# Unity Networking

## Network Manager

In order to create a LAN multiplayer game you start with a script attached to an empty game object in the scene. In this script you start the server and handle connections, when the host clicks a create game button it calls the function

Network.InitializeServer (int connections, int listenPort, bool useNat);

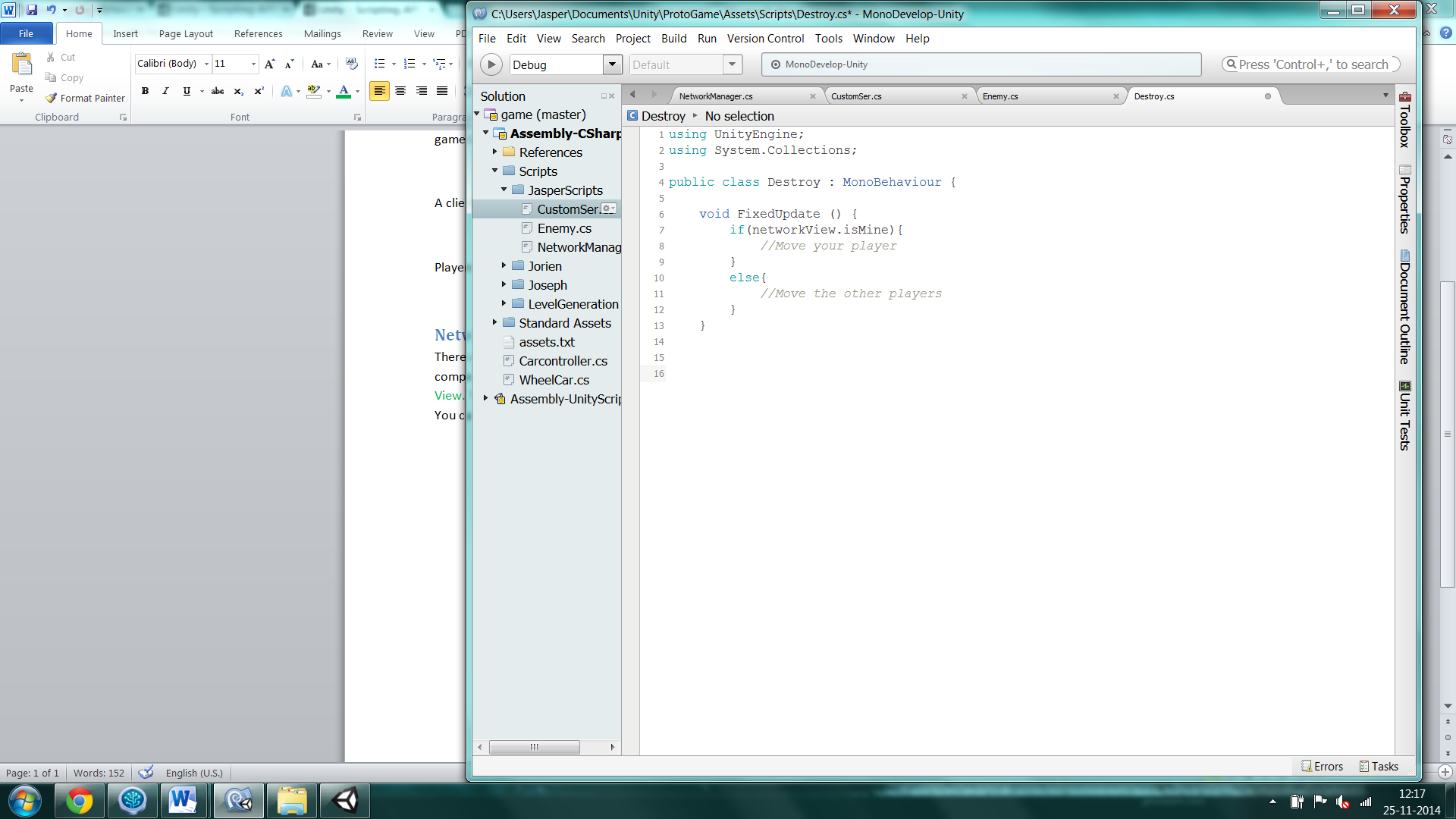
A client can connect by clicking a join button that calls the function

Network.Connect (string IP, int remotePort);

Players can be spawned using the messages OnServerInitialized() and OnConnectedToServer().

