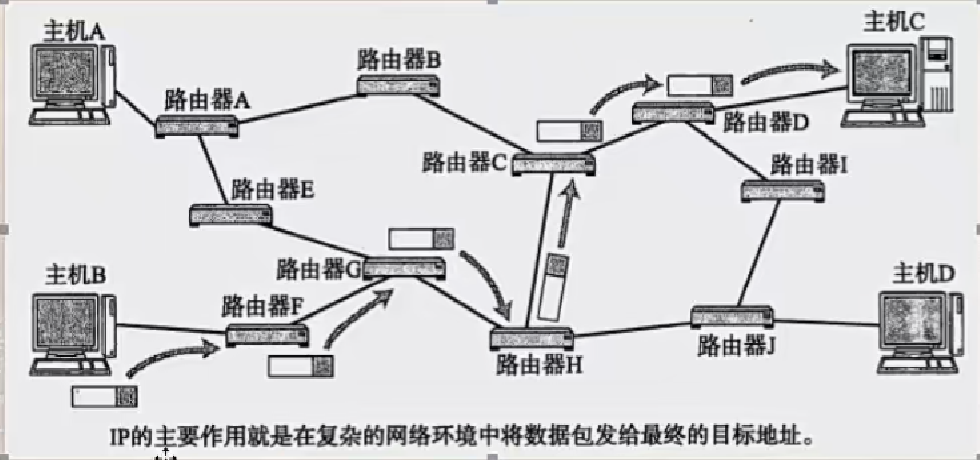
TCP/IP

IP的主要作用就是在复杂的网络环境中将数据包发给最终的目标地址

是一台电脑在网络上的位置，可以通过IP找到电脑

局域网IP win+R 输入cmd 输入ipconfig 查找

公网IP

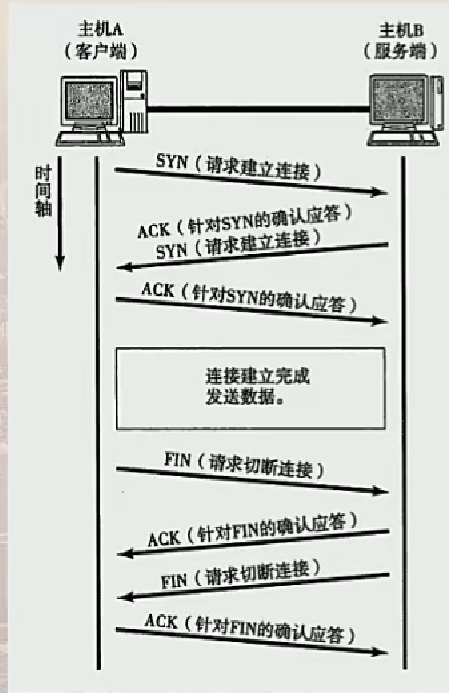


端口号

知名端口号0~1023

自定义/动态分配的端口号一般在49152~65535之间

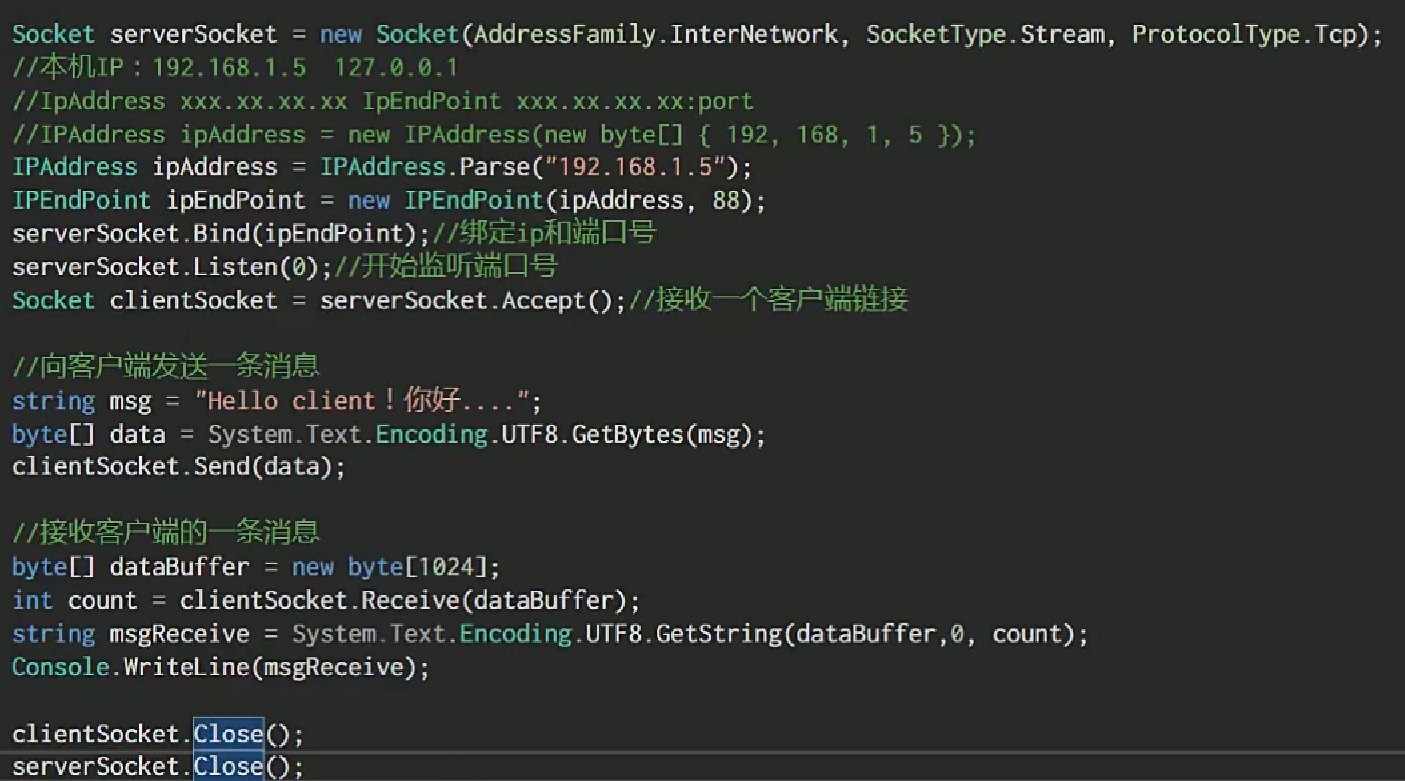
TCP三次握手



TCP传输信息稳定，不会丢失

UDP传输信息快，不稳定

**TCP服务**端



Bind 绑定IP和端口号

Listen 开始监听端口号

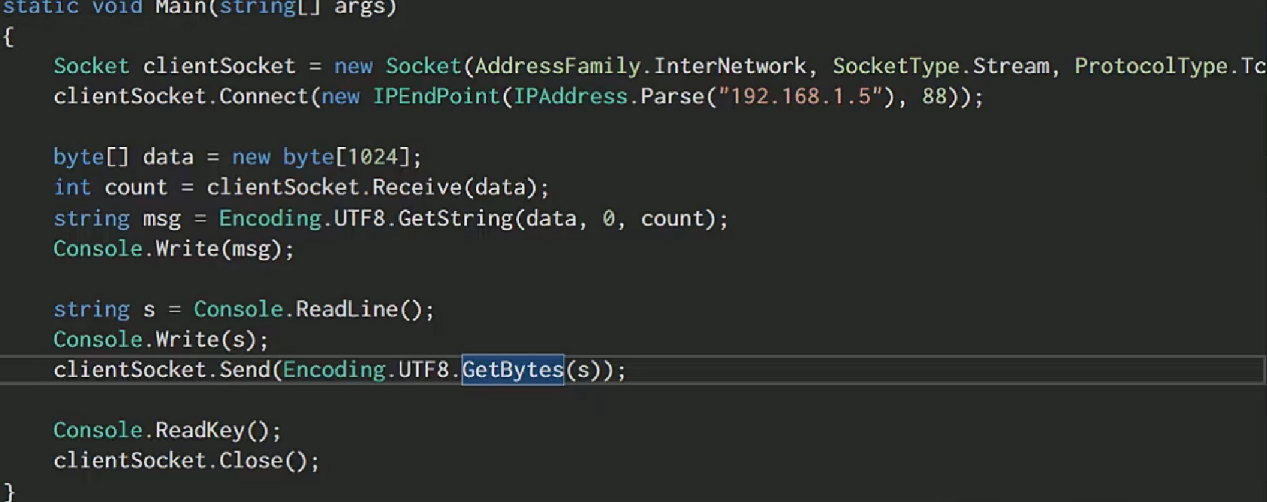
Accept 接受客户端链接

Send 发送消息

Receive 接收消息

