

Copilot

Sure, I can help with that. Here is a simple example of how you can achieve this using Python's tkinter library:

```
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.disable_add
        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_circle)

        self.adding = False

    def enable_add(self):
        self.adding = True

    def disable_add(self):
        self.adding = False

    def draw_circle(self, event):
        if self.adding:
            x, y = event.x, event.y
            r = 10 # radius
            self.canvas.create_oval(x-r, y-r, x+r, y+r, fill="blue")

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

In this code, we have two buttons: “Add” and “Delete”. When you click the “Add” button, it enables the addition of circles on the canvas with a left mouse button click. When you click the “Delete” button, it disables the addition of circles. The `draw_circle` function is bound to the left mouse button click event on the canvas, and it creates a new circle at the mouse position whenever the adding flag is `True`. The radius of the circle is set to 10, and the color is set to blue. You can adjust these parameters as needed.