```
# -*- coding: utf-8 -*-
#  thomas.nirrengarten

import matplotlib.pyplot as plt
import numpy as np
import RGB as RGB



i=1 #interfrange pour la plus petite longueur d'onde du spectre
x = np.linspace(-20*i,20*i,1000) #positions sur l'écran
wavelength=np.arange(380,780,1) #longueurs d'onde du spectre (nm)
nb_lambda=len(wavelength) #nombre de longueurs d'onde étudiées

###Initialisation de la fenêtre###

fig, ax = plt.subplots(2, 2,
                       gridspec_kw={
                           'width_ratios': [3, 1],
                           'height_ratios': [1, 1]},figsize=(14,14))
fig.set_facecolor('white')
ax1 = ax[0][0]
ax2 = ax[1][0]
ax3 = ax[0][1]
ax4 = ax[1][1]
fig.subplots_adjust(wspace=0.2,hspace=0.2)

###Initialisation des différents tracés###

ax1.set_xlabel(u"Position x sur l'écran (unité arbitraire)",fontsize=11)
ax1.set_ylabel(r"Intensité (ua)",fontsize=11)
ax1.set_yticks(np.linspace(0.,1,3, endpoint=True))
ax1.set_title("Eclairement total",fontsize=16)
ax1.axis([-20*i,20*i,0,1])
line_tot, = ax1.plot([],[],color='red',linewidth=1.0,linestyle='-',label='Itot')
ax1.legend(loc='upper right',fancybox=True,shadow=True)
title = ax1.text(0.5,0.88, "", bbox={'facecolor':'w', 'alpha':0.5, 'pad':5},
                 transform=ax1.transAxes, ha="center",fontsize=16)


ax2.set_xticks([])
ax2.set_yticks([])

ax3.set_title('Spectre au point cliqué')
ax3.set_xlabel("Longueur d'onde (nm)")
ax3.set_yticks([])
ax3.set_xticks(np.arange(21,400,100))
ax3.set_xticklabels((400,500,600,700))
posX=int(len(x)/2)

ax4.set_title('Spectre au point cliqué')
ax4.set_xlabel("Longueur d'onde (nm)")
ax4.set_ylabel('Intensité (ua)')
ax4.set_yticks([])
ax4.set_xticks(np.arange(21,400,100))
ax4.set_xticklabels((400,500,600,700))
ax4.set_xlim(0,nb_lambda)
ax4.set_ylim(0,1.5)
line_spec, = ax4.plot([],[],color='green',linewidth=2.0,linestyle='-')

###Calcul des différentes intensités

y=np.ones(200)
y_new=y[:,np.newaxis,np.newaxis]
RGBvalues=np.zeros((nb_lambda,3))
for i in np.arange(nb_lambda):
    RGBvalues[i,:]=RGB.WavelengthToRGB(wavelength[i])
```

```
RGBtot=np.sum(RGBvalues,axis=0)
RGBtot2D=np.ones(len(x))[:,np.newaxis]*RGBtot[np.newaxis,:]
RGBvalues_new=RGBvalues[np.newaxis,:,:]
wavelength_new=wavelength[np.newaxis,:]
xnew=x[:,np.newaxis]
Itot=0.5+0.5*np.cos(2*np.pi*xnew*wavelength[0]/wavelength_new)
ItotRGB=Itot[:,:,np.newaxis]*RGBvalues_new
ItotRGB_sum=np.sum(ItotRGB,axis=1)
ItotRGB_sum_norm=ItotRGB_sum/RGBtot2D
ItotRGB_sum_norm=ItotRGB_sum_norm[np.newaxis,:,:]
Itot_sum=np.sum(Itot,axis=1)
Itot_sum_norm=Itot_sum/np.max(Itot_sum)

I_spectre=ItotRGB[posX,:,:]
I_spectre_new=I_spectre[np.newaxis,:,:]
I_spectre_ecran=I_spectre_new*y_new
Iecran=ItotRGB_sum_norm*y_new

###Tracés initiaux des 4 subplots###

line_tot.set_data(x,Itot_sum_norm)

im2=ax2.imshow(Iecran)

im3=ax3.imshow(I_spectre_ecran)

line_spec.set_data(np.arange(nb_lambda),Itot[posX,:])

###Gestion d'un clic sur la figure d'interférences###

def onclick(event):
    posX=int(event.xdata)
    I_spectre=ItotRGB[posX,:,:]
    I_spectre_new=I_spectre[np.newaxis,:,:]
    I_spectre_ecran=I_spectre_new*y_new
    im3.set_array(I_spectre_ecran)
    line_spec.set_data(np.arange(nb_lambda),Itot[posX,:])
    fig.canvas.draw_idle()

cid = fig.canvas.mpl_connect('button_press_event',onclick)
```