
Subject: Groovey

Posted by [dudymas](#) on Sat, 11 Nov 2006 00:48:26 GMT

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

So, as a sort of proof of concept, I'll be trying to code this guy:

<http://code.google.com/p/gparticles/>

And just so you know, I also posted this topic at Dream In Code forums too... but mostly just to see if they are interested at all. I will proly use this topic as my 'usual' place of updates. Just so that if this turns out to be a success, I can look back and see a sort of journal and anyone else can do the same.

I think I might post the UML here if it seems like anyone is interested in watching the work happen and give me pointers or just laugh and discuss.

This is GOING to be hard. I know it. Crossplatform threading WITH OpenGL as well is going to have to be confusing... or at least it has to be because I'm not using java.

I have a macbook pro that's currently sick, though, so my main platform will not be tested first. I'll get it working on windows first.

Here's the particle models I want to get working in three months... they are given in the order I'll work on getting them done at:

Basic Gravity (A bunch of particles in a domain that bounce around happily. This will test two things: Threads don't starve or race. Particle model is at least working decently.)

Zombies (A bunch of people sit on the screen. Randomly spawn a single zombie. People have three behaviors: become a zombie if one touches you, run and panic if you see a zombie or another person panicking, and if everything is fine the people just walk around leisurely. Designed to test particle behavior system to make sure they have an efficient collision detection model and all that).

Whizz Bangers (Create a screen of self-attracting particles. They are given an overall whirlpool force that will continue to slosh them. As the particles begin to clump, set a critical mass size that causes them to behave like an atom bomb... eventually a bunch of clumps form and when they hit critical mass they explode violently. The cycle continues nonstop, because they just blow apart into the smaller beginning particles again. This will test both behaviors and models of force with both the exploding and the whirlpool. Finally, it will add a couple new features to test: can we get the particle driver to parse a config file with definitions of particles and can we make the particles change color and make smoke.)

And the rest of the particle sets I want to finish I'll keep secret for now... but I eventually want it so that anyone can make whatever group of particles they want and make them interact in complex or simple ways. This means I have to abstract each model (particles, behaviors, and models of forces) three ways:

What does the user see when they configure a particle system.

What does a computer see when it manipulates the system.
What does the graphics card see when it renders the system.

I'll be using differential equation solvers eventually... before you ask. And also some really funky random number generator foo and all.

So... the class structure is already looking rather funky. I mean... each time I look at it, it's like as if I suddenly hear "Get Funkayeeee, boyeeeeeee!" in my head. Or maybe that's just me needing more caffeine.

Yay! Doesn't this sound like fun?

And after all this project is finished... I have some crazy graphics tests for the whole thing (transforming bodies for the particles so they can look gooey, throbbing, smoldering, gaseous, probabilistic, magical, etc.)

On a final note, here's what I have to say about UPP so far, since this is the first time I've ever touched it:

I like it better than devcpp so far. As well, I love how you guys do all these packaging tricks. My gripes so far:

I wanna use boost libraries (for threading, primarily)... I also want to get UPP on mac if I can. As much as I love xcode, it's huge, and I'd like something freely available on windows as well as mac so that anyone can pick up my code and use it as thoroughly as possible with as little intervention from me or anyone else as possible.

You have a wonderful thing going here. It's simple, light, and potent. And so help me, I love the ALT-C thingy