Close 关闭连接

**07 TCP客户端**



Connect 建立连接

Receive 接收消息

Send 发送信息

Close 关闭

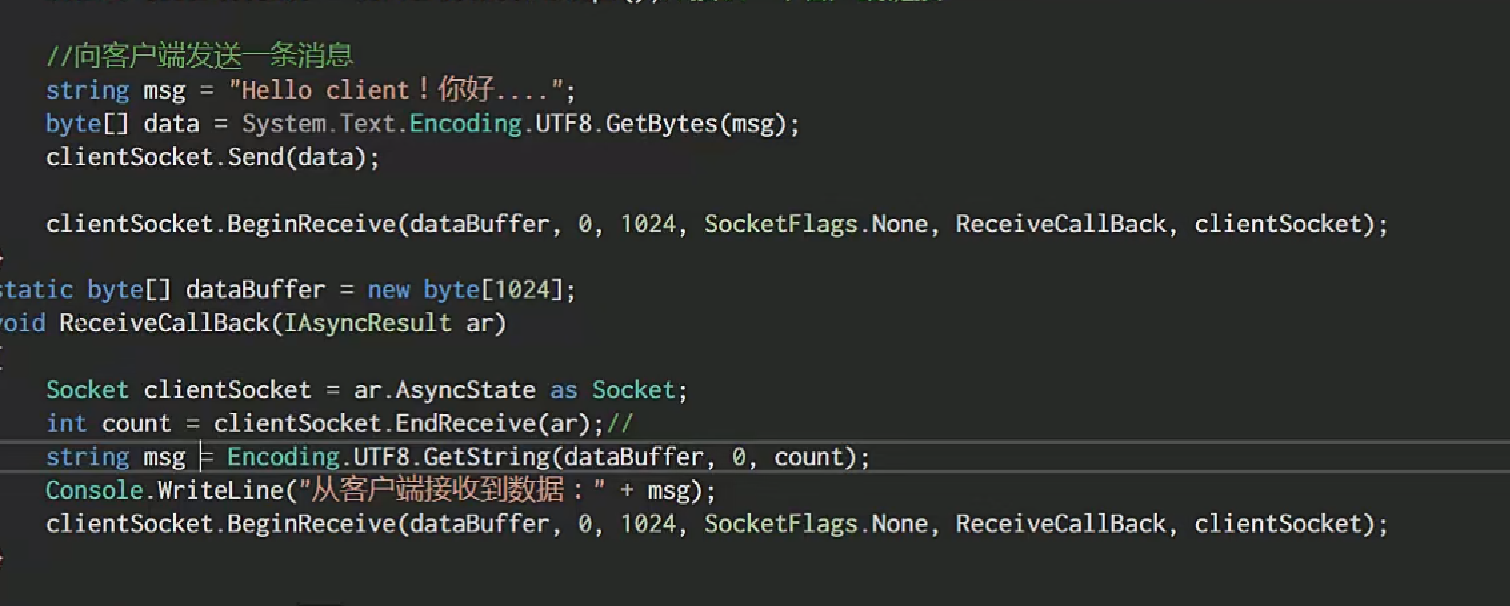
先接受，后发送

**08 实现服务器异步的消息接收**

BeginReceive 异步接收

EndReceive 结束接受并返回接受的数据

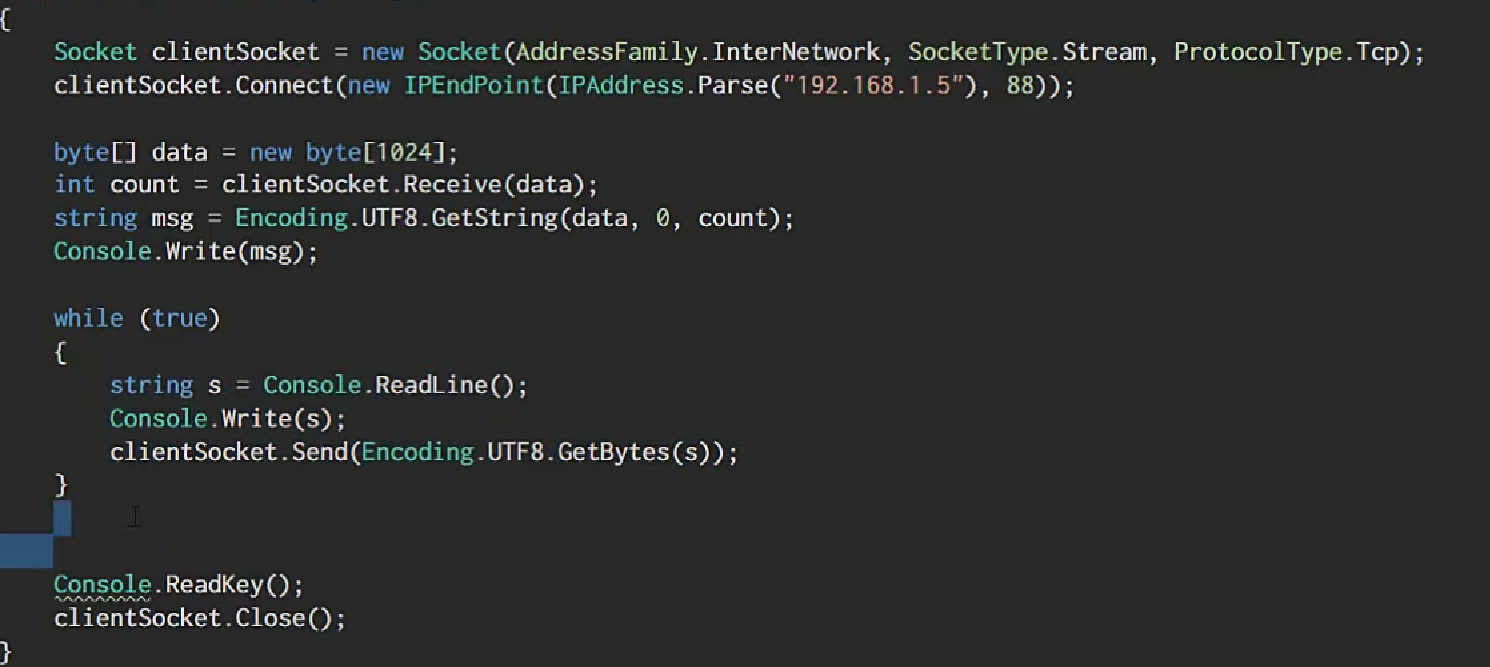
服务器



通过BeginReceive完成回调会进行一直接收数据，输入控制台

实现数据循环的接受与发送

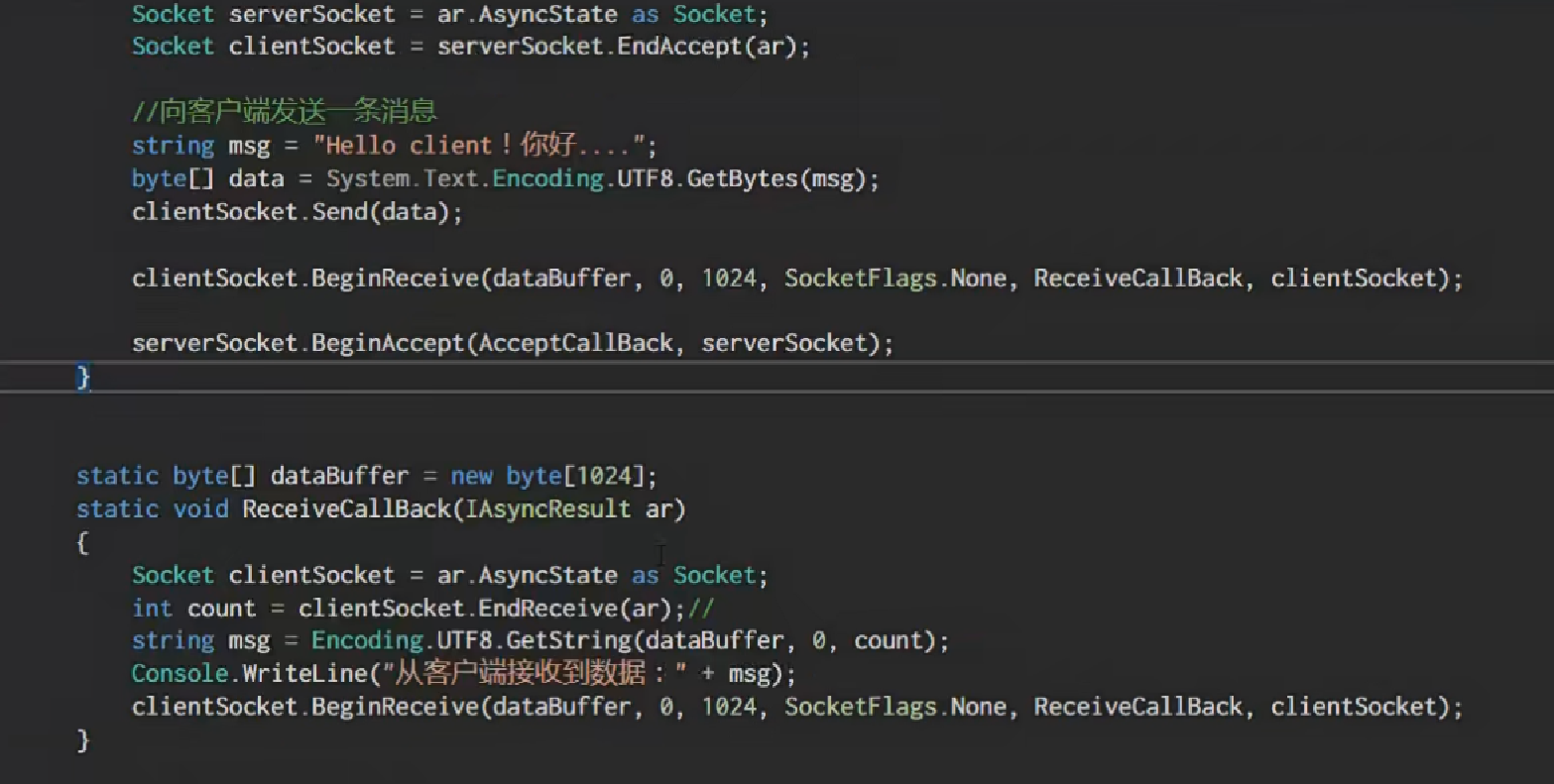
客户端



**09 修改服务器端开始异步处理客户端链接请求**

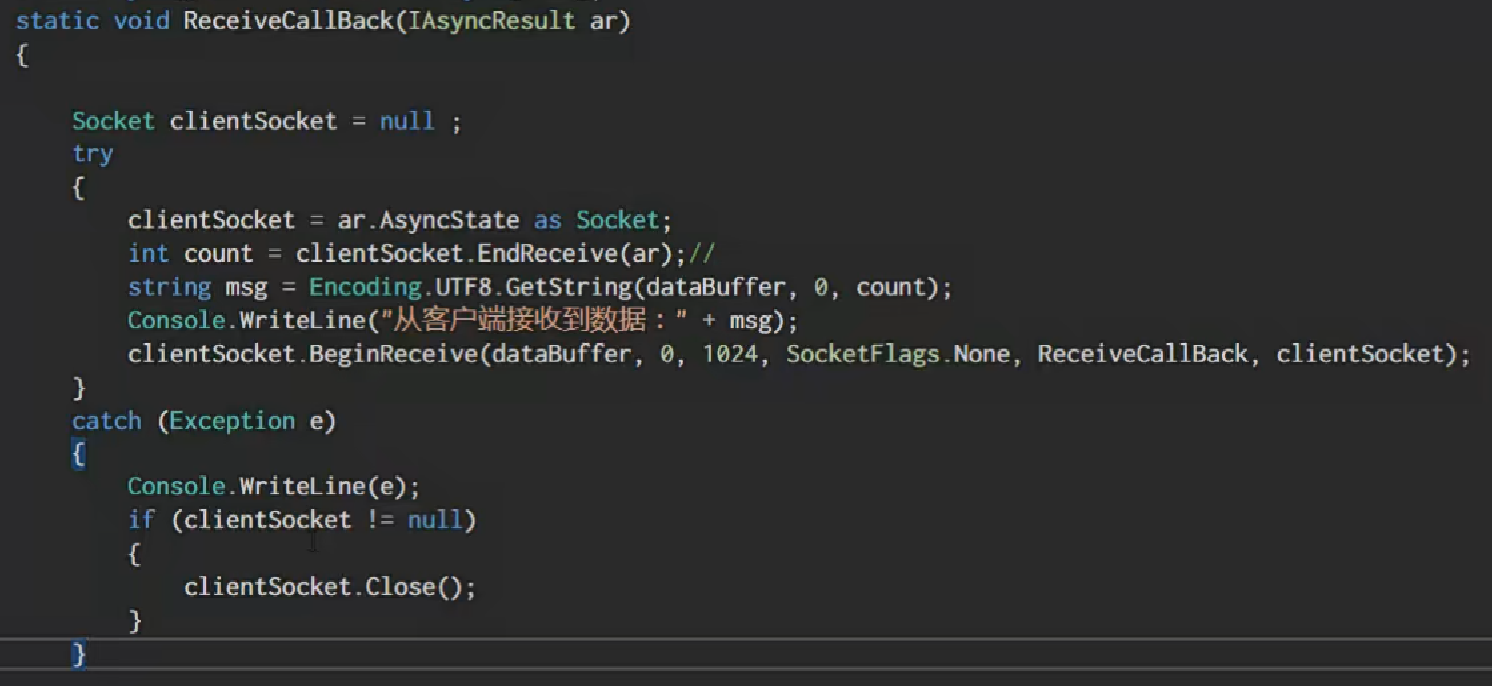
Accept 同步连接客户端，只会在链接到时才会进行下一步操作

