小tip：当在WBP中，想点击按钮就可以进入一个网站的时候，使用“启动URL”函数。

小tip2：鼠标光标的切换

将鼠标光标切换成图片或2D动画

在游戏中动态运用修改鼠标样式

多个玩家是有多个玩家控制器的，在玩家控制器里有对应的PlayerState

游戏模式是可以管理所有的控制器的。

RPC网络复制

正文：

Steam平台，远程联机，不需要服务器，两台电脑直连，实时对战

官方网址：

<https://partner.steamgames.com/doc/home>

联机同步、伤害同步

显示Steam游戏名、网络延迟

创建房间、加入房间、切换地图

42min

## 前置

创建第三人称项目，蓝图项目，

启动方式选择“新建编辑器窗口PIE”

网络模式选择以监听服务器运行

玩家数量选择2

创建新的游戏模式和玩家控制器，将玩家控制器赋值到游戏模式中

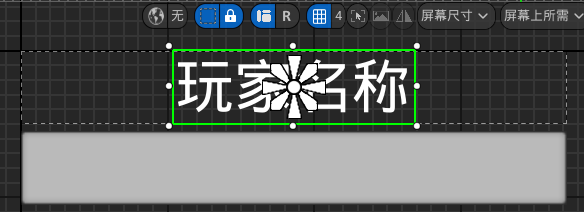
从BP\_ThirdPersonCharacter复制出一个新的角色作为GameMode中的默认角色。

（游戏模式重载不仅要在项目设置中进行设置，也要在世界场景设置中进行设置）

然后给角色头顶加上玩家名字，这个名字就显示Steam的玩家名字

创建WBP，命名为WBP\_UpHeadInfo

进入，加一个画布面板一个文本一个进度条（表示血条）



进入BP\_BaseCharacter，添加控件组件

“用户界面--空间”选择屏幕，“控件类”选择刚创建的WBP，在BP——BaseChar中设置一下和角色的相对位置。

此时可以开始游戏进行测试。注意要用PlayerStart，不要用BP的实例

完成基础东西

PlayerController的“获取受控pawn”函数

对于Pawn的“GetController”函数

## 技能同步与伤害同步

所有和伤害相关的东西，都要在服务器上执行。

我们首先完成一下数值系统。

在BaseCharacter中创建MaxHP和HP两个变量

## Steam在线联机

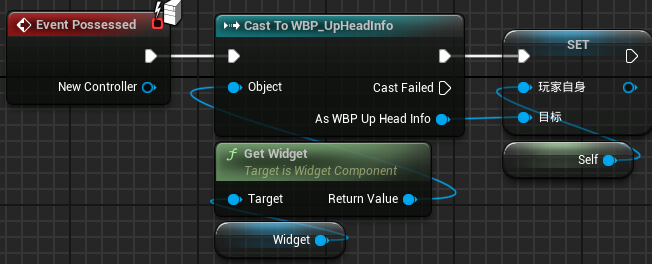
实现互相放技能

创建两个IA：Attack和Skill，

## 读取玩家名称

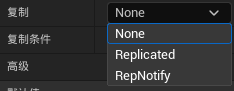
在WBP\_UpHeadInfo中添加BP\_BaseCharacter类变量，命名为“玩家自身”，传入信息

进入BP\_BaseChar，重载函数“已控制（EventPossessed）”，在已控制时，对控件组件调用“获取控件”函数，将BP自己赋值给WBP中的玩家自身变量。



然后进入到WBP。

在“玩家自身”变量的细节中，找到赋值



多人游戏的关键就是复制。服务器复制到客户端，客户端复制到其他的客户端和服务器。

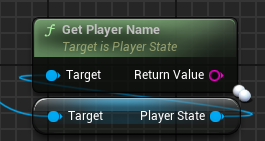
变量也是可以复制的，变量默认是不复制的，都是在客户端的本地。

但是“EventPossessed”是只在服务器上触发的。所以为了解决EventPossessed在客户端上也触发，就需要给“玩家自身”变量的复制设置为Replicated

