
Nubomedia-VTT-AR-Demo Documentation

Release 1.0

Markus Ylikerälä

October 14, 2016

1	Readme	3
1.1	Material	3
2	Developers Guide	5
2.1	Source Setup	5
2.2	Details of the Demo	5
2.3	KurentoClient	6
3	Installation Guide	7
3.1	Installing Optional ArMarkerDetectorDemo	7
4	Licensing and distribution	9

Contents:

Readme

This project is part of NUBOMEDIA www.nubomedia.eu

This document describes how to utilize the [ARModule](#)

Some material that can be utilized is given at the end.

Artifacts

Maven Central: <http://repo1.maven.org/maven2/fi/vtt/nubomedia/armodule/arfilterdemo/>

Npm: <https://www.npmjs.com/package/fi-vtt-nubomedia-kurento-module-armarkerdetector>

On the Server Side

To launch the test program clone this repository and execute

```
cd armoduledemos/ar3d
mvn compile exec:java -Dexec.mainClass="fi.vtt.nubomedia.kurento.Ar3DApp"
```

On the Client Side

Browse with WebRTC compliant browser (eg Chrome, Firefox) to the server where ar3d is launched http://IP_OF_AR3DHOST:8080/ Change the IP_OF_AR3DHOST and port (8080) if needed.

You should now see AR Demo so just follow the given instructions on that page.

Note For Ubuntu Desktop Users

On some environments, for some reason, it has been reported that since KMSv6 screen can get partially or totally blank. But because ssh works, it might be better to install ssh server beforehand to gain control to the host again.

1.1 Material

Augmented Models

Some models can be found eg from: [models](#)

Licensing and distribution

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

ALVAR 2.0.0 - A Library for Virtual and Augmented Reality Copyright 2007-2012 VTT Technical Research Centre of Finland Licensed under the GNU Lesser General Public License

Irrlicht Engine, the zlib and libpng. The Irrlicht Engine is based in part on the work of the Independent JPEG Group. The module utilizes IJG code when the Irrlicht engine is compiled with support for JPEG images.

Contribution policy

You can contribute to the Nubomedia community through bug-reports, bug-fixes, new code or new documentation. For contributing to the Nubomedia community, drop a post to the [Nubomedia Public Mailing List] providing full information about your contribution and its value. In your contributions, you must comply with the following guidelines

- You must specify the specific contents of your contribution either through a detailed bug description, through a pull-request or through a patch.
- You must specify the licensing restrictions of the code you contribute.
- For newly created code to be incorporated in the Nubomedia code-base, you must accept Nubomedia to own the code copyright, so that its open source nature is guaranteed.
- You must justify appropriately the need and value of your contribution. The Nubomedia project has no obligations in relation to accepting contributions from third parties.
- The Nubomedia project leaders have the right of asking for further explanations, tests or validations of any code contributed to the community before it being incorporated into the Nubomedia code-base. You must be ready to addressing all these kind of concerns before having your code approved.

Support

Support is provided through the [Nubomedia Public Mailing List]

[NUBOMEDIA]: <http://www.nubomedia.eu> [Nubomedia Public Mailing List]:
<https://groups.google.com/forum/#!forum/nubomedia-dev>

Developers Guide

This document describes details of the ArDemo that utilizes [ARModule](#)

2.1 Source Setup

The default directory where the demo is installed is the following:

```
~/nubomedia/armodule demos/ar3d
```

Fetch the source etc:

```
git clone https://github.com/nubomedia-vtt/armodule demos.git
cd ~/nubomedia/armodule demos/ar3d
curl -sL https://deb.nodesource.com/setup | sudo bash -
sudo apt-get install -y nodejs
sudo npm install -g bower
bower --config.analytics=false install --allow-root
mv bower_components src/main/resources/static/
```

2.2 Details of the Demo

Ar3D Kurento Client gets the data that is passes to the filter as json either from the browser ie javascript or from the file system For this demo the default json data is gotten from the browser if you execute:

```
mvn compile exec:java -Dexec.mainClass="fi.vtt.nubomedia.kurento.Ar3DApp"
```

But json data is gotten from the file system if executed with a valid json file eg [local.json](#)

```
mvn compile exec:java -Dexec.mainClass="fi.vtt.nubomedia.kurento.Ar3DApp" -Dexec.args="local.json"
```

Ar3D Kurento Client passes data related to the augmentation via key-value pairs eg {"model", "/opt/cube.ply"}, {"scale", 0.5f}. This is described in kmd of the ARModule [armarkerdetector.ArMarkerdetector.kmd.json](#) Please refer to this kmd to find out more about the syntax of json currently in use.