BeginAccept 开始异步的连接

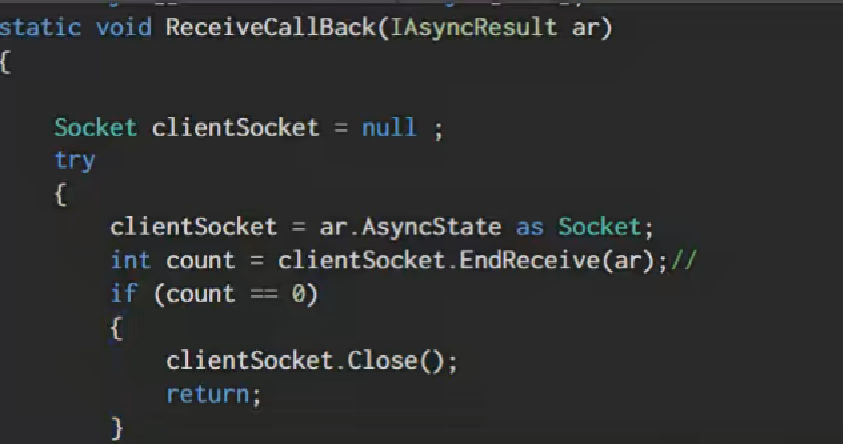
 serverSocket.BeginAccept(AcceptCallBack,serverSocket);重新调用，继续处理下一个客户端的连接

**10在服务器端处理客户端的连接正常关闭和非正常关闭**

非正常关闭处理



正常关闭处理



**11粘包分包**

包：数据

粘包：在发送信息时，信息量小，速度频繁的情况下，把数据合并成较大的数据包会被粘在一起

分包：在接收数据数据很大时，无法接受很大的完整的数据包，就会进行拆分

设置传输数据的范围

**13粘包分包问题的解决**

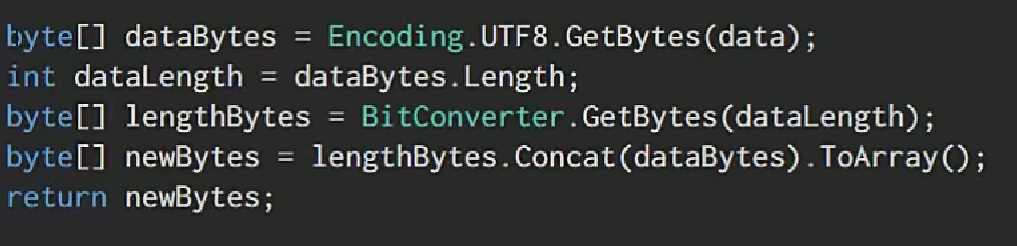
使用数据长度+数据的形式来进行传输，在接收信息的时候首先进行数据长度的判断，根据长度再去选择接收数据的长度，如若数据长度不够，则进行合并直至数据长度达到判断条件，最后进行接收。

