1. Objectives

- Debug and test unresolved issues, focusing on `processCurrentImage()` and selection box functionality.
- Implement a stable and functional selection box mechanism.
- Introduce and implement an undo feature for image modifications.
- Refactor code by separating multiple classes into individual headers and source files for improved modularity and maintainability.

---

2. Activities

- **Debugging and Fixing `processCurrentImage()` Issues:**
    - Resolved access violation by initializing `m_processImage` in the constructor using `std::make_unique<CGProcessImage>().`
    - Added null pointer checks in `processCurrentImage()` to prevent errors.
    - Ensured proper memory management for `m_processedData` and `inputMatrix` with allocation and cleanup handled in the destructor.
- **Selection Box Implementation:**
    - Attempted several fixes for the selection box:
        - Added logic in `handleMouseEvent` to finalize geometry upon mouse release.
        - Ensured proper rendering by marking the selection box as visible and finalizing its geometry in `updateSelection`.
        - Debugged coordinate mismatches between ImGui overlay and OpenGL rendering.
        - Transitioned rectangle drawing to OpenGL for improved stability and visual clarity.
        - Simplified shader logic for rectangle styling (solid red border, semi-transparent fill).
        - Verified OpenGL states, viewport alignment, and geometry dimensions through debugging output.
        - Studied online resources and forums for best practices in OpenGL-based rectangle drawing.

        - Temporarily removed all selection box functions to plan a reimplementation.

- **Undo Feature Implementation:**
    - Created a history stack (`m_undoHistory`) to store previous states of the image.
    - Implemented `pushToHistory()` to save the current state before each modification.
    - Added an `undo()` function to restore the most recent state from the history stack.
    - Integrated history saving into image processing, rotation, and calibration functions.
    - Cleared the history stack upon loading a new image.
    - Displayed appropriate status messages for undo actions or when the history stack is empty.
- **Code Refactoring:**
    - Separated multiple classes from `graphics_item.h` into individual header and source files.
    - Improved modularity and readability of the codebase.

---

3. Achievements

- Resolved access violation in `processCurrentImage()` and ensured robust memory management.
- Enhanced the undo functionality, enabling seamless restoration of previous image states across multiple operations.
- Simplified the codebase by refactoring `graphics_item.h` into separate files for better maintainability.
- Made progress in understanding and implementing a functional selection box, identifying key issues for further debugging.
- Enhanced OpenGL rendering logic for consistent and visually clear selection box implementation.

---

4. Problems & Solutions

1. **Problem:** Access violation in `processCurrentImage()` due to uninitialized variables.
    - **Solution:** Initialized `m_processImage` in the constructor, added null checks, and ensured proper memory management for dynamically allocated variables.
2. **Problem:** Selection box disappears after mouse release; conflicts between ImGui overlay and OpenGL rendering.
    - **Solution:** Transitioned selection box rendering to OpenGL, simplifying shaders and ensuring consistent coordinate usage. Temporarily removed functionality to reimplement with a more robust design.
3. **Problem:** Rectangle visibility issues in OpenGL rendering pipeline.
    - **Solution:** Verified OpenGL states, viewport settings, and alignment. Added debugging outputs for geometry and rendering calls. Delayed further implementation until critical issues are resolved.
4. **Problem:** Code complexity in `graphics_item.h`.
    - **Solution:** Separated classes into individual headers and source files, improving code organization and modularity.
5. **Problem:** No undo functionality for image modifications.
    - **Solution:** Implemented a history stack and integrated undo actions into all key modification processes, ensuring smooth reversibility.