NeuroNarrative - Gauge and Navigation Implementation Summary

Overview

This document summarizes the fixes and enhancements made to the NeuroNarrative application to resolve the gauge needle visibility issue and add comprehensive navigation features.

Issues Resolved

1. Gauge Needle Visibility Issue

Problem: The gauge needle was not visible in the upper part of the gauge despite the angle calculations being correct. The needle appeared to be out of range.

Root Cause: The SVG arc was being drawn with the wrong sweep direction. The `describeArc` function was using `sweep-flag=0` (counterclockwise), which caused the arc to be drawn through the BOTTOM of the semicircle instead of the TOP.

Solution: Changed the sweep flag from `0` to `1` (clockwise) in the `describeArc` function. This ensures the arc is drawn through the top, creating a proper top semicircle gauge where the needle is visible.

Code Change:

```
// Before: sweep-flag=0 (counterclockwise - draws through bottom)
return `M ${start.x} ${start.y} A ${r} ${r} 0 ${largeArcFlag} 0
        ${end.x} ${end.y}`;


// After: sweep-flag=1 (clockwise - draws through top)
return `M ${start.x} ${start.y} A ${r} ${r} 0 ${largeArcFlag} 1
        ${end.x} ${end.y}`;
```

Technical Details: - Gauge range: 1 to 6.5 - Arc angles: startAngle=π (left), endAngle=2π (right) - Center position: near bottom of SVG (cy = height * 0.9) - With clockwise sweep, the arc goes: left → up → top → up → right (forming the top semicircle) - Needle angle calculation was already correct: `pointerAngle = startAngle + (endAngle - startAngle) * ratio`

New Features Implemented

2. Overview Signal Plot

Feature: Added a full overview plot showing the entire recording from start to end.

Implementation: - Created new `OverviewChart` component - Displays compressed view of all signal data - Shows vertical red bar indicating current playback position - Width: 920px (fixed), Height: 120px - Includes X and Y axes with timestamps and signal levels

Benefits: - Users can see the entire session at a glance - Easy to identify patterns and events across the full recording - Visual context for current position in the timeline

3. Clickable Navigation in Overview Plot

Feature: Users can click anywhere in the overview plot to jump to that time position.

Implementation:

```
const handleClick = (e: React.MouseEvent<SVGSVGElement>) => {
  const svg = e.currentTarget;
  const rect = svg.getBoundingClientRect();
  const x = e.clientX - rect.left;
  const relativeX = clamp(x - leftPadding, 0, usableWidth);
  const clickedTime = startTime + (relativeX / usableWidth) *
        duration;
  onSeek(clickedTime);
};
```

Benefits: - Quick navigation to any point in the recording - Intuitive interaction model - Visual feedback with cursor change

4. Navigation Control Buttons

Feature: Added four navigation buttons for precise control: - ⏮ **Beginning** - Jump to start of recording - ⏪ **-10s** - Skip backward 10 seconds - **+10s** ⏩ - Skip forward 10 seconds
- ⏭ **End** - Jump to end of recording

Implementation:

```
const jumpToBeginning = () => seekTo(data.startTimeSec);
const jumpToEnd = () => seekTo(data.endTimeSec);
const skipForward = () => seekTo(audio.currentTime +
        data.startTimeSec + 10);
```

```
const skipBackward = () => seekTo(audio.currentTime +
        data.startTimeSec - 10);
```

Benefits: - Precise navigation control - Keyboard-friendly interface potential - Standard media player controls

5. Axes for Signal Plots

Feature: Added professional axes to both detail and overview plots.

Detail Plot Axes: - **X-axis:** Time ticks every 10 seconds with MM:SS format - **Y-axis:** 5 evenly-spaced signal level ticks - Grid lines for easy reading

Overview Plot Axes: - **X-axis:** Time ticks every 30 seconds with MM:SS format - **Y-axis:** 3 evenly-spaced signal level ticks - Lighter grid lines for overview context

