

Unity OWO Asset

The Unity Asset contains the necessary components to develop communication between the game and the OWO vest.

How to use the OWO SDK

1. Connection

The “OWOUDPClient” class is responsible for establishing a connection between the game and the OWO App. To use it, it needs to be built by another class.

There are two ways to establish the connection with the App when using this class:

- **Automatic IP:** Through the “FindServersInLANAndConnect” function, the “DeviceFinder” class will search the OWO App through the local network and it will establish a connection. This may not work if there is an App like OpenVpn opened or if the network does not allow this type of connection (broadcast).
- **Manual IP:** The user must enter the IP that the OWO App provides you when you press on the "Play" mode. To establish this connection use the “Connect” function of “OWOUDPClient” class.

If an error occurs during the connection, the “OnConnectionFailed” action will be called, and if the connection is successful, the “OnConnected” action will be called.

2. Send sensations

Once the connection has been made, sensations can be sent at any time through the “SendSensation” function.

To send a sensation it's necessary to specify two values:

1. The ID of the sensation.
2. The muscle in which you want to feel the sensation.

If the sensation is successfully sent, the “OnSensationSend” action will be called.

Sensations sorted by ID

Important: There are sensations that are not defined for certain muscles, in this case the App ignores the request and the user does not receive the sensation.

OWO allows the user to feel the same sensation on different muscles, individually. These sensations are:

ID	Name	Muscles affected
0	Ball	Any
1	Dart	Any (except left lumbar, right lumbar)
2	Dagger wound	Left abdominal, right abdominal
6	Axe	Left pectoral, right pectoral, left dorsal, right dorsal
7	Punch	Any (except left lumbar, right lumbar)
8	Grip	Left arm, right arm
9	Shot	Any (except left lumbar, right lumbar)
28	Gun recoil	Left arm, right arm
29	Heartbeat	Left pectoral

There are few sensations where only a specific group of muscles are affected. Depending on the target muscle, the user will feel it in one part of the body or another. The list of these sensations is:

ID	Name	Muscles affected
4	Shot with exit wound	<ul style="list-style-type: none"> • Left side when selected a left muscle • Right side when selected a right muscle
10	Insects	<ul style="list-style-type: none"> • Front ones when selected are a pectoral, abdominal or arm. • Back ones when selected a dorsal or lumbar

On the other hand, there are some sensations that can be felt on several muscles at the same time. Those sensations are:

ID	Name	Muscles affected
11	Free fall	left arm, right arm, left pectoral, right pectoral, left abdominal, right abdominal
12	Lift object	left arm, right arm
13	Lift heavy object	left arm, right arm, left abdominal, right abdominal, left lumbar, right lumbar
14	Fast driving	Left arm, right arm, left pectoral, right pectoral
15	Idle speed	Left arm, right arm, left abdominal, right abdominal, left dorsal, right dorsal
16	Insect bites	Left pectoral, right pectoral
17	Machine gun recoil	Left arm, right arm, left pectoral, right pectoral
18	Push heavy object	Left arm, right arm
19	Push object	Left arm, right arm
20	Abdominal wound	Left abdominal, right abdominal
21	Chest wound	Left pectoral, right pectoral
22	Oppression	Left dorsal, right dorsal
23	Strange presence	Left dorsal, right dorsal, left lumbar, right lumbar

Stop sensation

If you want to stop a sensation that is currently playing, you can send the sensation -1 to stop it.

ID	Name	Muscles affected
-1	Stop Sensation	Any

Muscles

The sensations affect different muscles. List of muscles sorted by ID.

ID	Muscle
0	Pectoral Right
1	Pectoral Left
2	Abdominal Right
3	Abdominal Left
4	Arm Right
5	Arm Left
6	Dorsal Right
7	Dorsal Left
8	Lumbar Right
9	Lumbar Left

Example

Here is a quick example of how OWOUDPCClient.cs could be implemented:

```
using UnityEngine;
using OWO;

Script de Unity | 1 referencia
public class OWOController : MonoBehaviour
{
    public static OWOController instance;

    private OWOCient owoClient;

    Mensaje de Unity | 0 referencias
    private void Awake()
    {
        if (instance == null) instance = this;

        owoClient = new OWOUDPClient();
    }

    /// <summary>
    /// Method called when the user introduces the ip via InputField.
    /// </summary>
    /// <param name="_ip"></param>
    0 referencias
    public void ConnectToOWO(string _ip)
    {
        owoClient.Connect(_ip);
    }

    /// <summary>
    /// Method called to connect automatically to a OWO App.
    /// </summary>
    0 referencias
    public void AutoConnect()
    {
        owoClient.FindServersInLANAndConnect();
    }

    /// <summary>
    /// Sends a sensation to the OWO vest.
    /// </summary>
    /// <param name="_sensationID"></param>
    /// <param name="_muscle"></param>
    0 referencias
    public void SendSensation(ushort _sensationID, OWOMuscle _muscle)
    {
        owoClient.SendSensation(_sensationID, _muscle);
    }
}
```