

May 12, 2016

**EE 356/CS 376**  
**Project 2 – Merging Images**

**September 8, 2017**  
**Due: September 18, 2017**

For this project you will start with two images of approximately the same size. The images must be color images and both must fit on the screen at the same time. When the user clicks on a *merge* button one of the images will be converted to the other by a series of incremental changes over a period of several seconds.

Each pixel in a color image is made up of three colors red, green, and blue and each color is specified by an 8 bit number. In hexadecimal each color ranges from 00 to 0xFF. Each pixel has a color specified by six hex digits. 0x000000 specifies all black and 0xFFFFFFFF specifies all white (equal amounts of red, green, and blue.) 0xFF0000 is red, 0x00FF00 is green, and 0x0000FF is blue. In decimal the number 0xFFFFFFFF is 16,777,215.

Suppose we want to merge a pixel of color 16711680 (red) on one image to the color 255 (blue) on the second image and we want to do this in 20 steps of one second each. We can write:

$$\frac{16711680 - 255}{20} = 835571.25$$

In the original image the pixel color is 16711680. A second later the pixel color is  $16711680 - 835571 = 15876109$ . In the third image the pixel color is  $16711680 - 2 * 835571 = 15040538$ . In the 20th image we set the color to 255. The viewer will see the red pixel merge into the blue pixel of 20 seconds.

Since you need to do this for every pixel in an image there is quite a lot of computation needed. There are two functions called SetPixel and GetPixel that allow you to alter one pixel at a time on a Writeable bitmap. These functions are too slow and should not be used. Instead, you can use the LockBits method to capture all of the pixels in an image. After this is done you can alter them and unlock them. See the LockBitsExamples notes. It is also possible to do this using a writeable bitmap and creating a back buffer but we will not use this method for this project.

*Minimum Requirements:*

1. Your project must use C# WPF.
2. Provide a file menu to allow a user to load two images to the screen.
3. Allow the user to set the number of increments to be used and the approximate time between increments.
4. Your project must use the LockBits method.

*Extras:*

1. Allow the user to drag and drop images onto the screen.
2. Include an *About* box that gives the author, date, etc.
3. Provide a list of instructions and help information.
4. Provide a mechanism so that the program runs forever merging each image into the other continuously.

Turn in the following electronically to your instructor

May 12, 2016

1. A Word document titled `EE356P2XXX.docx` (where `XXX` is your three initials) which contains:
  - Cover Sheet – The cover sheet should include your name, course number, project number, project title, and date handed in.
  - A short description of your project. Include an overview and any special features that you added that are not in the specifications.
2. A complete code file that can be executed. Your code file must have well documented source code.

Compress these two items into a zip file should have the title `EE356P1XXX.zip` where `XXX` are your three initials and email them to your instructor.