Shawon Razzak

CS 135: Final Project Design

Erinstagram! (Photo Editor!)

***Program Implementation:***

The program is an implementation of image processing that allows us to view, load, edit, and save a selected photo similar to social media apps like Instagram or Facebook. The user will be prompted with choices to utilize the photo editor program to their desire, such as uploading/selecting, viewing, and editing the photo of their choice. They are allowed various methods of editing, such as cropping, brightness modification, and rotation, on top of the option to decide if they want to save their design. The program continues until the user chooses to leave, allowing them to use the program as much as they desire.

***Data:***

* Image Choice – int 2D array composed of integers corresponding to a brightness value.
* Edited Image – int 2D array that stores the alterations made from the editing menu.
* File Pointers (height, width/ column, rows)
* First Menu Choice – Char
* Editing Menu Choice – Char
* Save Menu Choice – Char

***Function Algorithms (main & additional)***

Main():

***Functionality and Algorithm***: This function will guide the user throughout the program as well as receiving their input throughout the menu(s). It will display the first menu that allows the user to choose between loading a new image, displaying the current image, editing the image, and exiting the program. Using variables, formatted IO, and switch statements, it will receive and execute the user’s choices as they utilize the program. The program will also store and execute the user’s input in the editing menu, based on if they want to crop, alter brightness, or rotate the image. Additionally, it will also store and execute when the user decides whether or not they want to save the new image into a file (also executed by switch statements). The program will then bring them back to the first menu and continue to repeat via a “do while” statement until the user chooses to leave the program.

RetrieveNewImage():

***Functionality & Algorithm***: The purpose of this function is to retrieve a new image if that is what the user chooses to do in the first menu of the main function. The user will input the file name, check if the file is opened successfully and isn’t NULL, read and store the image into the program, and then close.

DisplayCurrentImage()

***Functionality & Algorithm***: The purpose of this function is to display the current image the program is “holding” using loops to go throughout the entire 2D array and displaying its respective pixel brightness. If there isn’t an image/ if it is NULL, then it should inform the user that there is currently no input image and prompt them back to the menu.

DimImage():

***Functionality & Algorithm***: The purpose of this function is to decrease the brightness value of each pixel by one, and thereby dim the image, if the user chooses this option from the edit menu. To achieve this, we can loop throughout the entire 2D/image array and lower the brightness value of each pixel until the entire image is dimmed. The program will also display the new version of the image afterwards.

BrightenImage():

***Functionality & Algorithm***: The purpose of this function is to increase the brightness value of each pixel by one if the user chooses this option from the edit menu. To achieve this, we can create a loop that goes through the entire 2d array and increases the brightness value for each pixel by one. The program will also display the new version of the image afterwards.

CropImage():

***Functionality & Algorithm***: The purpose of this function is to crop the image if the user chooses this option. To meet the user’s needs, we need to receive the cropping dimensions from the user, One we receive the cropping dimension, we need to confirm they within range, and crop the photo (however to be more user friendly, as they might not know the dimensions of the input photo or what cropping dimensions are, we can display the image dimensions and create a menu that allows them to choose a side, and how much they want to crop it in “pixels” and repeat the menu until they cropped all the sides of their choice and choose to leave). The program will also display the new version of the image afterwards.

SaveImage():

***Functionality & Algorithm***: The purpose of this function is to save the edited image the user produced if that is what they choose. If selected, it will prompt the user for a filename and store the data there using a loop and close the file, as well as notify when the process is complete. If there is an error, although rare, it should notify the user that there was an issue.

Extra Credit: Rotate90():

***Functionality & Algorithm***: The purpose of this function is to rotate the image 90 degrees. This function will process by swapping the height and width of the previous 2D array and making the “columns” of the previous array into the “rows” of the new array, and the “rows” of the previous 2D array into the “columns” of the new array. We can do this by creating a loop with conditions that stores the columns of the 2D array into the rows of the new 2D array. How the loop stores and organizes the pixels is also crucial as we don’t want to unintentionally invert the image (if we were flipping it 90 degrees clockwise, we want start from the leftmost column of the original array and read each of them from bottom to top, to store it from left-to-right in the new 2D array, if it was counter clockwise then we need to start from the rightmost column and read each of them from top to bottom. I provided a visual below). The program will also display the new version of the image afterwards.

Visual Example of Rotate90:

2D pixel Array of inputted image: Output 2D pixel Array after rotating it 90 degrees:

|  |  |  |
| --- | --- | --- |
| 1E | 2E | 3E |
| 1D | 2D | 3D |
| 1C | 2C | 3C |
| 1B | 2B | 3B |
| 1A | 2A | 3A |

Counterclockwise Clockwise

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1A | 1B | 1C | 1D | 1E |
| 2A | 2B | 2C | 2D | 2E |
| 3A | 3B | 3C | 3D | 3E |

|  |  |  |
| --- | --- | --- |
| 3A | 2A | 1A |
| 3B | 2B | 1A |
| 3C | 2C | 1C |
| 3D | 2D | 1D |
| 3E | 2E | 1E |