Walkthrough

Batch Grayscale Images

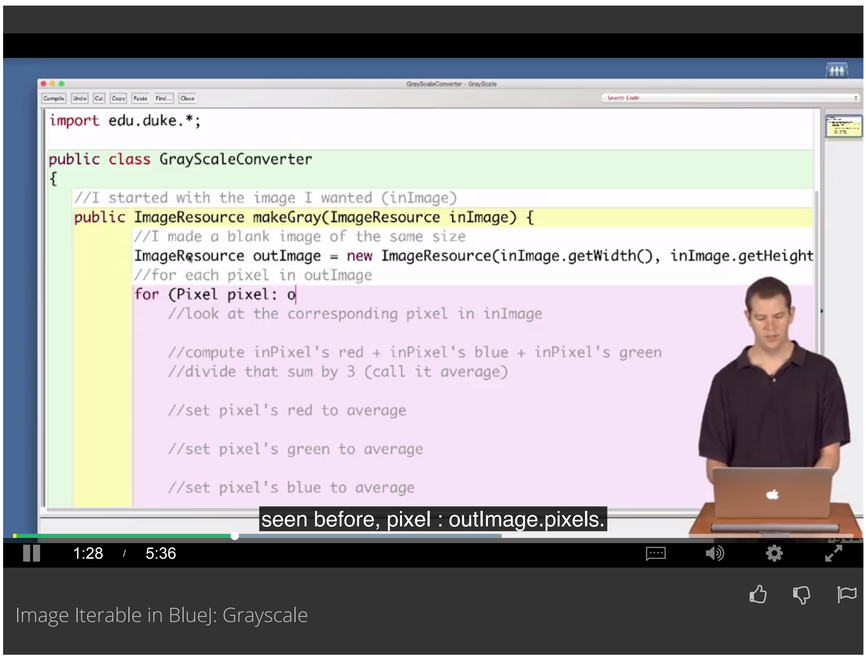

Image Iterable in BlueJ: Grayscale

Have a question? Discuss this lecture in the week forums.

Interactive Transcript

Search Transcript English

0:03
Now that we've developed our algorithm to convert to grayscale, we're ready to write code. We've started here with a class for the grayscale converter and we've already imported edu.duke.*. And before this video started, I went ahead and wrote in the steps that we came up with as comments in our code, to guide us as we write. The first thing we did in our algorithm was we started with the image we wanted inImage. In this case, that's going to be a parameter that we pass to our function, so that this function can operate on any image we want.

0:36
The next thing we did was we made a blank image of the same size as inImage. So if I want to make a new image, I'm going to declare an image resource. We called it outImage in our algorithm. And since I want to make a new thing I'm going to say new, the type of thing I want to make is an ImageResource. And when I create this ImageResource, I need to pass in information, in this case, telling it how big I want my image to be. This is going to be the width of inImage and the height of inImage.

1:16
The next thing we did in our algorithm was, we wanted to do something for each pixel in outImage. That's going to be a for loop, which you've seen before, pixel : outImage.pixels. And we're going to put curly braces around all of the steps of our algorithm that we want to do for each pixel. And you can see that BlueJ put this all in pink so we can easily see that these all are going to happen for each pixel.

1:48
In our algorithm, the next thing we did was we looked at the corresponding pixel in inImage. So, what we're going to want to do is, inImage.getpixel at the same location as pixel. So (pixel.getX(), pixel.getY());. Now, when we do this, we need to give it a name so we can use it again. That's going to mean we're declaring another variable, so it's going to be inPixel = inImage.getPixel. The next thing that we did was we said we wanted to compute inPixel's red + inPixel's blue + inPixel's green, divide that by 3 and call the result average. So I want inPixel's red, which is going to be inPixel.getRed() + inPixel's blue, inPixel.getBlue(), plus inPixel's green, inPixel.getGreen(). And then I want to divide by 3, now if I write /3 here, I've made a small mistake, because order of operations is going to make, inPixel.getGreen() /3. I wanna divide the whole thing by 3, so I need parentheses.

3:04
And we want to call this average, we're declaring a new variable. What type is this variable? Well it's going to be just a plain number so int average = all of that map. Now I want to set pixels red, so pixel.setRed(average);. And similarly, pixel.setGreen(average): and pixel.setBlue(average);. Those are all of the steps I wanted to do for each pixel, and then our last step at the bottom here says outImage is our answer. Whenever we know the answer to a function, we return that answer. So in this case, outImage is our answer, we just return outImage. So now I'm going to come up here and click Compile, at the bottom it says, Class compiled, no syntax errors.

3:58
You've frequently been testing your code by just making an object in the BlueJ main window and calling it's methods. But making an image resource there is a little bit tricky, so we're gonna write another method to help us test this out.

4:13
Public void, let's call it testGray(), just going to take no parameters, it's going to make an ImageResource, we'll call it ir = new ImageResource.

4:29
And that's going to pop up a dialog and ask us what image we want. And then we're going to make gray on this image resource, and then we're going to draw that. Now if we've forgotten the name of the method we just wrote we can scroll up and look, but I just remember it's called makeGray. Gonna compile, again no syntax errors, we obey the rules of the language, we don't yet know if our code works. Now I'm gonna come over here to the main window. I'm gonna hit new GrayScaleConverter. And it's gonna give me this object in my BlueJ.

5:08
Now I'm going to hit testGray. It's gonna pop up a dialog asking me what image I want. I'm just gonna choose these nice, colorful easteregs. But when I run it through my grayscale converter, I get easteregs in gray. Since my code passed this test case, I'm more confident that it works. And in general, the more test cases we run, the more confident we'll be in our code's correctness.

Downloads

- **Lecture Video** mp4
- **Subtitles (English)** WebVTT
- **Transcript (English)** txt

Would you like to help us translate the transcript and subtitles into additional languages?