此时Set节点右上角会多两个白球（变量的右上角有两个白球说明是实时更新的一个变量）

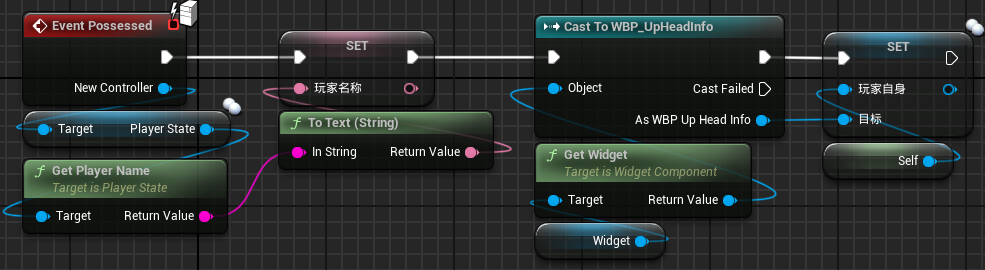


然后我们从EventPossessed函数返回的玩家控制器类，得到PlayerState并得到PlayerName



注意，这里的GetPlayerState的中文节点名称为“获取播放器命名”，属于翻译错误

然后在BP\_BaseChar中创建文本型变量命名为“玩家名称”，进行赋值



最后在WBP中对文本进行一次绑定



此时进入游戏就会发现：

服务器上会在头顶显示客户端的玩家名称，不显示服务器自己的

客户端上不显示玩家名（因为客户端不执行Possessed函数）

这里，服务器上是有玩家名字的，只是没有显示出来。为了在服务器上显示很简单，给EventPossessed加个延迟即可。

这里基本就完成了Steam的名称赋值。只要是在Steam上运行，就会自动赋值Steam的账号名。

但是这里不用延迟来实现。我们处理一下玩家控制器。

首先讲个知识点：游戏模式只存在于服务器，玩家控制器存在于所有机器上。

所以为了在所有机器上都能够看到玩家名称，在玩家控制器里实现设置的功能。

在玩家控制器中，重载函数“控制时EventOnPossess”，或者EventBeginPlay时调用。都可以。

区别是，这个实验项目默认控制器有控制的Pawn。如果控制器没有控制的Pawn，就尴尬了。

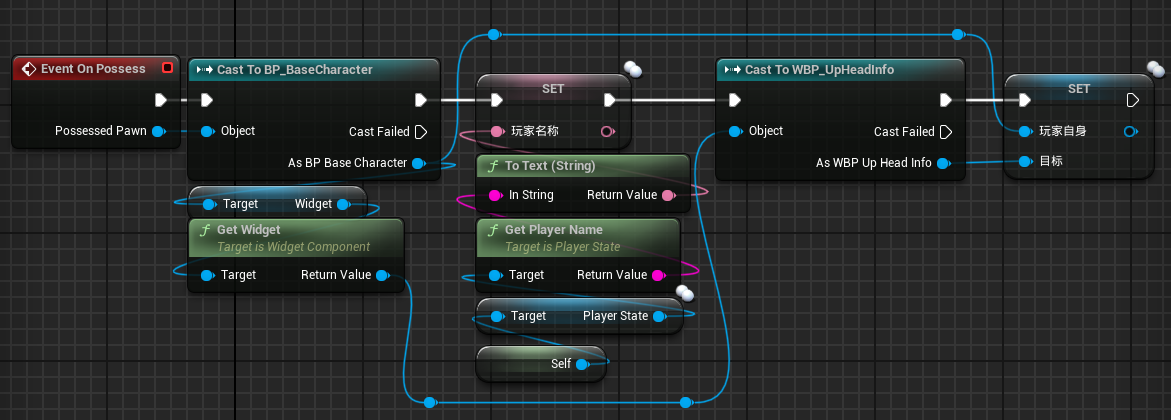
在这个实验项目中，回到BP\_BaseChar，在EventBeginPlay时调用相关功能。

注意这里的PlayerState的参数不是GetPlayerController函数的返回值，而是GetController函数的返回值



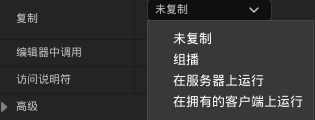
这样当然可以实现，但是不是正常流程。我们需要走复制的流程。

进入玩家控制器，当事件控制时，



将玩家名称变量设置为“Replicated”

在自定义事件的细节中，复制选项中有四个选项



未复制：谁调用谁执行

组播：由服务器同时给所有客户端发送这条指令

在服务器执行

在拥有的客户端上运行

## 创建房间、加入房间、切换地图

首先我们要创建大厅功能。

首先将我们现在的关卡改名为默认房间

创建一个新地图“战斗房间”

