```
'''
# READ BEFORE YOU START:

#Please don't change anything on the code beside your own planet.
#If you want to apply any change you must state that on the Discord.
# Please make a nice and sensible naming convention for your variables
#Please keep the code structures nice and tidy
# please do all the little edits on your own platform and then make sure it
works on your own device. and then after
that copy/pase them here
#For the most part do not run anything on google colab, because it is not
powerful and it crashes.
instead run it on your own system (or ask to check that if you prefer)

'''

#
==========================================================================
==
#
==========================================================================
==
# Team 7:
# START OF PHASE 1
#
==========================================================================
==
#
==========================================================================
==

#
==========================================================================
=================
# Team 7 : Define Your own corresponding constants here, use nice naming
convention as outlines
#
==========================================================================
=================

import numpy as np

G = 6.67e-11
Ms = 2.0e30  # sun
Me = 5.972e24  # earth
Mm = 6.39e23  # mars
Mc = 6.39e20  # unknown comet
AU = 1.5e11
daysec = 24.0 * 60 * 60  # days to seconds
```

```
e_ap_v = 29290  # earth velocity at aphelion
m_ap_v = 21970  # mars velocity at aphelion
commet_v = 7000


# top part of gravitational force equation is gravconst for each planet



gravconst_e = G * Me * Ms
gravconst_m = G * Mm * Ms
gravconst_c = G * Mc * Ms

# ======================================================================
# Team 7 : Setup Your own corresponding planet's starting Condition
# ======================================================================


# Star a   ################# Caution: Now this is our star! Not sun anymore!
Use this unstead of Ms
M_a = 0.67 * Ms
xa, ya, za = 0, 0, 0
xva, yva, zva = 0, 0, 0




# planet b
# to do.....
M_b = 8.32 * Me  # mass of b
a_b = 0.1162 * AU  # semimajor axis for b
per_b = 17.667087 * daysec  # period for b
e_b = 0.072  # eccentricity of orbit

gravconst_b = G * M_b * M_a

xb, yb, zb = 1.5 * AU, 0, 0  # position of b at aphelion
xvb, yvb, zvb = 0, 20000, 0  # velocity of b at aphelion

# planet c
# to do....
M_pc = 3.41 * Me  # mass of c
a_pc = 1646 * AU  # semimajor axis for c
per_pc = 29.79749 * daysec  # period for c
e_pc = 0.063  # eccentricity of orbit

gravconst_pc = G * M_pc * M_a
```

```
x_pc, y_pc, z_pc = 2 * AU, 0, 0
x_vc, y_vc, z_vc = 0, 17000, 0
# be wary bc c is defined for comet sections, maybe change those before
assigning c to planet


# Planet d
# to do.....
M_d = 0.55 * Me  # Planet d
a_d = 0.04298 * AU  # semimajor axis for D
per_d = 12.162183 * daysec  # period for planet d in seconds
e_d = 0.0700  # eccentricity
d_ap_v = 0.29
gravconst_d = G * M_d * M_a

x_d, y_d, z_d = 0.65 * AU, 0, 0
x_vd, y_vd, z_vd = 0, 20000, 0
# x_df, y_df, z_df = a_d, 0, 0
# x_vdf, y_vdf, z_vdf = 0, d_ap_v, 0 not sure if i need these i copied these
from template but using equations makes more sense to me

# Planet e
# To do.....
M_e0 = 0.72 * Me
a_e0 = 0.0680 * AU
per_e0 = 7.90754 * daysec
e_e0 = 0.07

gravconst_e0 = G * M_e0 * M_a
xe0, ye0, ze0 = 0.85 * AU, 0, 0
xve0, yve0, zve0 = 0, 22000, 0

# Planet f
# to do.....
M_f = 0.770 * Me
a_f = 0.0906 * AU
f_ap_v = 0.29  # m/s
gravconst_f = G * M_f * M_a
xf, yf, zf = 1 * AU, 0, 0  #
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
xvf, yvf, zvf = 0, 25000, 0  #
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

```
#
================================================================================
================================
# Team 7: please disregard this comet stuff ^^^^ that you see here and other
places in the code for the moment
# We'll get back to this if we got time
#
================================================================================
================================

# sun
xa, ya, za = 0, 0, 0
xva, yva, zva = 0, 0, 0

t = 0.0
dt = 1 * daysec  # every frame move this time

# ========================================================
# Team 7, please define your intial lists for your planets
# ========================================================



