

BIS 420 PROGRAMMING FOR DATA SCIENCE
PRAJAKTA POHARE
CHAPTER 17 EXERCISE 17.8
ILLINOIS STATE UNIVERSITY

Visual is a Python module that provides 3-D graphics. It is not always included in a Python installation, so you might have to install it from your software repository or, if it's not there, from <http://vpython.org>.

The following example creates a 3-D space that is 256 units wide, long and high, and sets the “center” to be the point (128, 128, 128). Then it draws a blue sphere.

```
from visual import *
```

```
scene.range = (256, 256, 256)
```

```
scene.center = (128, 128, 128)
```

```
color = (0.1, 0.1, 0.9) # mostly blue
```

```
sphere(pos=scene.center, radius=128, color=color)
```

color is an RGB tuple; that is, the elements are Red-Green-Blue levels between 0.0 and 1.0 (see http://en.wikipedia.org/wiki/RGB_color_model).

If you run this code, you should see a window with a black background and a blue sphere. If you drag the middle button up and down, you can zoom in and out. You can also rotate the scene by dragging the right button, but with only one sphere in the world, it is hard to tell the difference.

The following loop creates a cube of spheres:

```
t = range(0, 256, 51)
```

```
for x in t:
```

```
    for y in t:
```

```
        for z in t:
```

```
            pos = x, y, z
```

```
                sphere(pos=pos, radius=10, color=color)
```

1. Put this code in a script and make sure it works for you.
2. Modify the program so that each sphere in the cube has the color that corresponds to its

position in RGB space. Notice that the coordinates are in the range 0–255, but the RGB tuples are in the range 0.0–1.0.

3. Download http://thinkpython.com/code/color_list.py and use the function `read_colors` to generate a list of the available colors on your system, their names and RGB values. For each named color draw a sphere in the position that corresponds to its RGB values. You can see my solution at http://thinkpython.com/code/color_space.py.

```
from vpython import *
```

```
scene.range = 256
```

```
scene.center = vector(128, 128, 128)
```

```
color_rgb = vector(0.1, 0.1, 0.9) # blue
```

```
t = range(0, 256, 51)
```

```
for x in t:
```

```
    for y in t:
```

```
        for z in t:
```

```
            pos = vector(x, y, z)
```

```
            sphere(pos=pos, radius=10, color=color_rgb)
```

```
from vpython import *
```

```
scene.range = 256
```

```
scene.center = vector(128, 128, 128)
```

```
t = range(0, 256, 51)
```

```

for x in t:
    for y in t:
        for z in t:
            pos = vector(x, y, z)
            color_rgb = vector(x / 255, y / 255, z / 255)
            sphere(pos=pos, radius=10, color=color_rgb)

```

```

('CornflowerBlue', (0.392, 0.584, 0.929))

```

```

from vpython import *
from color_list import read_colors

```

```

scene.range = 1
scene.center = vector(0.5, 0.5, 0.5)

```

```

colors = read_colors()

```

```

for name, (r, g, b) in colors:
    pos = vector(r, g, b)
    sphere(pos=pos, radius=0.01, color=vector(r, g, b))

```

```

from vpython import *

scene.range = 256
scene.center = vector(128, 128, 128)

color_rgb = vector(0.1, 0.1, 0.9) # blue

t = range(0, 256, 51)
for x in t:
    for y in t:

```

```

        for z in t:
            pos = vector(x, y, z)
            sphere(pos=pos, radius=10, color=color_rgb)

from vpython import *

scene.range = 256
scene.center = vector(128, 128, 128)

t = range(0, 256, 51)
for x in t:
    for y in t:
        for z in t:
            pos = vector(x, y, z)
            color_rgb = vector(x / 255, y / 255, z / 255)
            sphere(pos=pos, radius=10, color=color_rgb)

('CornflowerBlue', (0.392, 0.584, 0.929))

from vpython import *
from color_list import read_colors

scene.range = 1
scene.center = vector(0.5, 0.5, 0.5)

colors = read_colors()

for name, (r, g, b) in colors:
    pos = vector(r, g, b)
    sphere(pos=pos, radius=0.01, color=vector(r, g, b))

```