**Getting the Connection**

By Florin, Wilson, Zach

1. Make a new project call it NetworkConnection

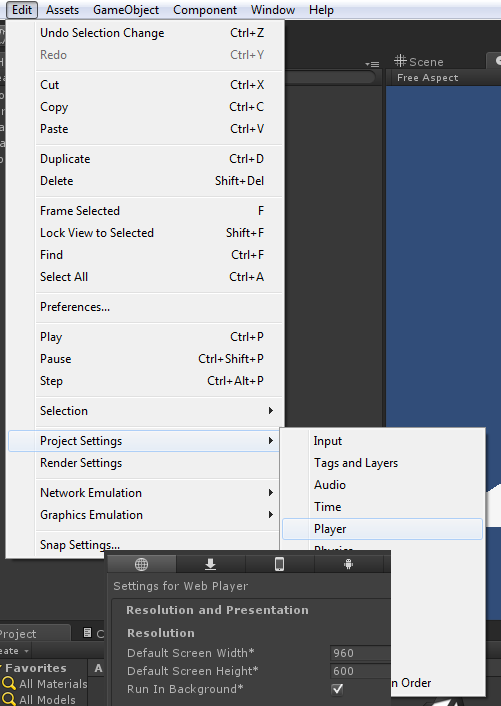
2. Go to Edit then Project Settings and click on Player

3. Click on the First tab (Global) in the Player Settings

4. Make sure Run in Background is checked

5. Click on the Second tab (Arrow) in the Player Settings

6. Make sure Run in Background is checked



7. Create a plane and add a Directional light

8. Create another Empty Game object and call it Spawner (This is where the player will spawn in when they connect)

9. Have the Y position of the spawner .5 higher than the plane Y position

10. Create and Java Script(JS) call it SpawnerJS and put it on the Empty Game object called Spawner

#pragma strict

public var playerPrefab : Transform;

function OnServerInitialized(){

Spawnplayer();

}

function OnConnectedToServer(){

Spawnplayer();

}

function Spawnplayer(){

//This Function Spawns player Obj.

var myNewTrans : Transform = Network.Instantiate(playerPrefab, transform.position, transform.rotation, 0);

//Spawns player where the Empty Game object called Spawner is

}

function OnPlayerDisconnected(player: NetworkPlayer) {

Debug.Log("Clean up after player " + player);

Network.RemoveRPCs(player);

Network.DestroyPlayerObjects(player);

}

//This removes player when they disconnect

function OnDisconnectedFromServer(info : NetworkDisconnection) {

Debug.Log("Clean up a bit after server quit");

Network.RemoveRPCs(Network.player);

Network.DestroyPlayerObjects(Network.player);

Application.LoadLevel(Application.loadedLevel);

}

11. Make an Empty Game object and call it Connect (this will hold the Connect Script)

12. Create and Java Script(JS) call it ConnectJS and put it on the Empty Game object called Connect

//Default Ip and Port

var connectToIP : String = "127.0.0.1";

var connectPort : int = 25001;

function OnGUI ()

//This is the GUI to start server or to connect to a server as a client

{

if (Network.peerType == NetworkPeerType.Disconnected){

GUILayout.Label("Connection status: Disconnected");

connectToIP = GUILayout.TextField(connectToIP, GUILayout.MinWidth(100));

connectPort = parseInt(GUILayout.TextField(connectPort.ToString()));

GUILayout.BeginVertical();

if (GUILayout.Button ("Connect as client"))

{

//Network.useNat = false;

Network.Connect(connectToIP, connectPort);

}

if (GUILayout.Button ("Start Server"))

{

//Network.useNat = false;

Network.InitializeServer(32, connectPort, false);

}

GUILayout.EndVertical();

}else{

if (Network.peerType == NetworkPeerType.Connecting){

GUILayout.Label("Connection status: Connecting");

} else if (Network.peerType == NetworkPeerType.Client){

GUILayout.Label("Connection status: Client!");

GUILayout.Label("Ping to server: "+Network.GetAveragePing( Network.connections[0] ) );

//Tells player if they are Client or Server

} else if (Network.peerType == NetworkPeerType.Server){

GUILayout.Label("Connection status: Server!");

GUILayout.Label("Connections: "+Network.connections.length);

if(Network.connections.length>=1){

GUILayout.Label("Ping to first player: "+Network.GetAveragePing( Network.connections[0] ) );

}

}

if (GUILayout.Button ("Disconnect"))

{

//Disconnect Button

Network.Disconnect(200);

}

}

}

13. Create a new script call it PlayerJS

#pragma strict

#pragma implicit

#pragma downcast

var speed : float = 5;

function Awake(){

if(!networkView.isMine){

enabled=false;

}

}

function Update(){

if(networkView.isMine){

InputMovement();

}

}

function InputMovement()

{

if (Input.GetKey(KeyCode.W))

rigidbody.MovePosition(rigidbody.position + Vector3.forward \* speed \* Time.deltaTime);

if (Input.GetKey(KeyCode.S))

rigidbody.MovePosition(rigidbody.position - Vector3.forward \* speed \* Time.deltaTime);

if (Input.GetKey(KeyCode.D))

rigidbody.MovePosition(rigidbody.position + Vector3.right \* speed \* Time.deltaTime);

if (Input.GetKey(KeyCode.A))

rigidbody.MovePosition(rigidbody.position - Vector3.right \* speed \* Time.deltaTime);

}

function OnSerializeNetworkView(stream : BitStream, info : NetworkMessageInfo)

{

if (stream.isWriting){

var pos : Vector3 = transform.position;

stream.Serialize(pos);

}else{

var posReceive : Vector3 = Vector3.zero;

stream.Serialize(posReceive);

transform.position = Vector3.Lerp(transform.position, posReceive,1);

}

}

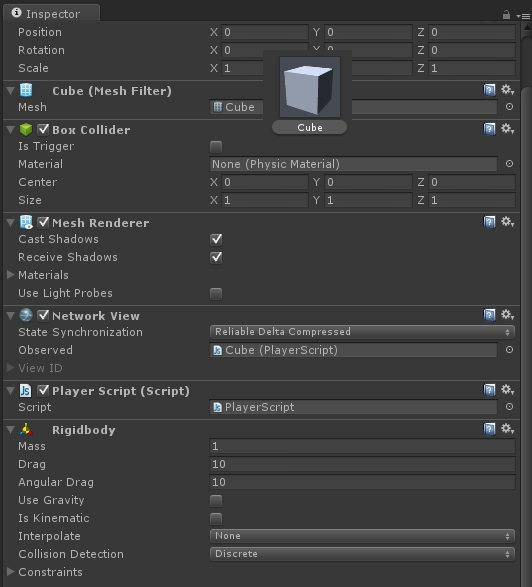
14. Create a Cube

15. Add a Network View, Rigidbody and the PlayerJS to the cube

16. Make the Cube a prefab

17. Drag the Player Script into the observed field of the Network View (I will explain this part)

(add cube prefab into spawn script)



18. Create another JavaScript called IPshower(Display IP)

19. Put the IPshower in the Empty Game object called Connect

#pragma strict

#pragma implicit

#pragma downcast

import System.Net;

function OnGUI () {

var strHostName:String = "";

strHostName = System.Net.Dns.GetHostName();

var ipEntry:IPHostEntry = System.Net.Dns.GetHostEntry(strHostName);

var addr:IPAddress[] = ipEntry.AddressList;

GUILayout.BeginArea(Rect(Screen.width/2+100,-250,400,Screen.height));

GUILayout.FlexibleSpace();

GUILayout.BeginHorizontal();

GUILayout.FlexibleSpace();

GUILayout.Label("Your IP:" +addr[addr.Length-1].ToString());

GUILayout.FlexibleSpace();

GUILayout.EndHorizontal();

GUILayout.Space(10);

GUILayout.BeginHorizontal();

GUILayout.BeginVertical();

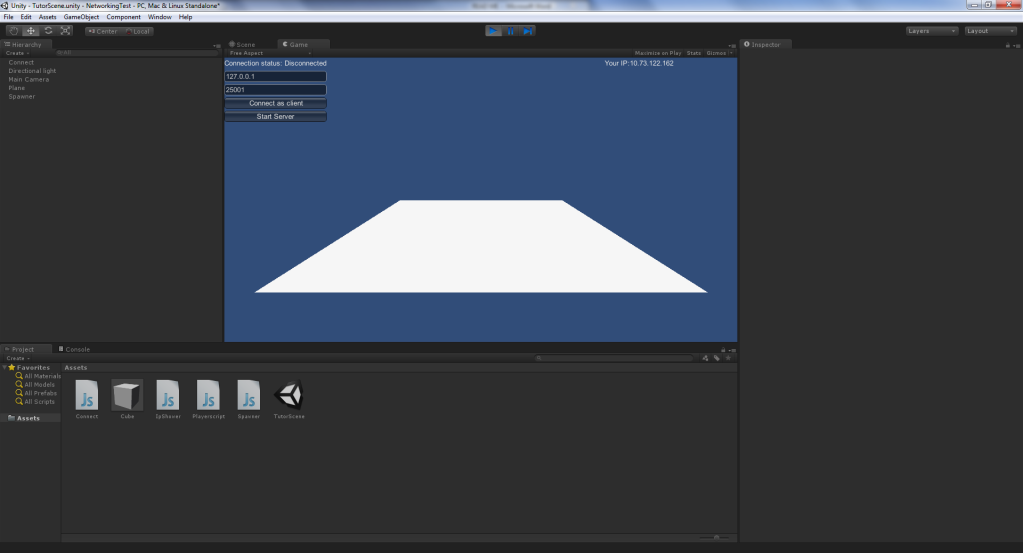
GUILayout.EndVertical();