# star a
xalist, yalist, zalist = [], [], []


# planet b
xblist, yblist, zblist = [], [], []

# planet c
# to do....
xpclist, ypclist, zpclist = [], [], []

# Planet d
xdlist, ydlist, zdlist = [], [], []

# Planet e
xe0list, ye0list, ze0list = [], [], []

# Planet f
xflist, yflist, zflist = [], [], []

# ============================================================
# Team 7: Start your own planet's simulation here:
# ============================================================

# start simulation
while t < 5 * 365 * daysec:
```

```
# planet b
# to do.....
# g-force on planet b
rx_b, ry_b, rz_b = xb - xa, yb - ya, zb - za
modr3_b = (rx_b ** 2 + ry_b ** 2 + rz_b ** 2) ** 1.5
fx_b = -gravconst_b * rx_b / modr3_b
fy_b = -gravconst_b * ry_b / modr3_b
fz_b = -gravconst_b * rz_b / modr3_b

xvb += fx_b * dt / M_b
yvb += fy_b * dt / M_b
zvb += fz_b * dt / M_b

xb += xvb * dt
yb += yvb * dt
zb += zvb * dt

xblist.append(xb)
yblist.append(yb)
zblist.append(zb)

# planet c
# to do....
rx_pc, ry_pc, rz_pc = x_pc - xa, y_pc - ya, z_pc - za
modr3_pc = (rx_pc ** 2 + ry_pc ** 2 + rz_pc ** 2) ** 1.5
fx_pc = -gravconst_pc * rx_pc / modr3_pc
fy_pc = -gravconst_pc * ry_pc / modr3_pc
fz_pc = -gravconst_pc * rz_pc / modr3_pc

x_vc += fx_pc * dt / M_pc
y_vc += fy_pc * dt / M_pc
z_vc += fz_pc * dt / M_pc

# update position
x_pc += x_vc * dt
y_pc += y_vc * dt
z_pc += z_vc * dt

# add to list
xpclist.append(x_pc)
ypclist.append(y_pc)
zpclist.append(z_pc)

# Planet d
# to do.....
rx_d, ry_d, rz_d = x_d - xa, y_d - ya, z_d - za
modr3_d = (rx_d ** 2 + ry_d ** 2 + rz_d ** 2) ** 1.5
```

```
fx_d = -gravconst_d * rx_d / modr3_d
fy_d = -gravconst_d * ry_d / modr3_d
fz_d = -gravconst_d * rz_d / modr3_d

x_vd += fx_d * dt / M_d
y_vd += fy_d * dt / M_d
z_vd += fz_d * dt / M_d

# update position
x_d += x_vd * dt
y_d += y_vd * dt
z_d += z_vd * dt

# add to list
xdlist.append(x_d)
ydlist.append(y_d)
zdlist.append(z_d)

# Planet e
# To do.....
# compute G force on planet e
rx_e0, ry_e0, rz_e0 = xe0 - xa, ye0 - ya, ze0 - za
modr3_e0 = (rx_e0 ** 2 + ry_e0 ** 2 + rz_e0 ** 2) ** 1.5
fx_e0 = -gravconst_e0 * rx_e0 / modr3_e0
fy_e0 = -gravconst_e0 * ry_e0 / modr3_e0
fz_e0 = -gravconst_e0 * rz_e0 / modr3_e0

xve0 += fx_e0 * dt / M_e0
yve0 += fy_e0 * dt / M_e0
zve0 += fz_e0 * dt / M_e0

# update position
xe0 += xve0 * dt
ye0 += yve0 * dt
ze0 += zve0 * dt

# add to list
xe0list.append(xe0)
ye0list.append(ye0)
ze0list.append(ze0)

# Planet f
# to do.....
# compute G force on planet f
rx_f, ry_f, rz_f = xf - xa, yf - ya, zf - za
modr3_f = (rx_f ** 2 + ry_f ** 2 + rz_f ** 2) ** 1.5
fx_f = -gravconst_f * rx_f / modr3_f
fy_f = -gravconst_f * ry_f / modr3_f
fz_f = -gravconst_f * rz_f / modr3_f
```

```
    xvf += fx_f * dt / M_f
    yvf += fy_f * dt / M_f
    zvf += fz_f * dt / M_f

