Objectives

- Continue addressing new R&D requirements for project development.
- Implement mouse-driven box selection functionality in the OpenGL viewport.
- Add rotation capabilities to the application, allowing clockwise and counterclockwise transformations.
- Enhance user interaction features, including zoom, pan, and dynamic UI updates.
- Investigate and resolve issues with the selection box rendering and functionality.
- Improve the UI layout and viewport utilization for better usability.

---

2. Activities

- **Box Selection Implementation:**
    - Designed a mouse-driven box selection feature:
        - Left-drag to create a semi-transparent selection box.
        - Selection coordinates stored in scene coordinates, accounting for zoom and pan.
    - Implemented a temporary overlay using ImGui for selection box rendering.
    - Added debugging messages to track the selection box workflow and adjusted the drawing algorithm.
    - Encountered issues with selection box stability; temporarily removed functionality for further debugging.
- **Rotation Feature Development:**
    - Implemented clockwise and counterclockwise rotation using matrix transformations.
    - Automatically updated the display to show rotated images and updated status text with new dimensions.
- **UI and Interaction Enhancements:**
    - Updated the main window to maximize based on device size using SDL_WINDOW_RESIZABLE and SDL_WINDOW_MAXIMIZED flags.
    - Adjusted the UI layout:
        - Status bar integrated into the top panel.
        - Moved zoom-level text to the right side of the status bar.

        - Removed bottom padding for full viewport utilization.
    - Refined mouse interaction:
        - Alt + Left-drag for panning.
        - Ctrl + Scroll for zooming.
        - Added checks for modifier key states using SDL_GetKeyboardState.
- **Image Processing Updates:**
    - Created `processCurrentImage()` for handling interlace and merge functions, automatically updating the display with processed data.
    - Enhanced window event handling for resize and maximize events, ensuring consistent viewport adjustments.
- **Pixel Tracking Development:**
    - Added mouse tracking for pixel values and coordinates, aiming to display 16-bit pixel details in the status bar.
    - Debugged image output issues causing delays in displaying pixel details.

---

3. Achievements

- Successfully implemented rotation functionality with seamless UI updates.
- Enhanced the main window and viewport to improve usability and layout efficiency.
- Developed an efficient overlay method using ImGui for rendering selection boxes on the OpenGL viewport.
- Improved interaction handling for zoom, pan, and selection, maintaining scene coordinate accuracy.
- Resolved type mismatch issues by adopting glm::vec2 for all scene coordinates.
- Encapsulated overlay logic into a modular `drawOverlay()` function for clarity and reusability.

---

4. Problems & Solutions

1. **Problem:** Selection box rendering failed to stabilize.

    - **Solution:** Used ImGui overlay for rendering, maintaining separation from OpenGL rendering. Corrected type mismatches with glm::vec2 and improved event handling logic.
2. **Problem:** Pan and zoom functions interfered with selection box interaction.
    - **Solution:** Isolated tracking for pan, zoom, and selection operations, ensuring proper handling of modifier keys (Alt, Ctrl).
3. **Problem:** Pixel tracking details not displayed.
    - **Solution:** Debugged image output view and postponed further implementation until other critical functions were stabilized.
4. **Problem:** Window resizing affected viewport consistency.
    - **Solution:** Enhanced window event handling to dynamically adjust the viewport and UI layout for maximize and resize events.
5. **Problem:** Selection box functionality disrupted by pan and zoom transformations.
    - **Solution:** Adjusted overlay rendering to dynamically align with zoom and pan, ensuring accurate transformations and seamless integration.