**Unity3D教程：Photon服务器入门（一）**

Posted on 2013年07月11日 by U3d / [Unity3D 基础教程](http://www.unitymanual.com/category/manual/unity3d-%e5%9f%ba%e7%a1%80%e6%95%99%e7%a8%8b) /被围观 26 次

首先去PhotonServer SDK下载服务器端SDK。解压出来是四个文件：

deploy：主要存放photon的服务器控制程序和服务端Demo

doc：顾名思义，文档

lib：Photon类库，开发服务端需要引用的

src-server：服务端Demo源代码

第一步：配置服务器端

打开deploy文件夹，看到包含了不同平台编译出的Photon目录，以“bin\_”为前缀命名目录，选择你的电脑对应的文件夹打开，看到PhotonControl.exe，运行后，可以在windows右下角看到一个图标，点击图标可以看到photon服务器控制菜单，这个以后再做详细介绍。

打开visual stadio，新建项目，选择c# 类库，应用程序名字叫做MyServer。

完成后，把我们的Class1.cs，改名为MyApplication.cs，作为服务器端主类.然后在当前项目添加引用，链接到刚才提到的lib文件夹目录下，添加以下引用：

ExitGamesLibs.dll,

Photon.SocketServer.dll,

PhotonHostRuntimeInterfaces.dll

然后新建一个类：MyPeer.cs，写法如下：

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
| 01 | **using** System; |
| 02 | **using** System.Collections.Generic; |
| 03 | **using** Photon.SocketServer; |
| 04 | **using** PhotonHostRuntimeInterfaces; |
| 05 | **namespace** MyServer |
| 06 | { |
| 07 | **public** **class** MyPeer : PeerBase |
| 08 | { |
| 09 | **public** MyPeer(IRpcProtocol protocol,IPhotonPeer photonPeer) |
| 10 | : **base**(protocol, photonPeer) |
| 11 | { |
| 12 | } |
| 13 | **protected** **override** **void** OnDisconnect(PhotonHostRuntimeInterfaces.DisconnectReason reasonCode, **string** reasonDetail) |
| 14 | { |
| 15 | } |
| 16 | **protected** **override** **void** OnOperationRequest(OperationRequest operationRequest, SendParameters sendParameters) |
| 17 | { |
| 18 | } |
| 19 | } |
| 20 | } |

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |
| --- | --- |
| 01 | **using** System; |
| 02 | **using** System.Collections.Generic; |
| 03 | **using** Photon.SocketServer; |
| 04 | **using** PhotonHostRuntimeInterfaces; |
| 05 |  |
| 06 | **namespace** MyServer |
| 07 | { |
| 08 | **public** **class** MyPeer : PeerBase |
| 09 | { |
| 10 |  |
| 11 | **public** MyPeer(IRpcProtocol protocol,IPhotonPeer photonPeer) |
| 12 | : **base**(protocol, photonPeer) |
| 13 | { |
| 14 |  |
| 15 | } |
| 16 |  |
| 17 | **protected** **override** **void** OnDisconnect(PhotonHostRuntimeInterfaces.DisconnectReason reasonCode, **string** reasonDetail) |
| 18 | { |
| 19 |  |
| 20 | } |
| 21 | *//Unity3D教程手册：www.unitymanual.com* |
| 22 | **protected** **override** **void** OnOperationRequest(OperationRequest operationRequest, SendParameters sendParameters) |
| 23 | { |
| 24 | } |
| 25 | } |
| 26 | } |

