You've accurately pinpointed a classic issue that arises when mixing Java-based applications (like Maple) with mixed screen orientations and anti-aliasing. Here's a breakdown of the problem and potential troubleshooting steps, keeping in mind the complexities of Java's graphics rendering:

Understanding the Problem:

- **Java and Graphics Rendering:**
  - Java's graphics rendering relies heavily on the underlying operating system's graphics libraries (e.g., Direct2D on Windows, Quartz on macOS).
  - When screen rotations are involved, the operating system must handle transformations and scaling, which can introduce inconsistencies, especially with anti-aliasing.
  - Java's handling of high DPI and multi-monitor setups has historically been problematic, leading to rendering artifacts.
- **Anti-Aliasing and Screen Rotation:**
  - Anti-aliasing smooths out jagged edges in text and graphics, but it requires precise pixel manipulation.
  - When a screen is rotated, the pixel grid changes, and the anti-aliasing algorithms may not handle the transformation optimally, resulting in blurry or distorted text.
- **Maple's Java Foundation:**
  - Maple, being a Java application, inherits the limitations and quirks of Java's graphics rendering.
  - This means that any issues related to screen rotation and anti-aliasing in Java will likely manifest in Maple as well.

Troubleshooting Steps:

1. **Java Runtime Environment (JRE) Updates:**
   - Ensure you have the latest version of the JRE that Maple is using. Newer JRE versions often include improvements to graphics rendering and multi-monitor support.
   - It is possible that Maple uses an embedded JRE, in that case, you will have to wait for a Maple update.
2. **Operating System Graphics Drivers:**
   - Update your graphics card drivers to the latest version. Outdated drivers can cause rendering problems.
   - Ensure that the operating system itself is updated.
3. **Maple's Graphics Settings:**
   - Check Maple's preferences or options for any graphics-related settings. Look

for options related to anti-aliasing, font rendering, or hardware acceleration.
    - Try disabling hardware acceleration inside of Maple.
    - Try disabling anti-aliasing inside of Maple.
4. **Operating System Scaling and Display Settings:**
    - Experiment with different display scaling settings in your operating system. Inconsistent scaling across multiple monitors can cause rendering issues.
    - Ensure that all monitors have the same scaling setting.
    - Ensure that the rotated display has the same scaling as the non rotated displays.
    - Try different screen resolutions.
5. **Java Command-Line Options:**
    - You can try modifying Java's command-line options to influence its graphics rendering.
    - Some potentially relevant options include:
        - -Dsun.java2d.opengl=true (Enables OpenGL rendering)
        - -Dsun.java2d.d3d=true (Enables Direct3D rendering on Windows)
        - -Dsun.java2d.noddraw=true (Disables DirectDraw on Windows)
        - -Dsun.java2d.dpiaware=true (Enables DPI awareness)
    - How you apply those options depends on how Maple is designed. If maple has a shortcut, you can try to edit the shortcut, and add those lines to the target field of the shortcut.
    - Be aware that these options can have unintended consequences, so test them carefully.
6. **External Display Management Software:**
    - Some third-party display management tools can provide more granular control over display settings, which might help resolve the issue.
7. **Report the Issue to Maplesoft:**
    - Since the problem is specific to Maple, report the issue to Maplesoft's support team. They may be aware of the issue or have a workaround.
    - Provide them with detailed information about your system, including the operating system, graphics card, monitor setup, and Maple version.
8. **Java version check:**
    - Check what java version Maple is using. If it is a very old version, that is likely the root of the problem.

Key Considerations:

- The exact solution may depend on your specific hardware and software configuration.
- Java's graphics rendering can be complex, and there may not be a single,

universal fix.
- Keep in mind that rotated displays will always represent a more complex situation for graphics rendering.

By systematically working through these troubleshooting steps, you should be able to narrow down the cause of the issue and find a suitable solution.