

API Module

Output / Input / Loop & Remote Control

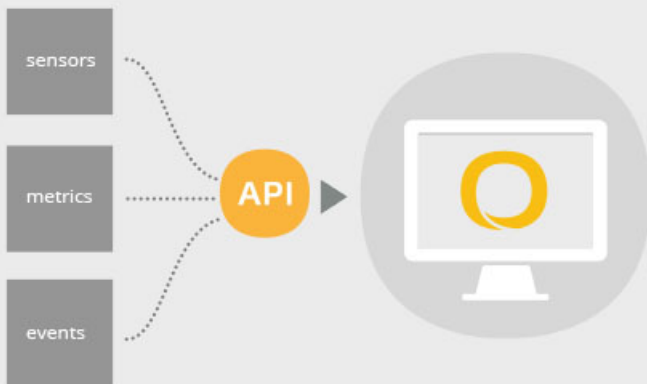
Pre-requisite: iMotions Core License



 **IMOTIONS®**

sales@imotions.com

Build & integrate anything with iMotions open API



API Input - Receive & synchronize in real time any external data stream into iMotions

The API input allows to connect and stream from a 3rd party software / hardware and receive data into iMotions. This data will be treated as any other data natively collected within iMotions & synchronized into the mix.

API Input (Receive) Example

A group of researchers at Stanford University wanted to use eye tracking to investigate driving while talking on cell phones. A car simulator was programmed to expose the drivers to certain events during a course. iMotions software recorded eye tracking data while receiving from the driving simulator data such as: Events, Throttle, Speed, Steer and Brake. This allowed the researchers to immediately correlate visual attention with essential driving parameters - both live and right after the test ended.



[Click here to see the use case](#)



API Input (Receive) Example

This gaming example shows the reception of 2 Unity events into iMotions mix. These events are treated in the same fashion and displayed in combination with the eye tracking and facial expressions biometric signals gathered through out the gaming session.

Shooter Game Update() method simple modification

```
void Update ()
{
    // Add the time since Update was last called to the timer.
    timer += Time.deltaTime;

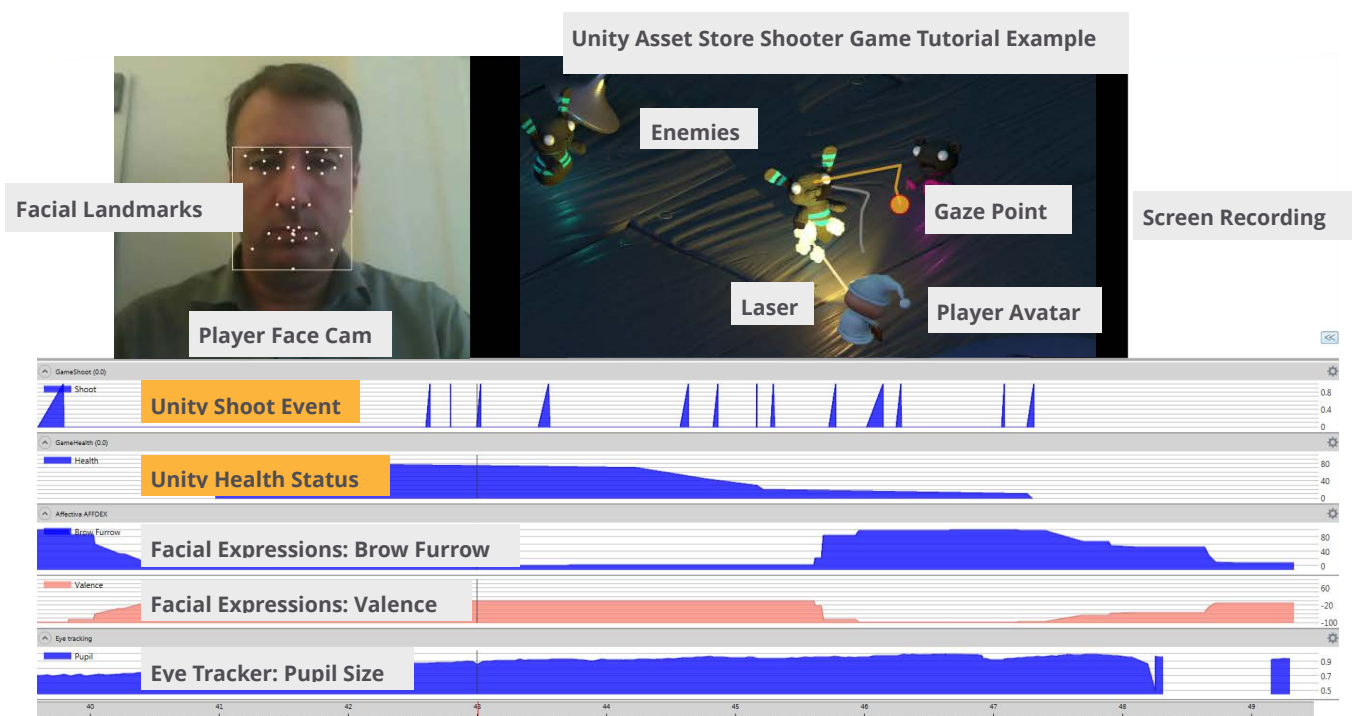
    // If the Fire1 button is being press and it's time to fire...
    if (Input.GetButton ("Fire1") && timer >= timeBetweenBullets && Time.timeScale != 0)
    {
        // Send Shoot Trigger to IMOTIONS ("1.0")
        string shooting = "E;1;GameHealth;1;0.0;;;GameShoot;" + "1.0" + "\r\n";
        SendUDPPacket("127.0.0.1", 8089, shooting, 1);

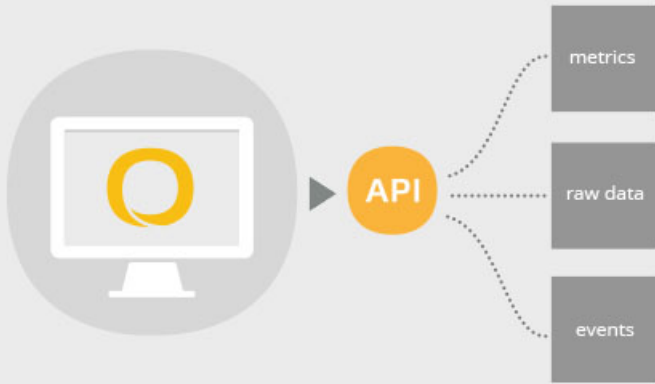
        // ... shoot the gun.
        Shoot();
    }

    // If no shooting just send 0s continuously
    string notshooting = "E;1;GameHealth;1;0.0;;;GameShoot;" + "0.0" + "\r\n";
    SendUDPPacket("127.0.0.1", 8089, notshooting, 1);

    // If the timer has exceeded the proportion of timeBetweenBullets
    // that the effects should be displayed for...
    if (timer >= timeBetweenBullets * effectsDisplayTime)
    {
        // ... disable the effects.
        DisableEffects ();
    }
}
```