接上，MyApplication.cs类这样写：

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 01 | | | **using** System; | |
| 02 | | | **using** System.Collections.Generic; | |
| 03 | | | **using** System.Linq; | |
| 04 | | | **using** System.Text; | |
| 05 | | | **using** Photon.SocketServer; | |
| 06 | | | **namespace** MyServer | |
| 07 | | | { | |
| 08 | | | **public** **class** MyApplication : ApplicationBase | |
| 09 | | | { | |
| 10 | | | **protected** **override** PeerBase CreatePeer(InitRequest initRequest) | |
| 11 | | | { | |
| 12 | | | **return** new MyPeer(initRequest.Protocol, initRequest.PhotonPeer); | |
| 13 | | | } | |
| 14 | | | **protected** **override** **void** Setup() | |
| 15 | | | { | |
| 16 | | | } | |
| 17 | | | **protected** **override** **void** TearDown() | |
| 18 | | | { | |
| 19 | | | } | |
| 20 | | | } | |
| 21 | | | } | |
|  |  | |  |

|  |  |
| --- | --- |
| 01 | **using** System; |
| 02 | **using** System.Collections.Generic; |
| 03 | **using** System.Linq; |
| 04 | **using** System.Text; |
| 05 | **using** Photon.SocketServer; |
| 06 |  |
| 07 | **namespace** MyServer |
| 08 | { |
| 09 | **public** **class** MyApplication : ApplicationBase |
| 10 | { |
| 11 |  |
| 12 | **protected** **override** PeerBase CreatePeer(InitRequest initRequest) |
| 13 | { |
| 14 | **return** new MyPeer(initRequest.Protocol, initRequest.PhotonPeer); |
| 15 | } |
| 16 |  |
| 17 | **protected** **override** **void** Setup() |
| 18 | { |
| 19 |  |
| 20 | } |
| 21 |  |
| 22 | **protected** **override** **void** TearDown() |
| 23 | { |
| 24 |  |
| 25 | } |
| 26 | } |
| 27 | } |

完成后，在解决方案资源管理器中选中当前项目，打开属性，选择生成选项卡，把输出路径改成bin\,然后就生成类库吧

复制当前项目下MyServer文件夹到deploy文件夹下，删除除了bin文件夹以外其他所有文件和文件夹，然后文本编辑器打开deploy\bin\_Win64\PhotonServer.config配置文件(我的是win7 64位机器，就选择这个),添加以下配置：

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | | <Application | | |
| 2 | | Name="MyServer" | | |
| 3 | | BaseDirectory="MyServer" | | |
| 4 | | Assembly="MyServer" | | |
| 5 | | Type="MyServer.MyApplication" | | |
| 6 | | EnableAutoRestart="true" | | |
| 7 | | WatchFiles="dll;config" | | |
| 8 | | ExcludeFiles="log4net.config"> | | |
| 9 | | </Application> | | |
|  | |  |  |

|  |  |
| --- | --- |
| 1 | <Application |
| 2 | Name="MyServer" |
| 3 | BaseDirectory="MyServer" |
| 4 | Assembly="MyServer" |
| 5 | Type="MyServer.MyApplication" |
| 6 | EnableAutoRestart="true" |
| 7 | WatchFiles="dll;config" |
| 8 | ExcludeFiles="log4net.config"> |
| 9 | </Application> |

这段代码放在放这里节点下面

Name：项目名字

BaseDirectory：根目录，deploy文件夹下为基础目录

Assembly：是在生成的类库中的bin目录下与我们项目名称相同的.dll文件的名字

Type：是主类的全称，在这里是：MyServer.MyApplication，一定要包括命名空间

EnableAutoRestart：是否是自动启动，表示当我们替换服务器文件时候，不用停止服务器，替换后photon会自动加载文件

WatchFiles和ExcludeFiles

完成后保存，运行托盘程序deploy\bin\_Win64\PhotonControl.exe，

（证书放在deploy\bin\_Win64\目录下）

下面开始编写客户端代码，首先从官网下载Unity SDK

打开Unity3D编辑器，首先把Photon-Unity3D\_v3-0-1-14\_SDK\libs\Release\Photon3Unity3D.dll导入到Unity中，新建脚本TestConnection.cs,脚本代码如下：

