

Copilot

Absolutely, you can make the canvas expand to fill the window by using the `fill` and `expand` options of the `pack` method. Here's how you can modify the code:

```
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.enable_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.enable_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.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 `fill` and `expand` options are set to `"both"` and `True`, respectively, for both the `Application` frame and the `canvas`. This makes them expand to fill their parent widget (the window for `Application`, and `Application` for the `canvas`). The `button_frame` is set to fill only in the `x` direction, so it expands horizontally but not vertically. Now, when you resize the window, the `canvas` will resize accordingly. Please note that the circles will not resize when the window is resized; they will stay the same size. If you want the circles to resize as well, you would need to implement additional logic. Also, please note that the circles will not move when the window is resized; they will stay in their original positions relative to the top-left

corner of the canvas. If you want the circles to move when the window is resized, you would need to implement additional logic. I hope this helps! Let me know if you have any other questions.