    # update position
    xf += xvf * dt
    yf += yvf * dt
    zf += zvf * dt

    # add to list
    xflist.append(xf)
    yflist.append(yf)
    zflist.append(zf)

    # the sun
    # update quantities how is this calculated?  F = ma -> a = F/m
    xva += -(fx_b + fx_pc + fx_d + fx_e0 + fx_f) * dt / M_a
    yva += -(fy_b + fy_pc + fy_d + fy_e0 + fy_f) * dt / M_a
    zva += -(fz_b + fz_pc + fz_d + fz_e0 + fz_f) * dt / M_a

    # update position
    xa += xva * dt
    ya += yva * dt
    za += zva * dt
    xalist.append(xa)
    yalist.append(ya)
    zalist.append(za)

    # update dt
    t += dt

## update the above section and use a instead of s for our system

print('data ready')

# print(xalist,yalist)


#
==========================================================================
==
#
==========================================================================
==
# Team 7 :
# END OF PHASE 1
# START OF PHASE 2
```

```
#
================================================================================
==
#
================================================================================
==

import numpy as np
import matplotlib.pyplot as plt
from matplotlib import animation

fig = plt.figure(figsize=(10, 10))
ax = plt.axes(projection='3d')
ax.axis('auto')

# ================================================================
# Team 7 : Here start to Define your Axis and correspondance:
# ================================================================

axis_size = 8  #
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
ax.set_xlim(-axis_size * AU * (1 / 4), axis_size * AU * (1 / 4))  #
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
ax.set_ylim(-axis_size * AU * (1 / 4), axis_size * AU * (1 / 4))  #
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
ax.set_zlim(-axis_size * AU * (1 / 4), axis_size * AU * (1 / 4))  #
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

# ax.set_aspect('auto')
# ax.grid()
datadict = {}

# sun
dataset_a = [xalist, yalist, zalist]


# planet b
dataset_b = [xblist, yblist, zblist]

# planet c
# to do....
dataset_pc = [xpclist, ypclist, zpclist]

# Planet d
# to do.....
dataset_d = [xdlist, ydlist, zdlist]

# Planet e
```

```
# To do.....
dataset_e0 = [xe0list, ye0list, ze0list]

# Planet f
# to do.....
dataset_f = [xflist, yflist, zflist]

# sun
datadict['a'] = dataset_a


# planet b
datadict['b'] = dataset_b

# planet c
# to do....
datadict['pc'] = dataset_pc

# Planet d
# to do.....
datadict['d'] = dataset_d

# Planet e
# To do.....
datadict['e0'] = dataset_e0

# Planet f
# to do.....
datadict['f'] = dataset_f

vis_dict = {}

# ============================================================
# Team 7: Here is the fun part, do your planet's drawings here:
# ============================================================


# star a
line_a, = ax.plot([0], [0], [0], '-g', lw=1)
point_a, = ax.plot([AU], [0], [0], marker="o", markersize=10,
markeredgecolor="#ff3300", markerfacecolor="#ff3300")
text_a = ax.text(AU, 0, 0, 'A')
vis_dict['a'] = [line_a, point_a, text_a]



# planet b
line_b, = ax.plot([0], [0], [0], linestyle='-', color="#0000ff", lw=1)
```

```
point_b, = ax.plot([AU], [0], [0], marker="o", markersize=7,
markeredgecolor="#0000ff", markerfacecolor="#0000ff")
text_b = ax.text(AU, 0, 0, 'B')
vis_dict['b'] = [line_b, point_b, text_b]

# planet c
# to do....
line_pc, = ax.plot([0], [0], [0], linestyle='-', color="#00cc00", lw=1)
point_pc, = ax.plot([AU], [0], [0], marker="o", markersize=5,
markeredgecolor="#00cc00", markerfacecolor="#00cc00")
text_pc = ax.text(AU, 0, 0, 'C')
vis_dict['pc'] = [line_pc, point_pc, text_pc]

# Planet d
# to do.....
line_d, = ax.plot([0], [0], [0], linestyle='-', color="#000000", lw=0.5)
point_d, = ax.plot([AU], [0], [0], marker="o", markersize=1.5,
markeredgecolor="#000000", markerfacecolor="#000000")
text_d = ax.text(AU, 0, 0, 'D')
vis_dict['d'] = [line_d, point_d, text_d]