这样我们就有两个关卡。

一般来说，这两个关卡需要有各自的GameMode、PlayerController等。

复制BP\_BaseGameMode，命名为BP\_LobbyGameMode

赋值BP\_BasePlayerController，命名为BP\_LobbyPlayerController，在GameMode中进行设置。

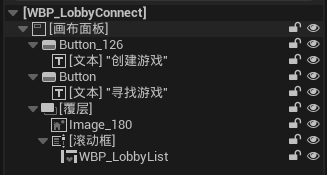
将默认房间的世界场景设置中的游戏模式进行设置

创建WBP\_LobbyConnect，在LobbyPC的EventBeginPlay时CreateWidget

创建WBP\_LobbyList



回到WBP\_LobbyConnect



## Steam联机功能

子系统

在UE项目，找到Config文件夹，打开DefaultEngine.ini文件

回到UE，在插件中找到

Online Subsystem Steam

Advanced Steam Sessions（不是必须的）

关于AdvancedSteamSession插件，首先需要安装，

在<https://vreue4.com/advanced-sessions-binaries>

下载对应的版本



这个项目是用的是UE5.4.4

下载相应的版本

下载完成后，打开UE的安装文件夹，找到插件的路径

本机路径为D:\Unreal Engine\UE\_5.4\Engine\Plugins

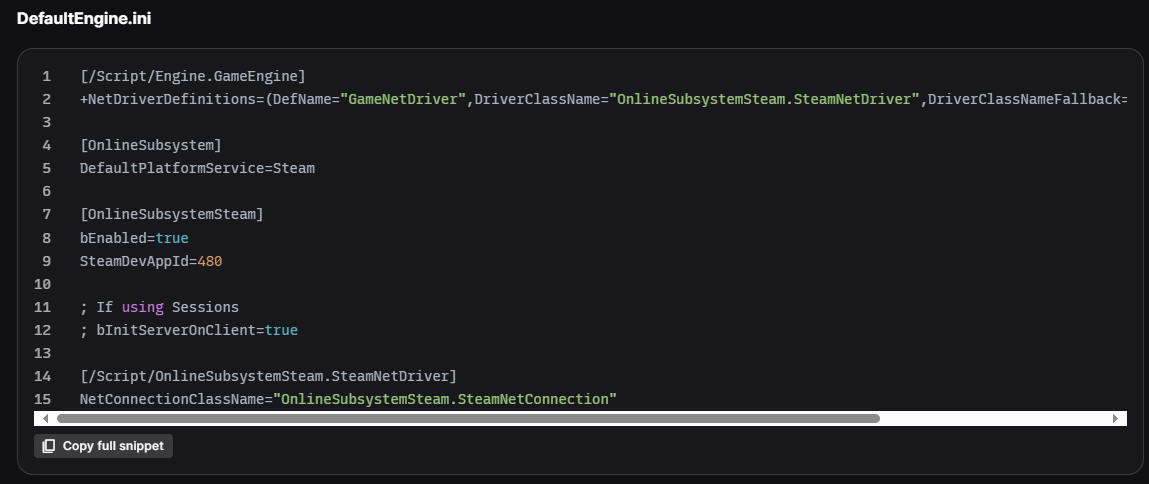
将下载后解压得到的名称为AdvancedSessions的文件夹，直接复制到Plugins文件夹下

然后重新启动一下UE，就可以启用这个插件了。

安装完插件，回到Steam文献库

<https://dev.epicgames.com/documentation/zh-cn/unreal-engine/online-subsystem-steam-interface-in-unreal-engine?application_version=5.5>

在DefaultEngine.ini中添加如下内容



[/Script/Engine.GameEngine]

+NetDriverDefinitions=(DefName="GameNetDriver",DriverClassName="OnlineSubsystemSteam.SteamNetDriver",DriverClassNameFallback="OnlineSubsystemUtils.IpNetDriver")

[OnlineSubsystem]

DefaultPlatformService=Steam

[OnlineSubsystemSteam]

bEnabled=true

SteamDevAppId=480

; If using Sessions

; bInitServerOnClient=true

[/Script/OnlineSubsystemSteam.SteamNetDriver]

NetConnectionClassName="OnlineSubsystemSteam.SteamNetConnection"