**14字符串和值类型如何转化成字节数据**

Int32转成字节数组BitConerter.GetBytes，可以转换值类型



**15在客户端发送数据的时候加上数据长度**

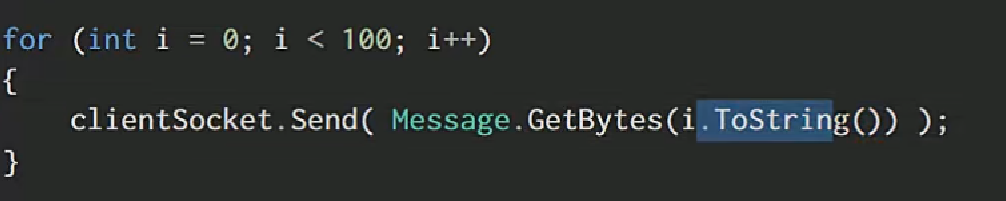


Encoding.UTF8.GetBytes()转化成数据数组

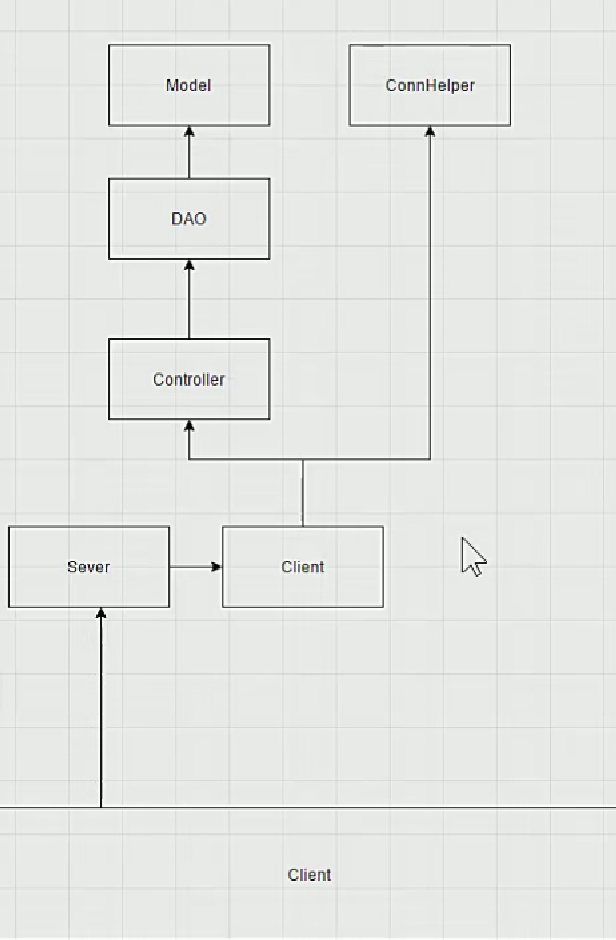
DataBytes.Length获取数据长度

BitConerter.GetBytes（dataLength）转化成Bytes的数组

lengthByrtes.Concat(dataBytes).ToArray()将两个数组进行连接并赋值给行的数组



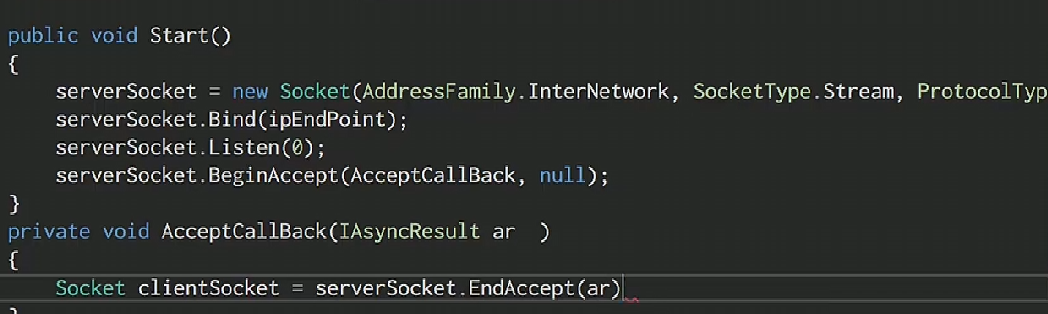
**23服务器端分层架构**



**25创建Server类，开始接受客户端连接**

开启服务器端需要指定IP和端口号，创建构造函数传递IP地址和端口号，启用互联网协议指定TCP协议

绑定IP和端口开启监听，用异步接收连接，异步回调，处理客户端连接，当接受到客户端连接就把clientSocket传递到Client中。再创建一个list管理所有连接的客户端，每次连接就添加到list中。调用client中的start方法。



**26创建Client类，处理跟客户端的数据通道**

需要clientSocket来进行客户端通信，将Server类中的clientSocket传递到Client中。

在start方法中建立数据接收方法，首先接收到数据的长度int值（判断数据长度是否为0，为0则调用close断开链接方法，将自身移除）

在关闭方法中，首先判断clientSocket是否为空，不为空则调用自身的close方法，在server中调用移除client的方法

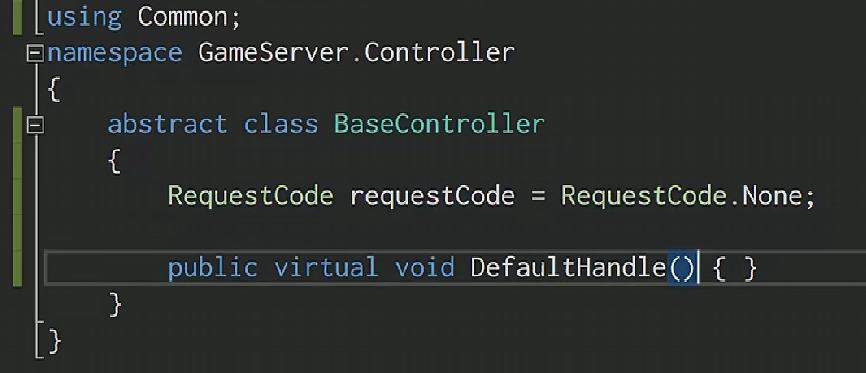
**27创建Message处理客户端消息的解析**

通过类去管理调用，在message中使用data进行接收，在client中创建新的message对象



**28开发Controller控制层**

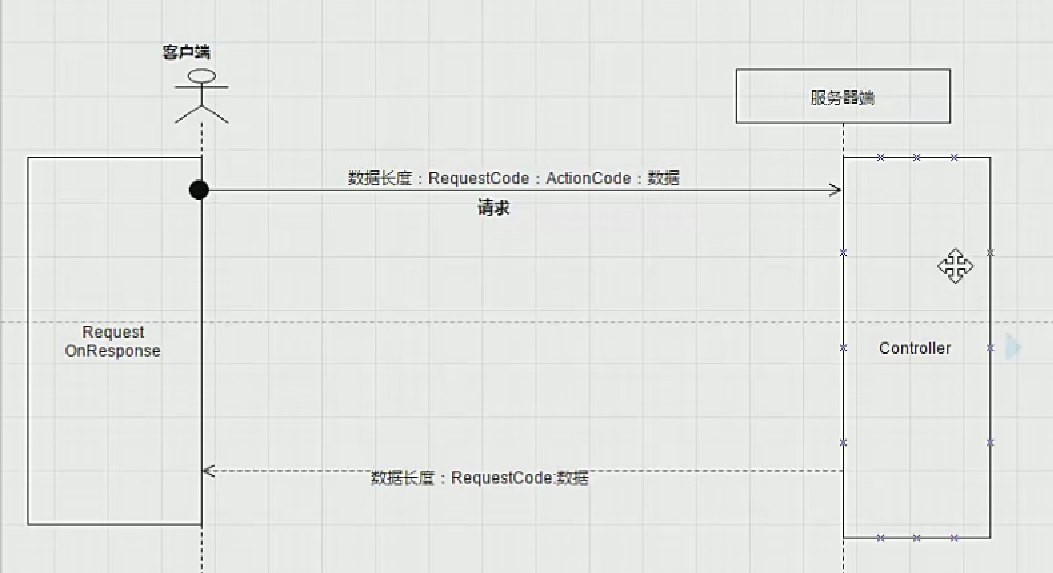
Client用Message来解析数据，用枚举来确定需要控制的数据。



**29客户端和服务器端的请求发起的流程图**

客户端向服务端发起请求的时候，数据组成有数据长度：RequestCode：ActionCode：数据，不同的客户端请求数据的组成也会不同，两个Code数据也都由Contorller进行处理，所以数据长度=数据长度+4+4。

服务器端给客户端响应的时候，传递的数据就变为数据长度：RequestCode:数据



**30创建ContorllerManager管理所有的控制器**

创建字典存储所有的Controller，定义初始化方法，实例化服务器端所有的controller存放到字典中，定义构造函数调用初始化方法。

在层集中，client接收到消息要对controller进行处理，需要通过Server进行controllerManager的调用。

在server类中初始化controllerManager

**31通过ControllerManager进行请求的分发处**

在游戏服务器端的BaseController 是用于处理请求的方法，根据客户端发送过来的数据来处理请求，在ControllerManager中通过HandleRequest方法来传递参数来调用BaseController方法

