Journal Report 3
9/16/19-9/20/19
Mikail Khan
Computer Systems Research Lab
Period 5, White

Daily Log

Monday September 2

I added friction. I had some issues with balls rolling backwards because some of the cases from the equation in the paper I was reading were wrong but I was able to logic it out.

Wednesday September 4

I added some basic input. First, I made it possible to drag around shapes with the mouse, and then I added a pause button. Dragging around shapes worked surprisingly well. At first I was just setting the shape's position to the mouse position, but then I switched it to setting the velocity equal to the offset between the mouse and the shape. Right now, you can only pick up shapes from the center (and the mouse snaps there) because I haven't added moment of inertia for arbitrary shapes/around points other than the center. Pausing was expectedly easy; the only slight problem is that you can drag shapes off the screen while paused since collisions don't happen.
Adding input made some of the problems with the simulation much more obvious. Right now, even though shapes resting on the ground are stable, they will vibrate a lot when stacked on top of each other. I spent the rest of the class trying to debug that.

Friday September 6

I had worked on trying to figure out why objects resting on top of each other vibrated so much on Thursday and all I'd really figured out was that turning friction off fixed it. During class, I found out that the reason is that when objects are very slightly overlapping friction gets messed up. To fix that, I started implementing continuous collision detection, which is when you use the exact time of collision to compute results instead of the time after the time step. I ran into some problems with memory management because the size of arbitrary shapes in memory is unknown, but the bigger problem is that I'm don't yet know what to do once I have the time of impact.

Timeline

Date	Goal	Met
Today minus 2 weeks	Rotational elastic collisions	Yes
Today minus 1 weeks	Resting collisions and gravity/friction	I thought that I'd had resting collisions working, but they only really work for objects resting on the ground. Friction is done
Today	Functional 2D mechanics with some basic input	Basic input, but I need to implement continuous collision detection (CCD)
Today plus 1 week	Continuous collision detection	I had planned to use this week to add a proper GUI, but I think adding CCD will take a while
Today plus 2 weeks	Proper GUI	

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.

It was really satisfying to add basic input and play pong against gravity, but it's pretty annoying that blocks won't stack properly. At first I thought that it would be a relatively easy fix, but adding CCD might be as hard as getting rotational collisions to work was. This sets me a week behind but I'm still a bit ahead of the general timeline I made last year (it does say that the mechanics simulation should be done by the end of next week, but because I've designed everything with a GUI in mind I don't think I'll need the three weeks I allocated).
Looking at last year's timeline/proposal, I really heavily based it on my gravity simulator. I'd thought that the GUI/input would be the hardest part, but adding the basic controls that I did on Wednesday really only took less than half an hour. A lot of the reason it's so much easier to add controls here is that I've been keeping it in mind the whole way whereas with my gravity simulator I didn't really have any sort of plan since I just wanted to learn Rust.
I've known about CCD for a while and have implemented it for e.g. Pong and Brickbreaker, but it's obviously way harder to think about when there's many shapes that might have more than one point of contact, along with rotational velocity etc. I thought that I wouldn't have to implement it here because things shouldn't go very fast and I'm working with a timestep of 300hz, but the combination of friction and rotational velocity make it necessary.