

Scene 2: Tapestry View - Implementation Prompt

Tapestry View - Implementation Prompt

TapestryView - Journey Path Visualization Implementation

Visualization Purpose

Create a flow visualization that shows how each attendee's journey through the survey questions creates unique paths, while revealing common patterns. This should emphasize that despite taking different paths, people often converge at similar destinations.

Pre-Implementation Investigation

Before coding, investigate:

1. Check if d3-sankey is available or if there's a preference for custom path drawing
2. Look for existing flow/path visualizations in the codebase
3. Find any bezier curve or path animation utilities
4. Examine color interpolation patterns used elsewhere
5. Check how other visualizations handle dense overlapping data

Core Concept

- Each person's survey responses create a path flowing left-to-right through all questions
- Path thickness represents the number of people taking that exact route
- Colors flow along paths based on years_at_medtronic or other attributes
- Paths merge and split, showing convergence and divergence of experiences
- The overall effect should be like looking at a woven fabric with distinct threads

Visual Design Specifications

Layout

- Horizontal flow from left to right

- Questions arranged as vertical columns
- Response options stacked vertically within each column
- Adequate spacing to prevent overcrowding
- Consider viewport with header space

Path Design

- Smooth bezier curves connecting responses
- Path width: 1px minimum, scaling up based on frequency
- Opacity: Higher for common paths, lower for unique ones
- Color: Gradient along path or solid based on attribute
- Anti-aliasing for smooth appearance

Visual Hierarchy

- Most common paths should be visually prominent
- Rare/unique paths visible but subdued
- Current focus path (on hover) should stand out clearly
- Background paths should create texture without noise

Key Features to Implement

1. Path Rendering System

- Calculate all unique paths through the data
- Aggregate identical paths to determine thickness
- Generate smooth curves between connection points
- Handle path bundling for cleaner visualization

2. Auto-Play Mode

- Sequentially highlight different journey patterns:
 - Start with most common path
 - Show paths by years_at_medtronic groups
 - Highlight surprising convergences
 - Trace individual unique journeys
- Smooth camera movements to focus on different sections
- Annotation callouts for interesting patterns

3. Interactive Mode

- Hover to highlight individual paths
- Click to isolate specific journey patterns
- Filter by any attribute to see subset of paths
- Ability to "follow" a path from start to end
- Show details of how many people took each path

4. Insight Display

- Dynamic statistics:
 - "Most common journey taken by X%"
 - "Y different paths lead to [same motivation]"
 - "New employees and veterans converge at [attribute]"
- Path diversity score
- Convergence points highlighted

Implementation Approach

Data Transformation

- Convert response data into path objects
- Each path needs: source points, destination points, count, attributes
- Pre-calculate path geometries for performance
- Group similar paths for aggregation

Rendering Strategy

Consider two approaches based on data density:

1. SVG paths for smaller datasets (better interactivity)
2. Canvas rendering for 600+ paths (better performance)
3. Hybrid: Canvas for background, SVG for interactive elements

Path Generation

- Use `d3.line()` or `d3.linkHorizontal()` for basic paths
- Custom bezier calculations for more control
- Consider path bundling algorithms for cleaner visualization
- Implement path smoothing to avoid sharp angles

Animation Patterns

Entry Animation

1. Questions fade in as vertical columns (0.5s)
2. Paths draw from left to right (2s total)
3. Thickness animates after paths complete (0.5s)
4. UI elements fade in (0.3s)

Auto-Play Sequences

- Highlight by frequency: fade all but top 10% paths
- Highlight by attribute: colorize paths by selected attribute
- Trace animation: follow random individual paths
- Convergence focus: zoom to where many paths merge
- Pattern reveal: show how different starts lead to same ends

Interactive Animations

- Hover: path lifts slightly and increases opacity
- Click: all other paths fade to 10% opacity
- Filter transition: smooth redraw of visible paths
- Zoom: maintain path smoothness during scale

Performance Considerations

Optimization Strategies

1. Path aggregation: combine identical paths into single elements
2. Level of detail: show fewer paths when zoomed out
3. Viewport culling: only render visible paths
4. Progressive rendering: draw in chunks for initial load
5. Use requestAnimationFrame for smooth animations

Data Structure Optimization

- Pre-calculate all path coordinates
- Store aggregated path counts
- Index paths by various attributes for quick filtering
- Consider WebGL (with regl) for massive datasets

Specific Interaction Patterns

Hover Behavior

- Highlight full path from start to end
- Show tooltip with journey details
- Temporarily increase path thickness
- Display how many share this exact path

Click Behavior

- Isolate clicked path and similar ones
- Show detailed breakdown of who took this path
- Option to "expand" to show individual variations
- Connect to other visualizations with same filter

Filter Controls

- Dropdown or toggle for each attribute
- Slider for "path frequency" threshold
- Option to show only unique paths
- Highlight convergence points

Edge Cases to Handle

1. Single-person paths (completely unique journeys)
2. Paths that skip questions (null responses)
3. Dense convergence points where many paths meet
4. Color accessibility for overlapping transparent paths
5. Mobile/touch interactions for precise path selection

Auto-Play Narrative Sequence

Design a compelling 2-3 minute sequence that tells the story:

1. "Everyone starts their journey..." (show all paths emerging)
2. "Taking different routes..." (highlight diversity)
3. "But notice how paths converge..." (zoom to convergence points)
4. "Veterans and newcomers often end up in the same place..." (color by years)
5. "Each thread in our tapestry is unique, yet woven together..." (full view)

Accessibility Considerations

- Keyboard navigation through major path groups

- Audio description mode for auto-play sequence
- High contrast mode that uses patterns instead of just color
- Path thickness minimums for visibility
- Alternative data table view available

Testing Priorities

1. Performance with 600 response dataset
2. Path rendering accuracy (no broken curves)
3. Interactive response time under 16ms
4. Memory usage during extended auto-play
5. Color contrast ratios for all path combinations
6. Touch gesture support for interactive displays

Key Questions for Implementation

1. Should paths be bundled at connection points or allowed to overlap?
2. How many simultaneous paths can be shown before it becomes noise?
3. Should the view support vertical scrolling for more response options?
4. How to handle real-time updates without disrupting the flow?
5. What's the best way to show percentage breakdowns at each stage?

Remember the Core Message

- Individual journeys that weave together into a collective tapestry
- Clear visualization of convergence despite different starting points
- Beautiful enough for passive viewing, informative enough for exploration
- The metaphor of weaving should be visually apparent