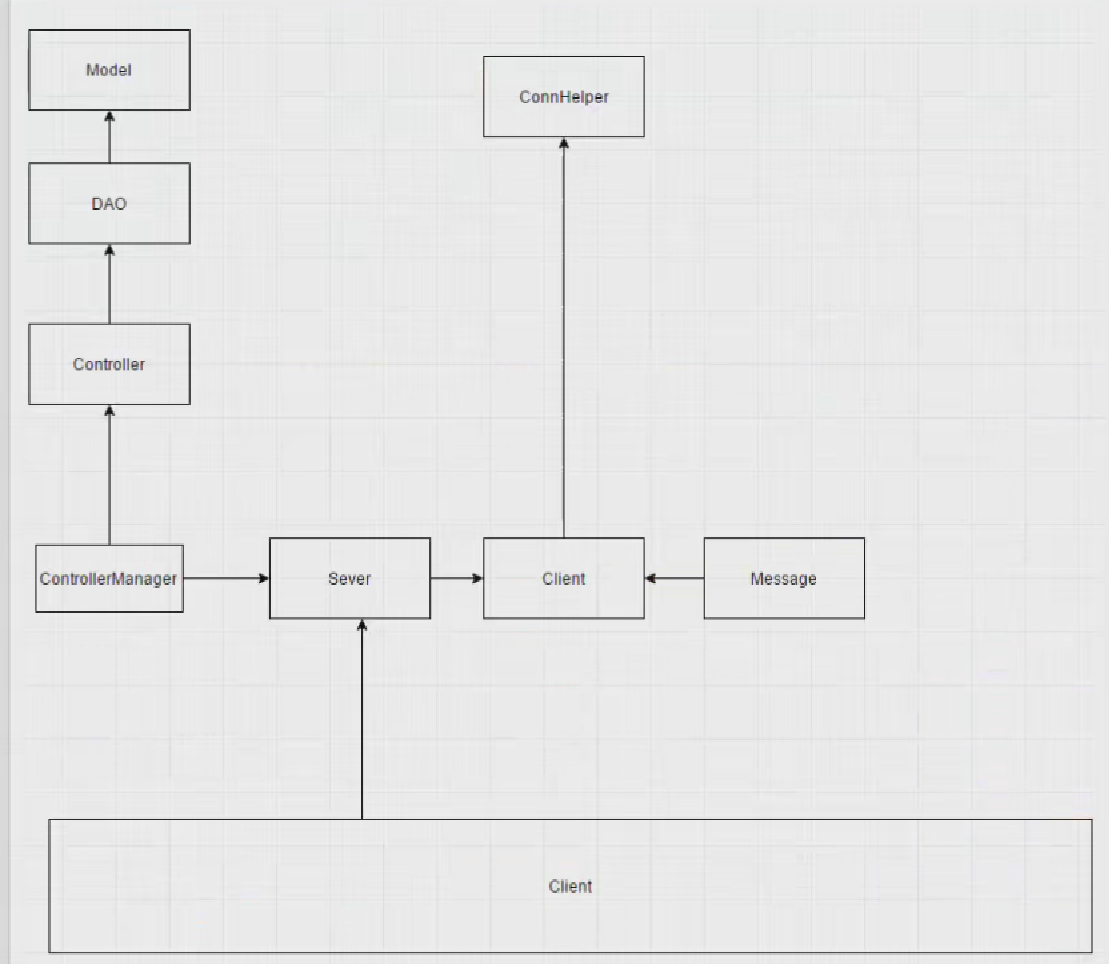
通过加入字典来返回bool值，查看是否返回成功，失败就进行输出报错日志然后return终止，成功的话就要处理请求，通过反射机制来调用方法名 ，使用GetType和GetMethod通过方法名来调用方法,判断得到的方法信息是否为空,输出对应的Type信息和Name。存在的话调用Invoke传入多个数据参数判断返回值是否个客户端响应。

**32客户端处响应的请求**

多传递参数，明确那个参数是调用客户端，哪个是传递请求还有对消息的广播，在传递信息的时候要传递三个参数。

判断参数是否为空，是的话终止返回，否的时候才需要传递给客户端。

在server端创建新的发送消息的方法，在传递信息的时候调用SendRespononse发送消息方法。



**33如何把客户端消息的解析传递给ControllerMa**

解析数据，判断数据长度是否大于4，小于四代表消息没有传递过来，大于四然后在获取后面的长度，则代表数据是完整的。

首先解析RequestCode，然后就是ActionCode，接着解析数据，对于数据的解析是从12字节后开始

在server中新加一个处理消息的方法，传递三个参数，把方法的调用传递给ControlManager

**34数据的打包和数据的发送到客户端**

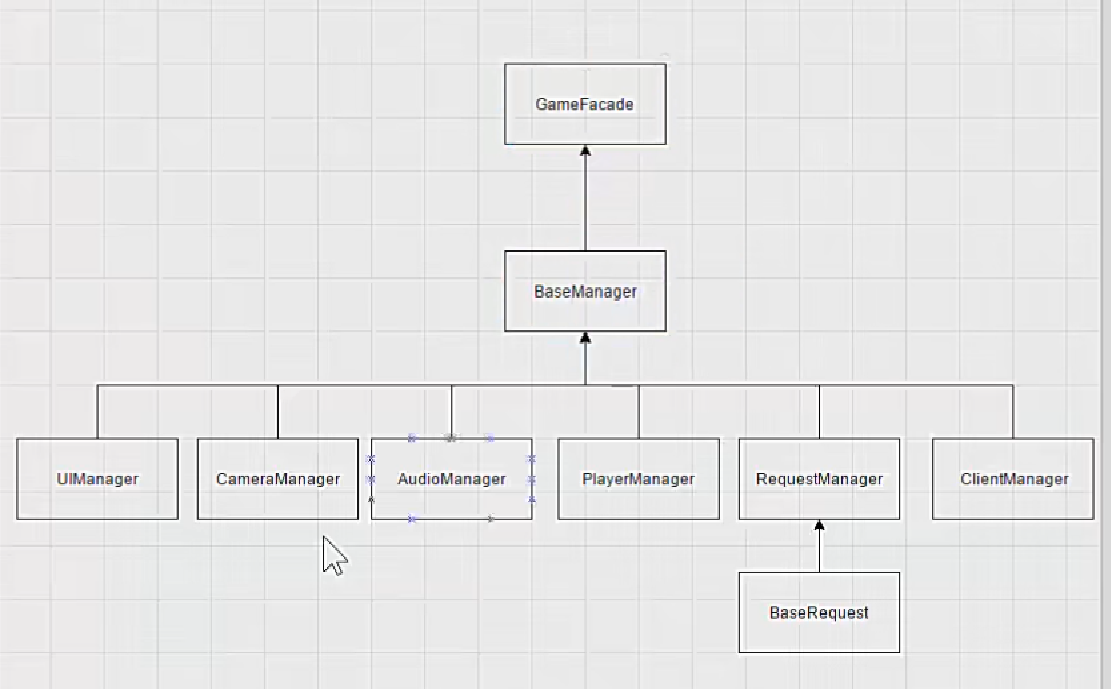
在Client端进行数据的包装和解析，使用静态方法完成包装的过程：

把数据转换成字节数组，数据长度也转换成数组，然后进行组装。发送给客户端，让客户端进行发送。

**35创建ConnHelper,数据库连接的创建和关闭**

创建连接字符串，创建连接方法通过连接字符串创建新的MySql，打开链接。创建关闭连接方法，调用close。在Client持有一个数据库的连接,在构造函数里进行连接，在关闭客户端的时候进行数据库的关闭。