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 01 | | | **using** UnityEngine; | |
| 02 | | | **using** System.Collections; | |
| 03 | | | **using** ExitGames.Client.Photon; | |
| 04 | | | **public** **class** TestConnection : MonoBehaviour,IPhotonPeerListener { | |
| 05 | | | **public** PhotonPeer peer; | |
| 06 | | | *// Use this for initialization* | |
| 07 | | | **void** Start () { | |
| 08 | | | peer = new PhotonPeer(**this**,ConnectionProtocol.Udp); | |
| 09 | | | } | |
| 10 | | | *// Update is called once per frame* | |
| 11 | | | **void** Update () { | |
| 12 | | | peer.Service(); | |
| 13 | | | } | |
| 14 | | | **void** OnGUI(){ | |
| 15 | | | **if**(GUI.Button(new Rect(Screen.width/2,Screen.height/2,200,100),"Connect")){ | |
| 16 | | | peer.Connect("localhost:5055","MyServer"); | |
| 17 | | | }*//Unity3D教程手册：www.unitymanual.com* | |
| 18 | | | } | |
| 19 | | | #region IPhotonPeerListener implementation | |
| 20 | | | **public** **void** DebugReturn (DebugLevel level, **string** message) | |
| 21 | | | { | |
| 22 | | | } | |
| 23 | | | **public** **void** OnOperationResponse (OperationResponse operationResponse) | |
| 24 | | | { | |
| 25 | | | } | |
| 26 | | | **public** **void** OnStatusChanged (StatusCode statusCode) | |
| 27 | | | {*//Unity3D教程手册：www.unitymanual.com* | |
| 28 | | | **switch**(statusCode){ | |
| 29 | | | **case** StatusCode.Connect: | |
| 30 | | | Debug.Log("Connect Success!"); | |
| 31 | | | **break**; | |
| 32 | | | **case** StatusCode.Disconnect: | |
| 33 | | | Debug.Log("Disconnect!"); | |
| 34 | | | **break**; | |
| 35 | | | } | |
| 36 | | | } | |
| 37 | | | **public** **void** OnEvent (EventData eventData) | |
| 38 | | | { | |
| 39 | | | } | |
| 40 | | | #endregion | |
| 41 | | | } | |
|  |  | |  |

|  |  |
| --- | --- |
| 01 | **using** UnityEngine; |
| 02 | **using** System.Collections; |
| 03 |  |
| 04 | **using** ExitGames.Client.Photon; |
| 05 |  |
| 06 | **public** **class** TestConnection : MonoBehaviour,IPhotonPeerListener { |
| 07 | **public** PhotonPeer peer; |
| 08 | *// Use this for initialization* |
| 09 | **void** Start () { |
| 10 | peer = new PhotonPeer(**this**,ConnectionProtocol.Udp); |
| 11 | } |
| 12 |  |
| 13 | *// Update is called once per frame* |
| 14 | **void** Update () { |
| 15 | peer.Service(); |
| 16 | } |
| 17 |  |
| 18 | **void** OnGUI(){ |
| 19 | **if**(GUI.Button(new Rect(Screen.width/2,Screen.height/2,200,100),"Connect")){ |
| 20 | peer.Connect("localhost:5055","MyServer"); |
| 21 | } |
| 22 | } |
| 23 |  |
| 24 | #region IPhotonPeerListener implementation |
| 25 | **public** **void** DebugReturn (DebugLevel level, **string** message) |
| 26 | { |
| 27 |  |
| 28 | } |
| 29 |  |
| 30 | **public** **void** OnOperationResponse (OperationResponse operationResponse) |
| 31 | { |
| 32 |  |
| 33 | } |
| 34 | *//Unity3D教程手册：www.unitymanual.com* |
| 35 | **public** **void** OnStatusChanged (StatusCode statusCode) |
| 36 | { |
| 37 | **switch**(statusCode){ |
| 38 | **case** StatusCode.Connect: |
| 39 | Debug.Log("Connect Success!"); |
| 40 | **break**; |
| 41 | **case** StatusCode.Disconnect: |
| 42 | Debug.Log("Disconnect!"); |
| 43 | **break**; |
| 44 | } |
| 45 | } |
| 46 |  |
| 47 | **public** **void** OnEvent (EventData eventData) |
| 48 | { |
| 49 |  |
| 50 | } |
| 51 | #endregion |
| 52 | } |

把脚本绑定到场景中物体上，运行后可以看到一个按钮，点击连接，如果连接成功会打印“Connect Success!”