COSC 1P02 Assignment 4

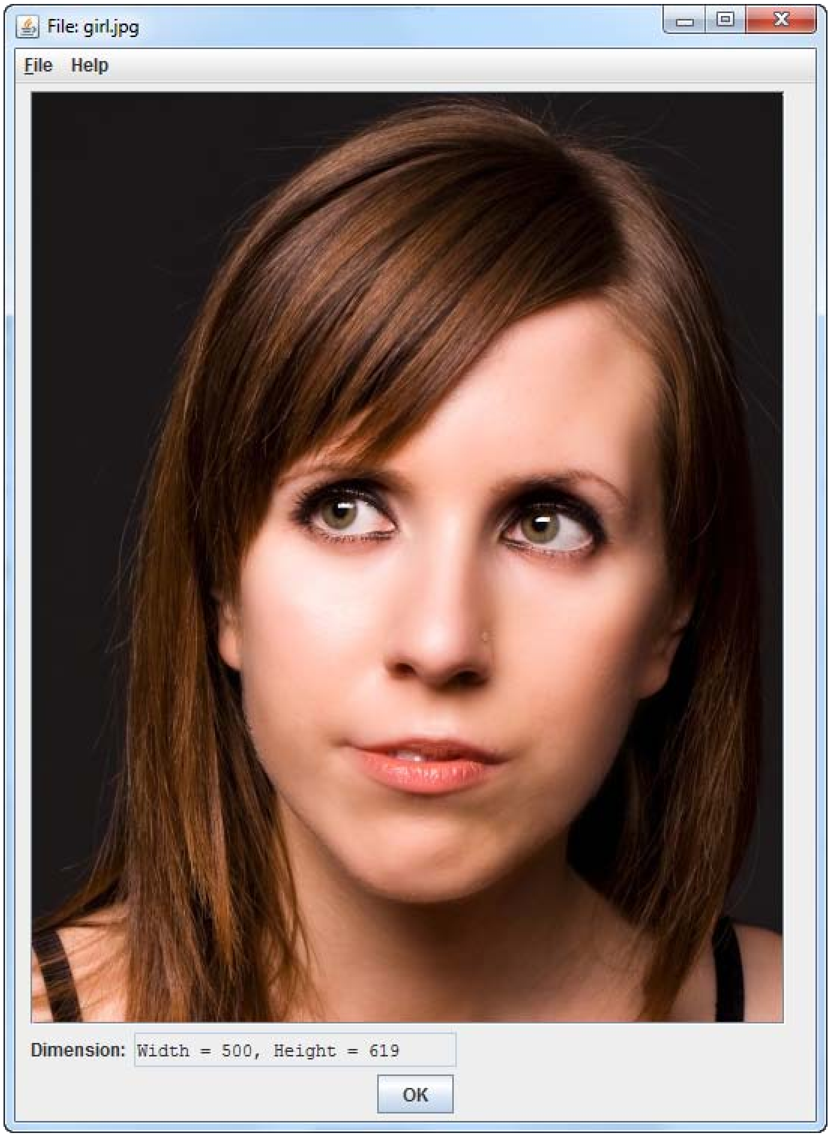
“The medium is the message.”

***Due: Nov. 11, 2016 @ 4:00 pm** (late date Nov. 14 @ 4:00 pm)*

The emphasis for this assignment is processing the pixels of a picture. In preparation for this assignment, create a folder called `Assign_4` for submission of the assignment and two subfolders: `Assign_4_A` and `Assign_4_B` for the two DrJava projects for the assignment.