**38游戏客户端架构分析**



UIManager：管理UI

CameraManager：控制摄像机跟随

AudioManager:控制声音的播放

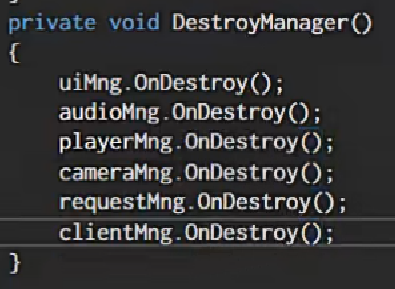
PlayManager:生成游戏角色

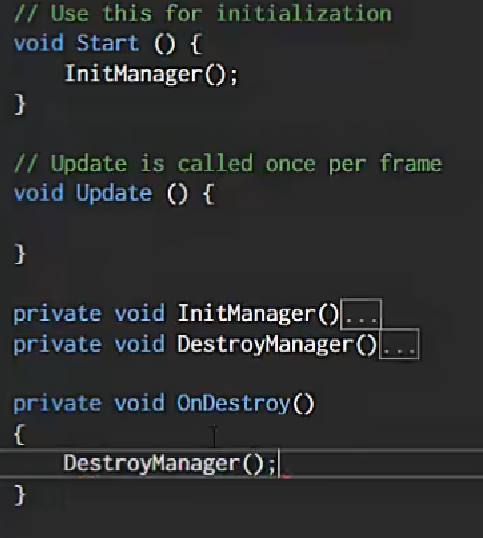
RequesManager:管理请求，向服务器端发起请求

ClientManager：与服务器端建立连接

**39创建客户端架构基础类**







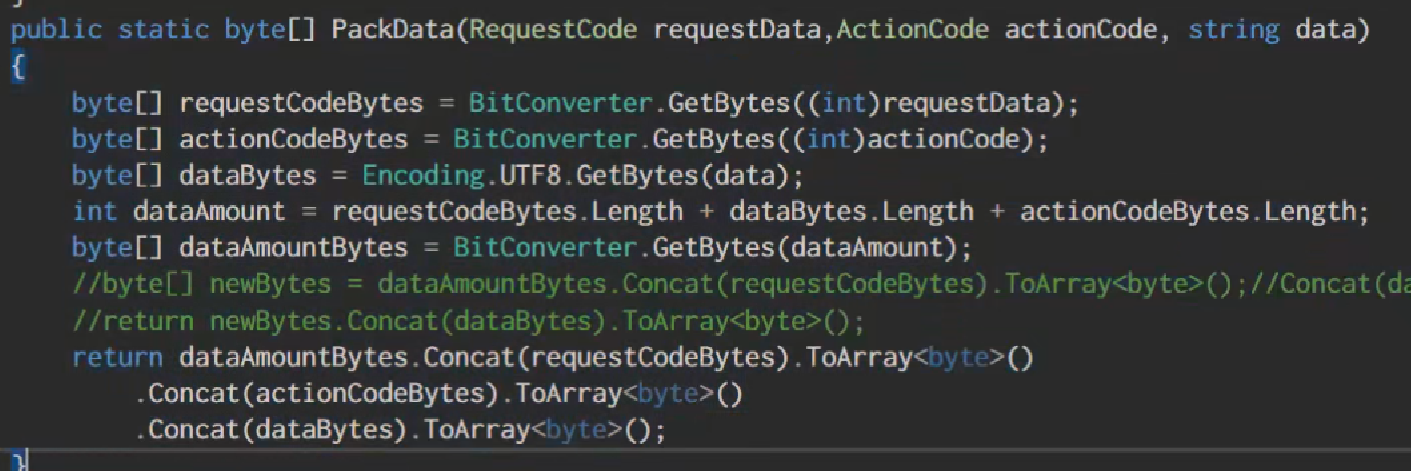
**40开发ClientManager,跟服务器端连接的建立**

建立IP和端口号，在Init方法里创建新的Socket用于接受IP和端口号

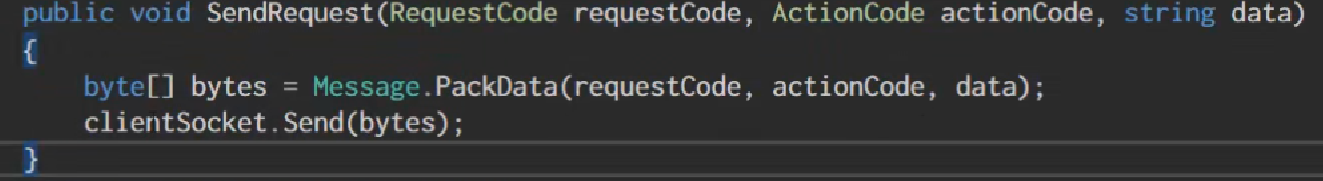
**42开发客户端向服务器端请求的发送功能**

在ClientManager提供客户端向服务器端发送请求，打包发送的数据

Message打包方法

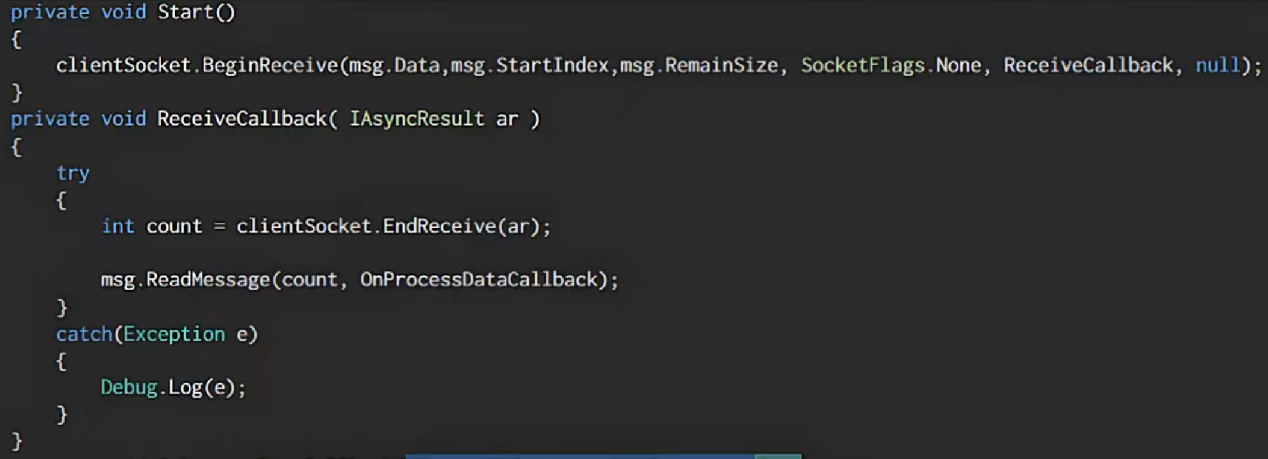


先打包数据，再进行发送

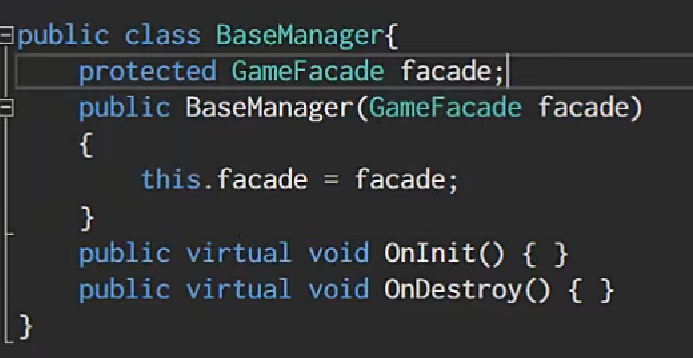


**43接受服务器的消息并解析**

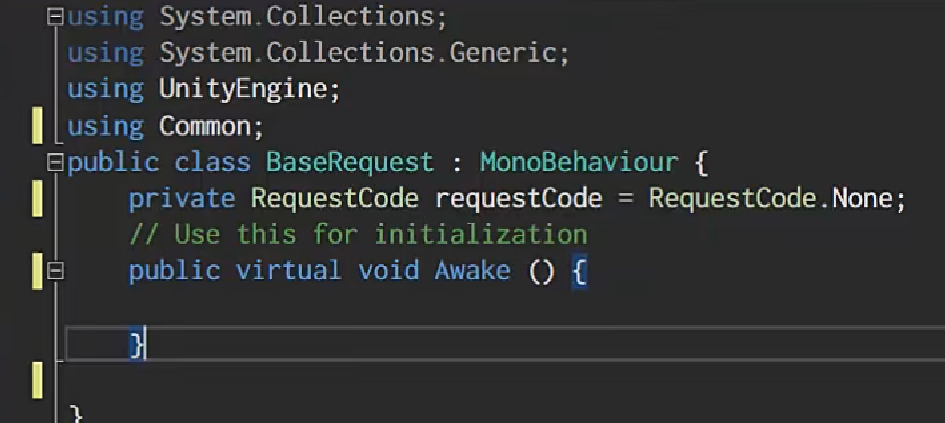
生成开始解析方法，先进行异步的消息接收



**44修改BaseManager**



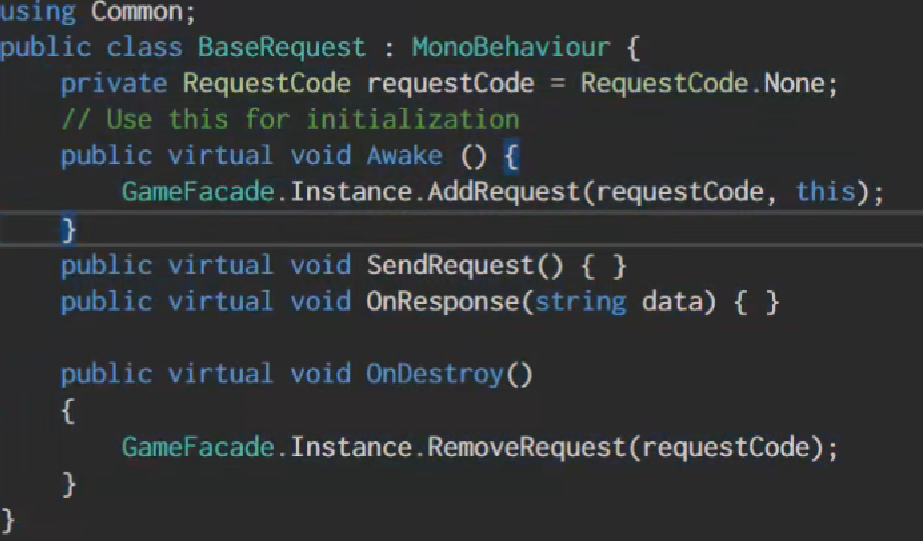
**45创建BaseRequset请求的基类**



**46Request对象的管理**

需要 请求的发起和请求的响应方法，服务器端发送过来的数据进行处理

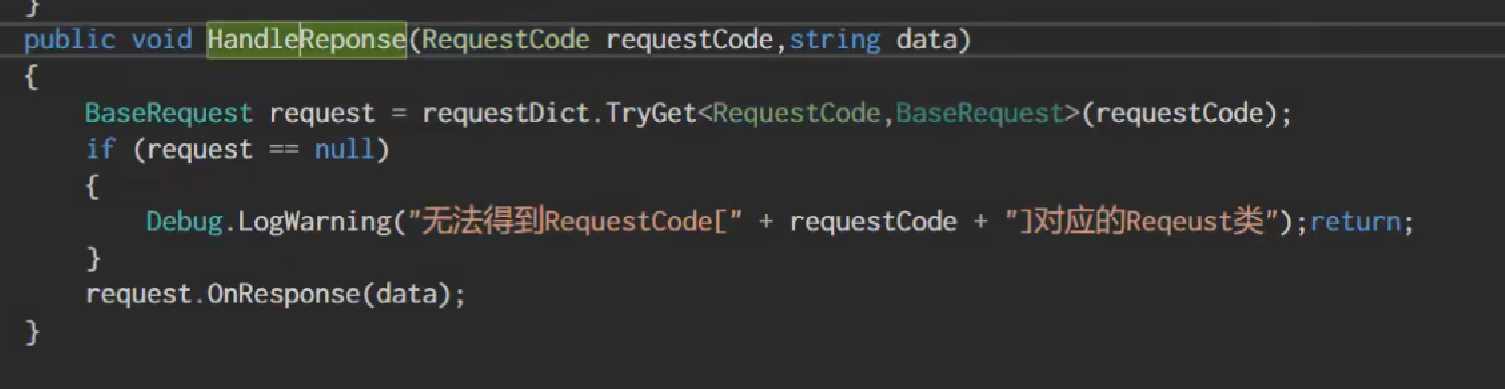
