```
## CSUMS 2-Body Problem CSUMS Spring 2013 A. O. Hausknecht
from visual import *
from visual.graph import *
#Create the scene
scene1 = display(title='2-Body Problem',
                 x=0, y=0, width=700, height=700, center=(0,0,4))
## Data Set
b1Mass = 1.5e30
b1P0   = vector(        0, 4.5e10, 0)
b1V0   = vector(   1.0e10,      0, 0) # Change 1: 1.0e10 to 0 => Looping orbits
b1Radius = 0.5e10
#
b2Mass = 1.5e30  # Change 2: Assuming Change 1 above,
                 # change 1.5e30 to 1.5e20 => Near Circular orbit
b2P0 = vector(      0, -4.5e10, 0)
b2V0 = vector(-1.0e10,       0, 0)
b2Radius = 0.5e10
#
def initialize():
    global body1, body2, g1, g2
    global t, dt, G
    #
    t = 0; dt = 0.1  # Time variables
    G = 6.67e-11     # Universal Constant of Gravitation

    # Create body1
    body1 = sphere(pos = b1P0 , radius = b1Radius, color=color.yellow,
                   make_trail=True, trail_type="curve")
    body1.m = b1Mass
    body1.v = b1V0
    body1.a = 0

    # Create body2
    body2 = sphere(pos = b2P0, radius = b2Radius, color=color.red,
                   make_trail=True, trail_type="curve")
    body2.m = b2Mass
    body2.v = b2V0
    body2.a = 0

    # Create plots of the bodies speeds
    gdisplay(x=700, y=0, width=700, height=700,
             title='speed vs. t', xtitle='t', ytitle='||v||',
      foreground=color.black, xmin = 0, xmax = 100)
    g1 = gcurve(color=color.yellow)
    g2 = gcurve(color=color.red)
```

```
46 #
47 def update():
48     global t
49     t += dt #Increment time
50     dirVector = body2.pos - body1.pos #Direction Vector
51     distSqr = mag2(dirVector) # Distance between bodies Squared
52     norm(dirVector) # Normalize the direction vector
53 
54     body1.a = G*body2.m/distSqr*dirVector # Acceleration of body1
55     body1.v += body1.a*dt    #Approximate velocity of body1 via Euler's Method
56     body1.pos += body1.v*dt  #Approximate position of body1 via Euler's Method
57     g1.plot(pos = (t, mag(body1.v)))
58 
59     body2.a = -G*body1.m/distSqr*dirVector # Acceleration of body2
60     body2.v += body2.a*dt    #Approximate velocity of body2 via Euler's Method
61     body2.pos += body2.v*dt  #Approximate position of body1 via Euler's Method
62     g2.plot(pos = (t, mag(body2.v)))
63 #
64 # Animation loop
65 initialize()
66 while (True):
67    rate(20)
68    update()
69 
70 
71 
72 
73 
74 
75 
```