

ADVANCED GESTURE RECOGNITION PLUG-IN FOR UNITY

Copyright (c) 2018 MARUI-PlugIn (inc.)

[IMPORTANT!] This plug-in is free for non-commercial use. Commercial use is prohibited. See the license statement below.

If you want to use this plug-in in a commercial application, please contact us at contact@marui-plugin.com for a commercial license.

[IMPORTANT!] The sample project in this package is for demonstration purposes only and not part of the plug-in. It contains parts of the Oculus Sample Framework (<https://www.assetstore.unity3d.com/en/#!/content/82503>).

Included files:

- GestureRecognition64.dll : The gesture recognition plugin. Place this file in your Unity project under /Assets/Plugins/
- GestureRecognition.cs : C# script for using the plugin. Include this file in your Unity project (for examples under /Assets/Scenes/Scripts/)
- sample_gestures.dat : Gesture database for a set of four sample gestures.
- all other files: sample Unity project. Used to illustrate how to use the gesture recognition plugin. The script using the plugin is /Assets/Scenes/Scripts/Sample.cs

NOTE: The sample project includes files from the Oculus

How to use:

Place the GestureRecognition64.dll file in your Unity project in the /Assets/Plugins/ folder, and the GestureRecognition.cs file in your scripts folder (eg. /Assets/Scenes/Scripts/).

In any of your project's scripts, create a "GestureRecognition" object.

```
> GestureRecognition gr = new GestureRecognition();
```

This object provides all the functions for recording, identifying and saving gestures.

(1) Place the GestureRecognition64.dll file in the /Assets/Plugins/ folder in your unity project and add the GestureRecognition.cs file to your project scripts.

(2) Create a new Gesture recognition object and register the gestures that you want to identify later.

```
GestureRecognition gr = new GestureRecognition();  
int myFirstGesture = gr.createGesture("my first gesture");  
int mySecondGesture = gr.createGesture("my second gesture");
```

(3) Record a number of samples for each gesture by calling startStroke(), contdStroke() and endStroke()

for your registered gestures, each time inputting the headset and controller transformation.

```
Vector3 hmd_p = Camera.main.gameObject.transform.position;  
Quaternion hmd_q = Camera.main.gameObject.transform.rotation;  
gr.startStroke(hmd_p, hmd_q, myFirstGesture);  
  
// repeat the following while performing the gesture with your controller:  
Vector3 p = OVRInput.GetLocalControllerPosition(OVRInput.Controller.RTouch);  
Quaternion q = OVRInput.GetLocalControllerRotation(OVRInput.Controller.RTouch);  
gr.contdStroke(p,q);  
// ^ repeat while performing the gesture with your controller.
```

```
gr.endStroke();
```

Repeat this multiple times for each gesture you want to identify.

We recommend recording at least 20 samples for each gesture.

(4) Start the training process by calling startTraining().

You can optionally register callback functions to receive updates on the learning progress

by calling `setTrainingUpdateCallback()` and `setTrainingFinishCallback()`.

```
gr.setMaxTrainingTime(10000); // Set training time to 10 seconds.  
gr.startTraining();
```

You can stop the training process by calling `stopTraining()`.

After training, you can check the gesture identification performance by calling `recognitionScore()` (a value of 1 means 100% correct recognition).

(5) Now you can identify new gestures performed by the user in the same way as you were recording samples:

```
Vector3 hmd_p = Camera.main.gameObject.transform.position;  
Quaternion hmd_q = Camera.main.gameObject.transform.rotation;  
gr.startStroke(hmd_p, hmd_q);  
  
// repeat the following while performing the gesture with your controller:  
Vector3 p = OVRInput.GetLocalControllerPosition(OVRInput.Controller.RTouch);  
Quaternion q = OVRInput.GetLocalControllerRotation(OVRInput.Controller.RTouch);  
gr.contdStroke(p,q);  
// ^ repeat while performing the gesture with your controller.  
  
int identifiedGesture = gr.endStroke();  
if (identifiedGesture == myFirstGesture) {  
    // ...  
}
```

(6) Now you can save and load the neural network.

```
gr.saveToFile("C:/myGestures.dat");  
// ...  
gr.loadFromFile("C:/myGestures.dat");
```

License:

This software is free to use for non-commercial purposes. You may use this software in part or in full for any project that does not pursue financial gain, including free software and projects completed for evaluation or educational purposes only. Any use for commercial purposes is prohibited.

You may not sell or rent any software that includes this software in part or in full, either in its original form or in altered form.

If you wish to use this software in a commercial application, please contact us at contact@marui-plugin.com to obtain a commercial license.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.