REACTIVE ENVIRONMENT FOR MIXING AND COMPOSITION
STEP BY STEP INSTALLATION INSTRUCTIONS

SETUP

In a terminal, go to your `'REMC'` folder (where you cloned our git repository) and type the following commands:

```
# mkdir Dependencies
# cd Dependencies
```

Step 1: Install a low-latency kernel

1. In a terminal, type the following command:

```
# sudo apt-get install linux-lowlatency
```

2. In order to use this kernel, you have to reboot your computer and select the linux-lowlatency kernel from the GRUB menu. (Note: if you don't see the linux-lowlatency kernel in GRUB, select "Previous linux entries" instead. You should be able to the linux-lowlatency in the list that follows).

Step 2: Install JACK Control

1. In a terminal, type the following command:

```
# sudo apt-get install jackd qjackctl libjack-dev
```

2. We need to configure JACK to use realtime scheduling. If the file `'/etc/security/limits.d/99-realtime.conf'` doesn't already exist, create it file by typing the following command in a terminal:

```
# sudo gedit /etc/security/limits.d/99-realtime.conf
```

3. Add the following two lines to that file:

```
@audio        - rtprio  99
@audio        - memlock unlimited
@audio        - nice    -15
```

4. In a terminal, type the following command:

```
# sudo groupadd audio
# sudo usermod -G audio yourUserID
# sudo adduser yourUserID audio
```

5. Log out, then log in again.

Step 3: Install Ardour

In a terminal, type the following command:

```
# sudo apt-get install ardour
```

Step 4: Install jmess

1. Go to the directory where you would like to install jmess

2. In a terminal, type the following commands:
```
# sudo apt-get install libasound2-dev g++ qt4-dev-tools
# wget http://jmess-jack.googlecode.com/files/jmess-1.0.1.tar.gz
# tar xvf jmess-1.0.1.tar.gz
# cd jmess-1.0.1/src
# ./build
# sudo make install
# cd ../..
```

Step 6: Install SuperCollider

In a terminal, type the following commands:

```
# sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys
FABAEF95
# sudo add-apt-repository ppa:supercollider/ppa
# sudo apt-get update
# sudo apt-get install supercollider supercollider-gedit
supercollider-dev supercollider-ide libsclang1
```

Step 7: Install Java Runtime Environment

In a terminal, type the following commands:

```
# sudo add-apt-repository ppa:webupd8team/java
# sudo apt-get update
# sudo apt-get install oracle-java8-installer
```

Step 8: Install SwingOSC

(Note: Our code requires SwingOSC, rather than Qt, to render all of its GUI elements correctly.)
In a terminal, type the following commands:

```
# wget
```

```
http://iweb.dl.sourceforge.net/project/swingosc/swingosc/0.70/SwingOS
C-0.70-Linux.zip
# mkdir SwingOSC
# unzip SwingOSC-0.70-Linux.zip -d SwingOSC
# cd SwingOSC
# sudo sh install_linux_local.sh
# sudo cp SwingOSC.jar /usr/bin/
# cd ..
```

Step 9: Configure SuperCollider to use SwingOSC

1. If the directory `'/home/yourUserID/.config/SuperCollider'` doesn't already exist, create it by typing the following command into a terminal:

```
# mkdir /home/yourUserID/.config/SuperCollider
```

2. In a terminal, type the following command:

```
# gedit /home/yourUserID/.config/SuperCollider/startup.scd
```

3. Add the following lines to the file:

```
GUI.swing;
SwingOSC.java = "/usr/lib/jvm/java-8-oracle/jre/bin/java";
SwingOSC.program = "/usr/bin/SwingOSC.jar";
g = SwingOSC.default;
g.boot;
q = q ? ();

ShutDown.run({
      SwingOSC.default.quit;
});

Server.default=l=Server.local;
l.boot;
```

Step 10: Install OpenNI

In a terminal, run the following commands:

```
# sudo apt-get install git freeglut3-dev libusb-1.0-0-dev doxygen
# git clone https://github.com/OpenNI/OpenNI
# cd OpenNI/Platform/Linux/CreateRedist
# sudo ./RedistMaker
# cd ../Redist/OpenNI-Bin-Dev*
# sudo ./install.sh
```

```
# cd ../../../../..
```

Step 11: Install SensorKinect

In a terminal, run the following commands:

```
git clone https://github.com/avin2/SensorKinect.git
cd SensorKinect
git checkout master
cd Platform/Linux/CreateRedist
sudo ./RedistMaker
cd ../Redist/*
sudo ./install.sh
cd ../../../../..
```

Step 12: Install NITE

If your computer has a 32-bit architecture, run the following command in a terminal:

```
# wget http://www.openni.org/wp-content/uploads/2012/12/NITE-Bin-
Linux-x86-v1.5.2.21.tar.zip
```

Otherwise, if your computer has a 64-bit architecture, run the following command in a terminal:

```
# wget http://www.openni.org/wp-content/uploads/2012/12/NITE-Bin-
Linux-x64-v1.5.2.21.tar.zip
```

(Note: You can check your computer's architecture by running the command "`uname -m`" in a terminal)

In a terminal, run the following commands:

```
# unzip NITE-Bin-Linux-*.zip
# tar xvf NITE-Bin-Linux-*.tar.bz2
# cd NITE-Bin-Dev-Linux-*
# sudo ./install.sh
```

Step 13: Build OSCeleton_for_REMC

In a terminal, run the following commands:

```
# cd ../../OSCeleton_for_REMC
# make
```

Step 14: Disable gspca_kinect module

Edit the "**`/etc/modprobe.d/blacklist.conf`**" file by running the following command in a terminal:

`# /etc/modprobe.d/blacklist.conf`

Add the following line to the end of the file:

blacklist gspca_kinect

(Note: this change will not take effect until you reboot your computer)

Step 15: Reboot computer and select low-latency kernel

In order to use the low-latency kernel you installed earlier, you have to reboot your computer and select the **linux-lowlatency** kernel from the **GRUB menu**.

(Note: if you don't see the linux-lowlatency kernel in GRUB, select "**Previous linux entries**" instead. You should be able to the **linux-lowlatency** in the list that follows).

That is all!

To begin using our Reactive Environment for NMP, please read "User Manual.pdf"