RequestManager类中，需要管理所有Request对象，通过RequestCode找到对应的Request对象发起请求，创建添加方法把Request对象添加进字典中。通过在GameFacade中创建单例模式的中介模式来对字典添加对象；再创建删除方法，通过RequestCode值来进行删除，还是在GameFacade的单例模式的中介设计方法来进行调用，最后在Request中添加对OnDestory的监听。



**47把消息转发给对应的Request处理**

在架构中网络接口解析服务器端的信息把解析的消息传递给对应的Request处理响应。

首先在ResquestManager中添加处理响应的方法，传入（ResquestCode来找到对应的Resquest，服务器端向我们发送过来的数据）



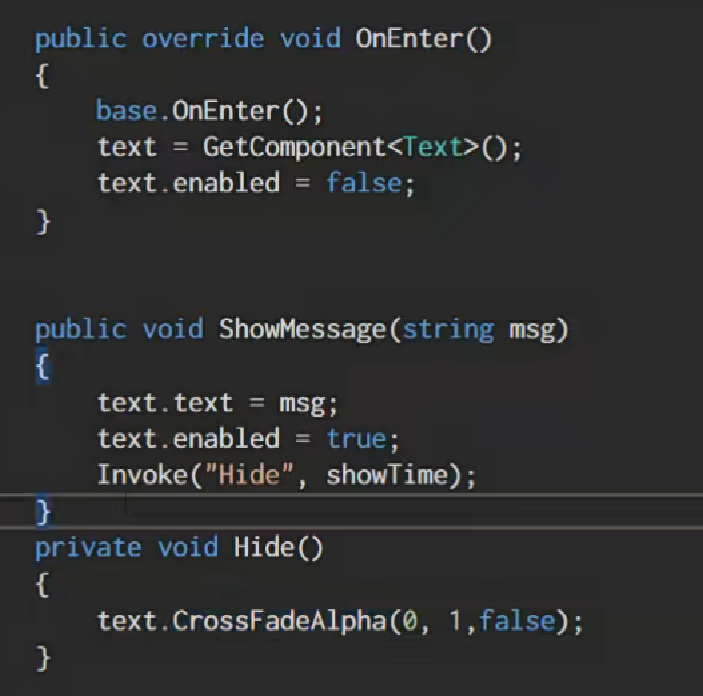
在GameFacade调用中转，clientManager调用处理服务器端的响应。

**48控制场景的视野漫游动画作为菜单界面背景**

通过动画组件控制主场景中摄像机的一个漫游状态

**54开发提示信息模块**

OnEnter在创建实例化时调用



**59设计数据库表**

第一张表存储用户信息，第二张是战绩、数据表

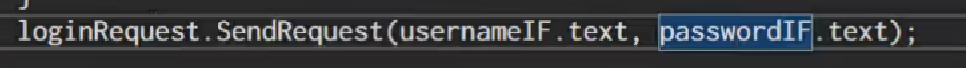
**60处理登录按钮的点击，校验账号信息是否为空**



判断账号密码是否为空，并根据对应的信息显示提示

**62在客户端发送登录请求**

创建方法传递账号密码，组拼账号密码发送数据，通过GameFacade中介去调用发送请求，将发起请求的方法写到父类中，通过调用方法使传递信息更加简洁



将账号密码传递到服务器中

**65在客户端处理登录的响应**

将数据转换成int类型再转换成枚举类型，调用数据解析方法，进行判断输入的数据和ReturnCode.Success，正确就进入创建房间界面，错误就提示信息。

**68在客户端发送注册请求**

进行注册的条件判断，对应不同条件输出不同提示信息。将注册信息按照同步输入到服务器中 ，在注册信息之后要求得到响应和请求，通过ReturnCode来判断是否注册成功。

