Objectives:

1. Debug and enhance functionality for the selection box and image processing.
2. Resolve access violations and memory management issues in the `processCurrentImage()` function.
3. Refine user interaction features, including an undo mechanism and improved mouse event handling.
4. Optimize code structure by separating multiple classes into individual header and source files.

Activities:

- **Debugging `processCurrentImage()` Function:**
    - Resolved access violation errors during `m_processImage` configuration by:
        - Initializing `m_processImage` using `std::make_unique<CGProcessImage>()` in the constructor.
        - Adding null pointer checks in `processCurrentImage()` to prevent issues.
        - Managing memory allocation for `m_processedData` and `inputMatrix` to avoid leaks, ensuring proper cleanup in the destructor.
- **Selection Box Debugging:**
    - Refined mouse event handling to finalize and render the selection box after releasing the mouse:
        - Updated `handleMouseEvent` and `updateSelection` for accurate geometry calculation and rendering.
        - Enhanced visibility management to ensure the selection box remains visible until explicitly cleared.
        - Integrated debugging outputs to track dimensions and rendering states.
    - Explored and attempted various approaches:
        - Shifted rendering from ImGui overlay to OpenGL, using shaders for solid borders and semi-transparent fills.

        - Verified viewport alignment, blending states, and coordinate transformations.
        - Referred to online resources for OpenGL rectangle drawing techniques but encountered persistent issues.
    - Temporarily removed selection box-related functions for reimplementation after further research.
- **Undo Functionality:**
    - Added an undo feature for image modifications:
        - Implemented a history stack (`m_undoHistory`) to store image states.
        - Created `pushToHistory()` to save the current state before modifications.
        - Added `undo()` to revert to the most recent state and integrated it into image processing, rotation, and calibration.
        - Cleared the history stack when loading new images.
        - Displayed status messages to inform users of undo actions or empty history stacks.
- **Code Organization:**
    - Separated multiple classes from `graphics_item.h` and its corresponding `.cpp` file into individual headers and source files for better modularity and maintainability.

Achievements:

1. Resolved access violation issues in `processCurrentImage()` by enhancing initialization and memory management.
2. Improved user interaction with the addition of an undo button, enabling seamless reversion of modifications.
3. Refactored code structure, promoting better organization and clarity.
4. Progressed on the selection box functionality, with partial fixes and plans for reimplementation.

Problems and Solutions:

1. **Problem:** Access violation in `processCurrentImage()` during memory configuration.
   - **Solution:** Initialized `m_processImage` properly, added null pointer checks, and managed memory allocation and cleanup.
2. **Problem:** Selection box disappears or behaves incorrectly after releasing the mouse.
   - **Solution:** Refined mouse event handling and rendering logic. Attempted OpenGL-based rendering with debug outputs to trace issues, but full resolution remains pending.
3. **Problem:** Overcomplicated selection box logic caused coordinate mismatches and rendering conflicts.
   - **Solution:** Simplified rendering pipeline, removed ImGui overlay for box drawing, and focused on OpenGL shaders for consistent results. Issues remain under investigation.
4. **Problem:** Undo functionality was unavailable for modifications.
   - **Solution:** Implemented a history stack with integrated undo functionality, providing robust state management.
5. **Problem:** Codebase was difficult to manage due to multiple classes in single files.
   - **Solution:** Separated classes into individual files, enhancing modularity and ease of maintenance.