```python
import sys
sys.path.append('../modules')

from modules import rayTracing as rt
import numpy as np
import project_inputs as pi
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from Basilisk.utilities import RigidBodyKinematics as rbk
from tabulate import tabulate

exp = rt.Experiment()
rays = pi.edge_ray
rays = pi.basical_paraxial_rays
exp.set_ray_starts(rays.X)
exp.set_ray_start_dir(rays.d)

inst = pi.cass
inst.surfaces[-1] = pi.grating
inst.surfaces.append(pi.cylindrical_detector)
# inst.surfaces = inst.surfaces[-2:]
exp.add_instrument(inst)

wavelength_list = np.arange(1200., 2100., 100.)
grating = inst.surfaces[-2]
detector = inst.surfaces[-1]
data_list = []
colors = ['violet', 'indigo', 'blue', 'green', 'yellow', 'orange', 'red', 'pink', 'black',
            'violet', 'indigo', 'blue', 'green', 'yellow', 'orange', 'red', 'pink', 'black',
            'violet', 'indigo', 'blue', 'green', 'yellow', 'orange', 'red', 'pink', 'black']

grating.set_order(0)
for wavelength in wavelength_list:
    grating.set_wavelength(wavelength)
    exp.reset()
    exp.trace_rays()
    data_list.append(detector.extract_image(exp.ray_hist[-1]))
fig0 = rt.save_3d_plot(inst.surfaces, exp.ray_hist)

grating.set_order(1)
for wavelength in wavelength_list:
    grating.set_wavelength(wavelength)
    exp.reset()
    exp.trace_rays()
    data_list.append(detector.extract_image(exp.ray_hist[-1]))

angstrom_per_mm = 1E7 / 3600. / 1000.
x1600 = data_list[-5][0, :]
dx_1600 = (np.nanmax(x1600) - np.nanmin(x1600)) * 1000.
resolution_1600 = dx_1600 * angstrom_per_mm
resolving_power_1600 = 1600. / resolution_1600
print('resolving power at 1600 A: %f' % resolving_power_1600)
x1200 = data_list[-9][0, :]
dx_1200 = (np.nanmax(x1200) - np.nanmin(x1200)) * 1000.
resolution_1200 = dx_1200 * angstrom_per_mm
resolving_power_1200 = 1200. / resolution_1200
print('resolving power at 1200 A: %f' % resolving_power_1200)
x2000 = data_list[-1][0, :]
```

```
 59 dx_2000 = (np.nanmax(x2000) - np.nanmin(x2000)) * 1000.
 60 resolution_2000 = dx_2000 * angstrom_per_mm
 61 resolving_power_2000 = 2000. / resolution_2000
 62 print('resolving power at 2000 A: %f' % resolving_power_2000)
 63 headers = ['Wavelength', 'Resolving Power']
 64 data = np.array([resolving_power_1200, resolving_power_1600,
    resolving_power_2000]).reshape([3, 1])
 65 index = np.array([1200, 1600, 2000]).reshape([3, 1])
 66 dat = np.hstack([index, data])
 67 table = tabulate(dat, headers, tablefmt='latex')
 68 with open('./resolvingTable.txt', 'w') as f:
 69     f.write(table)
 70     f.close()
 71 fig1 = rt.save_3d_plot(inst.surfaces, exp.ray_hist)
 72 plt.savefig('./figures/lab_view.png')
 73
 74 grating.set_order(2)
 75 for wavelength in wavelength_list:
 76     grating.set_wavelength(wavelength)
 77     exp.reset()
 78     exp.trace_rays()
 79     data_list.append(detector.extract_image(exp.ray_hist[-1]))
 80
 81 scat_fig = plt.figure(figsize=(20, 5))
 82
 83 ax2 = scat_fig.add_subplot('111')
 84 for order in range(3):
 85     for col, wavelength in zip(colors, data_list):
 86         # lab = 'order %s, wavelength %s' % (order, wavelength)
 87         ax2.scatter(wavelength[0, :], wavelength[1, :]*100, s=1,
    color=col)
 88 ax2.set_xlabel('RA [rad]')
 89 ax2.set_ylabel('height (along cylinder axis) [cm]')
 90 ax2.legend([str(int(w)) for w in wavelength_list], markerscale=6,
    title='[A]')
 91 ax2.text(0.575, 2., '0 order', bbox=dict(facecolor='black', alpha=0.
    05))
 92 ax2.text(0., 2., 'first order', bbox=dict(facecolor='black', alpha=0.
    05))
 93 ax2.text(-0.6, 2., 'second order', bbox=dict(facecolor='black', alpha
    =0.05))
 94 plt.savefig('./figures/spectrum.png')
 95 plt.show()
 96
 97
 98 # fig = plt.figure()
 99 # ax = fig.add_subplot('111', projection='3d')
100 # rt.plot_surfaces(ax, inst.surfaces)
101 # ax.set_xlabel('x')
102 # ax.set_ylabel('y')
103 # ax.set_zlabel('z')
104 # ax.set_aspect('equal')
105 # plt.show()
106
107
108
109
```