1. Objectives

- Continue fixing persistent issues in the box selection functionality.
- Ensure accurate viewport adjustments, mouse coordinate alignment, and rendering transformations.
- Implement and refine additional debugging tools to pinpoint remaining issues.
- Enhance OpenGL rendering for transparency, layering, and accurate object placement.

---

2. Activities

- **First Attempt to Fix Box Selection Issues:**
  - Implemented view scaling to maintain image proportions in the viewport.
  - Added aspect ratio handling logic to prevent image stretching or distortion.
  - Enhanced centering logic for proper image alignment within the viewport.
  - Debugged `TextureItem::setImage` to ensure successful OpenGL texture uploads:
    - Verified buffer bindings, texture IDs, and shader uniforms.
    - Corrected texture binding issues and ensured valid OpenGL texture creation.
    - Adjusted rendering logic to resolve z-ordering conflicts, ensuring images are rendered before overlays.
  - **Result:** Fixed image rendering issues; images now load correctly, fit the viewport, and respond to zoom and pan. However, selection box issues remained unresolved.
- **Second Attempt to Fix Box Selection Issues:**
  - Updated `RectItem::draw` with OpenGL blending (GL_BLEND) for transparent rectangle rendering:
    - Set initial rectangle colors (blue fill, red border) with 30% transparency.
    - Corrected shader logic and fixed compilation errors by aligning variables.
    - Verified blending and depth testing functionality for proper rectangle visibility.

    - **Result:** Semi-transparent rectangles with proper borders were rendered successfully.
- **Debugging and Updates:**
    - Added `PointItem` to render a red point at the mouse click location:
        - Integrated point rendering with rectangle drawing logic.
        - Addressed viewport adjustments to match mouse clicks with scene coordinates.
        - Configured OpenGL blending and point size for better visibility.
    - Added boundary checks for mouse interactions:
        - Defined viewport boundaries, considering control panel height and window dimensions.
        - Clamped mouse coordinates within valid boundaries for all events (button press, motion, wheel).
        - Cleaned up active drawing operations when the mouse exited the viewport.
    - **Issues Still Persist:**
        - Multiple boxes can be drawn if cleanup logic fails to remove the previous rectangle.
        - Mouse-click inaccuracies due to coordinate mismatches and viewport height miscalculations.
        - Simultaneous rendering of points and rectangles may result in overlap and conflicts.
        - Boundary enforcement needs refinement to eliminate gaps and overlaps near edges.

---

3. Achievements

- Successfully resolved OpenGL texture and shader issues, ensuring proper image rendering and transformations.
- Rendered semi-transparent rectangles with correct blending and styling.
- Added and tested point rendering at mouse click locations, improving debugging tools.
- Defined viewport boundaries and added boundary checks to restrict mouse interactions within valid areas.

- Improved understanding of viewport transformations and OpenGL rendering pipelines.

---

4. Problems & Solutions

1. **Problem:** Persistent mouse-click inaccuracies and misaligned transformations.
    - **Solution:** Debug viewport transformations, refine scaling and offset calculations, and verify coordinate mappings between mouse input and scene.
2. **Problem:** Multiple rectangles drawn without proper cleanup.
    - **Solution:** Add stricter logic to clear previous rectangles (`m_rectItem`) before creating new ones and verify cleanup operations.
3. **Problem:** Boundary enforcement inconsistencies near edges and control panel.
    - **Solution:** Refine boundary calculations to consider all offsets and dimensions accurately, ensuring no gaps or overlaps.
4. **Problem:** Overlap and conflicts between points and rectangles.
    - **Solution:** Separate logic for rendering points and rectangles, ensuring they do not interfere or share unintended dependencies.
5. **Problem:** Selection box rendering still not fully functional despite progress.
    - **Solution:** Temporarily pause additional features, focus on stabilizing existing functions (box drawing, viewport adjustments) before reintroducing complex interactions.