直接粘贴到最上面就行。

注意这里

SteamDevAppId=480

1.这里的480需要时你对应的游戏ID

0 ，。

默认的480是SpaceWar

注意这里

; bInitServerOnClient=true

把他前面的分号删掉

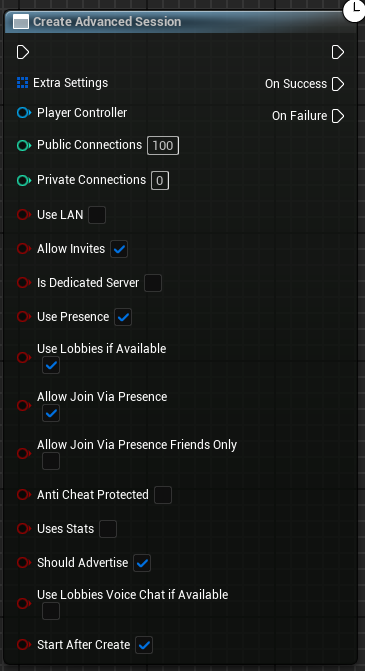
表示开启Steam的多人联机服务

此时若选择独立进程游戏方式启动，可以使用steam多人服务

## 创建房间

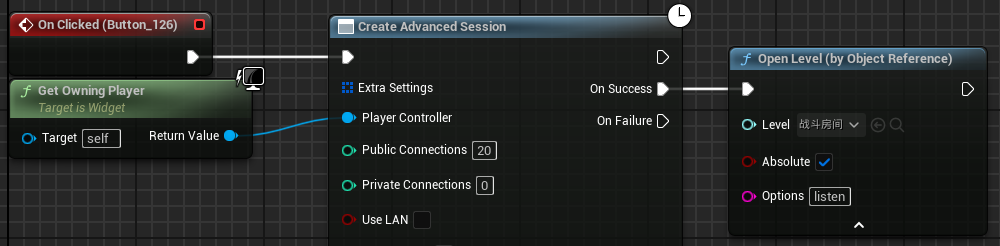
进入WBP\_LobbyConnect，当创建游戏按钮点击时，创建会话

使用的节点是CreateAdvancedSession



这里的PublicConnection表示人数上限

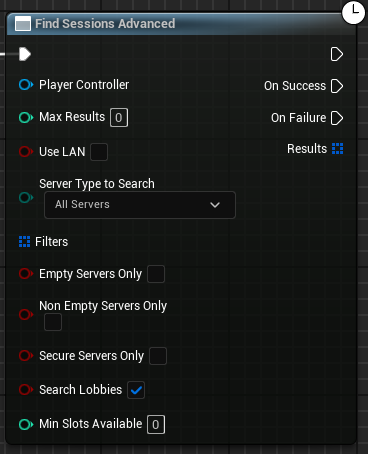
UseLan表示局域网，勾选了就不能使用Steam了。有局域网需求可以创建一个新的按钮。



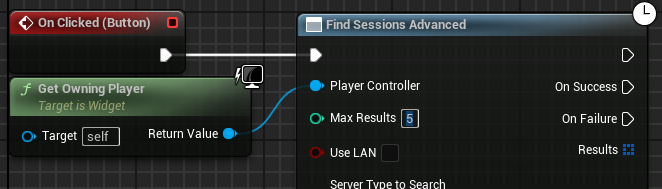
注意Options参数必须准确

当寻找游戏按钮点击时，加入会话

使用的节点是FindSessionsAdvanced

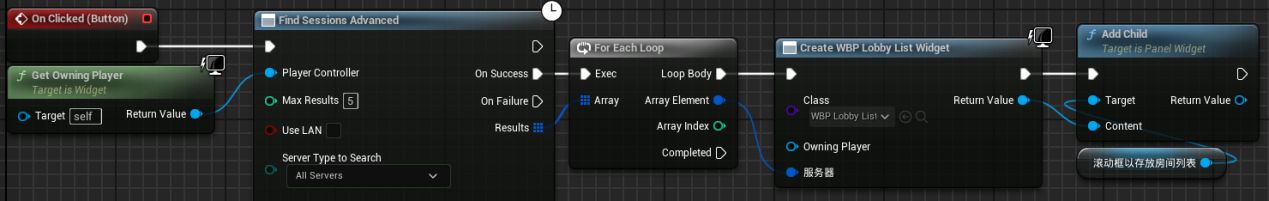


MaxResults表示最大搜索多少个



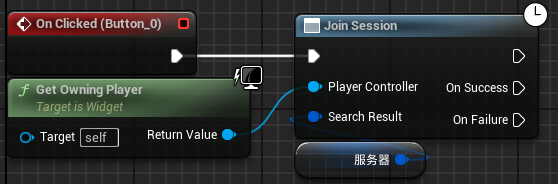
然后进入WBP\_LobbyList，创建变量，类型为“蓝图会话结果”，命名为“服务器”，可编辑实例生成时公开

