**Week 7 Module 10.2 Assignment: GUI ToDo**

Juan Macias Vasquez

Bellevue University

CSD325-H323 Advanced Python (2255-DD)

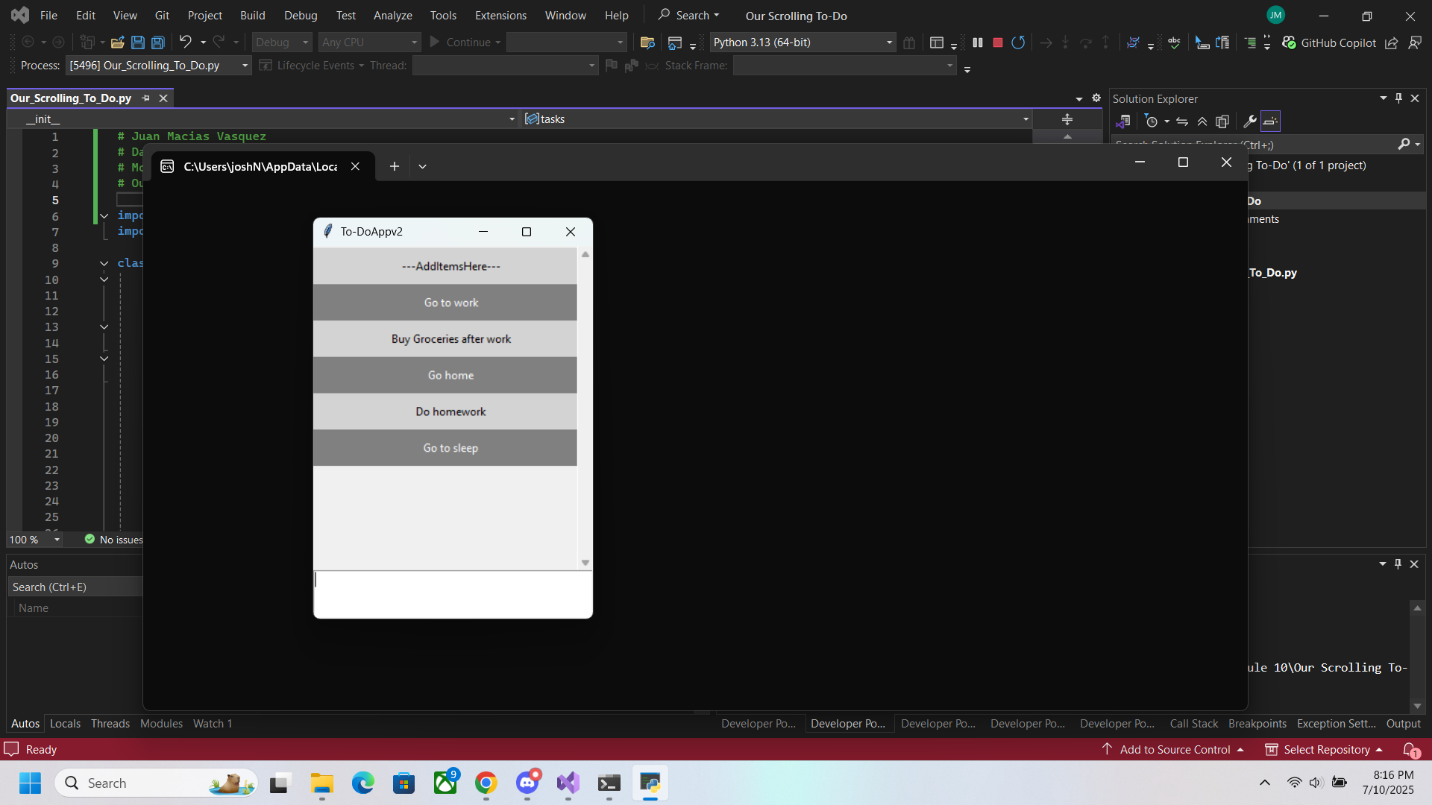
**Jack Lusby**

July 13th, 2025

Module 10.2 Assignment: GUI ToDo

Results Document

Pictures of original code testing



**Code of New version**

Changes to color, easier way to exit, and right click to delete task

# Juan Macias Vasquez

# Date 07/10/2025

# Module 10.2 GUI ToDo

# Macias\_To-Do.py

import tkinter as tk

import tkinter.messagebox as msg

class Todo(tk.Tk):

def \_\_init\_\_(self, tasks=None):

super().\_\_init\_\_()

if not tasks:

self.tasks = []

else:

self.tasks = tasks

# Set title and size

self.title("Macias-ToDo") # Replaced with my last name

self.geometry("300x400")

# Create menu bar

menubar = tk.Menu(self, bg="purple", fg="white")

file\_menu = tk.Menu(menubar, tearoff=0, bg="yellow", fg="black")

file\_menu.add\_command(label="Exit", command=self.destroy)

menubar.add\_cascade(label="File", menu=file\_menu)

self.config(menu=menubar)