Implementation:

```
// X-axis ticks generation
const xTicks = useMemo(() => {
  const tickInterval = 10; // seconds for detail, 30 for overview
  const ticks: { x: number; label: string }[] = [];
  for (let t = 0; t <= duration; t += tickInterval) {
    const x = leftPadding + (t / duration) * width;
    ticks.push({ x, label: formatTime(t) });
  }
  return ticks;
}, [duration, width, leftPadding]);

// Y-axis ticks generation
const yTicks = useMemo(() => {
  const numTicks = 5; // 5 for detail, 3 for overview
  return Array.from({ length: numTicks }, (_, i) => {
    const ratio = i / (numTicks - 1);
    const value = min + (max - min) * ratio;
    const y = topPadding + (1 - ratio) * usableHeight;
    return { y, value: value.toFixed(1) };
  });
}, [min, max, topPadding, usableHeight]);
```

Benefits: - Professional scientific visualization - Easy to read exact values and timestamps - Better understanding of signal dynamics

Integration and Synchronization

All Controls Work Together

- Audio playback position updates all visualizations
- Clicking overview plot updates audio position
- Navigation buttons update audio position
- All changes are logged for debugging
- State is properly managed through React hooks

State Management

```
const [currentTime, setCurrentTime] = useState(0);
const [isPlaying, setIsPlaying] = useState(false);

// Synchronized through:
useEffect(() => {
  const audio = audioRef.current;
  // ... listeners for timeupdate, play, pause, ended
}, [audioUrl]);
```

CSS Styling

Added professional styling for new components:

```
/* Navigation controls */
.navigation-controls {
  display: flex;
  gap: 0.75rem;
  justify-content: center;
  padding: 1rem 0;
  margin: 0.5rem 0;
  border-bottom: 1px solid rgba(148, 163, 184, 0.2);
}

.nav-button {
  border: 1px solid rgba(148, 163, 184, 0.4);
  background: rgba(30, 41, 59, 0.9);
  color: inherit;
  border-radius: 0.5rem;
  padding: 0.6rem 1.2rem;
  cursor: pointer;
  transition: background 0.2s ease, border 0.2s ease;
  font-size: 0.9rem;
}

/* Overview and detail chart sections */
.overview-chart, .detail-chart-section {
  margin: 1.5rem 0;
  padding: 1rem;
  background: rgba(15, 23, 42, 0.4);
```

```
  border-radius: 0.5rem;
  border: 1px solid rgba(148, 163, 184, 0.2);
}
```

Testing

The implementation has been: 1. ✅ Successfully built with `npm run build` 2. ✅ Tested in development mode with `npm run dev` 3. ✅ Committed to git with descriptive commit message 4. ✅ Pushed to GitHub PR #15

Files Modified

1. **frontend/src/components/SignalPreview.tsx**
   - Fixed `describeArc` function (sweep flag)
   - Added `OverviewChart` component
   - Added navigation button handlers
   - Enhanced `SignalChart` with axes
   - Updated main component layout
2. **frontend/src/styles.css**
   - Added navigation controls styling
   - Added overview chart styling
   - Added detail chart section styling

Technical Specifications

Gauge

- Range: 1.0 to 6.5
- Arc: π to 2π (top semicircle)
- Needle: Red line with circle at data point
- Baseline: Orange marker line

Overview Plot

- Width: 920px (fixed)
- Height: 120px
- X-axis ticks: Every 30 seconds
- Y-axis ticks: 3 levels
- Current position: Red vertical bar (3px width, 70% opacity)

Detail Plot

- Width: Dynamic based on duration (80px per second, minimum 720px)

- Height: 220px
- X-axis ticks: Every 10 seconds
- Y-axis ticks: 5 levels
- Current position: Red dashed line with circle marker

Navigation Timing

- Skip forward/backward: 10 seconds
- Jump to beginning: 0 seconds
- Jump to end: total duration

Future Enhancements (Optional)

1. **Keyboard shortcuts** - Add hotkeys for navigation (Space = play/pause, Arrow keys = skip)
2. **Zoom controls** - Allow zooming in/out of detail view
3. **Configurable skip duration** - Let users choose skip amount (5s, 10s, 30s)
4. **Event markers** - Show detected events in overview plot
5. **Multi-channel support** - Display multiple signals simultaneously
6. **Export functionality** - Save annotated plots as images

Conclusion

All requested features have been successfully implemented and tested: - ✅ Gauge needle now visible and functioning correctly - ✅ Overview plot with full signal view - ✅ Clickable navigation in overview - ✅ Navigation buttons for precise control - ✅ Professional axes on all plots - ✅ All controls synchronized and working together

The PR #15 has been updated with these changes and is ready for review.