Lab 6 Report

Javier Ramirez Moyano
Perception & Multimedia Computing

Write-up Questions

Part A

1.- What happens to the velocity of the object? How about the acceleration?

Just like in the real world, in this sketch, the acceleration of the object due to gravity remains constant through time, increasing the velocity of the object.

4.- Now you've created the basics of a shooting game. Can you think about other ways to make it more interactive and interesting? What other physics elements would you want to make it a fully developed game?

To make this game more interactive, we could add targets, both static and moving, that the user must attempt to hit with the cannonball. Another interesting physics element that could also be added would be collisions so that when two or more cannonballs collide, they bounce off each other with a velocity proportionate to their collision.

Part B

1.- When you change the parameters (M, K, D, R), how does the animation change?

Increasing the parameter M (mass) makes the motion of the spring slower and briefer, while lowering it makes it faster and last longer. The effect on the animation is the opposite for the parameter K (spring constant). Increasing K makes the motion of the spring faster, and it takes slightly longer to go back to its original shape. Decreasing it, on the other hand, makes the motion slower and a little bit shorter. Changes in parameter D (damping) seem to have the most drastic effect on the animation. Even a small change in this variable makes a really big change in the way the animation behaves. For example, increasing the parameter by 0.06 makes the spring oscillate for a much longer time while decreasing it by the same amount makes the oscillation very short, making the spring go back to its original shape almost instantly. Parameter R (rest position) does not change the oscillation time or speed, but it does change the position to which the spring returns after this oscillation.

3.- Can you think about more creative ways of using the springs to generate some interesting physical simulation effects? Describe at least one situation where you might use springs in your work.

One case in which springs could be used to create a physical simulation effect would be to create an animation of a ball hitting a surface, changing its shape, and then recovering its original shape after the impact. They could also be used in any simulation of an object that has spring-like qualities, for example, a car or motorcycle's suspension.

Part C

1.- How are these five steps implemented in this example?

The first step, generating new particles, is implemented in the *draw()* function, where a new particle is created by the function *addParticle()* every time the *draw()* function loops. Every time a new particle is created, new attributes are given to it in the *Particle* class constructor (step 2). These attributes include acceleration, velocity, position, and lifespan. In the *draw()* function, the *run()* function is called as well. *Run()* calls *particle.run()* on every particle in the system using a for loop, which calls the *update()* and *display()* functions, updating the attributes of every particle and displaying the particles with the updated attributes, therefore implementing steps 4 and 5. After calling the *particle.run()* function, the *run()* function checks if the recently updated particles have exceeded their allocated lifespan, and if they have, it removes them using the *splice()* function, implementing step 3.

2.- What are the different modes of this particle system? How are they implemented? When the mode changes, what changes for each particle?

The different modes are attract mode, repel mode, nearest points mode, and noise mode. In the *ofApp.cpp* file *setup()* function, the initial mode is set to *PARTICLE_MODE_ATTRACT*. In this same file, the *keyPressed()* function enables mode changes by pressing different keys. In the *demoParticle.cpp* file, the *update()* and *draw()* functions treat the particles differently depending on their mode by using conditional statements. These different modes change how the particles behave on the screen. For example, in attract mode, the particles will be attracted to the mouse, while in repel mode, it's the exact opposite.