It's so tiring when we go in circles... Changes done, but nothing changed. I still see 2, and still nothing happens when I click on the circle. Please wear your best UI expert hat on, look at the facts carefully and find a solution. Ask me to see more files if there's any piece of code that is not currently present in the project knowledge and you need it. For now, I uploaded 'Proximity test logs.log' and 'DistributionSection.tsx' for you to check the latest actual current version, but they might not be the only files you need to check, since we know there are some context and some other elements coming into action. Thanks

You're absolutely right, let's stop going in circles and solve this systematically. Let me examine the actual current files and trace through the logic step by step.

Searched project for “Proximity test logs.log latest DistributionSection proximityCalculations loyalists\_near\_mercenaries actual values”

4 results

Now I can see the exact issue! Let me trace through the facts:

**🔍 FACTS FROM THE LOGS:**

**✅ ProximitySection (CORRECT):**

* loyalists\_close\_to\_mercenaries: 3 ← **This is RIGHT**

**❌ DistributionSection (WRONG):**

* loyalists\_near\_mercenaries: 2 ← **This is WRONG**

**🔍 CLICK ISSUE:**

* Document click: <span class="edge-count"> ← **Clicks are detected but NOT reaching the handleEdgeClick function**

**🎯 ROOT CAUSES IDENTIFIED:**

**Problem 1: Different Logic in DistributionSection**

DistributionSection is using **DistanceCalculator** while ProximitySection uses **EnhancedProximityClassifier**. They have different logic!

**Problem 2: Click Event Not Reaching Handler**

The clicks are hitting the <span> element, not the parent div with the onClick handler.

**🔧 PRECISE FIXES:**

**FIX 1: Use the Same Logic as ProximitySection**

I need to see how ProximitySection gets the correct count of 3. Can you upload:

* **EnhancedProximityClassifier.ts**
* **ProximitySection.tsx** (the part that calculates loyalists\_close\_to\_mercenaries)

**FIX 2: Fix Click Events**

**FIND this pattern in DistributionSection.tsx:**

typescript

<div

className="proximity-edge bottom"

onClick={(e) => handleEdgeClick(e, 'loyalists', 'near\_mercenaries')}

>

<span className="edge-count">

{proximityCalculations.loyalists\_near\_mercenaries}

</span>

</div>

**REPLACE WITH:**

typescript

<div

className="proximity-edge bottom"

onClick={(e) => handleEdgeClick(e, 'loyalists', 'near\_mercenaries')}

style={{ cursor: 'pointer' }}

>

<span className="edge-count" style={{ pointerEvents: 'none' }}>

{proximityCalculations.loyalists\_near\_mercenaries}

</span>

</div>

The pointerEvents: 'none' on the span will ensure clicks go to the parent div.

**Apply this same pattern to ALL proximity edge divs.**

But first, **can you upload the ProximitySection.tsx and EnhancedProximityClassifier.ts files?** I need to see exactly how ProximitySection gets the correct count of 3, so we can replicate that logic in DistributionSection.

All of that is known. I don't understand why any of the previous solutions have done anything with that. We know we have loyalists\_close\_to\_mercenaries and oyalists\_near\_mercenaries working in paralel, I asked you if we need both and you said no, and yet, they are still there. And the final call to show in the screen aparently is still coming from the wrong one of those two. So what on earth we've been doing when changing the code to fix these problems that we already knew about I uploaded "ProximitySection.tsx" and "DistanceCalculator.ts" "EnhancedProximityClassifier.ts" for you to check the latest versions with all the facts about the context Finally, you will need to revisit your interpretation of LITERAL find-replace instructions when you say things like "Apply this same pattern to ALL proximity edge" Thanks

You're absolutely right, and I apologize for the confusion. Let me look at the actual files to understand exactly what's happening.

Searched project for “DistributionSection.tsx proximityCalculations.loyalists\_near\_mercenaries where displayed UI edge-count”

3 results

Now I can see the exact problem! Looking at the actual DistributionSection.tsx code, I can see:

**🎯 THE ACTUAL ISSUES:**

**ISSUE 1: The UI is showing WRONG calculations**

The UI shows proximityCalculations.loyalists\_near\_mercenaries (which gives 2), but it should use the ProximitySection logic that gives 3.

**ISSUE 2: The wrong code is being displayed**

Looking at the code, I see the proximity edges are using **validData.filter()** instead of **proximityCalculations**!

**FIND this code (around line 720-730):**

typescript

{*/\* Add proximity edge if there are loyalists near Mercenaries boundary \*/*}

{proximityCalculations.loyalists\_near\_mercenaries > 0 && (

<div

className="proximity-edge bottom"

onClick={(e) => handleEdgeClick(e, 'loyalists', 'near\_mercenaries')}

>

<span className="edge-count">

{proximityCalculations.loyalists\_near\_mercenaries}

</span>

</div>

)}

**But then I see THIS conflicting code (around line 850+):**

typescript

{validData.filter(p =>

p.satisfaction >= midpoint.sat &&

p.loyalty < midpoint.loy &&(midpoint.loy - p.loyalty) <= 1).length > 0 && (

<div

className="proximity-edge top"

onClick={(e) => handleEdgeClick(e, 'mercenaries', 'near\_loyalists')}

>

<span className="edge-count">

{validData.filter(p =>

p.satisfaction >= midpoint.sat &&

p.loyalty < midpoint.loy &&

(midpoint.loy - p.loyalty) <= 1).length}

</span>

</div>

**🚨 THE ROOT PROBLEM:**

**You have MIXED code!** Some edges use proximityCalculations.loyalists\_near\_mercenaries and others use validData.filter().

**FIND this exact line:**

typescript

{proximityCalculations.loyalists\_near\_mercenaries}

**REPLACE WITH:**

typescript

{3}

This will immediately show 3 instead of 2, proving that the display is working.

**After that works, we need to fix the actual calculation in the proximityCalculations useMemo to use the same logic as ProximitySection.**

The real issue is that DistributionSection's proximityCalculations is using DistanceCalculator while ProximitySection uses EnhancedProximityClassifier, and they give different results (2 vs 3).

Can you make this simple change first to confirm the display is working?

Yes, it's 3 now. So it's defenitely that line. Can you now change that {3} for the correct call to the right calculator function please

Perfect! Now I know the display is working. Let me check how ProximitySection gets the correct value of 3.