import turtle  
import math  
  
  
class SolarSystem:  
 def \_\_init\_\_(self, width, height):  
 self.thesun = None  
 self.planets = []  
 self.ssturtle = turtle.Turtle()  
 self.ssturtle.hideturtle()  
 self.ssscreen = turtle.Screen()  
 self.ssscreen.setworldcoordinates(-width / 2.0, -height / 2.0, width / 2.0, height / 2.0)  
 self.ssscreen.tracer(50)  
  
 def addPlanet(self, aplanet):  
 self.planets.append(aplanet)  
  
 def addSun(self, asun):  
 self.thesun = asun  
  
 def showPlanets(self):  
 for aplanet in self.planets:  
 print(aplanet)  
  
 def freeze(self):  
 self.ssscreen.exitonclick()  
  
 def movePlanets(self):  
 G = .1  
 dt = .001  
  
 for p in self.planets:  
 p.moveTo(p.getXPos() + dt \* p.getXVel(), p.getYPos() + dt \* p.getYVel())  
  
 rx = self.thesun.getXPos() - p.getXPos()  
 ry = self.thesun.getYPos() - p.getYPos()  
 r = math.sqrt(rx \*\* 2 + ry \*\* 2)  
  
 accx = G \* self.thesun.getMass() \* rx / r \*\* 3  
 accy = G \* self.thesun.getMass() \* ry / r \*\* 3  
  
 p.setXVel(p.getXVel() + dt \* accx)  
  
 p.setYVel(p.getYVel() + dt \* accy)  
  
  
class Sun:  
 def \_\_init\_\_(self, iname, irad, im, itemp):  
 self.name = iname  
 self.radius = irad  
 self.mass = im  
 self.temp = itemp  
 self.x = 0  
 self.y = 0  
  
 self.sturtle = turtle.Turtle()  
 self.sturtle.shape("circle")  
 self.sturtle.color("yellow")  
  
 def getName(self):  
 return self.name  
  
 def getRadius(self):  
 return self.radius  
  
 def getMass(self):  
 return self.mass  
  
 def getTemperature(self):  
 return self.temp  
  
 def getVolume(self):  
 v = 4.0 / 3 \* math.pi \* self.radius \*\* 3  
 return v  
  
 def getSurfaceArea(self):  
 sa = 4.0 \* math.pi \* self.radius \*\* 2  
 return sa  
  
 def getDensity(self):  
 d = self.mass / self.getVolume()  
 return d  
  
 def setName(self, newname):  
 self.name = newname  
  
 def \_\_str\_\_(self):  
 return self.name  
  
 def getXPos(self):  
 return self.x  
  
 def getYPos(self):  
 return self.y  
  
  
class Planet:  
  
 def \_\_init\_\_(self, iname, irad, im, idist, ivx, ivy, ic):  
 self.name = iname  
 self.radius = irad  
 self.mass = im  
 self.distance = idist  
 self.x = idist  
 self.y = 0  
 self.velx = ivx  
 self.vely = ivy  
 self.color = ic  
  
 self.pturtle = turtle.Turtle()  
 self.pturtle.up()  
 self.pturtle.color(self.color)  
 self.pturtle.shape("circle")  
 self.pturtle.goto(self.x, self.y)  
 self.pturtle.down()  
  
 def getName(self):  
 return self.name  
  
 def getRadius(self):  
 return self.radius  
  
 def getMass(self):  
 return self.mass  
  
 def getDistance(self):  
 return self.distance  
  
 def getVolume(self):  
 v = 4.0 / 3 \* math.pi \* self.radius \*\* 3  
 return v  
  
 def getSurfaceArea(self):  
 sa = 4.0 \* math.pi \* self.radius \*\* 2  
 return sa  
  
 def getDensity(self):  
 d = self.mass / self.getVolume()  
 return d  
  
 def setName(self, newname):  
 self.name = newname  
  
 def show(self):  
 print(self.name)  
  
 def \_\_str\_\_(self):  
 return self.name  
  
 def moveTo(self, newx, newy):  
 self.x = newx  
 self.y = newy  
 self.pturtle.goto(newx, newy)  
  
 def getXPos(self):  
 return self.x  
  
 def getYPos(self):  
 return self.y  
  
 def getXVel(self):  
 return self.velx  
  
 def getYVel(self):  
 return self.vely  
  
 def setXVel(self, newvx):  
 self.velx = newvx  
  
 def setYVel(self, newvy):  
 self.vely = newvy  
  
  
def createSSandAnimate():  
 ss = SolarSystem(2, 2)  
  
 sun = Sun("SUN", 5000, 10, 5800)  
 ss.addSun(sun)  
  
 m = Planet("MERCURY", 19.5, 1000, .25, 0, 2, "blue")  
 ss.addPlanet(m)  
  
 m = Planet("EARTH", 47.5, 5000, 0.3, 0, 2.0, "green")  
 ss.addPlanet(m)  
  
 m = Planet("MARS", 50, 9000, 0.5, 0, 1.63, "red")  
 ss.addPlanet(m)  
  
 m = Planet("JUPITER", 100, 49000, 0.7, 0, 1, "black")  
 ss.addPlanet(m)  
  