Add two lines of code to transfer Unity events to iMotions input API





API Output - Forward all synchronized collected data out of iMotions in real time

The API output allows to live export events, metrics, raw data, markers and triggers from iMotions to 3rd party software / hardware.

API Output (Forward) Examples

In this example a Chrome browser extension app was built. The app shows a schematic of facial muscles. All of the 19 Action Units provided by the Emotient facial expression analysis engine are indicated with a probability measure.

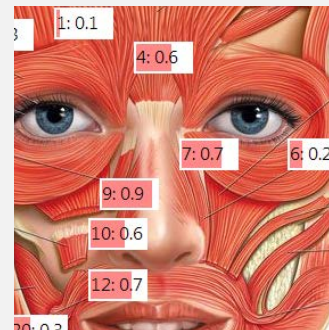
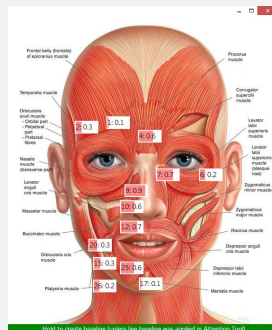
Computer iMotions Facial Expression Recording

Live broadcast of Emotient AU channels



Chrome Browser Extension

Real time visualisation of 19 Action Units mapped on to a face. Each AU is associated with a probability index.



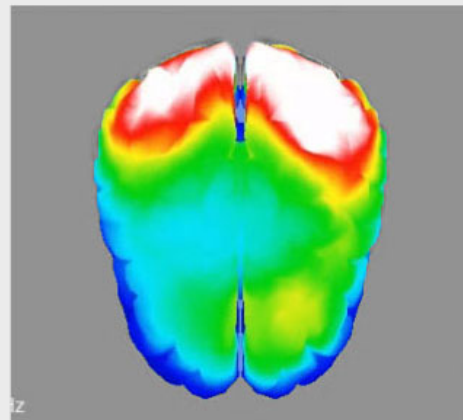
During EEG recordings the raw EEG data is sent to an external application that can live calculate & visualize the standard EEG metrics such as Alpha, Beta, Gamma and Delta frequency responses.