回到WBP\_LobbyConnect



此时以独立进程游戏启动多个进程，一个进程创建游戏，其他进程点击寻找游戏

然后进入WBP\_LobbyList，在按钮点击时加入会话

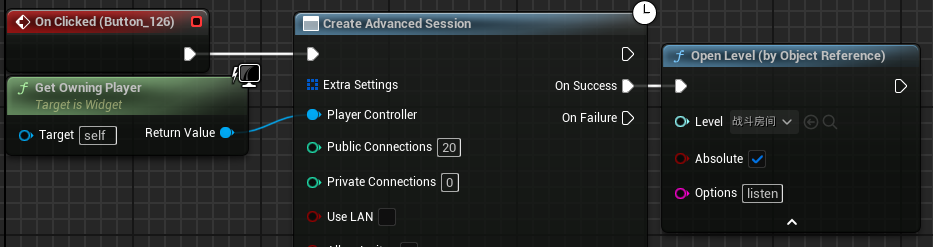


然后显示玩家数量、房间名称等。

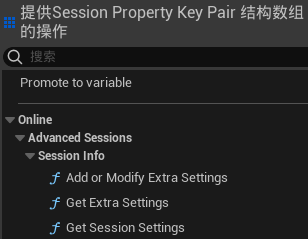
在WBP\_LobbyList中，我们有当前人数和最大人数两个通用文本。

在事件图表中，创建两个int型变量“当前玩家数”“最大玩家数”，可编辑实例生成时公开。将这两个值分别绑定到设计器中的两个通用文本。

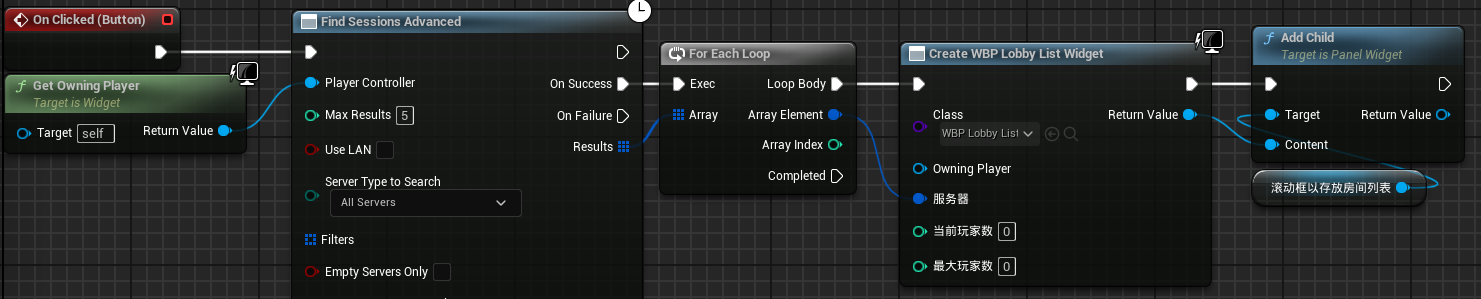
在WBP\_LobbyConnect中，当创建会话按钮点击时，我们有如下操作

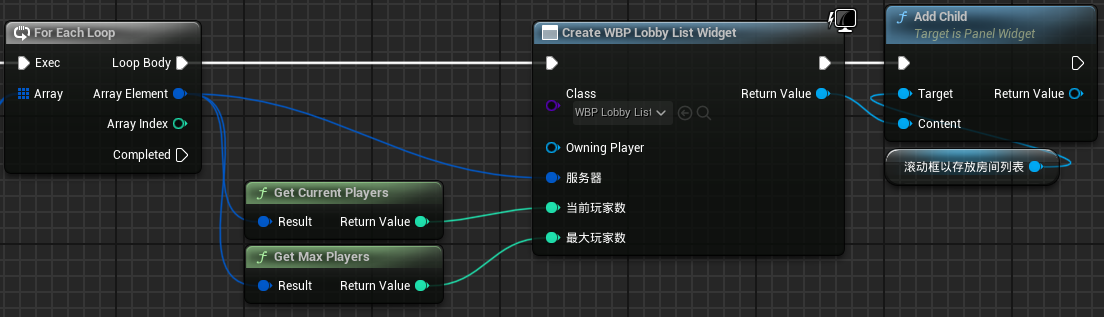


关注CreateAdvancedSession节点，这里的ExtraSettings参数，是插件自带的一个功能（其他的插件也会有类似的功能）。这个功能就是传入一些参数给服务器。也就是说服务器可以有一些自定义参数。

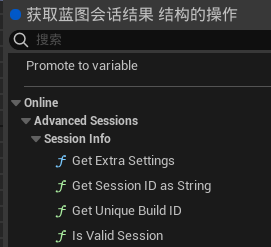


然后关注FindSessionsAdvanced函数的返回值。





这个返回值在forLoop之后得到的数组中的元素个体，



这个插件，在FindSessionsAadvanced的时候，如果房间人满了，自动找不到该房间

我们在给CreateWidget节点提供Get Current&Max Players的时候，就已经可以在WBP中看到正确的数据了。

补充：主机如果想动态的修改房间人数、是否允许邀请好友、设置开放关闭等，可以使用UpdateSession函数。他是立刻生效的。

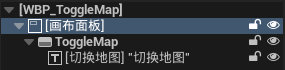
## 集体切换地图

我们已经在地图里了，现在希望可以让所有人一起去切换地图，

首先创建一个新的关卡，命名为“新房间”