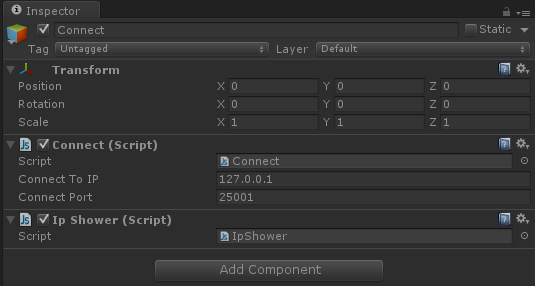
GUILayout.EndHorizontal();

GUILayout.FlexibleSpace();

GUILayout.EndArea();

}





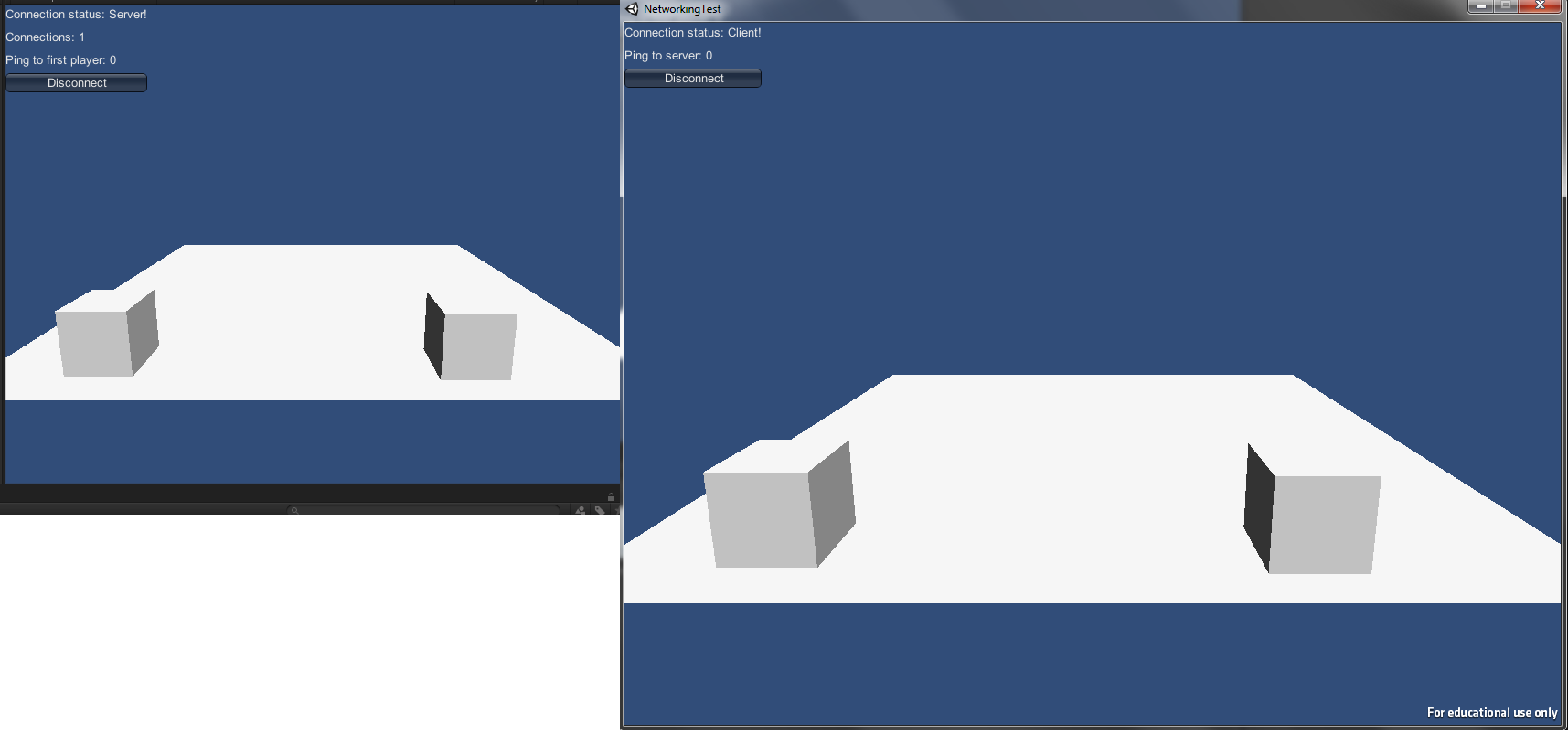
20. Now test the game by building an Exe (Ctrl B)

21. Make sure the scene looks exactly like the image at the end of page 9, there should not be a cube in the scene from the start, so delete the one you made earlier.

22. Run the Editor and run the Exe of the project

23. Make sure the server and the client are set to your machine’s IP address.

24. Use the Editor as the Host and connect with the Exe



Final Notes: An example project is provided with this tutorial, the javascript version of the “Playerscript” file is the one described in this tutorial. The c# version is a more advanced script that will be gone over in the following tutorial.