**M08 Software Solution**

**Codes:**

**Main Menu Window:**

import tkinter as tk

from tkinter import messagebox

class MainMenu(tk.Tk):

    def \_\_init\_\_(self):

        super().\_\_init\_\_()

        self.title("Green Team of Captain Game")

        self.geometry("400x300")

        # Labels

        self.label = tk.Label(self, text="Welcome to the Green Team of Captain Game!")

        self.label.pack(pady=10)

        # Buttons

        self.start\_button = tk.Button(self, text="Start Game", command=self.start\_game)

        self.start\_button.pack(pady=5)

        self.instructions\_button = tk.Button(self, text="Instructions", command=self.show\_instructions)

        self.instructions\_button.pack(pady=5)

        self.exit\_button = tk.Button(self, text="Exit", command=self.quit)

        self.exit\_button.pack(pady=5)

    def start\_game(self):

        messagebox.showinfo("Start Game", "Game is starting...")

    def show\_instructions(self):

        messagebox.showinfo("Instructions", "Here are the game instructions...")

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

    app = MainMenu()

    app.mainloop()

**Game Window:**  
import tkinter as tk

class GameWindow(tk.Toplevel):

    def \_\_init\_\_(self, parent):

        super().\_\_init\_\_(parent)

        self.title("Game Window")

        self.geometry("400x300")

        # Labels

        self.label = tk.Label(self, text="Game is in progress...")

        self.label.pack(pady=10)

        # Buttons

        self.pause\_button = tk.Button(self, text="Pause", command=self.pause\_game)

        self.pause\_button.pack(pady=5)

        self.quit\_button = tk.Button(self, text="Quit", command=self.quit\_game)

        self.quit\_button.pack(pady=5)

    def pause\_game(self):

        # Pause game logic here

        pass

    def quit\_game(self):

        self.destroy()

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

    root = tk.Tk()

    root.withdraw()  # Hide the root window

    game\_window = GameWindow(root)

    game\_window.mainloop()

**Input Validation:**  
  
def validate\_input(input\_value):

    if not input\_value.isdigit():

        raise ValueError("Input must be a number")

    if int(input\_value) < 0:

        raise ValueError("Input must be a positive number")

    return int(input\_value)

try:

    user\_input = input("Enter a positive number: ")

    validated\_input = validate\_input(user\_input)

    print(f"Validated input: {validated\_input}")

except ValueError as e:

    print(f"Error: {e}")

**Modular Approach:**  
def main():

    # Main function to run the application

    app = MainMenu()

    app.mainloop()

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

    main()

**Screenshots:**

**Main Menu Window:**

A screenshot of a computer

Description automatically generated

**Game Window:**

A screenshot of a computer

Description automatically generated

**Input Validation:**

A screenshot of a computer

Description automatically generated

**Modular Approach:  
  
A screen shot of a computer

Description automatically generated**