# Scrollable canvas and frames

self.tasks\_canvas = tk.Canvas(self)

self.tasks\_frame = tk.Frame(self.tasks\_canvas)

self.text\_frame = tk.Frame(self)

self.scrollbar = tk.Scrollbar(self.tasks\_canvas, orient="vertical", command=self.tasks\_canvas.yview)

self.tasks\_canvas.configure(yscrollcommand=self.scrollbar.set)

self.task\_create = tk.Text(self.text\_frame, height=3, bg="white", fg="black")

self.tasks\_canvas.pack(side=tk.TOP, fill=tk.BOTH, expand=1)

self.scrollbar.pack(side=tk.RIGHT, fill=tk.Y)

self.canvas\_frame = self.tasks\_canvas.create\_window((0, 0), window=self.tasks\_frame, anchor="n")

self.task\_create.pack(side=tk.BOTTOM, fill=tk.X)

self.text\_frame.pack(side=tk.BOTTOM, fill=tk.X)

self.task\_create.focus\_set()

# Instructional label

todo1 = tk.Label(

self.tasks\_frame,

text="--- Items Added --- \*\* Right Click Item to Delete \*\*",# Instructions in the label on how to delete a task.

bg="purple", fg="white", pady=10

)

todo1.bind("<Button-3>", self.remove\_task)

self.tasks.append(todo1)

for task in self.tasks:

task.pack(side=tk.TOP, fill=tk.X)

# Event bindings

self.bind("<Return>", self.add\_task)

self.bind("<Configure>", self.on\_frame\_configure)

self.bind\_all("<MouseWheel>", self.mouse\_scroll)

self.bind\_all("<Button-4>", self.mouse\_scroll)

self.bind\_all("<Button-5>", self.mouse\_scroll)

self.tasks\_canvas.bind("<Configure>", self.task\_width)

self.colour\_schemes = [{"bg": "purple", "fg": "white"}, {"bg": "gold", "fg": "black"}] #Menu and submenu with complementary colors (e.g., purple + yellow/gold).

def add\_task(self, event=None):

task\_text = self.task\_create.get(1.0, tk.END).strip()

if len(task\_text) > 0:

new\_task = tk.Label(self.tasks\_frame, text=task\_text, pady=10)

self.set\_task\_colour(len(self.tasks), new\_task)

new\_task.bind("<Button-3>", self.remove\_task) #Task deletion is triggered by right-click (<Button-3>).

new\_task.pack(side=tk.TOP, fill=tk.X)

self.tasks.append(new\_task)

self.task\_create.delete(1.0, tk.END)

def remove\_task(self, event):

task = event.widget

if msg.askyesno("Really Delete?", "Delete '" + task.cget("text") + "'?"):

self.tasks.remove(task)

task.destroy()

self.recolour\_tasks()

def recolour\_tasks(self):

for index, task in enumerate(self.tasks):

self.set\_task\_colour(index, task)

def set\_task\_colour(self, position, task):

\_, task\_style\_choice = divmod(position, 2)

scheme = self.colour\_schemes[task\_style\_choice]

task.configure(bg=scheme["bg"], fg=scheme["fg"])

def on\_frame\_configure(self, event=None):

self.tasks\_canvas.configure(scrollregion=self.tasks\_canvas.bbox("all"))

def task\_width(self, event):

canvas\_width = event.width

self.tasks\_canvas.itemconfig(self.canvas\_frame, width=canvas\_width)

def mouse\_scroll(self, event):

if event.delta:

self.tasks\_canvas.yview\_scroll(int(-1 \* (event.delta / 120)), "units")

else:

move = 1 if event.num == 5 else -1

self.tasks\_canvas.yview\_scroll(move, "units")

