

Design Document
Kace Alejandro
Image Processing

Data:

Field - number of “pixels” we will be approximating

Usage:

1. Display the main menu options.
2. Based on user input, execute the corresponding functionality.
3. If the user chooses to load a new image:
 - Prompt for the file name.
 - Attempt to read the image file.
 - Display the loaded image.
4. If the user chooses to edit the current image:
 - Display the edit menu options.
 - Based on user input, execute the chosen editing functionality.
 - Prompt the user to save the edited image.
 - If the user agrees, prompt for a file name and save the edited image.
5. If the user chooses to exit the program, terminate execution.

Functions:

main()

Data:

None

Functionality:

Display the main menu options.

Based on user input, execute the corresponding functionality (load image, edit image, save image, or exit program).

load_image()

Input Parameters:

Returned output:

Functionality: Prompt the user for the file name, attempt to read the image file then display the loaded image.

edit_image()

Input Parameters: None

Returned output: None

Functionality: Display the edit menu options. Based on user input, execute the chosen editing functionality, then prompt the user to save the edited image. If the user agrees, prompt for a file name and save the edited image.

save_image()

Input Parameters: None

Returned output: None

Functionality: Prompt the user for a filename to save the edited image.

crop()

Input Parameters: None

Returned output: None

Functionality: Provide the user with a way to specify which smaller section of the original image they would like to use to create an edited image.

dim()

Input Parameters: None

Returned output: None

Functionality: Create an edited image where each pixel is one step "dimmer".

brighten()

Input Parameters: None

Returned output: None

Functionality: Create an edited image where each pixel is one step "brighter".