

This directory contains the beginnings of a tutorial on how to take a RIB file, turn it into an eve file, and eventually turn it into a full blown piece of malleable media.

The object in question is a set of window blinds. I used Pixar's ShowPlace 2.1 Macintosh application (written by Dana Batali and the fine folks at Pixar) and used the "Blinds" plug-in written by Dan McCoy to generate six different versions of blinds. I saved each out as a RIB file, transferred them to a NeXT (which meant changing ctrl-Ms to ctrl-Js), and dropped them into a WW3DWell. I then saved each out as a .eve model and edited them a bit (taking out extraneous transforms, mainly). I then dropped them back into the WW3DWell and, using the shape browser, changed the names of the parts (double click on the shape browser in the WW3DWell's control panel and type in a new name), and saved the models back out again.

I then took one of the slats (a trimmed NURBS surface) and put it in a wwModel file wrapper and determined how much to translate it to center it about its origin (by selecting "draw origin" in the qrman controls of the WW3DWell's control panel and then manipulating the translate controls until the origin was centered), and added a little UI to rotate it.

I then took off the pelmet (the top part of the blinds assembly) and the batten (the heavy bottom part) and made them separate eve files.

Finally, I put them all together in blinds1.wwModel and tweaked and tweaked and tweaked... This model worked fine, and was pretty efficient (only had EveCmds where you could change stuff), but I was bummed that I couldn't pull the blinds up and down. Because the parts were all interconnected, it was difficult to just put in a few EveCmds and be done. So... I turned to the EveProc cmd, and redid it that way. This is much slower (especially on old black hardware), but is a much more flexible way of doing things. Now you can pull the blinds up and down; notice that the bottom slats behave differently when they are open and closed and scrunched together. This became blinds2.wwModel.

For fun, manipulate the blinds and render a sequence. Check out the groovy motion blur...

Again, in case I wasn't clear:

blinds.wwModel and blinds2.wwModel might, at first glance, seem to generate the same model. But they don't. The one generated by blinds2.wwModel is really a single EveProc; each time you change a paramter of the model, the entire procedure is reevaluated. This can take

a long time, but it's a really, really flexible way of doing things. The best part is that once you've debugged that procedure, it will probably only take you tens of minutes to turn it into an Objective-C class that conforms to the WWRenderable protocol.