## Network View

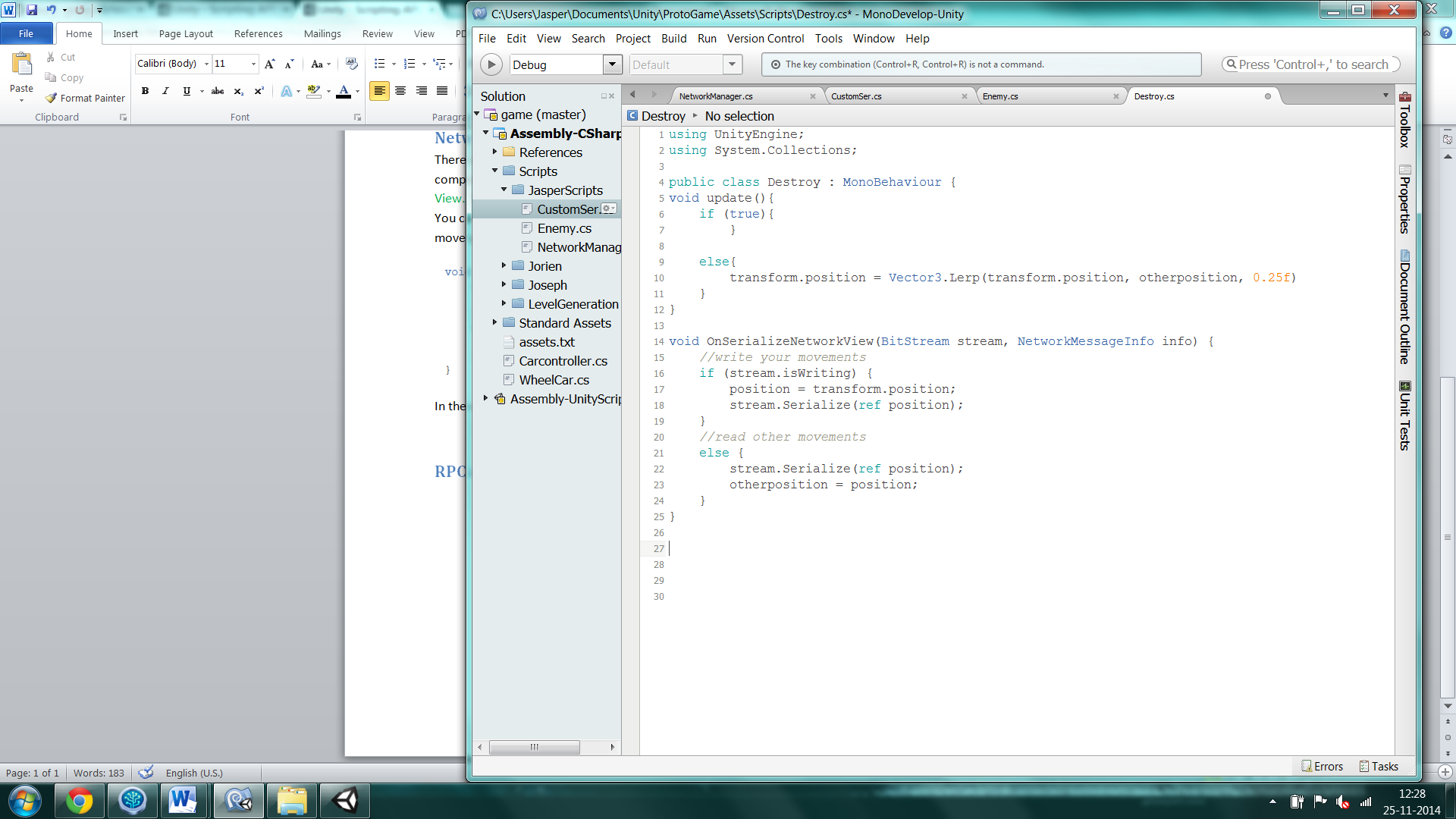
There are two ways to send data to all players the first one is Network View. Network View is a component you need to add to a game object using Add Component > Miscellaneous > Network View. The Network View observes a variable but can also observe a script which is what we need. You can use Network View to observe to movement of players. Using networkView.isMine you can move your own player while not moving the other players (see example).



In the else function you can use Vector3.Lerp() to make the other players movement smooth. To get the other player position you need to add the function

OnSerializeNetworkView(Bitstream stream, NetworkMessageInfo info)

(see example)



## RPC calls

The second way to send data is using RPC calls, using RPCs you can call a function on all players (host and clients). In order to use RPCs the game object needs a Network View but this Network View can still be used to observe something else. You can easily call a RPC using networkView.RPC("function name", RPCMode, parameters) but you need to write [RPC] above the function you want to call (see example).

