GUI Programming with Java

Session 5
Menus

- We will look at…
  - – Menus in SWING

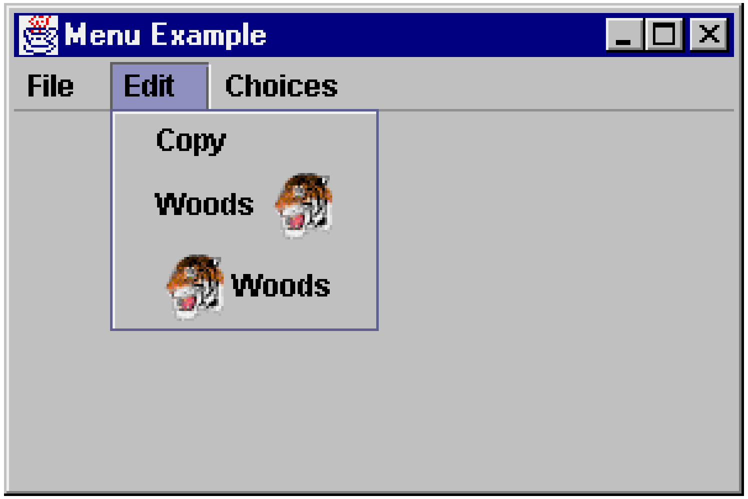
MENUS

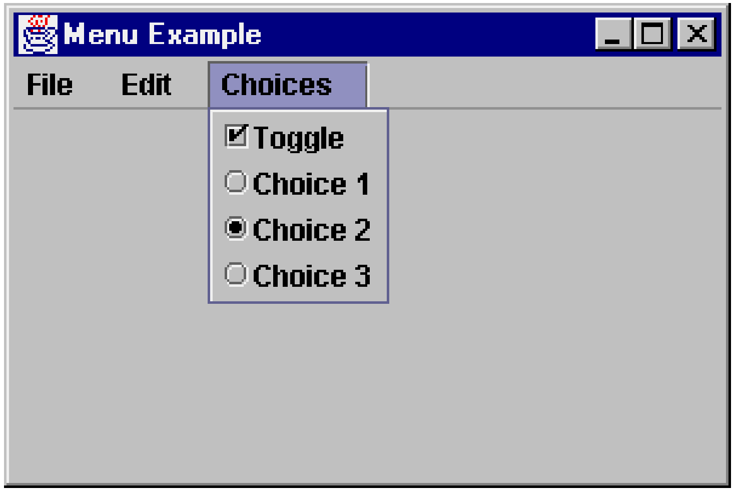
What is a Menu

- Menus are integral parts of GUI's.
- Menus make selection easier and are widely used in window applications
- They allow the user to perform actions without unnecessarily cluttering up the graphical user interface.
- IN SWING we can only apply menus to JFrame or to JApplet (both support the setJMenuBar method).

Menu Classes

- Java provides five classes to implement menus:
  - JMenuBar,
  - JMenu,
  - JMenuItem,
  - JCheckBoxMenuItem,
  - JRadioButtonMenuItem





What is a Menu Bar?

- A JFrame or JApplet can hold a *menu bar* to which the *pull-down menus* are attached.
- Menus consist of *menu items* that the user can select (or toggle on or off).
- Menu bars can be viewed as a structure to support menus
- A menu bar holds menus; the menu bar can only be added to a frame.

Adding a Menu Bar

- Following is the code to create and add a **JMenuBar** to a frame:

```
JFrame f = new JFrame();
f.setSize(300, 200);
f.setVisible(true);
JMenuBar mb = new JMenuBar();
f.setJMenuBar(mb);
```

- The Menu Bar has no menu’s on it at this stage so wont really be visible.

- For an alternative way of doing this please refer to sample 1 (MenuBar.JAVA) in the sample 1 folder.

Creating Menus

- We attach menus onto a JMenuBar.
- Use the following constructor to create a menu:
  - public JMenu(String myMenuItemName)
- The following code creates two menus, File and Help, and adds them to the JMenuBar mb:

```
JMenu fileMenu = new JMenu("File", false);
JMenu helpMenu = new JMenu("Help", true);
mb.add(fileMenu);
mb.add(helpMenu);
```

- The menus will not be seen until they are added to the menu bar. For an example please refer to sample 2 (menuBar2.java) in the sample 2 folder.

Creating Menu Items

- The following code adds menu items and item separators in menu fileMenu

```
fileMenu.add(new JMenuItem("new"));
fileMenu.add(new JMenuItem("open"));
fileMenu.addSeparator();
fileMenu.add(new JMenuItem("print"));
fileMenu.add(new JMenuItem("exit"));
```

- For an example please refer to sample 3 (menuBar3.java) in the sample 3 folder.

Creating Sub Menu Items

- You can add submenus into menu items.

- The following code adds the submenus “**Unix**,” “**NT**,” and “**Win95**” into the menu item “**Software**.”

```
JMenu softwareHelpSubMenu = new JMenu("Software");
JMenu hardwareHelpSubMenu = new JMenu("Hardware");
helpMenu.add(softwareHelpSubMenu);
helpMenu.add(hardwareHelpSubMenu);

softwareHelpSubMenu.add(new JMenuItem("Unix"));
softwareHelpSubMenu.add(new JMenuItem("NT"));
softwareHelpSubMenu.add(new JMenuItem("Win95"));
```

- For an example please refer to sample 4 (menuBar4.java) in the sample 4 folder.

