N-Body.py

```
importmatplotlib.pyplotasplt
importnumpyasnp
fromnumpyimportinf
importtime
defacceleration(limit,G,M,R):
xx=R[:,0:1]
yy=R[:,1:2]
zz=R[:,2:3]
Dx=xx.T-xx
Dy=yy.T-yy
Dz=zz.T-zz
#3x3->[[0,x2-x1,x3-x1],[x1-x2,0,x3-x2],[x1-x3,x2-x3,0]]

invR3=(Dx**2+Dy**2+Dz**2+limit**2)
#allelementsofinvR3raisedtopower-3/2
invR3=np.power(invR3,-1.5)
#replacingallinfinitiesprodcuedbyexponentiationprocess
invR3[invR3==inf]=0

Ax=G*(Dx*invR3)@M
Ay=G*(Dy*invR3)@M
Az=G*(Dz*invR3)@M
A=np.hstack((Ax,Ay,Az))
returnA


defmain():
N=100##ofparticles
t=0#time
et=3.0#endtime
dt=0.01#timestep
limit=0.1#limitlength
G=1.0#valueofGravitationalConstantusedforthesimulation

np.random.seed(int(time.time()))

M=100.0*np.ones((N,1))/N#totalMofparticlesis100
R=np.random.randn(N,3)#randomlyselectedpositionsandvelocities
vel=np.random.randn(N,3)
#hangonframeofc.o.m
vel-=np.mean(M*vel,0)/np.mean(M)
acc=acceleration(limit,G,M,R)

#timestep
Nt=int(np.ceil(et/dt))
pos=np.zeros((N,3,Nt+1))
pos[:,:,0]=R

plt.style.use('dark_background')
```

```
foriinrange(Nt):
#leapfrogintegration
vel+=acc*dt/2.0

#drift
R+=vel*dt

acc=acceleration(limit,G,M,R)
vel+=acc*dt/2.0
t+=dt
#updateandsavepositionsforplottingtrail
pos[:,:,i+1]=R

plt.cla()
xp=pos[:,0,max(i-50,0):i+1]
yp=pos[:,1,max(i-50,0):i+1]
#plotting
plt.scatter(xp,yp,s=1,color='red')
plt.scatter(R[:,0],R[:,1],s=10,color='blue')
plt.pause(0.001)
plt.show()
return0
if__name__=="__main__":
main()
```

Screen clipping taken: 3/29/2023 12:07 PM