# Springs

1. Implement all of the ForceGenerators in Chapter 6 except for those in the Stiff Springs section.
2. Make a clear, usable, two-dimensional demo of each generator.
   1. Make multiple demos if you cannot find a clear way to mash them all into one UI.
   2. Reuse generators for multiple particles if possible.
3. Extra credit worth no points: make a chain of particles the user can interact with, pulling the first particle around.
   1. More extra credit: make a 3D version.

Questions:

1. Explain Hook’s law.
   1. The force a spring gives is directly related to how far away from its “sweet-zone” the spring is extended/compressed. F = -k\*dt
2. What issues exist with stiff springs? How do we compensate for them?
   1. When you are dealing with really large spring constants (stiff springs), then when the spring is compressed (even for 1 frame), that next interval between updates the spring will push out an immense amount of acceleration which will look very wrong.
3. What optimization can we perform if two objects are attached by the same spring?
   1. Reuse the
4. How does the force for spring compression differ from spring extension?
   1. It doesn’t really—it’s just that the force the spring exerts on the objects is negative.
5. What is a limit of elasticity?
   1. The range of lengths that a spring will follow hook’s law. Beyond this range the spring will deform.
6. Something about Millington’s ParticleSpring class smells. Any idea what?
   1. He shouldn’t take the absolute value of the magnitude because otherwise the spring-compression force won’t work.
   2. Two particles attached by a spring should be able to be kept track of by the same spring—we shouldn’t need 2 separate springs for both objects.