PHZ4151C: Exercise 4

Computational Physics Lab
Due February 15

For each problem you will write one or more programs using the VPython module. For this exercise, programs will be developed and executed on the classroom Mac computers as the HPC system does not support VPython. These programs should follow our coding and formatting conventions outlined for our course. You must hand in copies of the program codes and any additional required materials.

In addition, you must submit the Python programs as an archive tgz file via email to phz4151c@hadron.physics.fsu.edu. On the HPC system, created a directory with the name <last_name>-exercise4/ where <last_name> is your last name. From the classroom Mac computers, use the secure copy command ***scp*** to copy your final VPython programs to your HPC <last_name>-exercise4/ directory then follow the usual procedure submitting an archive tgz file.

Place copies of only your Python programs in a directory called <last_name>-exercise4/where <last_name> is your last name and use similar command as provided in earlier exercises.

1. Visualization of the Solar System: The innermost six planets of our solar system revolve around the Sun in roughly circular orbits that all lie approximately in the same (ecliptic) plane. Here are some basic parameters:

Object	Radius of object (km)	Radius of orbit (millions of km)	Period of orbit (days)
Mercury	2440	57.9	88
Venus	6052	108.2	224.7
Earth	6371	149.6	365.3
Mars	3386	227.9	687
Jupiter	69173	778.5	4331.6
Saturn	57316	1433.4	10759.2
Sun	695500	–	–

Using the facilities provided by the visual package, create an animation of the solar system that shows the following:

a) The Sun and planets as spheres in their appropriate positions and with sizes proportional to their actual sizes. Because the radii of the planets are tiny compared to the distances between them, represent the planets by spheres with radii c_1 times larger than their correct proportionate values, so that you can see them clearly. Find a good value for c_1 that makes the planets visible. You'll also need to find a good radius for the Sun. Choose any value that gives a clear visualization. (It doesn't work to scale the radius of the Sun by the same factor you use for the planets, because it'll come out looking much too large. So just use whatever works.) For added realism, you may also want to make your spheres different colors. For instance, Earth could be blue and the Sun could be yellow.

b) The motion of the planets as they move around the Sun (by making the spheres of the planets move). In the interests of alleviating boredom, construct your program so that time in your animation runs a factor of c_2 faster than actual time. Find a good value of c_2 that makes the motion of the orbits easily visible but not unreasonably fast. Make use of the rate function to make your animation run smoothly.

Hint: You may find it useful to store the sphere variables representing the planets in an array of the kind described in the textbook on page 115.

For full credit turn in a printout of your final program.

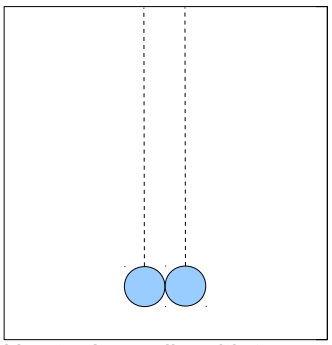
2. Visualizing a Classical Helium Atom: The classical view of a Helium atom consists of a nucleus of two protons and two neutrons being orbited by two electrons. Using the facilities provided by the visual package, create an animation of a classical Helium atom that shows the following:

a) A core nucleus consisting of two red spheres and two blue spheres representing the protons and neutrons.

b) Electrons spherically orbiting the nucleus where the orbiting paths cross orthogonally.

For full credit turn in a printout of your final program.

3. Newton's Cradle: Named after Sir Isaac Newton, Newton's cradle is a device that aptly demonstrates conservation of energy and momentum using a series of swinging spheres. Let's consider a cradle of just two spheres of equal weight with perfectly efficient elasticity and neglecting energy losses. When a sphere is lifted and released, it strikes the stationary sphere; a force is impacted onto the stationary spheres, transferring all the velocity to the sphere. The initially stationary sphere moves away with the impact velocity while the initially moving sphere becomes stationary which indicates all the momentum and energy are also transferred. The kinetic energy, as determined by the velocity, is converted to potential energy as it reaches the same height as the initial height and the cycle repeats.



Newton's cradle with two spheres at rest

a) Using the facilities provided by the visual package, create an animation of a two sphere Newton's cradle. Only be concerned with the motion of the two spheres(i.e. assume that the spheres are suspended using a massless transparent tether). Choose different colors for the spheres and a sphere radii approximately one-tenth of the pendulum length. Have the animation start with one sphere at an angle 0.5 radians from the vertical and let it evolve continuously. Hint: start off by animating simple harmonic motion of a single pendulum ($\theta(t)=\theta_0\cos(\sqrt{g/L}*t)$).

b) A more realistic animation should include frictional energy loses. So now create a new program, by modifying your original program, that approximates frictional damping. Model the damping by having the amplitude of the harmonic motion decay exponentially as $\theta_0=\theta_{0_{initial}}e^{-\mu t}$. Have your program initially set the value of μ to zero but allow the program to obtain a new value for μ from the command line. Use your program to determine an appropriate value for μ such that the animation exhibits about 12 collisions between the spheres before the motion is approximately stopped.

For full credit turn in a printout of your final programs and the value of μ which results in approximately 12 collisions.