Development Diary

16/10/2024 (10:00 - 14:00)
Today, our main goal was to begin implementing the `QueuePixel` class, which is crucial for loading pixels and moving forward with the project's first phase. However, we encountered some initial issues related to image loading using the library. After troubleshooting and resolving these problems, we shifted our focus back to the `QueuePixel` implementation.
Due to the number of methods involved and the errors that arose during the process (an image had to be added for testing purposes), we were unable to complete `QueuePixel` today. Nevertheless, we took the opportunity to draft the design for `OpStack` and its corresponding `Operation` elements, which will be used in the second phase of development.

17/10/2024 (10:00 - 14:00)
Today, we successfully tackled the first operation, `negative`. To implement this, we made use of a queue, and it functioned as expected without any significant issues.
Next, we started working on the user interface, creating a simple greeting system and an option menu that explains the available operations. Now, the interface can call the next operation in `OpStack`, execute it, and produce an output image with a proper file name.

19/10/2024 (16:00 - 20:00)
Our objective for today's session was to implement as many operations as possible. We initially focused on the `getFlipped` operation but ran into an unexpected issue that made us reconsider how we were using the tiny-image library and its data structures. This forced us to reevaluate and adjust our approach in order to fully understand the library's internal workings.

20/10/2024 (10:00 - 14:00)
Today's focus was on completing the implementation of all necessary data structures, as well as the `getDarken` and `getFlopped` operations. We managed to successfully implement both operations and wrap up the development of all data structures. However, we encountered difficulties with performing multiple operations in sequence and managing queue duplication. These issues will need further investigation before moving on to more complex operations like `sorting`.

22/10/2024 (12:00 - 14:00)
Today, we made adjustments to ensure our code followed the project requirements. This involved reducing the `main()` function to just three lines by offloading the user interface and class calls into a `Core` class. We also fixed several typos and resolved some issues related to inconsistent naming conventions across the project.
In addition, we introduced functionality to prompt users for the input image name and handle output image naming more efficiently.

23/10/2024 (10:00 - 14:00)

After reviewing our design, we decided not to use the `CircularSinglyLinkedList` to handle the `getDarken` operation due to its unreasonable complexity. Instead, we opted for a more straightforward approach, involving a stack, two queues, and a doubly linked list for handling different operations. This should provide better performance and simplify the code structure. Next we tackled the `getSort` operation and decided to use the double linked list to perform the operation. Having finished it we dived into the tiring process of documentation.

24/10/2024 (8:00-11:30)

Today's main objective was to advance in the documentation, we were able to almost finish commenting the code, we only have left `Operations`. Moving on to the documentation, we made the UML, Class diagrams, and explained the behavior of the program.

26/10/2024 (10:00-14:00)

After finishing commenting the code we came back to fix the `getSort` operation, and added multiple test images. Focusing on the documentation, today we tackled the data flow explanation, definition of operations of the ADT, explanation of classes, diagrams with source files and their relationships. Last but not least we started with the efficiency calculations.

27/10/2024 (9:00-13:00)

Today we dived into the documentation, finishing it, by doing the box diagrams, explanation of the ADT adaptations, possible enhancements and the reasoning behind our decisicions. We had to spend time searching for information to back our documentation process and the underlying hardships.