创建WBP\_ToggleMap

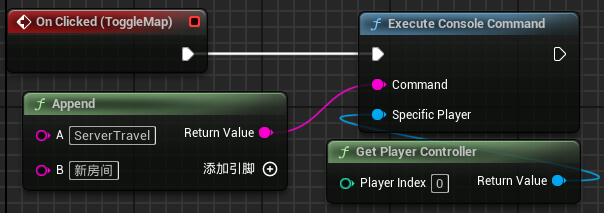
这个只能由服务器来切换





当按钮点击时，我们要让所有人一起切换地图。

这里要用一个新的节点：执行控制台指令ExecuteConsoleCommand



然后我们回到游戏模式，在游戏模式中创建控件

（我们在游戏模式下创建控件，这样就只有服务器能够看到这个控件）

（ServerTravel和地图名之间有一个空格）

进入游戏，点击按钮之后会发现我们不是每个人同时进入地图的。

此时我们要做一个无缝漫游

关注战斗房间的游戏模式是BP\_BaseGameMode，在GameMode的类默认值中，有“游戏模式--使用无缝漫游”，勾选即可。

这样就可以所有角色在同一时间切换地图。

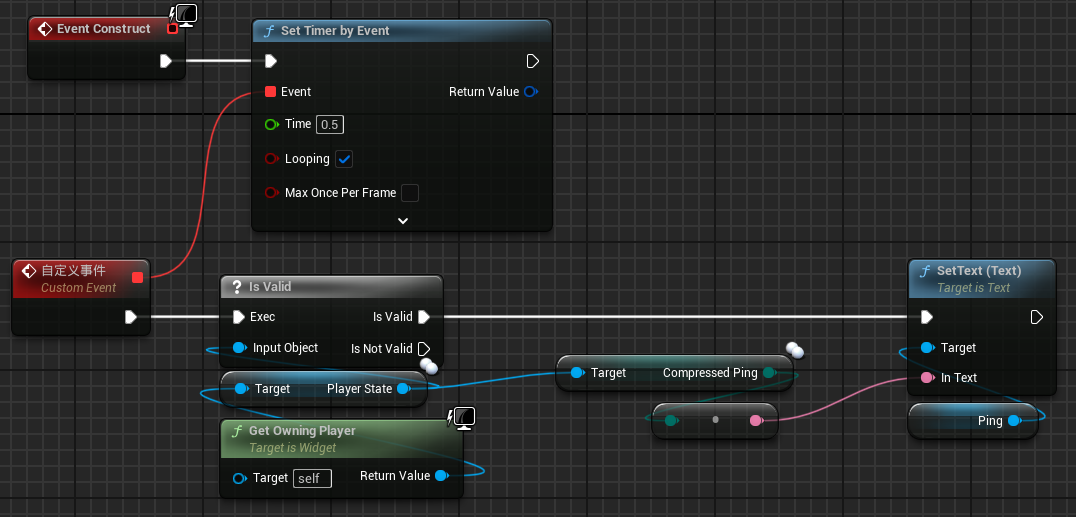