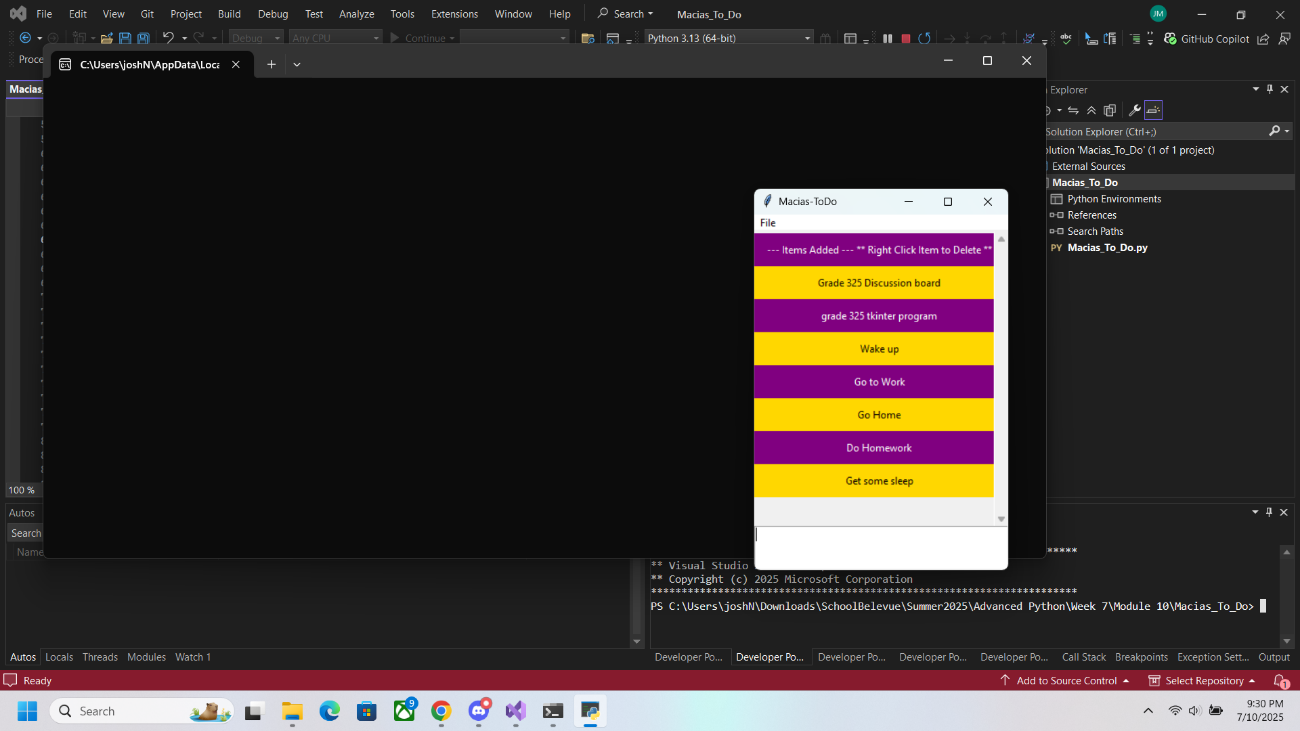
if \_\_name\_\_ == "\_\_main\_\_":

todo = Todo()

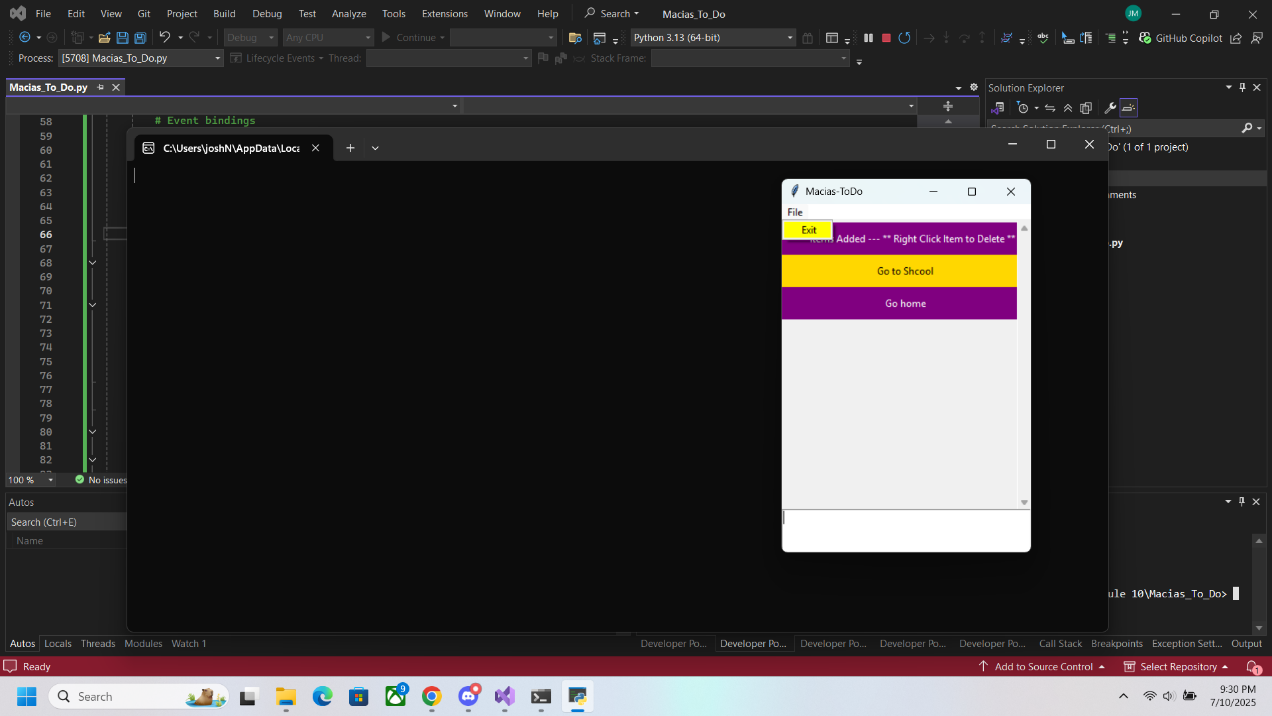
todo.mainloop()

**Pictures of updated version**

Adding to the list with new colors



Showing the new File exit button



Showing the delete task from to do list

