Homework

23

Green Screen & Composite

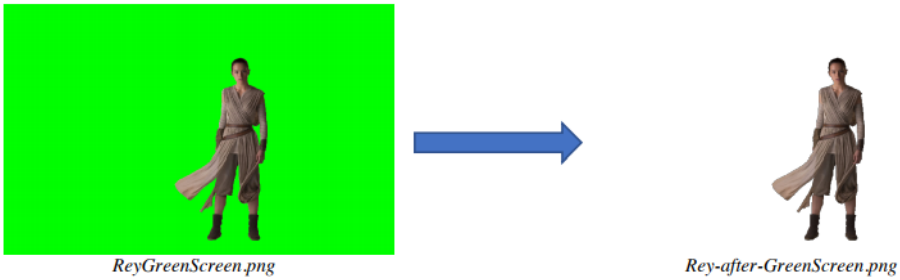
Last assignment you learned how to open a PNG file and process its pixels using pointers. In this assignment, you're going to put that to good use by implementing two of the big-screen movie effects: green-screen and composite.

Open the header file to see the prototypes for the functions you will write. The `greenScreen()` function will remove all green pixels, making them transparent. This is a `void` function that will modify the image passed as the first parameter. The code is already stubbed, so you can run `make test` and see that the code compiles.

You'll also see three extra folders: `input` which contains the photos we're going to start with, `expected`, which contains the photos as they should look, and `actual`, which contains the photos after your filters have been applied. If your code fails one of the tests, the `actual` folder will also have a rudimentary "diff" image you can examine. You can look at any of the photos just by double-clicking them in the IDE.

Green-screen is often used in movies to merge actors into a background. The technique uses a particular range of colors (such as green) to represent a background that can later be made transparent.

Your task in this function is to modify the image you are given, converting any `green` pixels into transparent ones, with no color information, such as the image below.



ReyGreenScreen.png *Rey-after-GreenScreen.png*

It is unlikely that any pixels will have pure `green` pixels (no `red` or `blue` with a `green` component of `255`), so you should treat a pixel as `green` if its `green` component is at least twice as large as the larger of its `red` and `blue` component. When you find a `green` pixel that meets these requirements, set all of its components—`red`, `green`, `blue` and `alpha` to `0`. If a pixel is not green, just skip it.

Here is some pseudocode you can follow:

```
Let p point the beginning of the image
Set end to point just past the end
While p != end
   If *(p + 1) is twice as large as max(red, blue)
      Clear all of the fields
   Increment p by 4
```

Composite

Now that you know how to green-screen a picture, let's combine that green-screen picture with a second image, to produce a **composite** picture like this:

For the `composite()` function, you'll be given two images, instead of one. The first image is the foreground, which you should pass to `greenScreen()`, where the green pixels will be removed. Then, combine the pixels using this plan:

```
Green-screen the foreground image
Set dest pointer to the foreground
Set src pointer to the background
Set end to point just past the end of dest
While dest != end
   If dest pixel is transparent
      Copy pixel at src to dest
   Increment src (*4)
   Increment dest (*4)
```

You should now be able to `make test` and all tests should pass. (Of course, if the Green Screen function doesn't pass, then Composite cannot pass.) Once you have 100%, then do `make submit` to turn in your assignment. **Make sure that your score is recorded** on the CS150 Homework Console.

As always, if you run into problems, bring your questions to the discussion board, or to my office hours