Getting Going with VPython

Steve Highland

A quick start on free (literally) and easy 3D simulations

Learn all about and get VPython by going to http://www.vpython.org/. It's *free*.

I found VPython pretty easy to get started with when I first heard of it from Ruth Chabay and Bruce Sherwood a few years ago at a summer AAPT meeting.

The brilliant idea behind this 3D programming environment is that all the "dirty work" of creating the objects and displaying them in a real-time interactive virtual world is done behind the scenes with virtually no work from you.

Installing VPython is a trivial two-step process on a PC, and a more complicated task on a Mac (I failed to make it work last fall, but maybe it's time I tried again).

→ You can manipulate the view while a VPython program is running, **zooming (click and drag both mouse buttons)** and **rotating (right click and drag)** the view any way you like.

On our PCs, get to VPython from the Start menu by going
Start menu --> Programs --> Python 2.4 --> IDLE (Python GUI)
This brings up an empty IDLE window that should look like

>   Untitled
>   File Edit Format Run Options Windows Help

Once you start up IDLE (supposedly an acronym for "Interactive DeveLopment Environment" but really a clever joke because Python is named after Monty Python, not a snake) you are ready to type program code into the window. The first program line is always—

```
from visual import *
```

To go further you'll need some help, but you'll find good tutorials and reference manuals at vpython.org. Demo programs are also included when you install VPython. Good luck!

Other things to know:
Pressing F5 will run the program in the IDLE window. (It will also automatically save the file over whatever previous version you last executed, so if you make changes and don't want to lose the old version, save it under a different name.)

To stop a program, just click the close box in the 3D display window. (If you close all the windows you'll have to start up IDLE again and retrieve your file from "recent files.")

Error messages show up in red in the (other) output window. Sometimes they are partly hidden by the 3D display window, so watch out.

Python can do seemingly weird things with integer division, so it is wise to type "one half" as 1.0/2.0 rather than ½. I just type all my physical quantity values with a decimal point to avoid trouble.

Program code for "x squared" is `x**2`.

I'd love to see your efforts at programming, and if you want to play with any of mine, just email me at highlasr@buffalostate.edu or shighlan@uslink.net.