import tkinter as tk

import tkinter.ttk as ttk

class CompLamp:

    def \_\_init\_\_(self, parent, width, order, color="red", \*args, \*\*kwargs):

        self.frame = ttk.Frame(parent.frame, \*args, \*\*kwargs)

        self.canvas = tk.Canvas(self.frame, width=width, height=width, bg="gray",highlightthickness=0)

        self.canvas.pack()

        self.color = color

        offset = width//8

        self.lamp = self.canvas.create\_oval(offset, offset,7\*offset,7\*offset,fill='black')

        self.frame.grid(row=order, column=0)

        self.state = "off"

    def turn\_on(self):

        self.state = "on"

        self.canvas.itemconfigure(self.lamp, fill=self.color)

    def turn\_off(self):

        self.state = "off"

        self.canvas.itemconfigure(self.lamp, fill='black')

class CompTrafficLight:

    def \_\_init\_\_(self, root, wd, initial\_color="red", \*args, \*\*kwargs):

        if initial\_color not in ("red", "yellow", "green"):

            raise ValueError(initial\_color + " is not a valid color")

        self.frame = ttk.Frame(root, width=wd, \*args, \*\*kwargs)

        self.frame.grid(row=0, column=0)

        self.color = initial\_color

        self.lamps = dict(zip(('red', 'yellow', 'green'),(CompLamp(self, wd, 0, 'red'),CompLamp(self, wd, 1, 'yellow'),CompLamp(self, wd, 2, 'green'))))

        self.lamps[self.color].turn\_on()

    def change(self):

        if self.color == 'red':

           new\_color = 'green'

        elif self.color == 'green':

           new\_color = 'yellow'

        elif self.color == 'yellow':

           new\_color = 'red'

        self.lamps[self.color].turn\_off()

        self.color = new\_color

        self.lamps[self.color].turn\_on()

    def resize(self, width):

        for lamp in self.lamps.values():

            lamp.resize(width)