**69在服务器端处理注册请求**

首先判断用户名是否存在，不存在就提那家到数据库中。

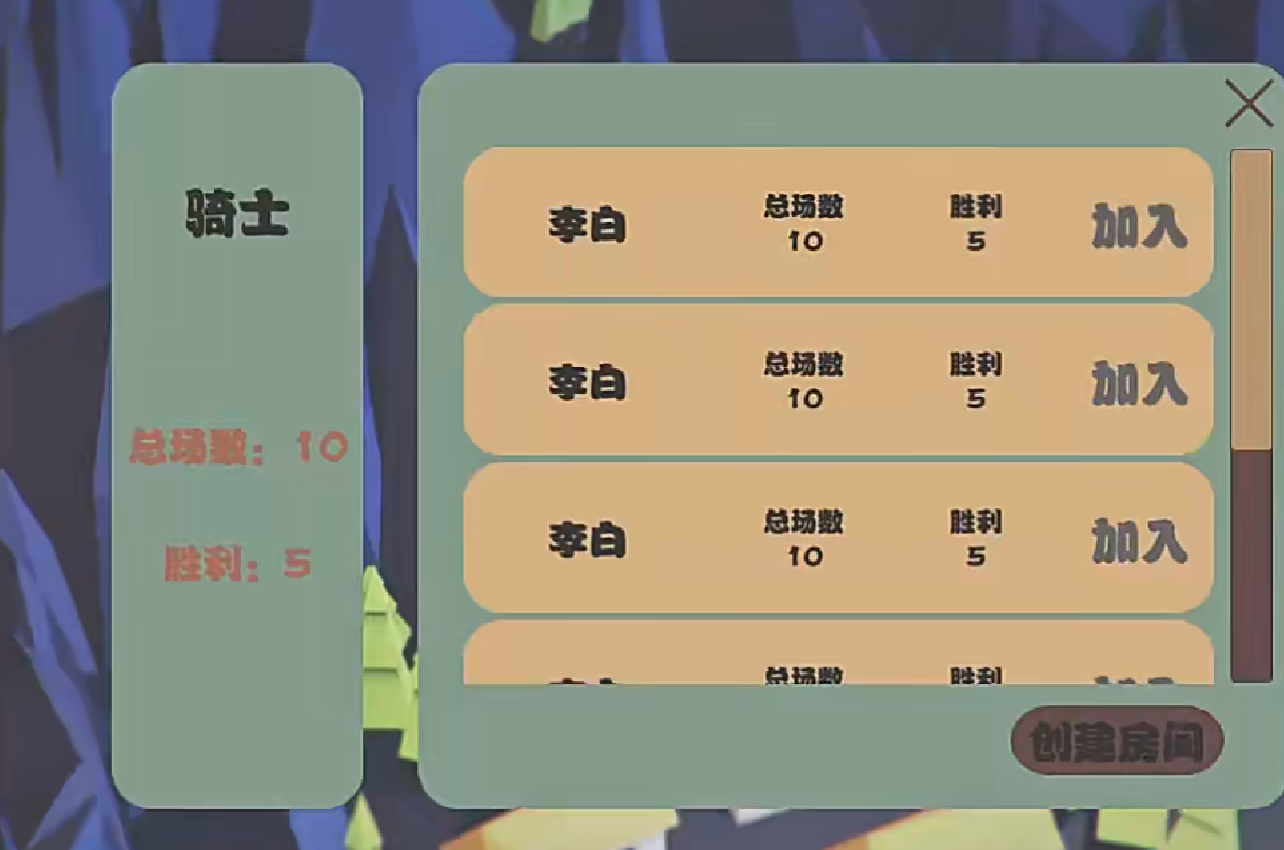
**71开发声音管理器**

在AudioManger中开发声音管理器,声明静态保存文件路径，统一调用， 设立播放设置，明确播放源，播放资源，音量大小，是否需要循环。

**74设计房间列表的UI界面**



**75开放房间中列表的滑动条核按钮**



**81在客户端处理响应保存战绩**

接受从服务器接收的信息，按照逗号进行分割，创建新的数据模型存放管理数据，通过调用构造函数去存储数据，通过数组一对一的存储数据，通过中介来调用方法

**83创建RoomItem控制一个房间项的显示和点击处理**

处理UIprefab的实例化，通过查找UI组件并注册每一个UI属性，进行初始化赋值。

**84创建房间列表**

加载房间个数，去实例化，将位置放在对应的UI布局之下，在加载的时候设置布局的高度，首先获取房间的个数，通过个数来改变布局的高度。

**85在服务器中创建跟房间相关的类与成员**

创建房间类，创建集合收集房间里的所有的客户端，明确房间在游戏中的状态：等待加入房间、等待战斗、战斗开始、结束状态，默认是等待加入状态。

在server建立对房间类的管理，在controller中处理房间类中的请求，建立请求类型：请求房间列表、创建房间、加入房间

**86开发房间界面UI**



**87开发设计房间面板的动画行为和按钮监听**

创建类来处理房间面板，设立开始按钮和退出房间按钮的监听，注册敌我双方的名字、总场数、胜利场数的属性，然后开始获取组件。

设立双方的战绩方法，开始初始化。

设立进入和离开的动画方法，注册组件属性，在进入方法中，给己方动画建立一个从左到右的动画行为，给敌方动画建立从右到左的动画行为，给开始和退出按钮安排了由小放大的行为；在离开方法中，直接就让动画行为退出。

**89发起创建房间的请求，在服务器端保存房间信息**

建立创建房间类，设定房间初始化，向服务器端发起创建房间请求，然后监听回调。

在服务器端创建RoomController，通过ControllerManager进行管理，建立方法处理客户端的CreatRoom，在server中管理创建房间功能，默认第一个创建者就是房主创建房间，在Room中设立一个添加客户端的功能，已经创建了一个房间。

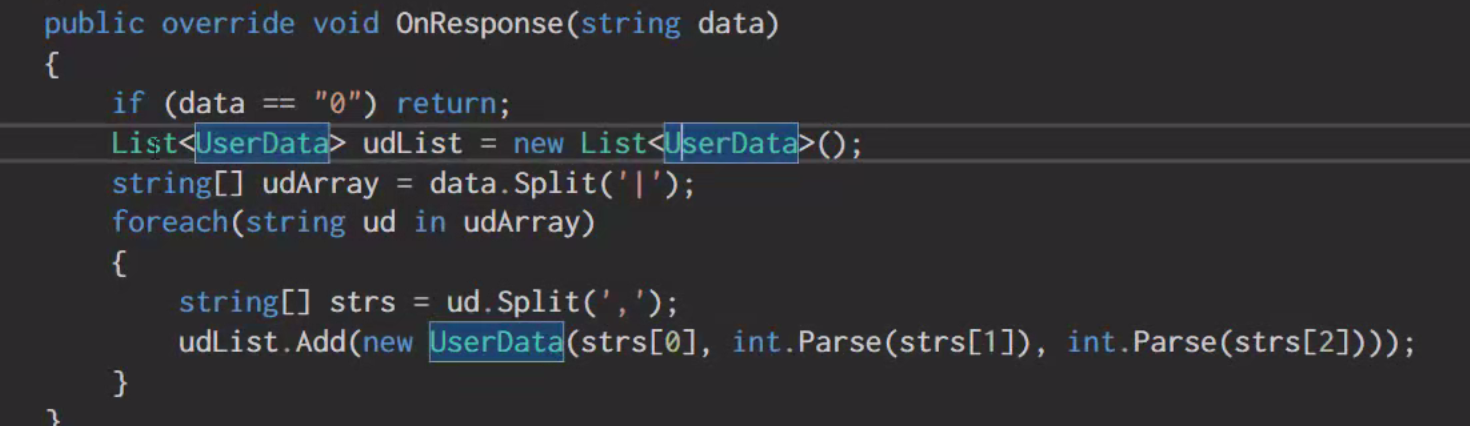
在客户端回调中解析传递的信息，成功创建就要把个人信息显示在蓝色界面中，红色面板清空等待用户加入

**90请求房间请求，在服务器端取得房间列表信息并返回**

创建Listist Roorequest，创建房主信息，传递房间对象信息，在请求房间列表的时候，通过sever取到房间的一个列表，通过room的属性取到client，再取得用户数据，把用户数据返回给客户端。

**91在客户端解析房间列表信息，并加载房间列表**

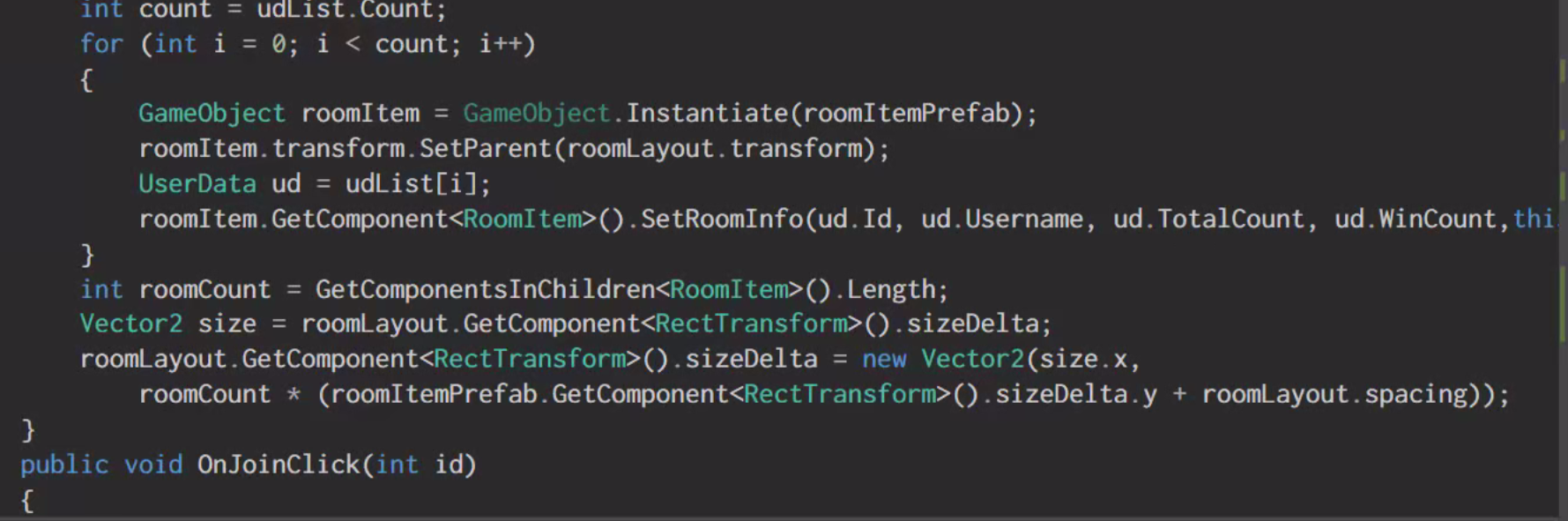
在客户端接受信息，解析显示。分割房间信息，得到房间对象，添加家到房间信息的列表，保存房间信息。



利用房间信息去创建列表，通过异步方法去进行加载，创建新的null的列表，将房间信息列表赋值给新列表，用新列表去进行调用，每调用一次，再次设为null，就不会重复调用。 在加载房间列表时，要加载的个数根据我们几个的个数来决定。

**92给房间添加ID，处理加入按钮的点击**

区分加入房间，需要把房间id从服务端传递给客户端，



**95处理加入房间请求的发起，和服务器端的处理**