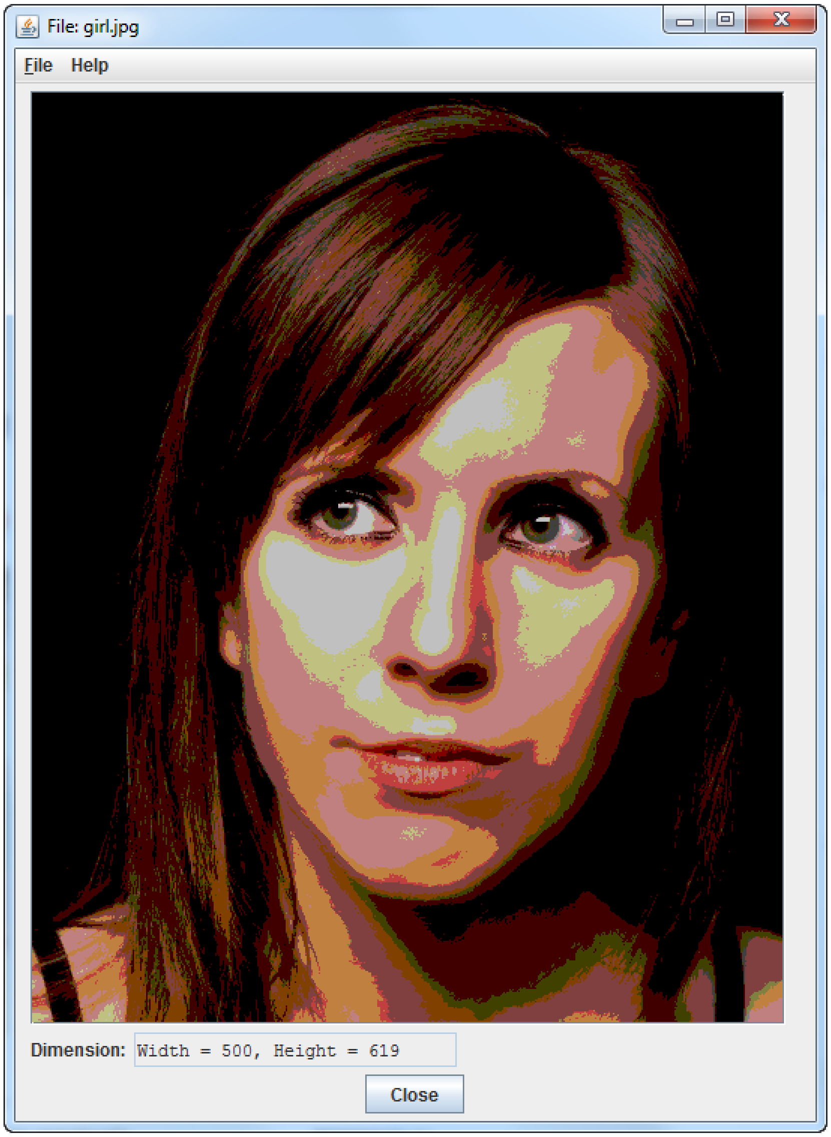
Part A – A Poster

When a poster is printed, it is sometimes printed with a reduced color palate, for example instead of 256 levels for each color component (i.e. 16 million colors), only a small number such as 4 (giving only 64 distinct colors) is used. The effect on the image is called “posterizing”. For example, the following picture:



Dimension: Width = 500, Height = 619

After being posterized would look like:



Dimension: Width = 500, Height = 619

Write a program that inputs a picture, displays it (waiting for the user) and then posterizes it to have only 4 levels per color channel. Use the `PictureDisplayer` that sizes to the size of the picture.

What we want to do is reduce each color channel from the range 0-255 (8 bits) to the range 0-3 (2 bits). We can do this by dividing the color channel value by 64. Using integer division, this will give us a value 0-3. However, since our actual display still uses 1 byte per color channel, values 0-3 will all look very much like black (very low color intensity). To make it look right, we need to scale the values back up to the original range (multiply by 64). This means that only 4 color channel values will occur: 0, 64, 128 and 192, imitating a 2-bit color palate.

Part B – A Night's Sky:

In lecture, we wrote a program to remove redeye from a photograph. In lab we wrote a program to do “Green Screening”. In each case we measured the color distance between the color of each pixel and `RED` (for redeye) or `GREEN` (for green screening), and then made a decision to change color by comparing that distance to a predetermined tolerance. The same technique can be used to remove a background of a single colour, or to replace other areas of consistent colouring. In this assignment, you'll be replacing a blue sky with a night sky. As an added feature you will randomly add stars to the night sky.

