Task 2 - Action Listeners

Prepared by William Reid
September 9, 2014

1 Action Listeners

Every action performed by the user must be attached to some type of **listener**. There are many different listeners ranging from **Mouse Listeners** to **Window Listeners**. The one we will focus on in this task is the most commonly used - the **Action Listener**. All listener types must be contained within their own private class, located at the bottom of the file they are used in:

```
private class <functionName> implements ActionListener{
    public void actionPerformed(ActionEvent action) {
       //perform functions in here related to this action
    }
}
```

2 Adding a Menu - (6 marks)

In your DrawFrame class, add a method named *addMenus()*. In this, create two new **JMenu** items, *File* and *Edit*.

For the file menu, add the following **JMenuItem** objects:

1. Open <Ctrl-o>
2. Save <Ctrl-s>
3. Close <Ctrl-x>

For the edit menu, add the following **JMenuItem** objects:

1. New <Ctrl-n>
2. Delete

Add these menu bars to the DrawPanel frame.

2.1 Adding Shortcut Keys - (5 marks)

Menu items can be assigned shortcut keys. This allows the user to quickly manipulate the program without having to use the mouse. In the *addMenus()* function, add the shortcuts referenced in the lists above. This will need to be done using **mnemonics** and **accelerators** - the following site should assist in explaining the differences and show their uses:

http://stackoverflow.com/questions/3717936/how-to-add-jmenubar-shortcuts

2.2 Adding Action Listeners - (10 marks)

We will now add an action listener to each of the menu items you created earlier. Every listener is created in two stages:

1. Create and implement the private listener class (shown in part 1 of this task).
2. Add the listener to the object for which it is intended (this is a single command, done after the menu item is created in the *addMenus()* function).

Add an action listener to every menu item we created earlier (every menu item will needs its own private listener class). For now, every action listener just needs to print out its name. i.e - *System.out.println("Open").* Clicking on each of the menu items should now print out the action that is being performed. Test your functions are operating correctly, including the use of the shortcut keys.

3 Adding a Mouse Listener - (8 marks)

In your DrawPanel class, implement a **Mouse Listener** private class. The mouse listener class requires you to implement ALL 5 functions. They can be found here:

http://docs.oracle.com/javase/tutorial/uiswing/events/mouselistener.html

Implement each of these in your DrawPanel class. For each listener print out the X and Y position of the mouse at the time of the event. Don't forget that this mouse listener needs to be added to something in order to make it operate correctly. Can you think of what object will use this listener? Test your new functions work correctly before moving on to the next section.

3.1 Adding a Mouse Motion Listener - (5 marks)

In your DrawPanel class, implement a **Mouse Motion Listener** private class. The mouse motion listener class requires you to implement its two primary functions. They can be found here:

http://docs.oracle.com/javase/tutorial/uiswing/events/mousemotionlistener.html

Implement both of these in your DrawPanel class. For each listener print a meaningful message out to the console. Have you remembered to correctly add the listener to the object that will use its functions?

4 Checking your Code (6 marks)

**Note: For each file - 1 mark each goes to comments, layout/structure and conforming to the function/variable naming standards. This does not include Draw.java as this has not been changed.*

Review your code for layout, structure, comments and how it measures against the supplied standards. When you have completed this check, ensure it is uploaded to the Git repository (you should be committing your work once you have completed each class).

Notify me that you have finished and I will check your work, giving feedback on how things can be improved, including your total mark out of **40**. After I have given feedback you may be asked to modify the code (based on my comments). Once you have completed this and notified me, will move on to the next task. Good luck!

5 Program Understanding

Do you know what we have tried to do with implementing **listeners** in these classes? Create a document called *Questions.txt* that lists questions of things you need clarification on. It may be useful to make this document before you start such that you can add and remove to it as you continue through these tasks.