# Planet e
# To do.....
line_e0, = ax.plot([0], [0], [0], '-g', lw=0.5)
point_e0, = ax.plot([AU], [0], [0], marker="o", markersize=4,
markeredgecolor="green", markerfacecolor="green")
text_e0 = ax.text(AU, 0, 0, 'E')
vis_dict['e0'] = [line_e0, point_e0, text_e0]

# Planet f
# to do.....
line_f, = ax.plot([0], [0], [0], linestyle='-', color="#ff33cc", lw=0.5)
point_f, = ax.plot([AU], [0], [0], marker="o", markersize=4,
markeredgecolor="#ff33cc", markerfacecolor="#ff33cc")
text_f = ax.text(AU, 0, 0, 'F')
vis_dict['f'] = [line_f, point_f, text_f]


# ================================================================================
# team 7: The animation's update happens here, please add your planet as well
# ================================================================================


def update(num, data_dict, vis_dict):
    # sun
    dataset_a = data_dict['a']
    line_a, point_a, text_a = vis_dict['a'][0], vis_dict['a'][1],
vis_dict['a'][2]
```

```
    line_a.set_data_3d(dataset_a[0][:num], dataset_a[1][:num],
dataset_a[2][:num])
    point_a.set_data_3d(dataset_a[0][num], dataset_a[1][num], dataset_a[2][num])
    text_a.set_position((dataset_a[0][num], dataset_a[1][num],
dataset_a[2][num]))


    # planet b
    dataset_b = data_dict['b']
    line_b, point_b, text_b = vis_dict['b'][0], vis_dict['b'][1],
vis_dict['b'][2]
    line_b.set_data_3d(dataset_b[0][:num], dataset_b[1][:num],
dataset_b[2][:num])
    point_b.set_data_3d(dataset_b[0][num], dataset_b[1][num], dataset_b[2][num])
    text_b.set_position((dataset_b[0][num], dataset_b[1][num],
dataset_b[2][num]))
    # planet c
    # to do....
    dataset_pc = data_dict['pc']
    line_pc, point_pc, text_pc = vis_dict['pc'][0], vis_dict['pc'][1],
vis_dict['pc'][2]
    line_pc.set_data_3d(dataset_pc[0][:num], dataset_pc[1][:num],
dataset_pc[2][:num])
    point_pc.set_data_3d(dataset_pc[0][num], dataset_pc[1][num],
dataset_pc[2][num])
    text_pc.set_position((dataset_pc[0][num], dataset_pc[1][num],
dataset_pc[2][num]))

    # Planet d
    # to do.....
    dataset_d = data_dict['d']
    line_d, point_d, text_d = vis_dict['d'][0], vis_dict['d'][1],
vis_dict["d"][2]
    line_d.set_data_3d(dataset_d[0][:num], dataset_d[1][:num],
dataset_d[2][:num])
    point_d.set_data_3d(dataset_d[0][num], dataset_d[1][num], dataset_d[2][num])
    text_d.set_position((dataset_d[0][num], dataset_d[1][num],
dataset_d[2][num]))

    # Planet e
    # To do.....
    dataset_e0 = data_dict['e0']
    line_e0, point_e0, text_e0 = vis_dict['e0'][0], vis_dict['e0'][1],
vis_dict['e0'][2]
    line_e0.set_data_3d(dataset_e0[0][:num], dataset_e0[1][:num],
dataset_e0[2][:num])
    point_e0.set_data_3d(dataset_e0[0][num], dataset_e0[1][num],
dataset_e0[2][num])
```

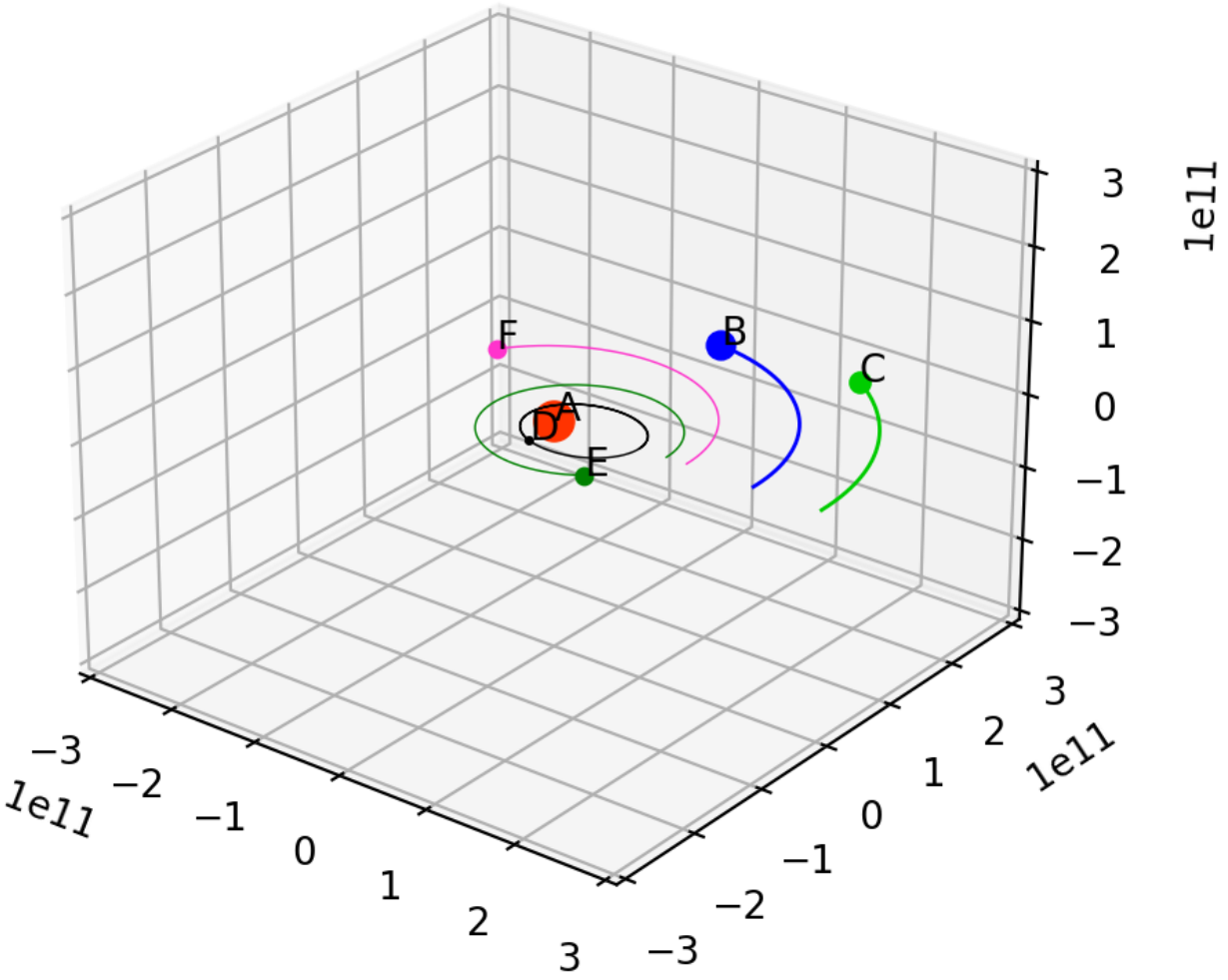
```
   text_e0.set_position((dataset_e0[0][num], dataset_e0[1][num],
dataset_e0[2][num]))

   # Planet f
   # to do.....
   dataset_f = data_dict['f']
   line_f, point_f, text_f = vis_dict['f'][0], vis_dict['f'][1],
vis_dict['f'][2]
   line_f.set_data_3d(dataset_f[0][:num], dataset_f[1][:num],
dataset_f[2][:num])
   point_f.set_data_3d(dataset_f[0][num], dataset_f[1][num], dataset_f[2][num])
   text_f.set_position((dataset_f[0][num], dataset_f[1][num],
dataset_f[2][num]))


#
================================================================================
==
#
================================================================================
==
# Team 7 :
# END OF PHASE 2
# (START OF PHASE 3??? Maybe???)
#
================================================================================
==
#
================================================================================
==


ani = animation.FuncAnimation(
   fig
   , update
   , len(xflist)
   , fargs=(datadict, vis_dict)
   , interval=1
)

plt.show()
```


F
B
C
A
D
E
3
2
1
0
−1
−2
−3
1e11
−3
−2
−1
0
1
2
3
1e11
3
2
1
0
−1
−2
−3
1e11