1. Objectives

- Refactor the crop functionality to use displayed image data within the window for accurate results.
- Address issues with coordinate mapping, scaling, and transformation to ensure the crop function reflects user selections accurately.
- Test and validate the functionality across edge cases like extreme zoom levels and non-uniform scaling.

---

2. Activities

- **Fixing Crop Functionality to Use Displayed Image Data:**
    - **Problem:**
        - The crop function used window coordinates, causing mismatches with the displayed image.
        - Transformations applied to fit the image in the window were not accounted for during cropping.
    - **Solutions:**
        - Updated cropToSelection to calculate coordinates using transformed display data.
        - Mapped selection rectangle coordinates from the view transformation matrix to the displayed image's coordinate space.
        - Enhanced GraphicsView to expose view transformation details for accurate coordinate calculations.
        - Modified TextureItem to manage and provide access to displayed image data for cropping operations.
        - Validated coordinates against the displayed image bounds to ensure proper cropping.
    - **Outcome:**

        - Cropping now operates on transformed coordinates, ensuring alignment with the displayed image.

- **Improving Crop Functionality with Display Scaling:**
    - **Problem:**
        - Incorrect coordinate calculations led to mismatches between user selections and cropped areas.
        - Scaling factors were not properly calculated or applied during cropping.
    - **Solutions:**
        - Calculated scaling factors for width and height:
            - displayScaleX = windowWidth / processedImageWidth
            - displayScaleY = windowHeight / processedImageHeight
        - Adjusted cropToSelection to:
            - Use scaling factors to map window coordinates to processed image coordinates.
            - Clamp dimensions within valid bounds using std::clamp with appropriate type casting.
        - Ensured m_imgData updates reflect the transformed coordinates and correct cropping.
        - Resolved compilation errors by including necessary headers like <algorithm>.
    - **Outcome:**
        - Cropping now correctly maps window view coordinates to processed image coordinates, matching user selections.

---

3. Achievements

- Successfully updated the crop function to use transformed display data, improving accuracy.

- Implemented display scaling factors to ensure consistent and precise coordinate mapping.
- Resolved issues with compilation errors and ensured proper clamping of dimensions.

---

4. Problems & Solutions

1. **Problem:** Cropping used window coordinates, causing mismatches with the displayed image.
   - **Solution:** Integrated view transformation and display scaling to calculate accurate crop coordinates.
2. **Problem:** Scaling factors were not correctly calculated or applied.
   - **Solution:** Calculated and used displayScaleX and displayScaleY for proper mapping of selection rectangles.
3. **Problem:** Precision discrepancies due to floating-point rounding errors.
   - **Solution:** Added clamping and type casting to minimize precision issues, though further testing is required for edge cases.
4. **Problem:** Edge cases, such as extreme zoom levels or non-uniform scaling, may cause inconsistencies.
   - **Solution:** Identified these cases for further testing and refinement.