

---

Subject: Re: Groovey

Posted by [dudymas](#) on Sun, 12 Nov 2006 22:09:48 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

thanks! I didn't realize there was an example already in there on threads.

I want the framework to be multi-platform, however, so unless I'm reassured that the threads in u++ work on mac, linux, and windows reasonably, I'm going to have to opt out of using it particularly. But, seeing as how I have no experience with threads yet, I could still definitely find a lot of help out of looking at the u++ code you mentioned there. I'll read through the example and see what insight it can give me.

I'll attach a look at the system I want to use... where one thread does either rendering (grabbing particles and drawing them to a buffer) or drawing (putting the buffer to the screen) and the other thread either manipulates particles (basically begins changing the particles) and updating the particles (committing the changes to the particles finally). Each thread can only switch between these two states, and yet I don't want to allow the system to update particles while they are being rendered, because that means that some of the particles will be changed before they are rendered, causing a loss in data or tearing on the window. And yet, the particles can be manipulated while there is rendering because none of the changes are committed to the particles yet... only logged.

So this state diagram (I admit, it's horrid to look at) is supposed to help show this will all the possible states the two threads can be in. The green lines show best case, yellow is nominal, and orange is unwanted and unexpected but still needs to be considered. Reddish means it's a rare scenario, but definitely still possibility. Finally, the fuchsia/violet lines mean the program is closed at any of those states.

Man, this is getting more fun each second.

So here's my plan:

I'll give the renderer states that rule over how the manipulation/update thread governs its transitions. Hopefully the renderer is going to be slow enough that it won't bully everything. We'll see.

### File Attachments

1) [Grooveyactivity.gif](#), downloaded 2247 times

---