

Final Project Design
Jay Tilman
Image Processing Program

Data:

Initial image - int 2d array

Updated (new) image - int 2d array

Menu choices - int

Image size specifications - int

User interface:

User starts program: a menu appears. User is prompted to choose an action to take. If user chooses to upload a new image, they are prompted for the file to get the image from. When user chooses to edit the image, edits are made, then both the old and the new image are displayed to the terminal window. When user chooses to exist, a goodbye message is given, then the program ends.

Algorithms for each Function:

main()

DATA: Arrays for the old and new image; variables for original image size, new image size, main menu and edit menu choices, file pointer to the original image, file pointer to save an image

FUNCTIONALITY: In a loop (until getMenuChoice returns zero): display the Menu, get the Menu choice.

If user chooses to load an image, prompt user for the file name; attempt to connect to the file storing the image. If the connection cannot be made, display an error message and end the program. If (reading) connection is successful, use loadNewImage().

If user chooses to display the image, display image to terminal using displayImage().

If user chooses to exit, display goodbye message and end loop (getMenuChoice will be equal to zero 0), leading to end of program.

If user gives an invalid choice, prompt for new choice.

If user chooses to edit, display the edit menu:

Prompt for and get user's edit choice.

If user chooses to crop, get specifications for the desired crop, then crop image to specifications.

Display the new and the old image.

If user chooses to dim or brighten the image, do so according to the appropriate functions below

Display the new and the old image

Prompt the user for whether to save the edit

If yes, prompt for the file name to save to. Establish a writing connection to this file, then save the image according to the saveNewImage function below.

Close the new (writing) file stream.

Close the old (reading) file stream.

displayMenu()

INPUT PARAMETERS: none

RETURNED OUTPUT: none

FUNCTIONALITY: display numerically labeled options for the user to choose from, including load new image, display current image, edit current image, and exit program

getMenuChoice()

INPUT PARAMETERS: none

RETURNED OUTPUT: int menuChoice

FUNCTIONALITY: Prompt the user for a choice and scan the user entered number for their choice and save it in the returned variable

loadNewImage()

INPUT PARAMETERS: int rowSize, int colSize, int imageArray, FILE* filePtr

RETURNED OUTPUT: none

FUNCTIONALITY: Assign all elements in imageArray to the corresponding values in the file

displayImage()

INPUT PARAMETERS: int rowSize, int colSize, int imageArray

RETURNED OUTPUT: none

FUNCTIONALITY: in a loop, display each value in imageArray with its corresponding value according to the system: 0 = " ", 1 = ".", 2 = "o", 3 = "O", and 4 = "0"

displayEditMenu()

INPUT PARAMETERS: none

RETURNED OUTPUT: none

FUNCTIONALITY: display numerically labeled options for the user to choose from, including crop image, dim image, brighten image

getEditChoice()

INPUT PARAMETERS: none

RETURNED OUTPUT: int editChoice

FUNCTIONALITY: Prompt user for a choice and scan the user entered number for their choice and save it in the returned variable

getCropSpecs()

INPUT PARAMETERS: int rowSize, int colSize, int imageArray, int* startRowPtr, int* endRowPtr, int* startColPtr, int* endColPtr

RETURNED OUTPUT: none

FUNCTIONALITY: prompt user for the coordinates (in terms of rows and columns in the original image) that they would like to crop the image to. Get these values and save them via the appropriate pointers.

cropImage()

INPUT PARAMETERS: int rowSize, int colSize, int imageArray, int startRow, int endRow, int startCol, int endCol

RETURNED OUTPUT: int newImageArray

FUNCTIONALITY: Assign newImageArray the values found between the rows in imageArray from startRow to endRow and the columns in imageArray from startCol to endCol

dimImage()

INPUT PARAMETERS: int rowSize, int colSize, int imageArray

RETURNED OUTPUT: int newImageArray

FUNCTIONALITY: Assign all values in newImageArray the value minus one at corresponding coordinates in imageArray, unless the value at a point equals zero, in which case assign the respective point in newImageArray zero.

brightenImage()

INPUT PARAMETERS: int rowSize, int colSize, int imageArray

RETURNED OUTPUT: int newImageArray

FUNCTIONALITY: Assign all values in newImageArray the value plus one at corresponding coordinates in imageArray, unless the value at a point equals four, in which case assign the respective point in newImageArray four.

saveNewImage()

INPUT PARAMETERS: int rowSize, int colSize, int imageArray, FILE* filePtr

RETURNED OUTPUT: none

FUNCTIONALITY: Print the newly created image to the newly created file with the name the user

specified