The models, images etc are loaded from the file system or via URL.

Thus, please check that the paths/URLs are correct ie that eg /opt/cube.ply is readable.

An example about the syntax

```
'{"id":0, "type":"3D", "strings":[{"model":"/opt/faerie.md2"}, {"texture":"/opt/faerie2.bmp"}], "flo
```

```
"id":0 = Alvar Marker 0 ie Zero Marker
```

```
"type":"3D" = render as 3D model (2D for flat models)
```

```
"model":"/opt/faerie.md2" = 3D model that to be rendered
```

```
"texture":"/opt/faerie2.bmp" = texture for the 3D model in md2 format
```

```
"scale":0.09} = scaling of the 3D model
```

2.3 KurentoClient

The API for Kurento Client to communicate with the AR filter is available at: <https://github.com/nubomedia-vtt/armodule/blob/master/ar-markerdetector/src/server/interface/armarkerdetector.ArMarkerdetector.kmd.json>

Basically, the following functionality is available:

```
ArMarkerdetector arFilter = new ArMarkerdetector.Builder(pipeline).build();
arFilter.setArThing(...);
arFilter.enableAugmentation(false/true);
arFilter.setMarkerPoseFrequency(false/true, 1);
arFilter.setMarkerPoseFrameFrequency(false/true, 10);
arFilter.enableMarkerCountEvents(false/true);
arFilter.addMarkerCountListener(...);
arFilter.addMarkerPoseListener(...);
```

Installation Guide

3.1 Installing Optional ArMarkerDetectorDemo

These are the instructions how to utilize the ar-markerdetector filter of of the [ARModule](#). Thus, after installation of ar-markerdeterctor is done, you can test the filter with this sample application.

It is assumed that the sample application will be installed to the same host where KMS is installed. Otherwise change the code of ar3d.

Installing

Execute the following commands if not already done:

```
sudo apt-get install xinit -y sudo apt-get install git -y sudo apt-get install maven2 -y sudo apt-get install libsoup2.4-dev  
-y sudo apt-get install openjdk-7-jdk -y
```

For Ubuntu Server Users

Run X:

```
xinit
```

If you're not authorized to run X, the following might help:

```
sudo dpkg-reconfigure x11-common
```

For Ubuntu Desktop Users

If you're not authorized to access X, then:

```
xhost +
```

On some environments, for some reason, it has been reported that since KMSv6 screen can get partially or totally blank. But if you are utilizing Ubuntu Server installation this is not an issue.

Finally, instead of xinit, there is the following alternative:

```
sudo apt-get install xserver-xorg-video-dummy  
wget xpra.org/xorg.conf  
Xorg -noreset +extension GLX +extension RANDR +extension RENDER -logfile ./0.log -config ./xorg.conf
```

If you have problems with port, just try another eg:

```
Xorg -noreset +extension GLX +extension RANDR +extension RENDER -logfile ./10.log -config ./xorg.conf
```

3.1.1 Models

Copy some sample models from:

```
https://github.com/nubomedia-vtt/armodule demos/tree/master/Models
```

into the default folder that is:

```
/opt
```

3.1.2 Artifact

On the server side

To install Java interface to arfilter, fetch artifact with artifactId arfilter from Maven Central Repository and install it:

<http://search.maven.org/>

Install the arfilter artifact, eg:

```
mvn org.apache.maven.plugins:maven-install-plugin:2.5.2:install-file -Dfile=arfilter-1.0.0.jar
```

You can also install JavaScript interface:

```
bower install fi-vtt-nubomedia-kurento-module-armarkerdetector
```

Fetch keystore:

```
wget -nd http://ssi.vtt.fi/ar-markerdetector-binaries/demo/keystore.jks
```

Fetch artifact with artifactId arfilterdemo from Maven Central Repository:

<http://search.maven.org/>

Execute artifact:

```
java -jar arfilterdemo-1.0.1.jar
```

On the client side

Browse with WebRTC compliant browser (eg Chrome, Firefox) to the server where ar3d is launched https://IP_OF_AR3DHOST:8443/ Change the IP_OF_AR3DHOST and port (8443) if needed.

You should now see AR Demo so just follow the given instructions on that page.

Licensing and distribution

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

ALVAR 2.0.0 - A Library for Virtual and Augmented Reality Copyright 2007-2012 VTT Technical Research Centre of Finland Licensed under the GNU Lesser General Public License

Irrlicht Engine, the zlib and libpng. The Irrlicht Engine is based in part on the work of the Independent JPEG Group. The module utilizes IJG code when the Irrlicht engine is compiled with support for JPEG images.