Setting Up a MasterServer- Part 5- Initalize a Server (call script for logining in on a trigger)

**Initializing a server** **The first thing you will want to set up is hosting games and joining games. To ini- tialize a server on the local machine, call Network.InitializeServer.** **This method takes three parameters: the number of allowed incoming connections, the port to listen on, and whether to use NAT punch-through. The following script initializes a server on port 25000 which allows 8 clients to connect:**

**using UnityEngine;**

**using System.Collections;**

**public class ExampleUnityNetworkInitializeServer : MonoBehavior {**

**void OnGUI()   {**

**if( GUILayout.Button( "Launch Server" ) )**

**{**

**LaunchServer();     }   }**

**// launch the server**

**void LaunchServer()**

**{**

**// Start a server that enables NAT punchthrough,**

**// listens on port 25000,**

**// and allows 8 clients to connect**

**Network.InitializeServer( 8, 25005, true );   }**

**// called when the server has been initialized**

**void OnServerInitialized()   {     Debug.Log( "Server initialized" );   }**

**}** **You can also optionally enable an incoming password (useful for private games) by setting Network.incomingPassword to a password string of the player's choice, and initializing a general-purpose security layer by calling Network.InitializeSecurity(). Both of these should be set up before actually initializing the server.**  **Note** **Note that incoming connections does not mean maximum player count, since it does not include the host (for example, if you allow 8 players to connect, it's possible for 9 players to play in the same room—8 clients plus the host).**