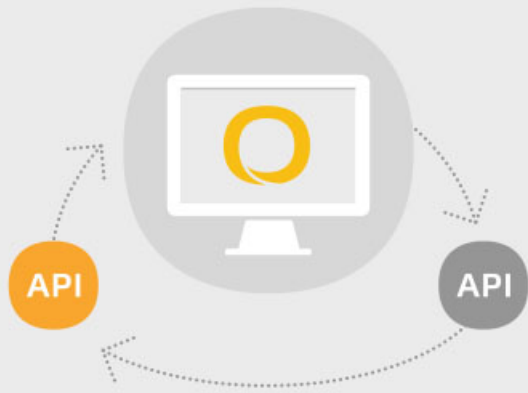
iMotions EEG Recording

Live broadcast

iMotions Listener Application

Real time calculation / visualization of EEG metrics: Delta, Theta, Alpha, Beta frequency responses



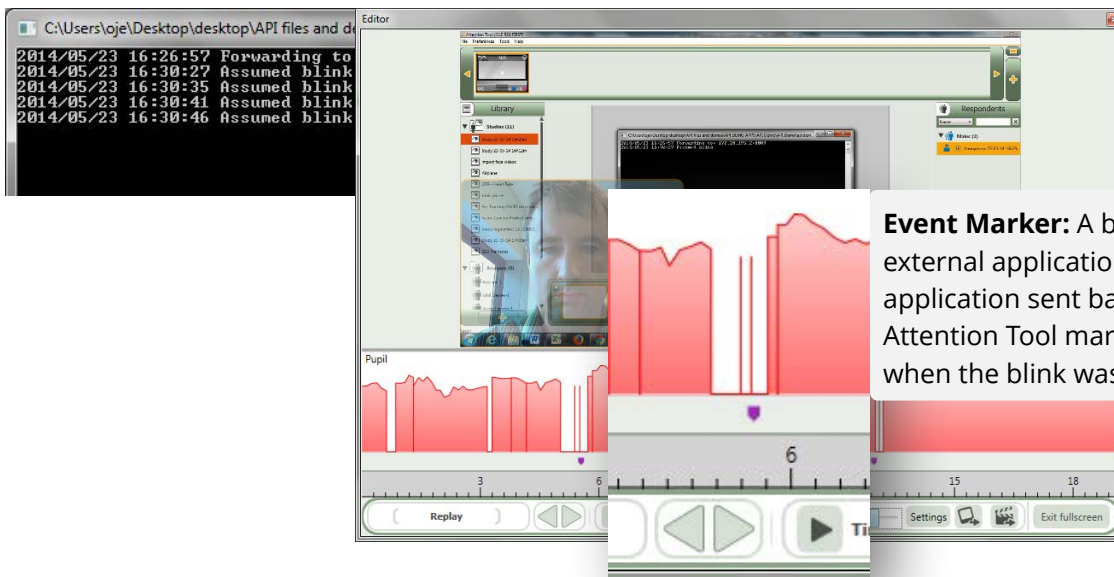
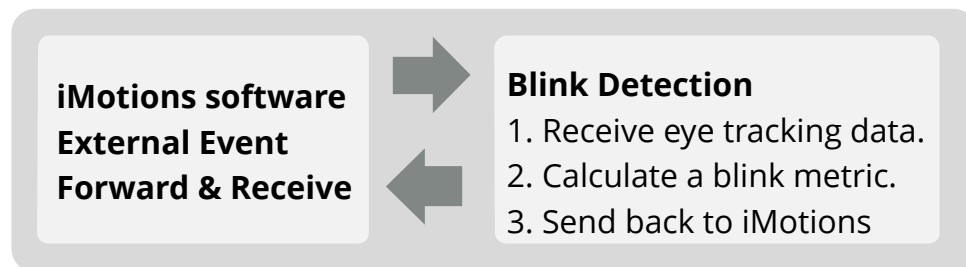


API Loop - Create sophisticated feedback loops of internal & external synchronized data

The combination of output and input API in iMotions can form a feedback loop. iMotions sends out data received by an external application which in turn processes the data and sends it back into iMotions.

API Output/Input (Loop) Example

In this example a simple application was built that can receive eye tracking data in order to detect eye blinks in the stream of data. If both left and right eye are having missing values at the same time a blink is detected. The blink event is communicated back to iMotions software as an event marker.