如果不用无缝漫游，他们就会先断开连接，然后每个人一个一个的进去。

## 显示Steam玩家名称

## 显示网络延迟

创建WBP\_Ping





在PlayerController中CreateWidget。

完成。

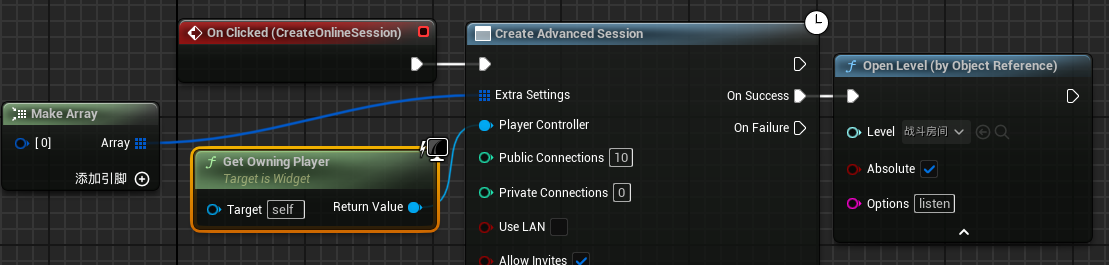
## 房间名称

在WBP\_LobbyList中创建文本型变量“房间名称”，可编辑实例生成时公开。

在设计器中给房间名字进行绑定。

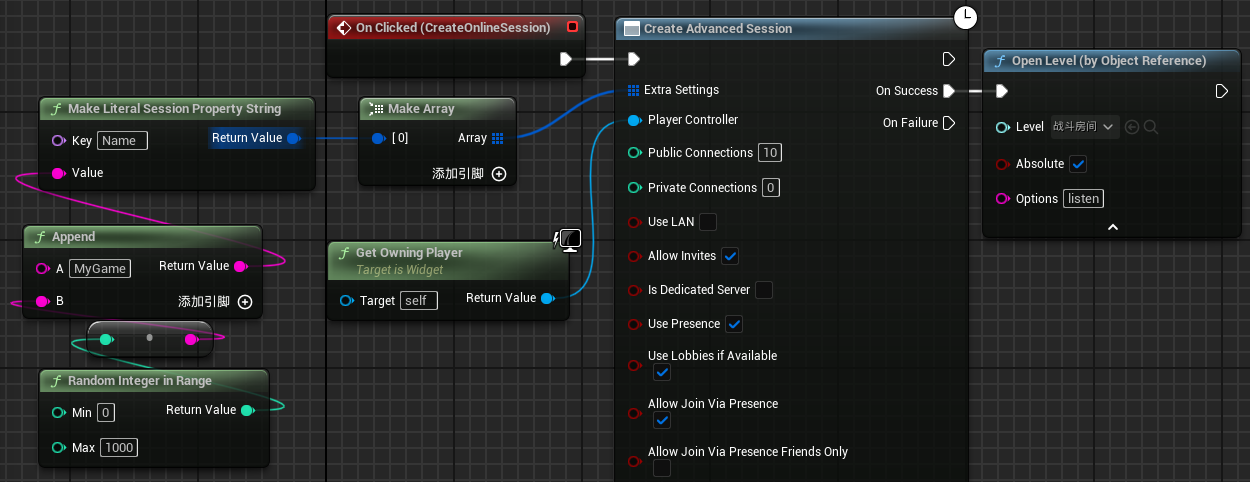
然后实现将起的名字传给服务器。

首先在WBP\_LobbyConnect中，对于CreateAdvancedSession的ExtraSettings参数，make Array

