ASSIGNMENT 5
DUE DATE: Saturday 28 May, 2011 11:59 PM

The purpose of this assignment is to understand how data is hidden in images using LSB substitution.

Description

In this assignment, you will write a Java program called StegoBreak.java that analyzes a given image for data hidden using LSB substitution. A set of seven BMP images are provided (download the assignment files from the assignment page). Each image uses a 24-bit RGB pixel format, i.e. 8 bits for red, 8 bits for green and 8 bits for blue. Most of them have one or more data files hidden inside using LSB substitution. The data could have been hidden using a single color channel (R, G, or B), two channels (e.g. BG, GR, etc.) or using all three channels. For example, Master Yoda’s image can be reconstructed by extracting the LSBs in the blue and green channels of every pixel in lena0.bmp.

The program will check for every possibility (single and multiple channels) to determine if a "known file" can be extracted. For simplicity, you may assume that when multiple channels are involved, bits are substituted in the blue, green and red order. For example, if the byte 10011010 is to be hidden using the green and blue channels in a 2x2 image, then it has been done as follows:

<table>
<thead>
<tr>
<th>Pixel at (0,0)</th>
<th>LSB(Blue)</th>
<th>LSB(Green)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixel at (1,0)</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Pixel at (0,1)</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Pixel at (1,1)</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

This limits the possibilities to BGR, BG, BR, GR, B, G and R.

What is a “known file”?

While any data file can be hidden using LSB substitution, your program is required to only check for JPG, BMP, DOCX, PDF and MP3 files. This can be done by first extracting the bits, then constructing the file, and then checking for a signature (the first few bytes) in the constructed file. The signatures corresponding to the specified file formats are as follows:
Values are in hex notation. Any constructed file not having one of these signatures can be discarded.

**Input and Output**

The program should take one argument – the name of file to analyze. The output should be the files extracted from it. Include information on the color channels as part of the output file name. For example, if `lena0.bmp` is the input, then `lena0-bg.jpg` is the output file. This will be a valid JPEG image of Master Yoda. The output file(s) should have the proper extension. If you extract an MP3 file then it should be saved with a file name ending in `.mp3`. The program may automatically delete any file it constructed but did not belong to one of the specified formats. The *File* class in Java lets you do this; but this is not a necessity.

**Approach**

1. Go over the `StegoMakeBG.java` program posted on the class website (May 5). This program shows how to load a BMP image in Java and then work with individual pixels.

2. Once you know how the red, green and blue channels of a pixel are extracted, find a way to extract the least significant bits from the color encodings. Think bit-shifting, bitwise-AND'ing, bitwise-OR'ing and such operations!

3. Once you have extracted enough bits to form one or more bytes, construct the bytes and write them to a file. Keep doing this until you have processed every pixel of the image.

4. Open the constructed file and check if the first few bytes match one of the given signatures. If so, rename the file to have the appropriate extension. Again, the *File* class can be used here. You may delete the file if no match is found.

5. Do these steps for every possible usage of the channels: BGR, BG, BR, GR, B, G and R. The scenario chosen will determine if the LSB from a certain color channel should be extracted or not in Step 2.

**Submission**

Follow file naming conventions and fully comment your code. Upload the `StegoBreak.java` file to Blackboard.
Grading

The assignment is worth **10 points**, divided as:
- processing one channel: 2 pts
- processing two channels: 2 pts
- processing three channels: 3 pts
- rest of the requirements: 3 pts

A program that *does not compile* is a program that *you did not submit at all*. Remember the GTA is not required to debug your program to give you partial points.

The late policy is available at http://cs.du.edu/2555/assignments.html. **You must work alone on this assignment.**