Virtual Environments: While a subject is navigating through a virtual environment the system can measure physiological responses like eye tracking or EEG in order to change the virtual environment based on these. For instance with eye tracking it is possible to detect if subject is looking at a particular object in the virtual environment. With EEG it is possible to measure cognitive workload in order to increase or reduce the task load in the environment.



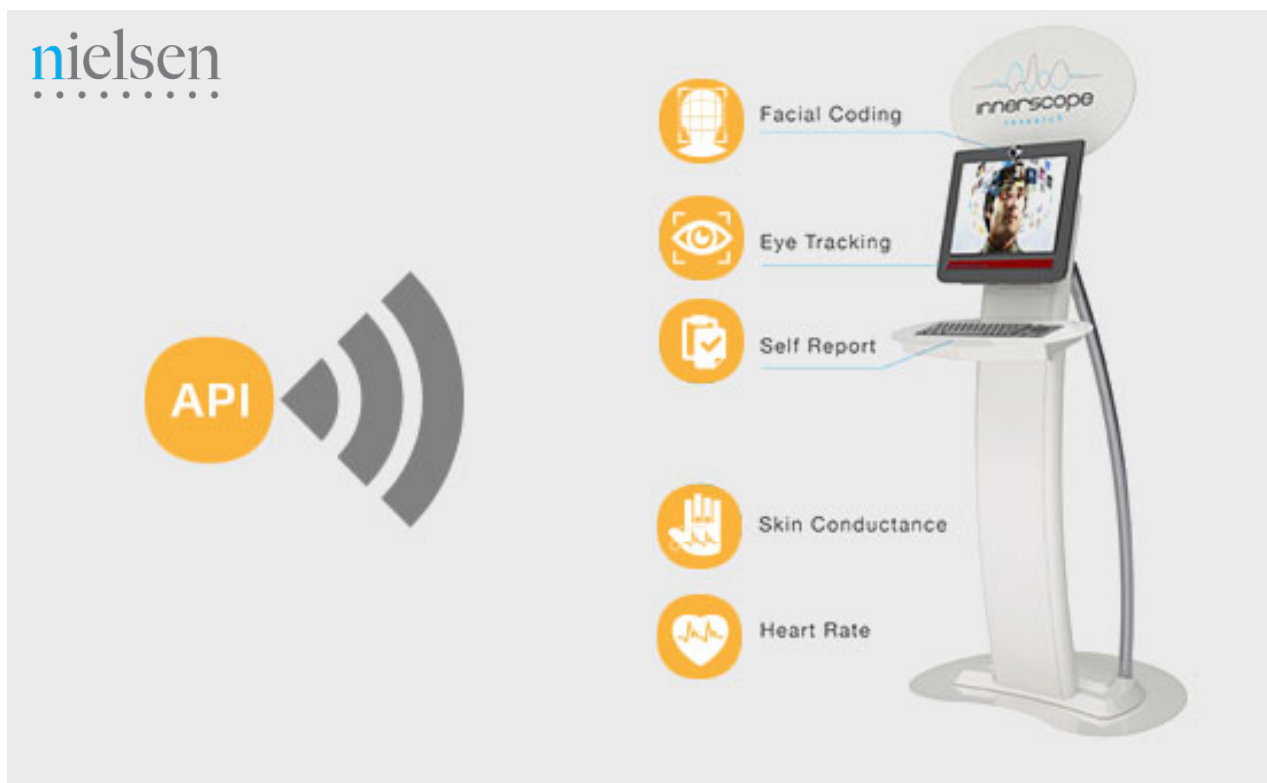
Control iMotions Remotely

The remote control API allows external programs to control iMotions i.e. perform certain tasks without interacting with iMotions user interface. The set of commands is targeted at allowing a user, with a large data collection network, to integrate iMotions into their existing study distribution infrastructure.

API Remote Control Example

Sensus Kiosks – Innerscope Research (Nielsen Consumer Neuroscience)

Nielsen Consumer Neuroscience uses iMotions to integrate multiple measures of conscious & non-conscious responses into their Sensus™ data collection concept. iMotions API is used to remotely control and manage all the network of kiosks from a central location. This is the first integrated consumer neuroscience kiosk, which delivers an unprecedented combination of market research tools in a single, scalable platform.



Distributing Studies In iMotions software

iMotions software allows the user to export the data for a study to a zip file. This file can then be loaded into a different system for analysis. If the study already exists on the analysis system, then the data for the respondents in the zip file are added to the existing study. The Save and Load commands allow these operations to be executed by a third-party software rather than from the UI.