# First VPython Programming Assignment

To hand these in, you should:

* Create a folder for each assignment
* In the upper left hand corner of that folder, if it says Private click to toggle it to Public.
* Put all your programs for this assignment in this folder, one for each problem. If you need to revise an existing program for another question, copy it, rename it and edit the copy.
* You will hand them in on the Assignments tab in D2L by sharing a link to this folder.

The best approach would be for each of you to do your own program, even if unsuccessful, and then compare within your group.

There are some instructional videos that are based on a previous version of VPython. There is also extensive documentation under “Help.” In particular, you can see the syntax for setting up objects like spheres and boxes. These are on the drop down lists on the left side of the window. There are also a number of example programs for you to look at.

The purpose here is to learn the basics of creating objects, looping, and deciding between alternatives.

1. View the first introductory video 3D Objects at vpython.org/video01.html and

1. Complete the challenge at the end.
2. Then write a program that represents the x, y and z axes by three long thin boxes of different colors that intersect at the center of the screen. In VPython, the z axis sticks out of the screen toward you. You can rotate the view by right-clicking and dragging anywhere in the scene.

2. View the second VPython video Variable Assignment at vpython.org/video02.html and complete the challenge at the end.

3. Watch the third VPython video Beginning Loops at vpython.org/video03.html and

1. Complete the challenge activity at the end.
2. Make a copy of your program from problem 1b and modify it so that the coordinate axes are represented by small boxes separated by spaces (e.g. like dotted lines). Think about using a loop to draw many boxes.

4. Watch the fourth VPython video Loops and Animation at vpython.org/video04.html and

1. complete the challenge activity at the end.
2. Consider the following VPython program:

from visual import \*

#make two objects

redbox = box(pos=vec(0,0,-1), size = vec(5, 5, 0.5), color = color.red, opacity = 0.4)

particle = sphere(pos=vec(-5, 0, -5), radius = 0.3, color = color.cyan, make\_trail = True)

#initialize variables; create a new velocity property for the particle object

particle.v = vec(0.5, 0, 0.5)

dt = 0.05

t=0

#animation loop

while t < 20:

#control the animation frame rate

rate(100)

#move the particle

particle.pos = particle.pos + particle.v \* dt

t = t + dt

See if you can figure out what this program will do before running it. Then modify it to start the particle at an initial position to the right of and in front of the red box, traveling to the left so that it passes in front of and parallel to the box. NOTE: you can’t just copy and paste this. Word introduces a lot of nonprinting characters that VPython will interpret as errors. In particular, a Word tab is not the same as a VPython indent. As usual, Microsoft lives in their own special little world.

5. Make a copy of the program from 4 and adjust the position and velocity of the particle as needed so that it collides with the surface of the red box at an angle and bounces off. You’ll need to check for when it reaches the red box. If you aren’t familiar with Python, you’ll have to search the web for “python if” or look at the example program called Bounce.