1. Connecting Unity Clients to a Dedicated Server | C# Networking Tutorial Part 1

Open up Visual Studio 2019 Select Console App For Project Tempelates And C#, Give it a name and save location. Let’s name it “GameServer”

In Program.cs (by right clicking you can create c# classes within the namespace GameServer. To right on visual studio)

Type this

using System;

namespace GameServer

{

class Program

{

static void Main(string[] args)

{

Console.Title = “Game Server”;

Console ReadKey();

}

}

}

Now create a class by right clicking on gameserver to the right in the top of the list. Name it “Server”

Type this

using System;

using System.Collections.Generic;

using System.Text;

using System.Net;

using System.Net.Sockets;

namespace GameServer

{

class Server

{

public static int MaxPlayers {get; private set;}

public static int Port { get; private set;}

public static Dictionary<int, Client> clients = new Dictionary<int, Client>);

private static TcpListener tcpListener;

public static void Start(int \_maxPlayers, int \_port)

{

MaxPlayers = \_maxPlayers;

Port = \_port

Console writeLine(“Starting server …” );

InitializeServerData();

tcpListener = new TcpListener(IPAddress.Any, Port);

tcpListener.Start();

tcpListener.BeginAcceptTcpClient(new AsyncCallback(TCPConnectCallback), null);

Console.WriteLine($“Server started on (Port)…”);

}

Private static void TCPConnectCallback(IAsyncResult \_result)

{

TcpClient \_client = tcpListener.EndAcceptTcpClient(\_result);

tcpListener.BeginAcceptTcpClient(new AsyncCallback(TCPConnectCallback), null);

Console.WriteLine($“Incoming connection from (\_client.Client.RemoteEndPoint)…”);

for (int i = 1; i <= MaxPlayers, i++)

{

if (clients[i].tcp.socket == null)

{

clients[i].tcp.Connect(\_client);

return;

}

}

Console.WriteLine($“(\_client.Client.RemoteEndPoint) failed to connect: Server Full!”)

}

private static void InitializeSeverData()

{

for(int i = I; I <= MaxPlayers, i++)

{

clients.Add(I, new Client(i));

}

}

}

}

Finally we are going to need to control the data flow from the client to your unity connection places. Right Click on GameServer on the right once again and then add a class, Call it “Client”.

Here is the data you need to copy to your visual studio.

using System;

using System.Collections.Generic;

using System.Text;

using System.Net;

using System.Net.Sockets;

namespace GameServer

{

class Client

{

public int id;

public TCP tcp;

public static int dataBufferSize = 4096;

public Client(int \_clientid)

{

id = \_clientid;

tcp = new TCP(id);

}

public class TCP

{

public TcpClient socket;

private readonly int id;

private NetworkStream stream;

private byte[] recieveBuffer;

public TCP(int \_id)

{

id = \_id;

}

public void Connect(TcpClient \_socket)

{

socket = \_socket;

socket.RecieveBufferSize = dataBufferSize;

socket.SendBufferSize = dataBufferSize;

stream = socket.GetStream();

recieveBuffer = new byte(dataBufferSize);

stream.BeginRead(recieveBuffer, 0, dataBufferSize, RecieveCallback, null);

// TODO : send welcome packet

}

Private void RecieveCallback(IAsyncReseult \_result)

{

try

{

int \_byteLength = stream.EndRead(\_results);

if (\_byteLength <= 0)

{

//TODO disconnect

return;

}

byte[] \_data = new byte(\_byteLength);

Array.Copy(recieveBuffer, \_data, \_byteLength);

//TODO handle data

stream.BeginRead(recieveBuffer, 0, dataBufferSize, RecieveCallback, null);

}

catch(Exception \_ex)

{

Console.writeLog(“Error receiving TCP data (\_ex)”);

After these steps comes the final one.. Connecting to the port and setting the Max Players Per the Server you have just created.

To Do This is Very important by passing it in the Program c# script. The script will be copied, but to look for the list of TCP and UDP port number, you must search Wikpedia for “List of TCP and UDP Port numbers”, you must past this port number as the server number in the new line of code. Here is an example

using System;

namespace GameServer

{

class Program

{

static void Main(string[] args)

{

Console.Title = “Game Server”;

// new line of code for first parameter as max player and second as port number

Server.Start(50, 26950);

Console ReadKey();

}

}

}

Chose to allow it through your firewall.

1. Now for the Client That ports to the server. We will be using Unity for this due to its mapability to other ports.