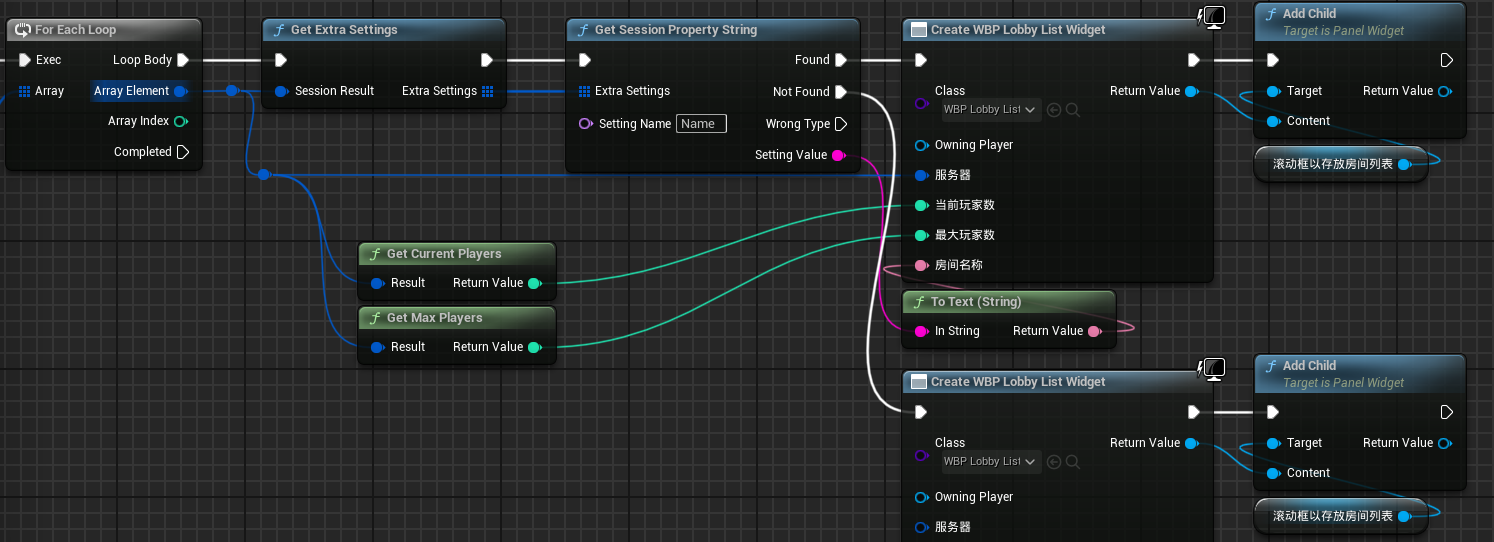


我们通过MakeLiteralSessionProperty节点来创建数组中的各个元素

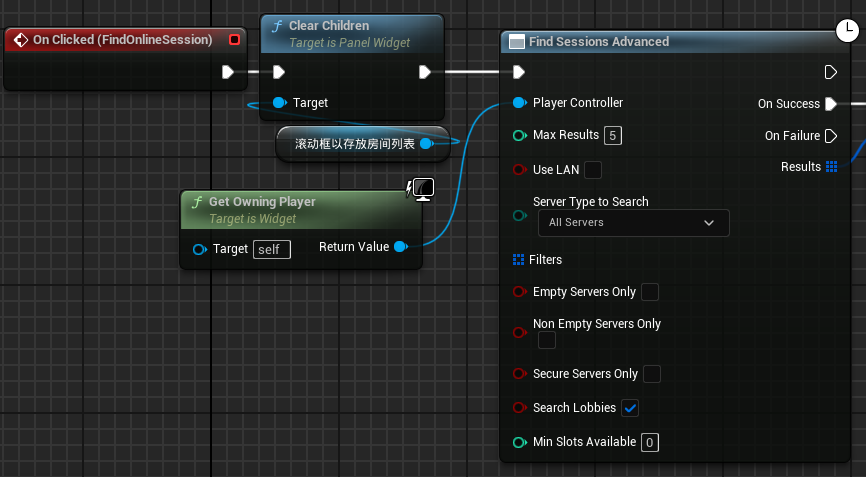
比如我们创建房间名称就使用MakeLiteralSessionPropertyString来创建



在FindSession后，我们会得到返回值。我们通过ForLoop遍历这个返回值后，可以得到数组中每个元素的本身，进而得到元素本身的属性



补充：每次点击寻找游戏按钮的时候，我们应该把存放房间列表的滚动框清空。



后续还需要做的联机模块：

房间内玩家掉线了

谁创建的游戏

房间人数已满

## 断开连接时销毁会话

如果一个回话结束了一定要断开连接，不然会搜索不到新的房间。

使用的是销毁会话DestroySession节点。