Create Checkbox menu items

- You can also add a JCheckBoxMenuItem to a JMenu.
- JCheckBoxMenuItem is a subclass of JMenuItem that adds a Boolean state to the JMenuItem, and displays a check when its state is true.
- You can click the menu item to turn it on and off.
- The statement following adds the checkbox menu item Check it.

```
helpMenu.add(new JCheckBoxMenuItem(“Check it”));
```

- For an example please refer to sample 5 (menuBar5.java) in the sample 5 folder.

Note: Some might consider checkboxes on menus a little outdated. They are considered by some to be bad examples of interface design (HCI)

Add Images to items

- You can add images to menu items (JMenuItem), menu checkboxes (JCheckBoxItem) and menu radio buttons (JRadioButtonMenuItem)

- You can add icons to items using the following code.

```
JMenuItem jmiNew, JmiOpen;
fileMenu.add(jmiNew = new JMenuItem(“New”));
fileMenu.add(jmiOpen = new JMenuItem(“Open”));
jmiNew.setIcon(new ImageIcon(“images/new.gif”));
jmiOpen.setIcon(new ImageIcon(“images/open.gif”));
```

- For an example please refer to sample 6 (menuBar6.java) in the sample 6 folder.

Add Images to items

- You can add images to menu items (JMenuItem), menu checkboxes (JCheckBoxItem) and menu radio buttons (JRadioButtonMenuItem)

- You can add icons to items using the following code.

```
JMenuItem jmiNew, JmiOpen;
fileMenu.add(jmiNew = new JMenuItem(“New”));
fileMenu.add(jmiOpen = new JMenuItem(“Open”));
jmiNew.setIcon(new ImageIcon(“images/new.gif”));
jmiOpen.setIcon(new ImageIcon(“images/open.gif”));
```

- For an example please refer to sample 6 (menuBar6.java) in the sample 6 folder.

Set Keyboard Mnemonics

- Setting a keyboard mnemonic for a menu item allows you to access that menu item by pressing the ALT key and the mnemonic key.
- We can add mnemonic keys to menus and menu items (including checkbox items etc).
- To add a mnemonic key to an item we use the following code
  - item.setMnemonic(key)
  - Example: helpMenu. setMnemonic(‘H’);
- For an example please refer to sample 7 (menuBar7.java) in the sample 7 folder.

Set Keyboard Accelerators

- One problem with keyboard mnemonics is that they only let you select menu items from the currently open menu.

- Key Accelerators however, let you select a menu items directly by pressing the CTRL key and the acclerator key. For example by using the following code you can attach the accelerator key CTRL+O to the open menu item

```
jmiOpen.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_O,
ActionEvent.CTRL_MASK);
```

- The setAccelerator method takes an object KeyStroke. The static method getKeyStroke in the KeyStroke class creates an instance of the keystroke.

- VK_O is a constant representing the O key and CTRL_MASK is a constant indicating that the CTRL key is associated with the keystroke.

- For an example please refer to sample 8 (menuBar8.java) in the sample 8 folder.

Adding Event Handling

- Event handling for menu items is pretty straightforward.
- Menu items generate ActionEvent objects. Your program must implement actionPerformed handler to respond to the menu selection.
- For an example of event handling for menus please refer to sample 9 (MenuBar9.java) in the sample 9 folder.

Enabling or Disabling Menu Items

- One way of protecting users from making mistakes is to disable menu items when they are not appropriate.
- To achieve this we can use the setEnabled method.
- The format of the method is setEnabled(boolean)
- Example: jmiNew.setEnabled(false)
- The above line of code will disable the new file menu item.

Summary

- IN SWING we can only apply menus to JFrame or to JApplet (both support the setJMenuBar method).
- Java provides five classes to implement menus:
  - JMenuBar,
  - JMenu,
  - JMenuItem,
  - JCheckBoxMenuItem,
  - JRadioButtonMenuItem
- A JFrame or JApplet can hold a *menu bar* to which the *pull-down menus* are attached.
- A menu bar holds menus; the menu bar can only be added to a frame.

Summary(2)

- The following code adds menu items and item separators in menu fileMenu

```
fileMenu.add(new JMenuItem("new"));
fileMenu.add(new JMenuItem("open"));
fileMenu.addSeparator();
fileMenu.add(new JMenuItem("print"));
fileMenu.add(new JMenuItem("exit"));
```

- You can add submenus into menu items.
- We can also add checkboxes and radio buttons to menus (and as sub items to menu items).
- By using the setIcon method we can add an icon to a menu item.
- Keyboard accelerators and mnemonics allow us faster access to menu items.
- Event handling is achieved via the use of the actionPerformed handler.

Labwork 5:

Task 1

Create a frame with a help menu. The contents of the help menu should look (not functionally simulate!!) the help menu from textpad.

Task 2

Create a frame with a file menu. Add three items (New, Open and Exit). When the user selects exit we should exit the system.

Get started on your assignment