**Program :**

**Normal Condition**

(1) **Each window of the GUI must have a scrollable text area, a set of 5 buttons, a pulldown menu and a popup menu.**

**Input:**

Run the Gui class. GUI should generate.

**Expected output:**

GUI contains a scrollable text area when the text area populates. A set of 5 buttons is present at the bottom of the window. A pulldown menu with title “File” is present and clickable. A popup menu shows when the user right clicks anywhere on the panel, outside of the GUI components.

(2) **New window - creates a new GUI window. The new GUI window should associate with another GreenhouseControls object.**

**Input:**

Click the pulldown menu and select “New window”. A new window should appear. Click “Browse” and select a .txt file in your directory. Click “Start”.

**Expected output:**

The text area of the opened window should print out the running program of the selected file.

(3) **Close window - closes the current window. If the GreenhouseControls object is running, a warning message should be displayed to ask the user to confirm closing of the window. If the current window is the only window opened, exit the application.**

**Input:**

Run the Gui class. GUI should generate.

Open two windows

On the second window, click “Browse” and select a .txt file in your directory. Click “Start”. While the program is running, click the pulldown menu again and select “Close window”. When the prompt appears, click “Yes”.

On the first window, click the pulldown menu and select “Close window”.

**Expected output:**
A prompt should pop up on the second window while the program is running to validate the option to exit the window.

The program should exit normally for the first window.

**(4) Open Events - opens an events file. It should bring up a file dialog and let the user chooses an event file. If the chosen file is not a valid event file, display an appropriate error message.**

**Input:**
Run the Gui class. GUI should generate.

Click the pulldown menu and select “Open Events File”. Select a non .txt file. Once prompted out, select a .txt file. Click “Start”.

**Expected output:**
A window will pop up on the first action showing “ERROR”, “Incorrect file type, select a .txt file”.

With a valid .txt file, the program does not encounter an error window and should run normally.

**(5) Restore - opens a dump.out file and restore the GreenhouseControls object. It should bring up a file dialog and let the user choose a dump.out file. If the chosen file is not a valid dump.out file, display an appropriate error message. This option should be disabled if the GreenhouseControls is running.**

**Input:**
Run the Gui class. GUI should generate.

Click “Browse” and select examples3.txt or examples4.txt file in your directory. Click “Start” and let the program run. Once the program finishes running, click the pulldown menu and select, “Load Dump File”. A popup will show, select a non .out file first. Once prompted, select the dump.out file in your directory.

**Expected output:**
The de-serialization event for the program should run from the dump.out file.

**(6) Exit - exit the application. If any of the opened GUIs is running a GreenhouseControls object, display a warning message to ask the user to confirm the exit.**

**Input:**
Run a .txt file.

While the program is running, click the pulldown menu and select, “Exit”.
Once prompted, click, “No”.  After the program finishes running, go into the pulldown menu again and select, “Exit” once again.

**Expected output:**
While the program is running, attempting to exit will display the warning message.

When the program is not running, clicking “Exit” will exit out the GUI.

**(7) Associate a keyboard shortcut with each of the above menu items.**

**Input:**
Keyboard shortcuts as follows:
N for New window
C for Close window
O for Open Events File

R for Restore
E for Exit

Run the Gui class. GUI should generate.
For all shortcuts … click on “File”
Press N
Press C
Press O, select a .txt file prompt
Press R, select a dump.out file prompt
Press E

**Expected output:**
All shortcuts should do their respective actions.

**(8) Start - start to run a GreenhouseControls object. This button should be disabled if the GreenhouseControls object is running.**

**Input:**
Run the Gui class. GUI should generate.

Click “Browse” and select a .txt file in your directory. Click “Start” and let the program run.

**Expected output:**
At GUI opening, Start will be disable. When a .txt file is selected, Start will be enabled. When the program is run, Start is disabled until program finishes running.

**(9) Restart - add a Restart object to rerun the current event file. This button should be disabled if no events file is read, or if the GreenhouseControls object is running.**

**Input:**
In the same GUI, click “Restart”.

Once the program finishes running, open a new window.

**Expected output:**
Restart is enabled when a .txt file is available. When the program is running, Restart is disabled.

In the new window, Restart is disabled because no .txt file is available.

**(10) Terminate - add a Terminate event to the running GreenhouseControls object. It should bring up a dialog to prompt for the delay time in milliseconds. This button should be disabled if the GreenhouseControls object is not running.**

**Input:**
Run the Gui class. GUI should generate.

Run any .txt file.
When the program starts, click “Terminate” and input a delay time of 4000.

**Expected output:**
When the program is not running, Terminate is disabled

On click of “Terminate”, a prompt should pop up asking to enter a delay time. On acceptance, a terminate event will be added into the running program and terminate the program at 4000msec.

**(11) Suspend - suspends all running event threads. This button should be disabled if the GreenhouseControls object is not running.**

**Input:**
Restart the program. When the program runs, click “Suspend”.

**Expected output:**
When the program is not running, Suspend is disabled

In the text area, the program should stop running and a message “Suspending” should print out.

**(12) Resume - resume all suspended event threads. This button should be disabled if the GreenhouseControls object is running.**

**Input:**
After some amount of time, while the program is suspended, click “Resume”.

**Expected output:**
When the program is running, Resume is disabled. When it is suspended, it is enabled.

The program should restart where it left off before suspension.

**(13) The popup menu should contain the following 5 submenus: Start, Restart, Terminate, Suspend, and Resume, with the same functionalities as the buttons.**

**Input:**
Load in a .txt file. Right click somewhere on the GUI panel and click, “Start”.
Once the program finishes running, right click again and click, “Restart”.
While the program is running, right click again and click, “Terminate”.
Restart the program again, right click again and click, “Suspend”.
Right click again and click, “Resume”.

**Expected output:**
The popup menu items should have the same functionalities and enable/disable behaviour as their button counterparts.