，25端口分配给SMTP(简单邮件传输协议)服务，80端口分配给HTTP服务，135端口分配给RPC(远程过程调用)服务等等。

动态端口的范围从1024到65535，这些端口号一般不固定分配给某个服务，也就是说许多服务都可以使用这些端口。只要运行的程序向系统提出访问网络的申请，那么系统就可以从这些端口号中分配一个供该程序使用。比如1024端口就是分配给第一个向系统发出申请的程序。在关闭程序进程后，就会释放所占用的端口号。

所以这里我们设置端口号为5055。

（4）、Protocol   
设置网络协议，主要分为两类UDP和TCP，其中TCP是可靠的网络传输，其需要的带宽大。而UDP是不可靠的网络传输，其需要的带宽较小。关于其他的差异，大家可以在网上查找其他资料。   
我们在这里选择TCP协议。

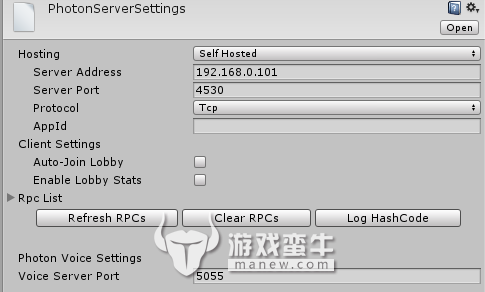


图 2-4 配置客户端

到这里客户端的配置就完成了，那就让我们打开一个Demo运行一下吧。   
下面是运行的截图

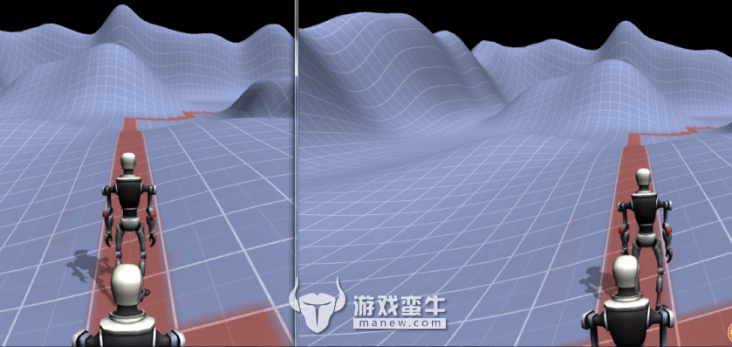


图 2-5 运行结果

**Unity3D利用Photon实现实时联网对战（三）创建和加入房间**  
Photon创建局域网的原理是通过一台机器创建房间，这台机器就是担负服务器的功能，负责与各个客户端之间的通信，而其也是一台客户端。   
之后其他机器加入房间，就是客户端。

直接上代码：

public class RoomController : MonoBehaviour{

private static byte Version = 0;

void Awake(){

//定义房间中的所有客户端是否应与主客户机保持相同的级别，一样的加载水平，当更新或加入时，所有客户端都会加载新场景。

PhotonNetwork.automaticallySyncScene = true;

//如果状态是创建并可以连接

if (PhotonNetwork.connectionStateDetailed == ClientState.PeerCreated)

//连接设置，设置客户端的版本号

PhotonNetwork.ConnectUsingSettings(Version + "." + SceneManagerHelper.ActiveSceneBuildIndex);

}

void Start() {

StartCoroutine(CreateOrJoinRoom());

}

/// 创建或者加入房间的协程

IEnumerator CreateOrJoinRoom()

{

//如果运行工程，直接创建或者加入房间的话，需要等待几秒先创建房间，否则会报错。

yield return new WaitForSeconds(0.5f);

Debug.Log("CreateOrJoinRoom");

if (!PhotonNetwork.inRoom)

PhotonNetwork.JoinOrCreateRoom("RoomOne", new RoomOptions { MaxPlayers = 10 }, TypedLobby.Default);

}

/// 加入房间后执行的方法，一般用来检测是否创建或者加载场景

public void OnJoinedRoom() {

Debug.Log("join the room");

}

public void OnFailedToConnectToPhoton(DisconnectCause cause) {

Debug.LogError("Cause: " + cause);

}

/// /退出程序，则退出房间

public void OnApplicationQuit() {

PhotonNetwork.LeaveRoom();

}

}

在这里介绍几个常用的方法：

1、CreateRoom   
其功能就是创建房间，执行该代码的机器即担负服务器也担负客户端的功能。   
CreateRoom(“RoomOne”, new RoomOptions { MaxPlayers = 10 }, null);其参数中第一个参数是房间名称；第二个参数是创建房间的选择，这里MaxPlayers 设置的为该房间的最大连接数；第三个参数为大厅类型，在这里不做设置。这样就可以创建一个房间啦。

但是在创建房间之前最好判断一下该程序是否已经在房间内，这里就说到Photon的一个缺点啦，如果在退出程序时不设置离开房间，等到再次创建房间时可能保留了上次创建的房间，这样房间名字是重复的，就会发生错误。这是为了避免这个因素，所以在创建房间之前做判断。   
所以上面的代码也在OnApplicationQuit方法中调用离开房间的方法，即退出该程序时，也退出该房间。

2、JoinOrCreateRoom   
其功能就是加入或者创建房间，如果之前已经创建了一个房间（创建的房间和加入的房间必须房间名称一致），就直接加入该房间即可；如果之前没有创建该房间，则直接创建房间。其参数和CreateRoom方法的相同。

3、JoinRoom   
其功能是加入房间，如果之前已经创建了一个房间（创建的房间和加入的房间必须房间名称一致），就直接加入房间即可；如果之前没有创建该房间，则会报错。   
JoinRoom(“RoomOne”)，其参数为房间名。

4、JoinRandomRoom   
其功能是加入一个随机的房间，如果之前已经创建了一个或多个房间，其随机加入一个房间；如果之前没有创建该房间，则会报错。   
该方法没有参数。

这就是一些常用的创建和加入房间的方法。

下面介绍几个常用的回调方法：

1、OnCreatedRoom   
在创建房间之后自动调用该方法，在该方法中一般做人物角色的生成和初始化以及其他一些初始化等。

2、OnJoinedRoom   
在加入房间之后自动调用该方法，在该方法中一般做人物角色的生成和初始化以及其他一些初始化等。

3、OnFailedToConnectToPhoton   
如果连接Photon失败，则会调用该方法。

上面的方法都是在Photon中用到的频率比较高的，那下一篇就介绍如何生成角色。