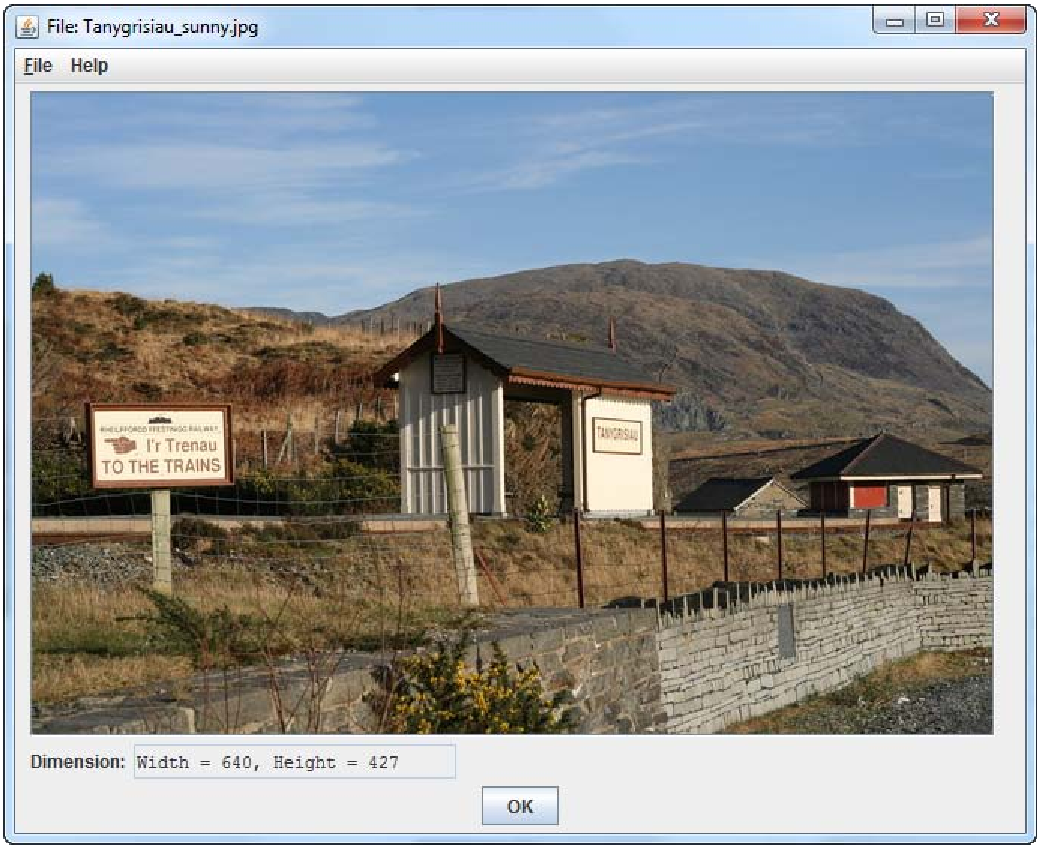
The methodology is straightforward. Examine each pixel of the picture and if it is within a reasonable distance of the **sky blue** (`Color(8431307)`) change the colour of the pixel to a very **dark blue** (`Color(1054800)`). However, approximately once in every 2,000 sky pixels processed (i.e. with probability `0.0005`), instead of using **dark blue**, use a **white** colour (`Color(16777088)`), producing a star in the sky.

Hints:

- Use the `PictureDisplayer` constructor that sizes to the size of the picture.
- You will process all of the pixels.
- You must check to see if the pixel color is close to **sky blue** and change the color if so. (You will have to experiment to find a tolerance that works. Something around `65` works for the supplied picture.)
- When replacing the color, you need to randomly (with probability `0.0005`) use **white** rather than **dark blue**. (Generate a random double in [0,1) and, if it is less than 0.0005, use **white**.)
- Make sure the user has time to see the original picture, as well as the processed picture, and save the final modified result in a new file.

You may try it on your own photo that includes a skyline if you wish, but you will then need to determine the best 'blue' to use.

