Copilot

Absolutely, you can modify the `enable_add` and `enable_delete` methods to toggle the enabled state of the buttons when they are clicked. 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(fill="both", expand=True)
        self.create_widgets()

    def create_widgets(self):
        self.button_frame = tk.Frame(self)
        self.button_frame.pack(side="top", fill="x")

        self.add_button = tk.Button(self.button_frame)
        self.add_button["text"] = "Add"
        self.add_button["command"] = self.toggle_add
        self.add_button.pack(side="left")

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

        self.canvas = tk.Canvas(self, bg='white')
        self.canvas.pack(fill="both", expand=True)
        self.canvas.bind("<Button-1>", self.draw_or_delete_circle)
        self.canvas.bind("<Configure>", self.resize_circles)

        self.adding = False
        self.deleting = False
        self.circles = []

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

    def toggle_delete(self):
        if self.deleting:
            self.deleting = False
            self.delete_button.config(relief="raised")
        else:
            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:
            id = self.canvas.create_oval(x-r, y-r, x+r, y+r, fill="blue")
            self.circles.append((id, x, y, r))
        elif self.deleting and overlapping:
            for circle in self.circles:
                if circle[0] in overlapping:
                    self.canvas.delete(circle[0])
                    self.circles.remove(circle)
    def resize_circles(self, event):
        for circle in self.circles:
            id, x, y, r = circle
            self.canvas.coords(id, x-r, y-r, x+r, y+r)

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

In this updated code, I've renamed the `enable_add` and `enable_delete` methods to `toggle_add` and `toggle_delete`, respectively. These methods now check if the corresponding mode is already enabled, and if it is, they disable it; otherwise, they enable it and disable the other mode. Now, when you click a button that is already enabled, it will become disabled, and the other button will remain disabled. I hope this helps! Let me know if you have any other questions.