Copilot

Absolutely, we can modify the code to reflect the enabled state of the “Add” button and to delete a circle when the “Delete” button is enabled and the mouse is clicked inside a circle. Here’s how you can do it:

```
import tkinter as tk

class Application(tk.Frame):
    def __init__(self, master=None):
        super().__init__(master)
        self.master = master
        self.pack()
        self.create_widgets()

    def create_widgets(self):
        self.add_button = tk.Button(self)
        self.add_button["text"] = "Add"
        self.add_button["command"] = self.enable_add
        self.add_button.pack(side="left")

        self.delete_button = tk.Button(self)
        self.delete_button["text"] = "Delete"
        self.delete_button["command"] = self.enable_delete
        self.delete_button.pack(side="right")

        self.canvas = tk.Canvas(self, width=400, height=400)
        self.canvas.pack()
        self.canvas.bind("<Button-1>", self.draw_or_delete_circle)

        self.adding = False
        self.deleting = False

    def enable_add(self):
        self.adding = True
        self.deleting = False
        self.add_button.config(relief="sunken")
        self.delete_button.config(relief="raised")

    def enable_delete(self):
        self.adding = False
        self.deleting = True
        self.add_button.config(relief="raised")
        self.delete_button.config(relief="sunken")

    def draw_or_delete_circle(self, event):
        x, y = event.x, event.y
        r = 10  # radius
        overlapping = self.canvas.find_overlapping(x-r, y-r, x+r, y+r)
        if self.adding:
            self.canvas.create_oval(x-r, y-r, x+r, y+r, fill="blue")
        elif self.deleting and overlapping:
            self.canvas.delete(overlapping)

root = tk.Tk()
app = Application(master=root)
app.mainloop()
```

In this updated code, the “Add” button is sunken when it is enabled and the “Delete” button is sunken when it is enabled. This gives a visual indication of which mode is currently active. When the “Delete” button is enabled and the left mouse button is clicked inside a circle, that circle is deleted. The `find_overlapping` method is used to find any circles at the mouse click position, and if any are found, they are deleted. Please note that this will delete all circles that overlap with the click position, not just the topmost one. If you want to delete only the topmost circle, you can use `find_closest` instead of `find_overlapping`.