

EE 356/CS 376

Project 01 – ASCII Art

Due: September 5, 2018

For this project you will convert a black and white image into an image made entirely from black ASCII characters on a white background. To begin you need to choose a font and a font size. For each letter you can write a program which determines the weighting of dark and light pixels. For example, Figure 1 shows a letter S as dark pixels on a white background and the letter S with white pixels on a dark background. The total number of pixels in each letter is 63 since the format is 9 x 7. For the letter with black pixels on a white background the weighting is $15/63 = 0.2381$. For the letter with white pixels the weighting is $(63-15)/63 = 0.7619$ which is $1 - 0.2381$. After finding the ratio for every letter, number, and punctuation mark in the font you should store this in an array ordered by ratio.

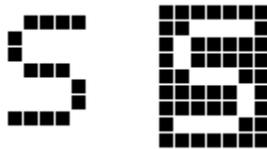


Figure 1

The letter S in pixels.

The next step is to choose an image that has a gray scale to match your font size. In the example of Figure 1 I have 63 pixels giving me potentially 63 levels of black so this would work well with an image that has 64 gray levels. (You can use Paint.net available free to change an image to black and white and set the gray scale.) Divide the image into rectangles each of which is the same size as the font you have chosen to use. Calculate the darkness level of each rectangle in the image and replace that rectangle by a letter that is closely matched to its darkness level. If you have several letters that have the same darkness, it is probably best to choose the one that has the best distribution of dark pixels (not all on one side for example).

You should give the user the option of choosing the font and the font size. Provide a minimum of three different fonts in at least two different sizes.

Minimum Requirements:

1. Your project must use C# WPF.
2. Provide a file menu that allows the user to save a converted picture and to load a new picture. Your file menu should also have an *exit* option to quit the program. Your program does not have to function on color pictures. It may be limited to gray scale photos of a particular size in bmp format.
3. Provide a font menu to allow the use to choose the font. You must have at least three different fonts.
4. Provide a font size menu to allow the use to choose the font size. You must have at least two different font sizes.
5. Provide appropriate means to allow the user to draw or redraw the screen. If a user draws a work a picture and changes the font your program should allow the redraw without restarting the program.

Extras:

May 7, 2018

1. Allow the user to choose additional fonts.
2. Allow the user to choose additional font sizes.
3. Include an *About* box that gives the author, date, etc.
4. Provide a list of instructions and help information.
5. Allow different size photos.
6. Provide conversion of color photos to gray scale.

Note that you can have your program calculate the darkness levels for the font and font size chose when the program runs or, you can have this done in advance and stored in a file which your program can read when it starts. Either option is OK.

Turn in the following electronically to your instructor

1. A Word document titled `EE356P1XXX.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.