GUI from Hell

ISTE-121 – Homework 2

Introduction:

You have seen enough GUIs to know the difference between good and bad interfaces. You will create a user interface that will make the users life as miserable as possible. The end result should be a "user interface from Hell".

You should try to be as creative as possible. Things like moving buttons, text fields that occasionally drop the letter "e" from a word, random font or color changes, GUIs that look different each time you start them or horrible color schemes are a good start.

Requirements:

1. Your GUI must use *at least* four different controls. By different controls, this means four different control classes. Four JButtons do not count as four controls that is one control since they are all from the same class.
   a. Each 'control' must control or 'do' something.
      i. Doing something does not mean a JOptionPane, it must have some action or cause a change to take place.
   b. **One of the controls must be a menu**, with at least one submenu item in it. For example: File ➔ Exit, or Help ➔ About would count as a control.
   c. List these controls and their actions in the GUI.txt file.

2. Your GUI must use an anonymous inner class for at least one control.

3. Your GUI must use a WindowListener or setDefaultCloseOperation method-call to handle the WindowClosing event on your JFrame. I need some way to kill these GUIs when I'm done looking at them. The close window button (in the upper right of the Frame) can simply cause the program to end. The program and all other 'things' it does need to completely stop running.

4. You must use the Swing package to construct your interface.

5. All controls are active controls that must respond to user input in some way. The user input could be any of the following:
   a. Button presses
   b. Mouse events, such as a mouse entering a button or text area.

c. Keyboard events
d. List selections
e. Active Radio and/or CheckBox on-click action events.
f. Any other recognized Java events
g. Look at the Java SwingSet demo and source code for ideas. Cite any code that is not of your own creation.

6. Be as creative as possible. You could look at paper forms, online data entry systems or other sources.

7. Sound and form movement are not considered a 'control', though they are not restricted from being used, and can be the result of a "control" being used. Let me know if your program will use sound ahead of time, so I can make sure the speakers work for my computer.

Restrictions *(that we have had to add to protect professors and graders from your GUI's)*:

1. Your program shall only change what displays on the running computer's monitor. Your program:
   a. Shall **NOT** change the monitor's resolution, or other attributes
   b. Shall **NOT** access files, which includes but not limited to: deleting, modifying, adding, renaming the running computer's data, hard drive, files on the hard drive, external drives, network drives, memory, boot sector, etc, etc, etc.
   c. Shall **NOT** access the Internet in any way, including but not limited to running programs off the Internet, accessing email, etc.
   d. Shall **NOT** cause or require a reboot of the running computer.
   e. Shall **NOT** start so many frames/windows/threads the controls freeze up. Remember, my computer may run slower than your computer, thus freezing up may take a few thousand less frames/threads than your computer.

2. Your program must have a way to easily exit and stop itself.
   a. This may not immediately be from the "Close icon" (red X in the upper right of a frame for PCs or red dot in the upper left frame for Macs), but the means to exit must be documented in GUI.txt (see next section), and take nor more than **two** EASY operations to exit.

Submit to dropbox:

To the dropbox:

1. Your very well **javadoc** documented code *.java and *.class files.
2. A text file, **GUI.**txt, (not Word *.doc file) containing all the evilness your program possesses. This GUI.txt file should contain at least these sections:
    a. Your name
    b. The controls used (at least 4 different classes)
        i. *If* you have multiple “locations” for the controls in your program, tell me where each control is located.
    c. (optional) What the program would have been if it didn’t go to the dark side (Example: Student registration, Mortgage calculator, just an evil useless application, etc.)
    d. Short description of how to navigate and use the program. How to use all the great “features” you’ve added.
    e. What makes your program the most evil in the class.
    f. LAST how to **easily** exit the program.

Submitting the Assignment:

This assignment is due as specified in myCourses. To the myCourses conference drop box for this course, submit all .java, .class files and the GUI.txt file for this assignment in **one ZIP file**. Other compression methods such as 7z or rar are not accepted.

If you want extra points:

A list of the top evil programmers will be written on the board, you may add your name to the list of evilness. Each of the finalists will demo their own programs describing the evilness of each. If you have an evil program you can add your name to the list and demo at anytime before the voting starts. The class will determine which programs meet the Best In Class, which receives the bonus points. Bonus points for this homework are awarded as follows:

- 1st place = 25 points
- 2nd place = 15 points
- 3rd place = 10 points

R·I·T

Rochester Institute of Technology
Golisano College of Computing and Information Sciences
Department of Information Sciences & Technology

ISTE-121 – Homework 2 – GUI from Hell – Grade Sheet

Name: ____

Criteria	*Max Pts*	**Pts Earned**
Program meets stated requirements		
At least four different controls used, as described in GUI.txt	20	
One control must be a Menu with submenu For example: “File ➔ Exit”, Help ➔ About	10	
Each control exhibits *unique* behaviors Two controls cannot do the same thing.	20	
Swing used to construct the GUI	10	
Frame closes cleanly, as in GUI.txt	10	
Program follows Java naming conventions	10	
Proper coding style used: indentation, use of white space for readability,	10	
The program contains adequate Javadoc documentation. File Header, classes, methods	5	
A GUI.txt file is provided that describes how the GUI works and how to kill it off when done.	5	
Points earned for homework:	**-**	
Best in Class (25, 15, 10 points bonus):		
Deductions:		
Late deductions		
Other deductions		
Total points		

Note: If your program fails to compile, you will receive a zero for the entire assignment. A clean compilation means that the compiler generates no error messages. (Warning messages are OK as long as the class file gets created.)

Additional Comments: