Unity’s Multiplayer library (**UNET**) is the **completely free to use networking library** which is used for the Unity Game Engine. So, we will be using this library to build the multiplayer logic and element of our game.

Note: ( UNET is slowly being phased out and replaced (we should have our project finished before then) so things may potentially be buggy and slow) Source for additional information – (<https://support.unity3d.com/hc/en-us/articles/360001252086-UNet-Deprecation-FAQ>)

**Multiplayer Solutions/ Proposed Choices**

**Unity Services is** the additional optional framework of services Unity provide onto of the UNET library, which provide their own services, hosting CCU, Matchmaking, Relay servers etc. Which avoids the need for a dedicated server.

I see conflicting and information on whether the 20 players limit for Unity Personal, is free to use, but in conclusion, I see it’s free to use in the “development stages” but if you want to get players outside or/and it’s considered “ready to release”. We would have to pay for the bandwidth we use. This may or not pose a problem, since this is a strictly educational project. Please see links (<https://answers.unity.com/questions/1396250/do-i-have-to-pay-for-unity-multiplayer-if-im-just.html>) & (<https://unity3d.com/unity/features/multiplayer>) & (<https://forum.unity.com/threads/unity-multiplayer-launch-faq-march-2016.392474/>) for additional information.

If we don’t want to use Unity’s matchmaking services, (We want to write a LAN or Peer 2 peer solution etc) we would have to write our own networking logic, port forwarding, etc. ( But to our advantage there are many free libraries and open source libraries on the Unity Asset store as well as tutorials that can help us with this.

A paragraph I found directly sums up the difference between UNET & Unity Services

“With that being said, you **DO NOT** need to use Unity Services to make a multiplayer game in UNET, and you can host your own dedicated servers through Amazon etc. or even host your own servers at home for your game and never have to pay any sort of CCU/Subscription cost/etc.  
However, if you decide to host your own servers, you will not have some nice features that come with Unity Services such as matchmaking, relay servers, etc. which you would have to figure out yourself (Which is not overly difficult, and several Unity Asset Store packs to help with this) if you required them and were not hosting through Unity.” - Source (<https://forum.unity.com/threads/can-we-please-clear-this-confusion-up-unet.503219/>)

There are options of a Peer 2 Peer & LAN network using UNET with tutorials . Apparently for a peer2peer network, all you need is the player’s IP addressed to be entered. (<https://answers.unity.com/questions/1501868/making-own-server-instead-of-using-unity-matchmaki.html>)

“The specific Unity feature that allows you to facilitate lan connections is UnityEngine.Networking.NetworkDiscovery. It is completely independent from both the HLAPI (UNET) and the LLAPI (transport-layer), so it can be used with any networking library you decide to use.

Unity also published an example project for using NetworkDiscovery: https://forum.unity3d.com/threads/networkdiscovery-sample.354928/

Unity's networking library will also work without a central server (for both the HLAPI (UNET) and LLAPI (transport layer) versions), but you need to do connection creation yourself.” - (<https://www.reddit.com/r/Unity3D/comments/5adedl/how_can_i_start_making_mobile_peertopeer/>)

Things need to set up Multiplayer in a Unity Game - Source https://docs.unity3d.com/Manual/UNetGettingStarted.html)

* A **Network Manager**
* A **user interface** (for players to find and join games)
* Networked **Player Prefabs** (for players to control)
* **Scripts**  
  and **GameObjects**  
  which are **multiplayer-aware**

In summary, I suggest we implement a LAN network built on top of UNET, using free libraries and tutorials we can find. Which I believe offers the best trade-offs.

Extra Information

A very good summary, on the different options of multiplayer

**Peer-to-Peer**

Players each share equal authority and networking tasks; no player is a "host". Generally used in simpler non-synchronous games.

**Client-Server with Client as Host: (Not to be confused with Peer2Peer)**

This is the way the standard Unity Networking system (UNet) operates, though it is also possible to run a Unity standalone player in headless mode on a dedicated server and use UNet.

Note: I confused this with Peer-to-Peer networking for a long time, which led to many cases of confusion.

* No dedicated server required, so significantly cheaper
* Able to play via LAN, without connection to internet
* Harder to prevent hacking/cheating
* Not HTML5 (WebGL) compatible, since a WebGL game cannot be a host
* Must be more meticulous when testing, since host-client operates slightly differently from other clients (e.g. reduced latency for host)

**Client-Server with Dedicated Server:**

* Expensive to maintain (depending largely on scale; in my specific example, based on Unity's cost estimate—which is based on expected number of users, average message size, and messages per second—anywhere from $50 - 500 per month)
* Must be connected to internet
* Easier to prevent hacking/cheating (since everything must go through the non-client dedicated server)
* HTML5 (WebGL) compatible, since the host is the non-WebGL dedicated server (though certain other missing WebGL components, like microphone use, may make this a moot point for my specific game)
* Testing is more equivalent for all players since all players interact with the server equally

**Unity Multiplayer** (https://unity3d.com/services/multiplayer) : The standard to use with Unity is Client-Server with a player as the Server. While this is relatively new, I've gotten it to work, and it's relatively well-documented. The Unity Matchmaking Service (which is free) is likely the best way to go about finding players to play with. This can also be set up with a dedicated server (as mentioned above, by running a Unity standalone player in headless mode on a dedicated server and using UNet).

**Amazon Web Services** (https://aws.amazon.com/documentation/sdk-for-unity/) :

**Photon Networking (PUN**) (http://u3d.as/2ey) : Seems to be easiest, most expandable, and well-documented. Uses Client-Server model with dedicated server. The Photon Cloud can be used (free version includes 20 CCU), however, code cannot be executed on the server, so hacking a game using this wouldn't be too difficult. Photon can also be used with your own dedicated server that you could, like with Unity Multiplayer, have Unity playing on, which would prevent issues with hacking.

Resources & Links for Additional Information

https://www.reddit.com/r/Unity3D/comments/5adedl/how\_can\_i\_start\_making\_mobile\_peertopeer/

<https://www.reddit.com/r/Unity3D/comments/8he04r/peer_2_peer_multiplayer/>

<https://stackoverflow.com/questions/36091976/can-i-use-the-unity-networking-hlapi-without-paying-for-the-unity-multiplayer-se>

<https://forum.unity.com/threads/lan-with-unet.346182/>

<https://docs.unity3d.com/Manual/UNetGettingStarted.html>

https://docs.unity3d.com/Manual/UNetConverting.html

<https://docs.unity3d.com/Manual/UNetConcepts.html>

https://docs.unity3d.com/Manual/UNetOverview.html

https://docs.unity3d.com/Manual/UNetManagerHUDLanMode.html

<https://docs.unity3d.com/Manual/UNetUsingTransport.html> - Peer2Peer

<https://docs.unity3d.com/Manual/UNetManager.html>

<https://stackoverflow.com/questions/53819720/unity-alternative-to-unet-for-p2p-server>