

## Simple GUI application attempted.

**Source:** <http://coding.derkeiler.com/Archive/Java/comp.lang.java.help/2004-07/1245.html>

---

**From:** flesh\_eating\_dragon (*dragon\_at\_netyp.com.au*)

**Date:** 07/23/04

Date: 23 Jul 2004 05:13:05 -0700

Some assistance in the development of a simple GUI application would be appreciated.

Not long ago I completed a computer science degree, and right now I want to get a bit more experience and confidence by trying to write applications that are just a bit beyond what I had to write at university. However, in the early stages of designing the application that I have in mind, I have hit a brick wall. I can't write it the way that I thought at first would work, because of the ColorModel classes having no 'set' methods, so I'd like to know what `_will_` work.

Please imagine the following scenario (this is not a description of the application, but it is a simpler illustration of the same problem). Suppose you are writing a graphical application that simulates a digit on a traditional LCD display, as for example on a handheld calculator. This should be achievable using a Raster that encodes something akin to the following:

```
011110
200003
200003
044440
500006
500006
077770
```

and a changeable ColorModel, in which 0 always represents the BACKGROUND colour and the other indexes 1 to 7 represent colours that we will call ON and OFF. For example, if the digit to be displayed is "4", then indexes 2, 3, 4 and 6 will be set to represent colour ON whilst indexes 1, 5 and 7 will be set to represent colour OFF. Similarly, if the digit to be displayed is "3", then indexes 1, 3, 4, 5 and 7 will represent ON and indexes 2 and 6 will represent OFF.

It would be nice if the ColorModel classes had 'set' methods, because then you could make the relevant class (let's call it DigitDisplayImage) a subclass of Image, and the rest would be elegant and easy. But it doesn't, although I feel there `_should_` be a similar

comp.lang.java.help: Simple GUI application attempted.

solution. Then you could write `setIndex(int, Color)` methods in the `DigitDisplayImage` class, after which the following code example would work to display the digit "4".

```
digitDisplay.setIndex(1, OFF);  
digitDisplay.setIndex(2, ON);  
digitDisplay.setIndex(3, ON);  
digitDisplay.setIndex(4, ON);  
digitDisplay.setIndex(5, OFF);  
digitDisplay.setIndex(6, ON);  
digitDisplay.setIndex(7, OFF);
```

This would be lovely.

In the absence of `ColorModel` classes with 'set' methods, how, in practise, should I go about achieving a similar functionality? (In the real application, the corresponding object needs a lot more than eight index values and will be called upon to display far more than just digits between 0 and 9, but I think it is better to discuss the problem with reference to this simplified example.)

Thanks,

Adrian.