 m = Planet("Pluto", 1, 500, 0.9, 0, .5, "orange")  
 ss.addPlanet(m)  
  
 m = Planet("Asteroid", 1, 500, 1.0, 0, .75, "cyan")  
 ss.addPlanet(m)  
  
 def movePlanets(self):  
 G = .1  
 dt = .001  
  
 for p in self.planets:  
 p.moveTo(p.getXPos() + dt \* p.getXVel(), p.getYPos() + dt \* p.getYVel())  
  
 rx = self.thesun.getXPos() - p.getXPos()  
 ry = self.thesun.getYPos() - p.getYPos()  
 r = math.sqrt(rx \*\* 2 + ry \*\* 2)  
  
 accx = G \* self.thesun.getMass() \* rx / r \*\* 3  
 accy = G \* self.thesun.getMass() \* ry / r \*\* 3  
  
 p.setXVel(p.getXVel() + dt \* accx)  
  
 p.setYVel(p.getYVel() + dt \* accy)  
  
  
class Sun:  
 def \_\_init\_\_(self, iname, irad, im, itemp):  
 self.name = iname  
 self.radius = irad  
 self.mass = im  
 self.temp = itemp  
 self.x = 0  
 self.y = 0  
  
 self.sturtle = turtle.Turtle()  
 self.sturtle.shape("circle")  
 self.sturtle.color("yellow")  
  
 def getName(self):  
 return self.name  
  
 def getRadius(self):  
 return self.radius  
  
 def getMass(self):  
 return self.mass  
  
 def getTemperature(self):  
 return self.temp  
  
 def getVolume(self):  
 v = 4.0 / 3 \* math.pi \* self.radius \*\* 3  
 return v  
  
 def getSurfaceArea(self):  
 sa = 4.0 \* math.pi \* self.radius \*\* 2  
 return sa  
  
 def getDensity(self):  
 d = self.mass / self.getVolume()  
 return d  
  
 def setName(self, newname):  
 self.name = newname  
  
 def \_\_str\_\_(self):  
 return self.name  
  
 def getXPos(self):  
 return self.x  
  
 def getYPos(self):  
 return self.y  
  
  
class Planet:  
  
 def \_\_init\_\_(self, iname, irad, im, idist, ivx, ivy, ic):  
 self.name = iname  
 self.radius = irad  
 self.mass = im  
 self.distance = idist  
 self.x = idist  
 self.y = 0  
 self.velx = ivx  
 self.vely = ivy  
 self.color = ic  
  
 self.pturtle = turtle.Turtle()  
 self.pturtle.up()  
 self.pturtle.color(self.color)  
 self.pturtle.shape("circle")  
 self.pturtle.goto(self.x, self.y)  
 self.pturtle.down()  
  
 def getName(self):  
 return self.name  
  
 def getRadius(self):  
 return self.radius  
  
 def getMass(self):  
 return self.mass  
  
 def getDistance(self):  
 return self.distance  
  
 def getVolume(self):  
 v = 4.0 / 3 \* math.pi \* self.radius \*\* 3  
 return v  
  
 def getSurfaceArea(self):  
 sa = 4.0 \* math.pi \* self.radius \*\* 2  
 return sa  
  
 def getDensity(self):  
 d = self.mass / self.getVolume()  
 return d  
  
 def setName(self, newname):  
 self.name = newname  
  
 def show(self):  
 print(self.name)  
  
 def \_\_str\_\_(self):  
 return self.name  
  
 def moveTo(self, newx, newy):  
 self.x = newx  
 self.y = newy  
 self.pturtle.goto(newx, newy)  
  
 def getXPos(self):  
 return self.x  
  
 def getYPos(self):  
 return self.y  
  
 def getXVel(self):  
 return self.velx  
  
 def getYVel(self):  
 return self.vely  
  
 def setXVel(self, newvx):  
 self.velx = newvx  
  
 def setYVel(self, newvy):  
 self.vely = newvy  
  
  
def createSSandAnimate():  
 ss = SolarSystem(2, 2)  
  
 sun = Sun("SUN", 5000, 10, 5800)  
 ss.addSun(sun)  
  
 m = Planet("MERCURY", 19.5, 1000, .25, 0, 2, "blue")  
 ss.addPlanet(m)  
  
 m = Planet("EARTH", 47.5, 5000, 0.3, 0, 2.0, "green")  
 ss.addPlanet(m)  
  
 m = Planet("MARS", 50, 9000, 0.5, 0, 1.63, "red")  
 ss.addPlanet(m)  
  
 m = Planet("JUPITER", 100, 49000, 0.7, 0, 1, "black")  
 ss.addPlanet(m)  
  
 m = Planet("Pluto", 1, 500, 0.9, 0, .5, "orange")  
 ss.addPlanet(m)  
  
 m = Planet("Asteroid", 1, 500, 1.0, 0, .75, "cyan")  
 ss.addPlanet(m)  
  
 numTimePeriods = 20000  
 for amove in range(numTimePeriods):  
 ss.movePlanets()  
 ss.freeze()  
createSSandAnimate()