Introduction

The purpose of this homework is to get you acquainted with some of the development tools that we will be using this quarter and have you write your first midlet to run on a (simulated) cell phone.

We have provided you with a sample midlet suite in two different release formats. One release file is binary only. This contains just the files needed to run the midlet: spma.jad, spma.jar, and release-notes.txt The second release file contains the files needed to build the midlet from scratch. This is the traditional source release that a programmer might want in order to develop a follow on version of the software.

You have two tasks for this homework:

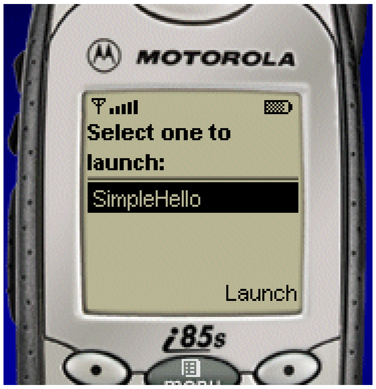
1. Implement an image viewer midlet like the one I showed in class and add it to the suite.
2. Extend the Ant build.xml file to add a target named “dist” to create release files like the ones provided to you.

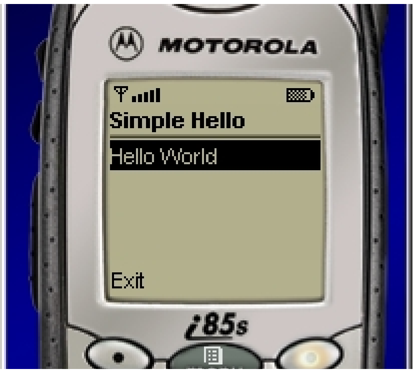
Preparation:

1. If you are working on a non-lab machine, install the course software as described on the class web site software page. If you are on a lab machine, verify that the various parts and pieces are available once you have logged on.

2. Download the zip files for this homework from the web site. Unzip the binary archive. If the WTK is installed and associated with jad files correctly you should be able to double click on /build/bin/spma.jad and start the emulator running with the sample midlet. [On my Win2K system, the open action associated with the JAD file type is c:\apps\WTK104\bin\emulatorw.exe -gui -Xdescriptor "%1".]

3. Unzip the source archive. Using a command line window, change directories to the spma directory containing build.xml, then type the command “ant”. This should recompile the code and rebuild the jad and jar files and put them in the build/bin directory. You should be able to type “ant run” and again see the midlet running in the emulator.





Midlet Task

Your first task is to write a midlet that implements the ImageViewer capability.

1. There is a running binary version of ImageViewer available from the web site, so you can experiment with it to see how it works.

2. The main class of the application is appmain.ImageViewer, as specified in the jad file. You can change this if you like, but that is what it is when delivered to you.

3. When the application runs, it presents a list of names of images. When the user selects one of the images, it is displayed on the screen. Back and Exit commands are available, and the user can keep looking as long as the images hold his interest.

4. The images are built from png files that are stored in the jar file when the application is packaged. The sample pngs are provided to you in the src directory.

5. Build a list of image names based on the application properties in the jad file. You can assume that the list of image numbers will be a contiguous list, that is, there will be information about Image-1, and Image-2, and Image-3, etc, with no gaps. A gap marks the end of the list. Look at Hello.java to see a very basic example of building a List to show on the screen, and also how to attach Command objects to it.

6. Refer to the Learning Wireless Java book, MIDP API documentation or other resource to learn how to create Image and ImageItem objects and display them. An ImageItem is displayed in a Form.

7. Add at least two more images to the list of possible images and make sure that your midlet can find and display them correctly.

8. Update the release-notes.txt file in the doc directory. In the release notes, include a brief description of the features of your midlet. Also describe any known bugs or limitations.

Ant Task

Your second task is to implement a new ant target “dist” to build the distribution files.

1. Edit build.xml and add a new target named dist. The dist target depends on the targets “build” and “buildjavadoc” to create a clean build of the midlet.

2. In “dist”, create the archives "${dist.dir}/${midlet.name}-${build.number}-bin.zip" and "${dist.dir}/${midlet.name}-${build.number}-src.zip" using the zip task. The property names shown are interpreted by Ant to create files with names like spma-20-bin.zip and spma-20-src.zip.

3. The binary distribution should contain only the files spma.jad, spma.jar, and release-notes.txt. The jad and jar files should be located in ${midlet.name}/build/bin within the archive, and release-notes.txt should be located in ${midlet.name}/doc within the archive. (In other words, maintain the original directory structure.)

4. The source distribute should contain the useful contents of:

   a) The top-level directory (build.xml, build.properties, and build.number)
   b) The src directory including the png files and package subdirectories with all java files.
   c) The doc directory (release-notes.txt).
   d) The build/javadoc directory and any subdirectories.

   Again, preserve the original directory structure so that when this archive is unpacked the next programmer has a working configuration. You should be able to unpack this archive and rebuild the program by entering the ant command.

Turn in

Using the Ant new target you have written, build the binary and source archive files. Verify that the zip files do indeed have all your files in them correctly, then follow the turnin link on the web site and turn in the two files.

This homework is due before midnight, Tuesday April 8. Don’t be late, the turnin server closes automatically!