

Objectives:

1. Implement advanced features for the R&D project, including box selection, image manipulation, and enhanced UI functionalities.
 2. Debug and improve the box selection and rotation features for usability and accuracy.
 3. Refine the UI layout and functionality for an optimal user experience.
 4. Integrate mouse tracking and ensure seamless interaction with the OpenGL viewport.
-

Activities:

- **Box Selection Implementation:**

- Designed the box selection functionality using mouse events:
 - Left-click and drag to create a semi-transparent blue rectangle (50% transparency).
 - Finalized selection when the mouse button is released, storing coordinates in scene coordinates.
- Added debug messages to track the process, adjusted the algorithm, but temporarily removed the function due to persistent failures.

- **Rotation Functionality:**

- Added clockwise and counterclockwise image rotation:
 - Clockwise:
$$\text{rotated}[j][\text{height}-1-i] = \text{original}[i][j]$$
$$\text{rotated}[j][\text{height}-1-i] = \text{original}[i][j]$$
 - Counter-clockwise:
$$\text{rotated}[\text{width}-1-j][i] = \text{original}[i][j]$$
$$\text{rotated}[\text{width}-1-j][i] = \text{original}[i][j]$$
 - Display updated with new dimensions, and status text reflects changes.

- **UI Enhancements:**

- Modified the window creation flags to allow resizing and maximization:
 - Added `SDL_WINDOW_RESIZABLE | SDL_WINDOW_MAXIMIZED`.
 - Enhanced event handling for resize and maximize actions, ensuring accurate viewport adjustments.

- Improved button organization and relocated the status bar to the bottom, moving zoom-level text to the right side of the status bar.
- Integrated the status bar into the top panel and removed bottom padding, maximizing viewport usage.
- Updated mouse interactions:
 - **Pan:** Press Alt and drag the left mouse button.
 - **Zoom:** Press Ctrl and scroll using the mouse roller.
- **Mouse Tracking and Selection Box via OpenGL and ImGui:**
 - Implemented OpenGL and ImGui overlay integration for selection box rendering:
 - Used `drawOverlay()` method to handle ImGui overlay for selection box visualization.
 - Maintained OpenGL rendering for the image and added ImGui drawing utilities for the selection box.
 - Ensured accurate coordinate conversion and real-time adjustments for zoom and pan transformations.
 - Fixed bugs related to type mismatches (e.g., `ImVec2` vs. `glm::vec2`) and unified coordinate handling.
 - Debugged selection box behavior, addressing initialization and member variable inconsistencies.
- **Other Enhancements:**
 - Added `processCurrentImage()` to handle interlace and merge functionality.
 - Integrated rotation and viewport functionalities into a cohesive system.
 - Ensured modularity in UI layout and interaction handling for future extensibility.

Achievements:

1. Successfully implemented and debugged the rotation function with real-time updates.
2. Enhanced the application's UI layout for better usability and viewport efficiency.
3. Established an integrated system for handling zoom, pan, and selection box interactions using OpenGL and ImGui.
4. Debugged and resolved multiple inconsistencies in rendering and interaction logic.

Problems and Solutions:

1. **Problem:** Selection box was not displaying or behaving correctly.
 - **Solution:** Added debug messages, reworked the draw algorithm, and implemented an ImGui overlay for better integration. Fixed type mismatches and ensured real-time coordinate adjustments.
2. **Problem:** Persistent bugs in the selection box drawOverlay functionality, causing the box to not remain fixed.
 - **Solution:** Encapsulated overlay logic into a modular function, corrected variable initialization, and streamlined coordinate transformations to align with pan and zoom operations.
3. **Problem:** Inconsistent viewport adjustments during window resizing.
 - **Solution:** Enhanced event handling for resize and maximize actions, ensuring seamless updates to viewport dimensions and layout.
4. **Problem:** Mouse tracking for pixel values and coordinates not working.
 - **Solution:** Identified issues with the image output view and delayed implementation until other functionalities are stabilized.
5. **Problem:** Conflicts between OpenGL rendering and ImGui overlay.
 - **Solution:** Separated rendering responsibilities, leveraging ImGui utilities for overlays without interfering with OpenGL's rendering pipeline.