The Unity asset contains the necessary components to develop communication with the OWO vest.

It contains the main script **OWOUDPClient.cs** which is necessary. There are two sample scenes: They are inside the demos folder, one is an example of how to properly connect with OWO App, the other is an example of how to send sensations to the haptic jacket.

To use the connection and access the vest you only need the script mentioned above **OWOUDPClient.cs**. It is a script that we place in a gameobject of the Unity scene, that script contains the necessary to connect with OWO and send sensations.

How to use OWO asset

1. Invoke the **Connect(string _ip)** function (OWOUDPClient.cs): If an error occurs, the action **OnConnectionFailed** will be called and if you connect to the application, **OnConnected** will be called.

2. Once we are connected to the server if we want to send a sensation to the vest we must use the OWOUDPClient.cs **SendSensation()** function which contains two parameters:
    a. idSensation, the numerical identifier of the sensation.
    b. idMuscle, the numerical identifier of the muscle.

3. If you are not connected to the server, the action **OnConnectionFailed** will be called.

Important: There are sensations that are not defined for certain muscles in this case the application ignores the request and the user receives nothing.

Note: OnClientFailedToConnect and OnClientStarted actions are not delegated as the asset lets the developer do whatever they see fit on its call.

SensationsID and MusclesID:
https://docs.google.com/document/d/1NOCX77HHMmR00X4oAhb3PN6dBTypEb9IM0PoVVomqkY/edit?usp=sharing

Integration Example:

Here there is a video explaining how to easy implement OWO into a shooter project:

OWO integration