**Just add multiplayer**

While working on *Champions of Aerial,* I took on the task of researching if it would be possible to add online multiplayer to the game. Online multiplayer is not something that we had planned to do when we started this whole project, but it’s a feature that playtesters have requested we should add. No one have specifically said “You **must** include online multiplayer” and of course we don’t absolutely **have** to add it, but I will admit that it would have been a sweet addition to the game.

I find it interesting whenever someone use the word “just” when they talk about this feature. When someone makes the suggestion “Just add multiplayer to your game” or in the form of a question “Why don’t you just add multiplayer to your game?” the word “just” not only implies that the suggestion is a really obvious thing we should already have thought of, but also that the suggestion is a really easy thing to do. Actually, it turns out that adding online multiplayer to a game is not as easy as people might think.

To add online multiplayer to a game you first need to implement the server that all the players in the same game can connect to. That could be a *Dedicated server* if you have lots and lots of money, or it could be a *Hosting* type of connection where every player in the same game connects to one of the players’ computer, which in turn would act as the server. Once you have successfully built a system that acts as the server for the game, you need to implement clients that each computer use to communicate to the server. Clients are used for sending a package of information from the inputs that you do on your computer to the computer who acts as the server (or the separate server if using *Dedicated servers*), which then updates the game and sends packages of updates to every other client, telling them how your inputs have changed the game. This is constantly happening from every client that is connected to that server. That by itself would not be impossible to build by yourself, but now we are getting to the fun parts.

Sending a package of information from one computer to another over the internet isn’t instantaneous. It will take perhaps a split part of a second. While playing offline you expect a game to update itself at a good 60 time every second (which is more commonly known as frames-per-second, or fps) but most people can accept 30. Anything below 30 make the game look and feel choppy. Now imagine having clients that have to send packages of new inputs to a server, which takes those inputs and updates the game, which then sends back new packages consisting of the updated game to the clients so the client on your computer can finally update your game. Now is a good time to point out that the server and the client could exist in different parts of the planet. Normally a server-to-client communication can handle around 9 updates every second, which would mean that games played online would be very choppy. But 9 is a lot less than 30 and the games we play online doesn’t look choppy, how is this achieved? Well, let me tell you.

When the client on your computer sends off a package of new inputs to the server, the client performs a bunch of calculations and tries to predict how the update will look like. So before the client gets a response from the server, your game can perform an update based on your clients prediction. Once the client receives the actual update from the server, it simply implement the corrections. This makes your game able to perform 30 to 60 updates every second and in turn look a lot smoother, even though your client in reality only receives around 9 updates every second. The better predictions your client can calculate, the less corrections are needed when the update from the server are received. But if you have built a poor client, those predictions will cause your updates to jump players around all over the place.

All of the server-to-client communication can of course only occur if the server and the clients know where to send the packages. The clients need to know where the server exist, and the server need to know where the clients exist. In most online games where a *Hosting* multiplayer exist, a sort of *lobby room* takes care of establishing these connections. You as a player creates a *room* where other players can join your game (your server), and you can find other *rooms* from your game to join. But this *lobby room* actually exist on another server, a *Dedicated server* that the game in advance have been built to know how to connect to. So, if you don’t want to pay for this *lobby room* to exist on a *Dedicated server* where players can easily establish these connections with each other, you would have to create some other way for the server and the clients to find one another or have the player type in the IP-addresses of other players manually.

So unfortunately *Champions of Aerial* will not have a built-in online multiplayer, and whenever someone asks “Why don’t you just add online multiplayer to your game?” I can now explain why that would be a difficult thing for us to do.

*Robin*