BIOPAC Systems, Inc.

PRODUCT SHEET

info@biopac.com
support@biopac.com
www.biopac.com

UNITY INTERFACE FOR ACQ*KNOWLEDGE*

Unity® Interface for Acq*Knowledge*® allows you to easily to connect your Unity3D projects with BIOPAC acquisition hardware and analysis software.

- Create your virtual environment using industry-standard Unity.
- Connect and configure your project with Acq*Knowledge* in real time.
- Control Acquisition from Unity to Custom Markers, Digital, and Analog I/O.
- Deploy to your devices. Immerse your users and capture biofeedback for analysis.
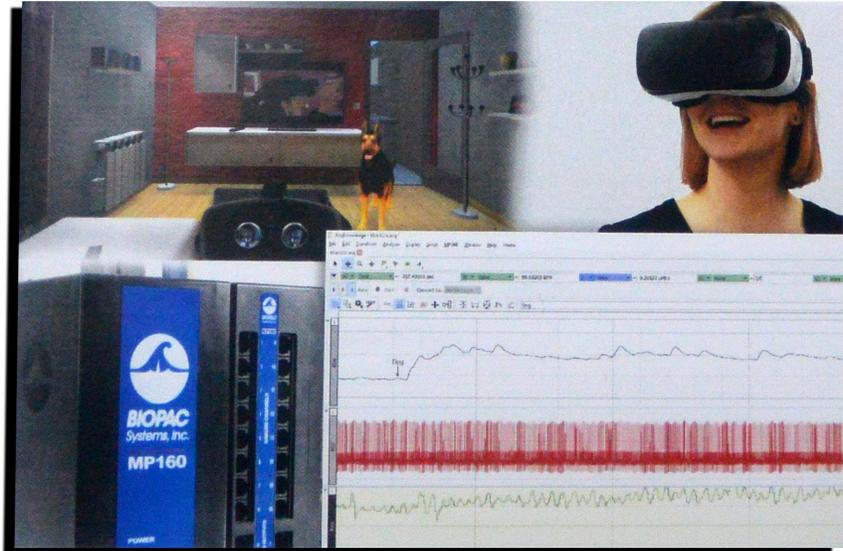
This Unity3D package allows Unity developers to access the Acq*Knowledge* network API (Network Data Transfer) v5.0 from Unity scripts without having to deal with the details of network communication, threading, and data access.

The Unity Interface for Acq*Knowledge* is compatible with recent versions of Windows and macOS.

This interface utility from The ReviaÎ Assets does not include Unity Software, Acq*Knowledge* software, or Acq*Knowledge* Network Data Transfer (NDT) license.

Unity Interface works on recent versions of Windows and MacOS and all recent version of Unity (5.5+).

Make sure the network connection is activated in Acq*Knowledge* with NDT (Network Data Transfer license).

Start Acq*Knowledge* with a small setup of digital and analog I/O you can find in the samples folder.

Watch the Unity VR Interface Demo Video for an overview of hardware setup and interface information.

SAMPLE USES

- **Buttons** | Create a small script with buttons to start, stop, or toggle the acquisition from within Unity; use the connector Component.
- **Marker events** | Again, we will write a few line of script using GenericEventSender and its õInsertGlobalEventö function. The script can be configured with default arguments set from the Unity Inspector, or use some custom values directly set from script. Here for example, we can trigger the rock fall, and send an event at the same time. You can also send other event with script-defined values.
- **Analog & Digital inputs** | Use InputChannel component, with the GetValueOrDefault function. The channels are preconfigured in the Unity inspector window. Here you can see that it will read the voltage of the battery, and detect the button press. For example, having 3.2 V from the batteryí the DIGITAL value goes from 0 to 5 Ví 0 to 5 V againí
- **Outputs** | Use OutputChannel component to set some data from the script, here for example, we will generate a Sine curve to send some data to Acq*Knowledge*. Here again, the Unity inspector will be used to configure the Output channel number. When we press õplayö, we can see the sine function, and the LED indicator
- **Digital Output** | This is the same with boolean values. Toggle the output with a GUI buttoní see it on the Acq*Knowledge* graph and the LED goes off, on, off again.