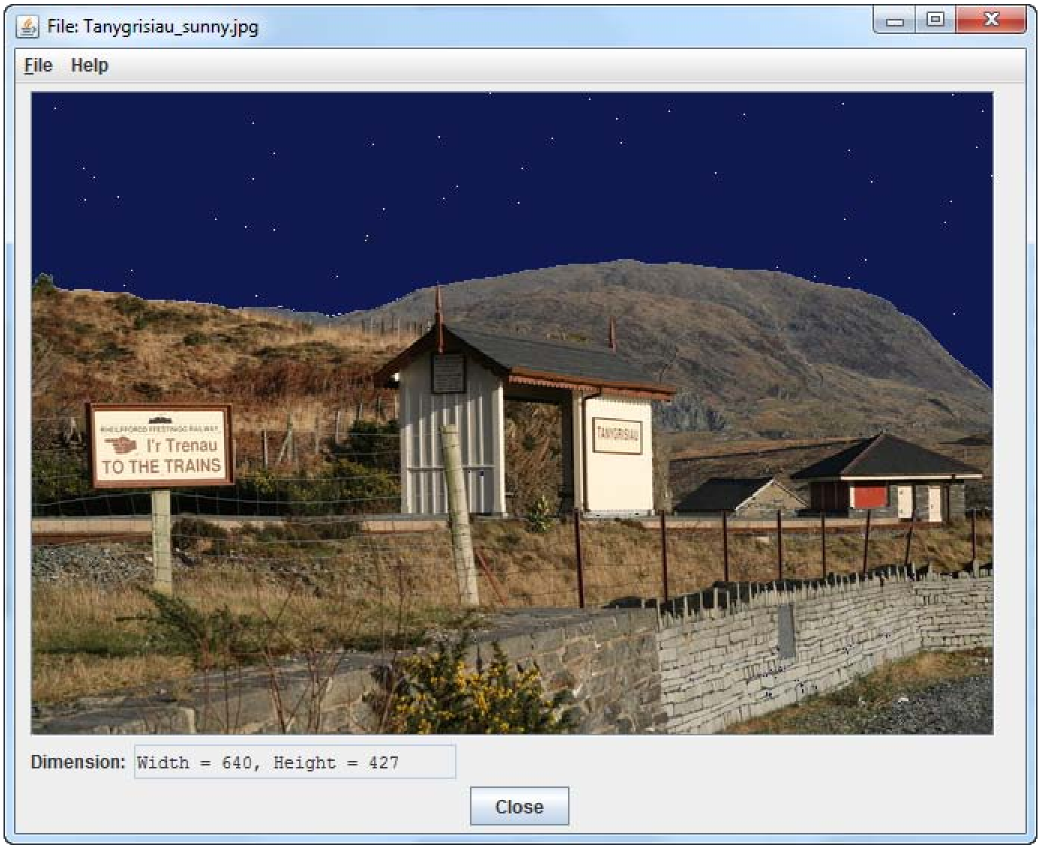
Before:



File: Tanygrisiau_sunny.jpg

Dimension: `Width = 640, Height = 427`

After:



Dimension: Width = 640, Height = 427

Submission:

Details regarding preparation and submission of assignments in COSC 1P02 are found on the COSC 1P02 Sakai Site as `Assignment Guidelines` under `Course Documents`. This document includes a discussion of assignment preparation, programming standards, evaluation criteria and academic conduct (including styles for citation) in addition to the detailed assignment submission process copied below.

To prepare and submit the assignment electronically, follow the procedure below:

1. Ensure your submission folder (`Assign_4`) has two subfolders (`Assign_4_A` and `Assign_4_B`) that contain the two DrJava projects for the assignment.

2. Using DrJava, print (to `CutePDF Writer`) the `.java` file of each of the parts for your assignment using the name *`ClassName`*`.pdf` where *`ClassName`* is the class name (i.e. same name as the `.java` file) and save the `.pdf` files at the **top level** of the submission folder (i.e. directly within `Assign_4`).

3. Run each program. For Part A use the file: `girl.jpg` and for Part B use the file `Tanygrisiau_sunny.jpg`. When the display is finished (i.e. `Close` button visible), select `Print Image of Window...` from the `File` menu on the `PictureDisplayer`, direct the output to `CutePDF Writer` and save the `.pdf` file at the **top level** of the submission folder (i.e. directly within `Assign_4`) using an appropriate name (e.g. `Part_A.pdf` or `Part_B.pdf`).

4. Create a `.zip` file of your submission by right-clicking on the top level folder (i.e. `Assign_4`) and selecting `Send to/Compressed (zipped) folder`. A zipped version of the folder will be created. Use the default name (`Assign_4.zip`).
5. Log on to Sakai and select the COSC 1P02 site.
6. On the `Assignments` page select `Assignment 4`. Attach your `.zip` file (e.g. `Assign_4.zip`) to the assignment submission (use the `Add/Remove Attachments` button and select `Browse`). Navigate to where you stored your assignment and select the `.zip` file (`Assign_4.zip`). The file will be added to your submission. Be sure to check the `Honor Pledge` checkbox. Press `Submit` to submit the assignment. You should receive a confirmation email.

DrJava

The `.zip` folder you submit should contain the project folders for the two parts, including all files relevant to the project—the `.java` and `.class` files for the assignment—and the `.pdf` files for program listings and output at the top level.

Other Platforms

If you are using an IDE other than DrJava to prepare your assignment, you must include the `.java` source files and the `.pdf` files described above for each part as well as an executable file (likely `.class` or`.jar`) that will execute on the lab machines.