

# cool New Stuff

266379\_PixelRule.tiff ↩

Well, for those of you who have seen this stuff before are probably thinking: "so what's new in 2.0?" Well, the short answer is that the tcl palette has been cleaned up a bit and had a few more objects added that make it very, very powerful, especially for doing more multi-mediaish UIs. The 3D palette has been expanded in a very significant way to allow it to deal with animation. This is wildly exciting...

Here's some highlights, off the top of my head:

A new object called `WWSimpleMovieView`. I can't believe no one ever wrote this object before - it really shows off `NEXTSTEP` well, especially given how easy it was to write. It basically lets you have 3 layers in a view: a background view (with color and transparency), a movie layer (which takes an anim, and eventually, a QT movie, and has transparency), and an image layer (which also has transparency). The movie can run over the image, or vice versa. You can control the frame rate of playback on the movie, and you can control when and how the movie plays back. Finally, if you cmd-click it, you can inspect the object (and change stuff) while it's running.

This is a way useful object, especially when you subclass it for your own nefarious uses (see my WWMovieVarView and WWMovieProcView)...

new icons. I spent a lot of time on the new icons - hope you like them.

A 3D text object written by Ian Wilkinson at Canon UK. This one is still under development, but I for one think this is sooo, sooo cool! What's also neat is that it serves as a nice template for how you as a programmer can extend the WW3DKit. Note that it motion blurs correctly - this is much harder to do (not for you; I've done the hard work already) than you might think at first, given the generality of the WW3DKit. Thanks Ian!

Animation done right! I've spent many months thinking about the "correct" way to integrate animation into WavesWorld, and I'm very happy with the state of things. One of the neatest things about the way animation is handled is that things like motion blur come for free. I need more modeling tools to convince the unwashed, but if you're at all a CG weenie, you can do lots of cool stuff already.

drag-and-drop now does the "right thing" - it used to be that when you

dragged a filename over a WW3DWell it just put up the copy icon and then when you dropped it then it determined if it knew what it was. Now it checks the whole time, and the default icon is like a "move" while you have to hold down the alt-key to get copy now. Also, it does the right thing there - if you just drag a model in, it replaces the old one, if you alt-drag, it makes that new model the child of the currently selected shape already in the well.

An interesting Timer object which grooves well with tcl. This object still needs some work (needs to have optional threading, needs an IBEEditor, needs conditional lifetime, etc.),but it's still useful now, and is fun to play with for animation stuff.

Archiving now really works! Thanks again to Ian for this one; turns out there is nasty bugs in NeXT's archiving if you use the "[1234f]" format for lots of numbers (even on the same machine), and once I changed all those statements to use the NXRead/WriteArray() routines, everything seems hunky dory. This means I should finally be able to write usable plug ins for other apps (hello Andrew and Guido). A big thanks to Ian for doing the legwork on this one.

A new kind of animation command - the EveProc. One of the limitations in 1.x was the fact that you could only put atomic

commands inside an EveCmd - in other words, you could only vary things in a command, you couldn't vary a whole sequence of commands. The EveProc lets you do this. It's complicated to explain quickly - look at the canonicalRIB/solid quadrics in the Examples directory for more info.

More examples! Lots and lots of examples.

The beginnings of a good intro to the 3D stuff in Documentation.

Lots of shader source code! Thanks to my friend Larry Gritz for allowing me to include many of brilliant shaders. Also, I've included shader source from Darwyn Peachey and Ken Musgrave from the book they coauthored (with a bunch of other smart folks) about procedural modeling and texturing.

Support for Larry Gritz's most awesome shareware RenderMan compatible rendering tools, the "Blue Moon Rendering Tools". Larry single handedly implemented a radiosity and raytracing RenderMan compatible renderer and then made it available to the world! What a prince!