```
# All lines like this one that begin with "#" are
# comments.
# All other lines are program statements.

# The next line is an internal revision control id:
# $Date: 2007/11/08 17:19:19 $ $Revision: 1.2 $
# Copyright (c) 2007 David M. Harrison

# Import the visual library.
from visual import *

# These four lines control the size of the window of
# the animation and the scale. The details of
# these lines are not important for our purposes.
scene.autoscale = 0
scene.height = 400
scene.width = 800
scene.range = vector(60, 60, 60)
```

```
# Create the green ball that will execute simple
# harmonic motion by numerical integration.
greenBall = sphere (color = color.green, radius = 2)

# yellowBall will execute simple harmonic motion
# using a sine function.
yellowBall = sphere (color = color.yellow, radius =
2)

# The initial x position of the balls: this is
# the equilibrium position.
x = 0
```

```
# Position the balls. pos is a built-in of VPython,
# and lists the (x,y,z) coordinates. The x axis
# is horizontal, y axis is vertical, and the z
# axis is perpendicular to the plane of the screen.
# We place the green ball just below the center of
# the scene, at y - -10.
greenBall.pos = (x,-10,0)

# yellowBall is above the first ball: it's y
# coordinate is 10, just above the center
# of the scene.
yellowBall.pos = (x, 10, 0)

# The initial x component of the velocity of
# the balls: all other components are zero.
vx = 150

# The spring constant
k = 9.0
```

```
# The mass of the balls
mass = 1.0

# The amplitude of yellowBall's motion
ampl = sqrt(mass/k) * vx

# The time
t = 0

# This is the time step
dt = 0.005
```

```
# This causes the following indented lines
# to be executed forever in a loop.
while 1 == 1:

    # Set the rate of the animation
    rate(1/dt)

    # The acceleration in the x direction.
    a = -(k/mass) * x

    # Update the speed using the acceleration. Note
    # that we "recycle" the variable vx,
    # replacing the old value with the new one.
    vx = vx + a*dt

    # Update the x position of the ball using
    # the speed.
    x = x + vx*dt
```

```
# Position greenBall at the new x position
greenBall.pos = (x, -10, 0)

# Update the time
t = t + dt

# Now we calculate simple harmonic motion using
# a sine function and position yellowBall
# using the result of the calculation
x2 = ampl * sin( sqrt(k/mass)* t)
yellowBall.pos = (x2,10,0)
```