Journal Report 17

2/10/20-2/14/20
Mikail Khan
Computer Systems Research Lab
Period 5, White

Daily Log

Monday February 10

I started adding a way to actually create wavelets which is difficult because each wavelet needs to be part of the main physics world as well as the collision world. There's also some extra stuff that the collision library wants that's annoying such as rotation, which will matter when I'm adding rectangular barriers etc., but for wavelets is unnecessary.

Wednesday February 12

I added basic drawing of wavelets, integration, and started adding collisions on the border of the screen. I'm using verlet integration again, and I think that there should be no integration error since the acceleration will either be constant or instantaneous due to a collision. It's funny how much code there is to make such simple bounces work; I usually write something similar just to test game engine libraries, but because of all the stuff I'm planning to add there's a lot of baggage.

Friday February 14

I started working on collisions with other waves. I don't think they should be hard collisions; I think there should be some overlap between wavelets, but I added a strength component to each wavelet so that the repulsion force can scale differently. Hopefully this is where the collision detection library becomes useful, because I think that normal radius check collisions would be pretty slow when there's hundreds of wavelets.

Timeline

Date	Goal	Met
Today minus 2 weeks	FBDs	Acceleration graphs
Today minus 1 weeks	Start wave lab sim	Yes
Today	Basic wavelets and collisions	Wall collisions work and wavelet collisions against each other are mostly done but not complete
Today plus 1 week	Working wave simulation	
Today plus 2 weeks	Adjustable Barriers	

Reflection

In narrative style, talk about your work this week. Successes, failures, changes to timeline, goals. This should also include concrete data, e.g. snippets of code, screenshots, output, analysis, graphs, etc.

The wave sim is starting to make some progress. Wavelet on wavelet collisions might take some work to get right but it should be doable. I think I got most of the actual simulation boilerplate stuff out of the way so progress should be getting faster. I was able to copy paste a lot of stuff from the gravity sim without too much adjustment since they're similar in that almost everything is circles. I did some research on reducing integration error in n body gravity sims and found that it's impossible to reduce to zero. With a set single body orbit it's easy, but with more than 2 bodies in the system you have to use numerical integration .I didn't look too far into the math but apparently n-body systems can be modeled through Hamiltonian mechanics for which you need a symplectic integrator. Verlet is the 2nd degree symplectic integrator but there is a fourth degree integrator which would have much lower error, but still not perfect, and, more importantly, time is not reversible. It's still possible to use a fourth degree symplectic integrator for reverse time, but you have to hardcode it to use different equations depending on the sign of the time. Assuming that a fourth degree integrator is about as accurate as verlet with four times the timestep, I'm not sure that purely based off of integration it would be faster since the equations are so much more complex. However, the biggest computational cost in the simulation is collision checking, so a fourth degree integrator would definitely make the simulator overall more accurate. I'll probably switch to the fourth degree integrator, but honestly I don't think it will make a difference for users since you need such a highly elliptical orbit for it to matter. Timestep and number of timesteps to run per frame are adjustable though.