

Objectives

1. Implement a full-dimension image display within the control panel, allowing users to see the image in its actual size.
2. Develop a "Fix Zoom" functionality within the zoom controls to retain the current zoom level during processing.
3. Update control panel and image display features to support the new zoom functionality without reverting to the default size during fixed zoom.
4. Address feedback from the team lead to implement a "detect and remove line" function using a double 2D pointer, storing row and column data efficiently.

Activities

- **Zoom Fix Implementation:**
 - Created a "Fix Zoom" function within the zoom controls to retain the current zoom level and enable function application while zoom is active.
 - Updated image classes to store zoomed images and track whether zoom is fixed, allowing smooth toggling between zoom modes.
 - Modified the control panel to support toggling between fixed and unfixed zoom states, updating button labels and status indicators.
- **Bug Fixes & Adjustments:**
 - Noticed and fixed issues where a black line appeared during fixed zoom, and the zoom level reverted to 1.0x upon unfixing.
 - Disabled zoom-in, zoom-out, and reset buttons when the zoom is fixed to prevent accidental zoom adjustments.
 - Changed button colors for better UX: blue indicates zoom mode is active, and red indicates fixed zoom with a prompt message to remind users to unfix zoom first.
- **Code Optimization:**
 - Separated the zoom functionality into individual header and source files, restructuring code for better readability and modularity.
 - Removed `detectDarkLines()` within `fixZoom()` to prevent detected lines from affecting output during fixed zoom operations.
- **Line Detection Task:**
 - Started working on the new "detect and remove line" function using a double 2D pointer to store row and column data.

- Planned to use a struct within the 2D pointer to hold row and column values, aiming for efficient data handling.
- Began converting the line detection and removal process into the 2D pointer format, testing the approach to accurately detect and remove lines.

Achievements

- Successfully implemented "Fix Zoom" functionality, enabling users to retain zoom level during image processing and improving the zoom usability.
- Updated the `updateImageDisplay()` to consistently display labels for detected lines at any zoom level, enhancing visibility.
- Improved UX by adding clear button status indicators (color-coded) to communicate zoom mode states and fixed zoom.
- Set up foundational code for the "detect and remove line" function with a double 2D pointer for efficient line handling in images.

Problems & Solutions

- **Problem:** Fixed zoom occasionally displayed an additional black line during processing.
 - **Solution:** Identified the issue with the `detectDarkLines()` function and removed it from `fixZoom()` to prevent unintended effects on output.
- **Problem:** Zoom level reverted to 1.0x when zoom was unfixed.
 - **Solution:** Saved the pre-fix zoom level and used it to restore the appropriate zoom level after unfixing.
- **Problem:** The zoom functionality reset after deactivation, and a dialog to retain zoom state was initially planned but found ineffective.
 - **Solution:** Removed the dialog function and adjusted zoom behavior so that zoom level remains while zoom mode is active but resets correctly when deactivated.
- **Problem:** Button color did not always update to reflect the current zoom state.
 - **Solution:** Improved button interactivity by dynamically updating colors to match active zoom state and fixed zoom, offering better visual feedback.
- **Problem:** Structuring the "detect and remove line" functionality with double 2D pointers was complex.

- **Solution:** Started designing a struct within the 2D pointer to store row and column data, allowing efficient data organization and making progress in code conversion.