<https://docs.unity3d.com/540/Documentation/Manual/UNetLobby.html>

# Multiplayer Lobby

**多人大厅**

Many multiplayer games have a staging area for players to join before playing the actual game. In this area - often called the “lobby”, players may be able to pick options and be able to set themselves as ready for the game to start.

大多数多人游戏都有一个让玩家进入实际游戏之前的等待区域 。这个区域-通常称为“大厅”，玩家可以设定选项并且可以设置他们已经准备好开始游戏了。

The NetworkLobbyManager is a specialized [NetworkManager](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkManager.html) that provides a lobby for Unity Multiplayer games. It includes:

NetworkLobbyManager是Unity为多人游戏提供的一种专用的[NetworkManager](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkManager.html) 。它包括：

* Limit on number of players that can join
* 限制可以加入的玩家数量
* Support for multiple players per client with a limit on number of players per client
* 支持一个客户端多个玩家，并限制一个客户端的玩家数量
* Prevent players from joining game in-progress
* 阻止 玩家加入正在进行的游戏
* Per-player ready state, so that game starts when all players are ready
* 每位玩家的准备状态，以至于所有玩家准备好时开始游戏
* Per-player configuration data
* 每位玩家的配置数据
* Re-joining the lobby when the game is finished
* 当游戏结束时重新加入大厅
* Virtual functions that allow custom logic for lobby events
* 虚方法可以为大厅事件定义自定义逻辑
* A simple user interface for interacting with the lobby
* 一个与大厅交互的简洁用户界面

Below are the NetworkLobbyManager virtual functions called on the server:

以下是NetworkLobbyManager在服务器上调用的虚方法：

* [OnLobbyStartHost](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyManager.OnLobbyStartHost.html)
* [OnLobbyStopHost](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyManager.OnLobbyStopHost.html)
* [OnLobbyStartServer](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyManager.OnLobbyStartServer.html)
* [OnLobbyServerConnect](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyManager.OnLobbyServerConnect.html)
* [OnLobbyServerDisconnect](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyManager.OnLobbyServerDisconnect.html)
* [OnLobbyServerSceneChanged](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyManager.OnLobbyServerSceneChanged.html)
* [OnLobbyServerCreateLobbyPlayer](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyManager.OnLobbyServerCreateLobbyPlayer.html)
* [OnLobbyServerCreateGamePlayer](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyManager.OnLobbyServerCreateGamePlayer.html)
* [OnLobbyServerPlayerRemoved](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyManager.OnLobbyServerPlayerRemoved.html)
* [OnLobbyServerSceneLoadedForPlayer](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyManager.OnLobbyServerSceneLoadedForPlayer.html)
* [OnLobbyServerPlayersReady](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyManager.OnLobbyServerPlayersReady.html)

All of the above server functions have empty default implementations, except for OnLobbyServerPlayersReady, which calls ServerChangeScene with the PlayScene.

上述所有的服务器函数都有默认的空实现，除了OnLobbyServerPlayersReady，它将在进入游戏场景时调用ServerChangeScene

Below are the NetworkLobbyManager virtual functions called on the client:

以下是NetworkLobbyManager在客户端调用的虚方法：

* [OnLobbyClientEnter](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyManager.OnLobbyClientEnter.html)
* [OnLobbyClientExit](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyManager.OnLobbyClientExit.html)
* [OnLobbyClientConnect](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyManager.OnLobbyClientConnect.html)
* [OnLobbyClientDisconnect](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyManager.OnLobbyClientDisconnect.html)
* [OnLobbyStartClient](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyManager.OnLobbyStartClient.html)
* [OnLobbyStopClient](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyManager.OnLobbyStopClient.html)
* [OnLobbyClientSceneChanged](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyManager.OnLobbyClientSceneChanged.html)
* [OnLobbyClientAddPlayerFailed](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyManager.OnLobbyClientAddPlayerFailed.html)

All of the above client functions have empty default implementations.

