minclient

2015-03-16

Daniel Pohl

minclient is a minimal client that interacts with minrt for in-home streaming. It is not meant as a fully featured software, but as a reference implementation.

The software comes **without official support**. But you could try to reach some of the team members, e.g. the author of this document with firstname.lastname@intel.com.

The main target for this software is Android 4.x or higher. The hardware needs to support ETC1 textures, which is available for most modern smartphones, tablets and some smartwatches. The code contains some stubs for manual software decompression of ETC1, which one could implement for debugging reasons on a system that does not handle ETC1 natively.

# Building and installing the apk

You should first try to be able to build Android NDK applications like the hello-jni and hello-gl2 sample applications that come with the NDK. Once this works, you can build with:

1. ndk-build -j 8 from within minclientandroid\jni.
2. ant debug -Dbuild.compiler=javac1.7 from within minclientandroid
3. adb install -r qwrt-mini-debug.apk from within minclientandroid\bin

# Launching and configuring

The first time you launch minClient on your mobile device, it will create a qwrt.cfg file in your /sdcard/ folder on the Android device. The file is filled with default values that are defined in Engine.cpp. Open this up in a text editor on Android (e.g. Jota Text Editor).

### 1-Server Setup

Set the IP and the port of the server that runs minRT, e.g.

server0 192.168.0.10 2000

### Multi-Server Setup

More than one machine can create pixels and minClient combines them. They need to be split into rectangular areas. Probably not the most efficient way, but this works reasonably well and should be seen as a reference implementation. Following is the setup with four servers that generate together a 1280x720 image.

server0 192.168.0.10 2000

server1 192.168.0.11 2000

server2 192.168.0.12 2000

server3 192.168.0.13 2000

rectLeftServer0 0

rectRightServer0 959

rectBottomServer0 0

rectTopServer0 1079

rectLeftServer1 960

rectRightServer1 1919

rectBottomServer1 0

rectTopServer1 1079

rectLeftServer2 640

rectRightServer2 959

rectBottomServer2 0

rectTopServer2 720

rectLeftServer3 960

rectRightServer3 1279

rectBottomServer3 0

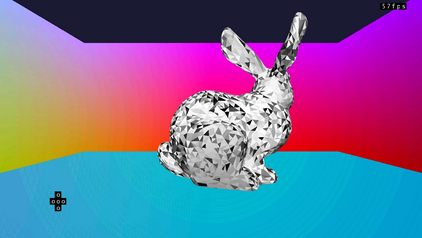
rectTopServer3 720

The client will connect to those servers in the order that they are defined. If multiple servers are used, rectMode is set to true internally.

### Troubleshoot

If the connection doesn’t work, try to ping from the server to the phone and vice versa. Try disabling fire walls.

# Controlling the demo



In the demo there is a little up/down/left/right touch area in the left bottom area. Next to it, but not visible, is another area which controls the view (comparable to mouse look). These are not tweaked for nice usage, just to get some basic interaction done. If the rendered resolution does not fit the displayed resolution, then the effective touch areas might lie outside of the rendered screen. In Android, the resolution should always be the fullscreen resolution, but sometimes lower resolutions work as well while the remaining part of the screen is made black.

# Exiting the demo

Touching the screen quickly five times within a few seconds exits the client.