# TODO:

1. ApparentlySparseArry should replace all integer keyed hashmaps, got to copy the code
2. I saw some time spent with ReferenceQueue in profiler, so maybe wek refs are not very performant?
3. I think the behaviors and renderer might be running during load up, so the cpu is spending all it’s time rendering? Maybe?
4. Make new project for morrowind kick off
5. Do the joglespipeline project fork now before forgetting, then sort out pulling
6. Various nif files cause a crash, find and stop
7. Oblivion shaders appear to not like textures missing or something, see anvil for examples
8. W/Adreno-ES20: <\_\_load\_uniform\_float:841>: GL\_INVALID\_OPERATION appears in logcat a lot! Web says some setuniform is to blame, only happens a couple of times a second, so not major
9. I accidentally gave it the skyrim trees with the wrong sort of textures in it, but it only spews a tiny gl error on load attempt, I feel I should detect wrong textures and report them properlymaybe as flags in the texture loader classes(allowed) though notice KTX never mentions to anyone what sort of texture it is, just the pipeline compressed call
10. I like this idea [Dynamic Resolution Rendering on OpenGL\* ES 2.0 | Intel® Developer Zone](https://software.intel.com/en-us/articles/dynamic-resolution-rendering-on-opengl-es-2)

# Notes:

Android FileChannel.map throws exception when offset is greater than 2gig, not size, offset! So obviously a 32bit operating system and so file system nothing above 2 gb

<http://stackoverflow.com/questions/10126845/handle-screen-rotation-without-losing-data-android>

Ok first up sdk23 new permission must ask to access external storage using new

<http://inthecheesefactory.com/blog/things-you-need-to-know-about-android-m-permission-developer-edition/en>

gear. Need to use the Compat thingy style so it’ll do version checking for me

Heap size maximum work around

<http://www.badlogicgames.com/wordpress/?p=2367>

Note I’ve added a thing that lets bitmaps go out of limit, but I need directbytebuffers to be out of limit

**try** {  
 BitmapFactory.Options opts = **new** BitmapFactory.Options();  
 Field field = opts.getClass().getField(**"inNativeAlloc"**);  
 field.setBoolean(opts, **true**);  
}**catch** (IllegalAccessException e){  
 e.printStackTrace();  
} **catch** (NoSuchFieldException e) {  
 e.printStackTrace();  
}

# FAQ

What about beth?

They had a go at openMW and just said no android – because free and open source

What about openMW – did you copy?

Yeah, nay.

OpenMW are still actively developing?

Between them and beth

Lawyer from beth?

No just like openMW I’m not distributing the IP

Work with beth?

Yep hope so, money for them and art direction

What about interface?

Yes tricky, definitely prefer no keyboard/mouse

Just notes on android renderer mixup

I think I need to install a renderer on android that runs off the GLWindowListener display caller thingy, How often does that get called I wonder?

That way the display callback that owns the glContext doesn’t need to be released etc, but I need to make sure that renderer knows about glwindow callback before addNotify as I want all calls to be done through it.

Maybe a static? Yes I think so. Right before canvas3d create or universre of view or anything, in the init of a glwindow itself

Ummmm…..

It does not like creating a listener on a listener thread callback, so I need to create it before hand somehow!!

My other idea is to only instanite the glwindow in the create new context, possibly using that shared context madness thingy