所有呈现的客户端方法都有空的默认实现。

**Lobby Player Objects**

**大厅玩家对象**

There are two kinds of player objects - each which has a prefab slot in the NetworkLobbyManager. The slots can be seen in this screenshot:

有两种玩家对象：每一种都在NetworkLobbyManager有一个预制体栏位。它可以在截图上看到：



The LobbyPlayer is created from the LobbyPlayerPrefab when a player joins the lobby:

当一个玩家加入大厅时将从LobbyPlayerPrefab创建LobbyPlayer:

* One LobbyPlayer for each player
* 每一个玩家对应一个LobbyPlayer
* Created when client connects, or player is added
* 当客户端连接，或者玩家加入时创建
* Exists until client disconnects
* 直到客户端断开连接时将一直存在
* Holds the ready flag for this player for the lobby
* 玩家在大厅时将保持准备标志。
* Handles commands while in the lobby
* 在大厅时处理命令
* Add user scripts to this prefab to hold game-specific player data
* 添加到这个预制体的用户脚本将保持游戏专有的玩家数据
* This prefab must have a [NetworkLobbyPlayer](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyPlayer.html) component
* 这个预制体必须拥有一个[NetworkLobbyPlayer](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyPlayer.html) 组件

**Minimum Players**

**最少玩家数**

The “Minimum Players” field represents the minimum number of “Ready” players in the Lobby to start the Match with. If the number of connected clients is more than the “Minimum Players” value, then waiting for all connected clients to become “Ready” will start the Match.

“最少玩家数”字段代表大厅可以开始比赛的最少玩家数量。如果连接的客户端数量大于最少玩家数量，等待连接的所有客户端都成为“Ready”状态时将开启比赛。

For example if “Minimum Players” is set to 2:

举个例子，如果最少玩家数量设为2：

* Start one instance of the game and start Host. Then in game Lobby UI press “Start” for your player. You will still be in Lobby mode because minimum number of Ready players to start game is 2.
* 作为主机启动了一个游戏实例 ，如果玩家按下了游戏大厅UI 的“Start”，因为启动游戏的最少玩家数是2，所以仍将在大厅模式。
* Start two more instances of the game and start Clients there. It doesn’t matter that “Minimum Players” set to 2. Wait for all - 3 in this case - connected players to become Ready.
* 启动了2个客户端的游戏实例 。如果“最少玩家数量”设置为2也没有关系。等待这种情况中的3个连接的玩家都准备好。
* Press “Start” in Lobby UI for one player. Two players are Ready, but still in Lobby mode.
* 当只有一个玩家按下了大厅的“开始”UI时，如果两个玩家都准备了，但是将仍处于大厅模式。
* Press “Start” in the Lobby UI for the last player and now all players moved to Game mode.
* 当最后一个玩家按下了大厅的“开始”UI，现在将移动所有的玩家到游戏模式中。

**GamePlayer**

**游戏玩家**

The GamePlayer is created from the GamePlayerPrefab when the game starts:

当游戏开始时GamePlayer将从GamePlayerPrefab创建。

* One GamePlayer for each player
* 每一个玩家对应一个GamePlayer
* Created when game scene is started
* 当游戏场景开始时创建
* Destroyed when re-entering lobby
* 当重新进入大厅时销毁
* Handles commands while in the game
* 在游戏中处理命令
* This prefab must have a [NetworkIdentity](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkIdentity.html) component
* 这个预制体必须拥有一个 [NetworkIdentity](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkIdentity.html) 组件

The [NetworkLobbyPlayer](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyPlayer.html) component is used for LobbyPlayer objects. It supplies some virtual function callbacks that can be used for custom lobby behaviour

[NetworkLobbyPlayer](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyPlayer.html) 组件用于LobbyPlayer对象。当游戏进入大厅时，客户端会调用OnClientEnterLobby函数

public virtual void OnClientEnterLobby();

public virtual void OnClientExitLobby();

public virtual void OnClientReady(bool readyState);

The function [OnClientEnterLobby](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyPlayer.OnClientEnterLobby.html) is called on the client when the game enters the lobby. This happens when the lobby scene starts for the first time, and also when returning to the lobby from the game-play scene.

在游戏进入大厅的客户端将调用[OnClientEnterLobby](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyPlayer.OnClientEnterLobby.html)方法。这发生在大厅场景第一次启动时，从游戏场景回到大厅时也会发生

The function [OnClientExitLobby](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyPlayer.OnClientExitLobby.html) is called on the client when the game exists the lobby. This happens when switching to the game-play scene.

在游戏退出大厅时调用函数[OnClientExitLobby](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyPlayer.OnClientExitLobby.html)。这会在切换到游戏运行场景时发生。。

The function [OnClientReady](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyPlayer.OnClientReady.html) is called on the client when the ready state of that player changes.

