CS25610 / CSM5610 Worksheet Three 1999-00

David Price

March 19, 2000

1 Introduction

This worksheet is intended to provide an opportunity to gain practice with

- creating Java Applets
- providing practice at using the Java AWT
- interacting with the browser in which the Applet is running

NOTE WELL: You will need to consult your lecture notes and online manuals and web pages during this practical.

2 Your Task

The overall idea is to create a set of web pages together with a Java Applet that allow a user to "load" other files and either display them in browser windows and frames or to play them via the audio system.

An example of the overall screen layout of my solution to this worksheet is attached at the end of this document, or available on-line on the course materials web site. Your application does not have to look identical to mine, but it should have the same features and if complete, the same hierarchy of web pages.

2.1 The components of the task

Overall you have three subtasks to complete

- The Web Pages

  The Applet you produce should run within a web page using the Netscape browsers on the Intel machines running Solaris in the Solarium.

  To begin with, you can embed the Applet of a simple, single web page.

However, another component of the practical is intended to gain practice using "Frames" on web pages and via them understand some extra features available to Java Applets.

If you have never used Frames before, then at the end of this worksheet I give a simple example of a "top" web page which includes as its sole contents two frames.

The eventual idea for your web pages is as follows

- top level

  At the top level you have a web page which includes two frames, each loaded from separate web pages. One of the web pages which is loaded into one of the frames should be a simple straightforward web page. The second web page, loaded into the second frame, should not be a simple web page, but should itself be a page which contains two more frames.

- lower level

  The lower level page mentioned above which itself includes two frames should be constructed as follows. One of its two frames should again be a simple web page, the second of the frames should be the one into which your Java Applet is loaded.

- Examples

  The screen shot of my application included at the end of this worksheet is precisely constructed in the way described above.

  * my top level web page
    My top level page merely includes commands to indicate that it has two frames. Into those frames I load the web page which starts

    "This frame is the upper one..."

    and the second area which contains all of the rest of the displayed material. The web page I immediately load contains just another specification for a page which itself loads two frames.
  * my embedded frames
    The lower part of my display therefore includes the two frames which contain the words

    "Computer Science, Prifysgol ...."

    and the words

    "Dave Price's Applet test page."
  * the applet page
    The last page mentioned above includes the appropriate commands to embed the Applet. You will find these in your lecture notes. However, to be helpful I've included a simple example at the end of this document.

- The Java Applet

  Your Java Applet should be written using various facilities available to applets as well as general AWT facilities.

  The applet visual interface has four main areas.

  - File Selection

    One area, at the top in my Applet, allows the user to type in the final components of the URL that references some information the user would like to load. In my tests, I have allowed the user to enter the name of plain text files, html pages, images in the form of gif files, names of directories and audio clips in the form of ".au" files. All url endings specified by the user are assumed to be relative to the *documentbase* from which the web page was loaded.

– Selection of Window For Display

This part of my interface is displayed towards the left on my GUI.

Applets are allowed to interact with the browser in which they are running via an AppletContext interface. An applet can get hold of its AppletContext by calling the method *getAppletContext()*.

One particular method available via the AppletContext is that you can request your hosting browser to open and display other material reached via a URL. When you ask the browser to do this for you, there is a facility to indicate in which "browser window or frame" you would like the accessed material to appear.

The API specification for one of the *showDocument()* methods on an AppletContext is as follows

```
public void showDocument(URL url,
                         String target)

   Requests that the browser or applet viewer show the Web
   page indicated by the url argument. The target
   argument indicates in which HTML frame the document
   is to be displayed. The target argument is interpreted as
   follows:

    ``_self"
            Show in the window and frame that
            contain the applet.
    ``_parent"
            Show in the applet's parent frame. If the
            applet's frame has no parent frame, acts
            the same as ``_self".
    ``_top"
            Show in the top-level frame of the
            applet's window. If the applet's frame is
            the top-level frame, acts the same as
            ``_self".
    ``_blank"
            Show in a new, unnamed top-level
            window.
    name
            Show in the frame or window named
            name. If a target named name does not
            already exist, a new top-level window
            with the specified name is created, and the
            document is shown there.


   An applet viewer or browser is free to ignore
   showDocument.
   Parameters:
      url - an absolute URL giving the location of the
      document.
      target - a String indicating where to display the
      page.
```

Thus, as you will see, you are allowed to specify the window required by means of you supplying a specific string as the second parameter. You will now hopefully appreciate the complicated set of web pages and frames that you have been asked to define. Of the five choices available, you are asked to implement the first four as I have in my example. As an extra, you might like to add names to your frames and implement the fifth option later.

