still not done...

266379_PixelRule.tiff ¬

This is an incomplete list of things that need to still get done. Please note that many of these are rather cryptic remarks intended to jar my memory, and not necessarily for public understanding. Here's some highlights, off the top of my head, and in no particular order:

"Option"s should go directly to the camera; it should have a list of Option objects that it renders in the WorldBegin stuff...

also, "Declare"s and stuff like that, before the WorldBegin, should go to the camera. Maybe the camera should have a list of renderable commands too...

Before loading a nib, I should save the "main menu" or whatever the heck it's called. Then after loading the nib, if the "main menu" is set ot something new, I should add this as a submenu as "from nib..." of something like that, and restore the original menu.

make it so that Lights aren't moot (i put a disclaimer with the warning now)...

fix color bug on matrices of sliders - it's ugly and annoying...

you can crash IB if some of the parameters on the WWThumbwheel are both zero. Need to find and fix those...

Need a WWPasteboard type. You should be able to cut and paste all over the place. Need to be able to cut and paste images and movies from WWSimpleMovieViews. You should also be able to cut and paste RIBCommands. This should be part of the WWRenderable protocol.

Should be able to decide whether to ignore or hang onto comments in RIB files.

Need to make sure the rest of the Map routines work like TextureMap does.

Any of the shader rib commands (Surface, Displacement, etc.) should try and split their args if there is only one, so that you can easily hand in args...

Want to have the symbolic description of an EveCmd or EveProc evaluate to something null'ish but valid. This would allow you to build up a scene that wasn't mutable (i.e. like visualizing-a-shot scene) - Could use ArchiveRecord I guess...

allow controls back on the 4x4 of a WW3DShape; allow both relative and absolute.

RIBCommandList's bounding box is sometimes wrong, as is EveProc's.

maxSampleBandwidth stuff isn't done for all classes; need a good sweep through them.

Need autovisualizing for scenes and shots.

still don't draw grid in the Well.

don't understand why PAThumbwheel still draws number in its cell (although it gets covered by the rest; but on a big slider you can see the drawing).

I should fork a thread when rendering; this "render in foreground" stuff is bullshit.

make vector args to shaders editable, and stop printing out that info...

the bounding box of a RIBCommandList seems mucked up (see

WW3DText).

Fix up and tell people about the camera interpolation stuff.

Actually implement and UI-ify the "keys" thang for tcl - allow arbitrary binding of tcl commands to events ala Tk.

There is still no UI for manipulating lights. One of the problems with this is the code I wrote for WW3DLights to draw themselves - it's wrong. Basically, the lights are pointing the wrong way. Obviously my trig is wrong, and I'm lame, but I haven't had the time to fix it. The place holder UI I had on the WW3DWell is still around, and may or may not be the UI I use when I get around to fixing this.

Need to document various undocumented protocols, such as the WW3DWell delegate, and other delegate protocols.

Still need to figure out the best way to allow RIBCommands to allow themselves to be inspected. Given the way I'm doing animation, it doesn't make sense to allow you to change a RIBCommand, but you could change an eve command...

When you drop an object into a WW3DWell, sometimes the rendering style changes (i.e. smooth shaded goes to wireframe or faceted). The next time the well's camera is redrawn it goes back, but I'm still not clear on what's going on here.

The WWSimpleMovieView should drop frames to keep up. It also should have a facility for reporting actual frame rate. Actually, NeXT should just release NEXTIME so I could incorporate it and not have to deal with shit like this...

Need to make the WWTTTimer be able to run in its own thread, and need to implement "synchWithScene" so that it takes advantage of this.

Need to do a smarter parsing of the dependencies of UI elements - if a $ isfollowed by a {, follow it to the }, and evaluate that variable then,so things like ${shape} will work correctly.   This will allow me to build generic UIs that bind themselves to the variables at load time.

Make sure that everything knows how to archive itself, WWTCLClosedCmd doesn't, right now...

The camera should be able to see what the highest sampling frequency of the WWAnimatable objects in the scene that it's looking at.

allow "awake with control panel visible"

make a shot inspector:
 - shot name
 - frames per second
 - initial Scene time
 - final scene time
 - initial set of camera parameters
 - tcl proc for interpolating camera parameters to next frame

allow initiating a camera shot to:
- RIB
- qrman
- prman (puts them in an .anim directory)
- qt (brings up NEXTIME)

- need to put adaptive rendering quality back in.
- two level of qrman
- two levels of prman (automatically)
   - set shadingRate (and other quality parameters)
   - set delay for initial start-up

shadows...

Sound: when a sound gets dropped into a scene, it has an extent in time and has a preferred sampling rate.   For each sample point, a new WWSample gets generated...   Too wasteful.

finish extending the expr language

add types!!

make an interface for a floating camera instance which can be applied and executed

make the 4x4 of the shapes be samples (SHOULD I? Good reasons not to...)

add the volume view stuff as a set of categories to the WWSimpleMovieView

Need a slate for the camera shot thang...

Have a WW3DShotTemplate object, which has a WW3DShot instance, but the actual WW3DShot instance is short lived, while the WW3DShotTemplate is long lived, and it's the thing that has the inspector.

make RIBCommandList able to do interpolation (with some epsilon before it synthesizes a new sample)

figure out why having a "Basis" call in a RIB file causes WW3DTester to die in opening up a Open panel in the add shader thang...

add askEM, askBM, tellEM, tellBM,

implement synchWithScene so it works for threads.

Whenever a new sample is generated for an EveCommand, you should mark the command (and it's correspond shape) as dirty.

Also, rather than constantly updating the controls, it would be good to have a flag about "dirty" or "out of date" info, and grey it. Have a button there which allows you to synch, but if we grey it out, then both the camera, and the shape info would be more accurate.

Think about an interpolation function between two samples in a WWSampleList; In a real sense, they are two key frames; think about what is interesting about key frames, you have u(start) and u(finish), now you just want u(somewhereInBetween). It's like a motor program, except in a motor program you don't get the final value.

a WW3DShape is really nothing more than a compound RIBCommand, right? It's a RIBAttributeBegin, RIBAttribute (name), RIBSurface, etc...

- Is that true, though? It also contains rib command lists and hierarchy pointers...

Make WWTCLInterp archive default parameters for proc definitions

Actually write some class documentation (hah!)