在玩家改变准备的状态的客户端会调用[OnClientReady](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyPlayer.OnClientReady.html)函数。

**Adding the Lobby to a Game**

**在游戏中添加大厅**

Process for adding a NetworkLobby to a multiplayer game (without using the multiplayer-lobby asset package):

添加一个NetworkLobby到一个多人游戏的过程（不使用多人大厅资源包）

* Create a new lobby scene
* 创建一个新的大厅场景
* Add the scene to the build settings, as the first scene
* 添加场景到构建设置中，作为第一个场景
* Create a new game object in the new scene, rename it to LobbyManager
* 创建一个新的游戏对象到新场景中，重命名为LobbyManager
* Add the NetworkLobbyManager component to the LobbyManager object
* 增加NetworkLobbyManager组件到LobbyManager对象上
* Add the NetworkManagerHUD component to the LobbyManager object
* 增加NetworkManagerHUD组件到LobbyManager对象上
* Open the inspector for the NetworkLobbyManager component
* 打开NetworkLobbyManager组件的展示面板
* Set the LobbyScene slot of the NetworkLobbyManger to the scene that contains the LobbyManager object
* 设置NetworkLobbyManager的LobbyScene预制体栏位为包含LobbyManager对象的场景
* Set the PlayScene slot of the NetworkLobbyManager to the main game-play scene for the game
* 设置NetworkLobbyManager的PlayScene预制体栏位为游戏主要的游戏玩法场景
* Create a new gameObject and rename it to LobbyPlayer
* 创建一个新的游戏物体并重命名为LobbyPlayer
* Add the NetworkLobbyPlayer component to the LobbyPlayer
* 为LobbyPlayer添加NetworkLobbyPlayer组件
* Create a prefab for the LobbyPlayer and delete the instance from the scene
* 为LobbyPlayer创建一个预制体并把场景的实例删除
* Set the LobbyPlayerPrefab slot to the LobbyPlayer prefab
* 设置LobbyPlayer预制体到LobbyPlayerPrefab预制体栏位
* Set the GamePlayerPrefab slot to the prefab for the player in the main game
* 在主游戏为玩家设置GamePlayerPrefab 预制体栏位
* Run the game
* 运行游戏
* Save the scene.
* 保存场景

This version of the NetworkLobbyManager uses the OnGUI user interface like the [NetworkManagerHUD](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkManagerHUD.html). For a better user interface use the multiplayer-lobby asset package.

这个版本的NetworkLobbyManager使用类似于[NetworkManagerHUD](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkManagerHUD.html)的OnGUI用户界面。

为使用多玩家大厅资源包提供一个更好的用户界面。

The NetworkLobbyManager has many virtual function callbacks that can be used for custom lobby behaviour. Most important is [OnLobbyServerSceneLoadedForPlayer](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyManager.OnLobbyServerSceneLoadedForPlayer.html) which is called on the server for each player as they transition from the lobby to the main game. This is the ideal place to apply settings from the lobby to the player’s game-play object.

NetworkLobbyManager有为自定义大厅行为的多个虚方法回调。其中最重要的是[OnLobbyServerSceneLoadedForPlayer](https://docs.unity3d.com/540/Documentation/ScriptReference/Networking.NetworkLobbyManager.OnLobbyServerSceneLoadedForPlayer.html) 方法，它在每个玩家从大厅转移到主游戏时调用。这是将玩家在大厅的设置应用到游戏场景中对象的理想位置。

// for users to apply settings from their lobby player object to their in-game player object

public override bool OnLobbyServerSceneLoadedForPlayer(GameObject lobbyPlayer, GameObject gamePlayer)

{

var cc = lobbyPlayer.GetComponent<ColorControl>();

var player = gamePlayer.GetComponent<Player>();

player.myColor = cc.myColor;

return true;

}

**Sample Project**

**示例工程**

There is a sample project on the Unity asset store that uses the NetworkLobbyManager and provides a GUI for the lobby. This can be used as a starting point for making a multiplayer game with a lobby.

在Unity资源商店有一个使用NetworkLobbyManager并且提供大厅GUI的示例工程。它可以是你创建多人游戏的大厅的起点。

[Lobby Sample Project](https://www.assetstore.unity3d.com/en/#!/content/41836)