It's clearly "unfriendly" to ask a user to specify his/her selection by typing the specific strings as documented in the API for showDocument(). Thus, you are required to provide easy to click options as I have done in my example. The area including the clickable items is in my example defined as an object which extends a AWT panel. I call my class that implements this, *CheckPanel*. On that panel I use *Checkbox*s for the items that a user may select and all are attached to CheckboxGroup so that as I select one, the others become automatically selected. My CheckPanel class has one available method, and that returns a single String object referencing the appropriate information as needed for the second parameter to showDocument() when I invoke it.

- The Load Request Button

  Towards the right of my GUI I have a single button labelled "Load". You GUI should have such a similar button too. On clicking the button, the Applet should acquire the filename from the "File Selection" are of your GUI and combine that with the "documentbase" to form a URL. You will need to use the *getDocumentBase()* method and the URL constructor to do this.

  Having completed the above, the code obeying the Load button should then acquire the indication of the required window from the CheckPanel and then invoke the *showDocument()* method to cause the request information to be displayed.

- The Message Area

  There should be a section on your display where the Applet can output messages. On my example, the area is placed at the bottom of the GUI and is implemented as a *TextArea* inside a ScrollPane. You can implement this anyway you choose.

- The Data Files

  You will need to place all the classes for your Applet and the web pages in an area of your web space. You will also need to place within that area (or subdirectories of it) some example text, html or images files that you can allow the user to view via your application.

  I've provided a few example gif (image) and au (audio) files in the directory reached via the URL

  http://www.aber.ac.uk/~dap/cs25610

  you may copy these to your web area.

2.2 Testing your Applet

Using your Applet you should be able to gain an understanding of the hierarchy of your web pages. By selecting an appropriate option in the CheckPanel area and providing a valid name in the File Selection area, you should be able to get the browser to display the selected information upon different web frames or windows.

Your Applet should remain running through your tests and you will often need to use the "back" button of the browser to return to the main page of your Applet.

If you send output to *System.out* or *System.err* then that will get re-directed to the Java Console of the browser. On Netscape, you can select to view the Java Console by selecting the appropriate

option from the tools component of the menu that drops down from the *Communicator* menu item on the main Netscape menu line. You may find that using the method *println()* on System.err is a good way of generating diagnostics to help debug your Applet.

3 Effort Allocated to the Worksheet

It is highly unlikely, probably impossible for you to complete all of this worksheet in your two hour practical this week. This is completely intentional. You should expect to spend some "own time" work to complete the worksheet. Another, new worksheet will be provided for the practical next week.

4 Assessment of this Worksheet

These worksheets for the modules CS25610 and CSM5610 are NOT assessed.

The worksheets and the demonstrated practicals are provided as part of the educational offering of the module. The examination questions for these modules may be based on material covered in practicals in addition to lecture material and background reading drawn to your attention by the lecturers or by links on the course materials web site.

5 Example Web Pages

5.1 http://www.aber.ac.uk/~dap/cs25610/top.html

```
<html>
<head>
<title>Page that includes just two frames</title>
</head>
<FRAMESET ROWS = 150,*>
<FRAME SRC = frame_one.html SCROLLING = Auto>
<FRAME SRC = frame_two.html SCROLLING = Auto>
</FRAMESET>
</html>
```

5.2 http://www.aber.ac.uk/~dap/cs25610/frame_one.html

```
<html>
<head>
<title>This will be in the upper frame</title>
</head>
<body BGCOLOR="#DD0000">
```

```
A simple web page.

</body>
</html>
```

5.3 http://www.aber.ac.uk/~dap/cs25610/frame_two.html

```
<html>
<head>
<title>This will be in the lower frame</title>
</head>
<body BGCOLOR="#0000DD">

Another simple web page.

</body>
</html>
```

5.4 http://www.aber.ac.uk/~dap/cs25610/applet_page.html

```
<html><head><title>Applet test page
</title></head>
<body>
<h1>Applet test page.</h1>
<applet code="TheApplet.class" width=600 height=400>
</applet>
</body></html>
```