

## J2ME\_Customizing\_Game\_Sprites\_Color

Often in games a nice feature to give to users is the possibility to customize their own character, for example changing colors of the different parts (hairs, eyes, shirt, and so on).



To support this features there are 2 possibilities:

- Include a different image for each color of each different part
- Replace colors by code

The second options will save you the effort to create these multiple images, and will strip down your JAR size. Also, it will allow you to support a lot more colors.

You can see a simple midlet showing the following snippet of code in action on [this page](#).

Here is the code that takes your input image, the color to be replaced and the new color, and returns a new image.

### *Some notes on this code:*

- The `Image.getRGB` method() returns colors as effectively represented by phone, so they can be different from the original colors of your image. This is the reason why we use the `getDeviceColor` to get the effective color to be replaced, since on different phones it can be represented differently.
- The limit of this approach is that if 2 different colors of your image map to the same color (because the phone has a limited range of available colors), both of them will be replaced with the new color.
- A simple optimization of this code will be to store the `rgb[]` data in a variable, to reuse it for all the color replacement operations

```
import javax.microedition.lcdui.Image;

public class ColorChanger
{
public static Image changeColor(Image source, int fromColor, int toColor)
{
int imageWidth = source.getWidth();
int imageHeight = source.getHeight();

int[] rgb = new int[imageWidth * imageHeight];

int deviceColor = getDeviceColor(fromColor);

getRGB(rgb, 0, 0, imageWidth, imageHeight);

for(int i = 0; i < rgb.length; i++)
{
if(rgb[i] == deviceColor)
{
```

## J2ME\_Customizing\_Game\_Sprites\_Color

```
[i] = toColor;          rgb
}
}
return Image.createRGBImage(rgb, imageWidth, imageHeight, true);
}
static int getDeviceColor(int color)
{
Image fake = Image.createRGBImage(new int[]{color}, 1, 1, false);

int[] rgb = new int[1];

    getRGB(rgb, fake, 0, 0, 1, 1);

return rgb[0];
}
}
```