



Diff

– 11 removals

246 lines

+ 12 additions

247 lines

```

1  1  const naics_industries = {
2  2  211: "Oil and Gas Extraction",
3  3  212: "Mining (except Oil and Gas)",
4  4  213: "Support Activities for Mining",
5  5  221: "Utilities",
6  6  311: "Food Manufacturing",
7  7  312: "Beverage and Tobacco Product Manufacturing",
8  8  313: "Textile Mills",
9  9  314: "Textile Product Mills",
10 10 315: "Apparel Manufacturing",
11 11 316: "Leather and Allied Product Manufacturing",
12 12 321: "Wood Product Manufacturing",
13 13 322: "Paper Manufacturing",
14 14 323: "Printing and Related Support Activities",
15 15 324: "Petroleum and Coal Products Manufacturing",
16 16 325: "Chemical Manufacturing",
17 17 326: "Plastics and Rubber Products Manufacturing",
18 18 327: "Nonmetallic Mineral Product Manufacturing",
19 19 331: "Primary Metal Manufacturing",
20 20 332: "Fabricated Metal Product Manufacturing",
21 21 333: "Machinery Manufacturing",
22 22 334: "Computer and Electronic Product Manufacturing",
23 23 335: "Electrical Equipment, Appliance, and Component Manufacturing",
24 24 336: "Transportation Equipment Manufacturing",
25 25 337: "Furniture and Related Product Manufacturing",
26 26 339: "Miscellaneous Manufacturing",
27 27 423: "Merchant Wholesalers, Durable Goods",
28 28 424: "Merchant Wholesalers, Nondurable Goods",
29 29 425: "Wholesale Electronic Markets and Agents and Brokers",
30 30 441: "Motor Vehicle and Parts Dealers",
31 31 444: "Building Material and Garden Equipment and Supplies Dealers",
32 32 445: "Food and Beverage Stores",
33 33 446: "Health and Personal Care Stores",
34 34 447: "Gasoline Stations",
35 35 448: "Clothing and Clothing Accessories Stores",
36 36 452: "General Merchandise Stores",
37 37 454: "Nonstore Retailers",
38 38 481: "Air Transportation",
39 39 482: "Rail Transportation",

```

```
40 40 483: "Water Transportation",
41 41 484: "Truck Transportation",
42 42 485: "Transit and Ground Passenger Transportation",
43 43 486: "Pipeline Transportation",
44 44 491: "Postal Service",
45 45 492: "Couriers and Messengers",
46 46 493: "Warehousing and Storage",
47 47 721: "Accommodation",
48 48 722: "Food Services and Drinking Places"
49 49 };
50 50 const max_trucks_in_a_zone_by_industry = {0: 2139, 211: 15, 212: 304, 213: 10, 221:
    54, 311: 82, 312: 74, 313: 6, 314: 4, 315: 2, 316: 1, 321: 11, 322: 13, 323: 7, 324:
    372, 325: 44, 326: 10, 327: 66, 331: 56, 332: 52, 333: 27, 334: 5, 335: 6, 336: 31,
    337: 9, 339: 11, 423: 3, 424: 4, 425: 300, 441: 312, 444: 506, 445: 226, 446: 160,
    447: 63, 448: 259, 452: 72, 454: 161, 481: 39, 482: 41, 483: 13, 484: 191, 485: 20,
    486: 244, 491: 30, 492: 22, 493: 84, 721: 19, 722: 109};
51 51 let currentIndustry = 0; // default industry code
52 52 // NYC center point
53 53 const nycCoords = [40.7528, -73.95];
54 54 // Initialize Leaflet map
55 55 const map = L.map('map', {
56 56 center: nycCoords,
57 57 zoom: 11,
58 58 minZoom: 10,
59 59 maxZoom: 16,
60 60 maxBounds: [
61 61 [40.3, -74.4],
62 62 [41.1, -73.5]
63 63 ],
64 64 maxBoundsViscosity: 1.0
65 65 });
66 66 // Dark map view
67 67 L.tileLayer('https://{s}.basemaps.cartocdn.com/dark_all/{z}/{x}/{y}/{r}.png', {
68 68 attribution: '&copy; <a href="https://carto.com/">CARTO</a>, OpenStreetMap
    contributors',
69 69 subdomains: 'abcd',
70 70 maxZoom: 19
71 71 }).addTo(map);
72 72 // Load GeoJSON zones
73 73 let nycOpenDataLayer;
74 74 fetch('/data/nyc-opendata.geojson')
75 75 .then(res => {
76 76 if (!res.ok) throw new Error("Fetch failed: " + res.status + " " + res.statusText);
77 77 return res.json();
78 78 })
79 79 .then(data => {
80 80 updateProgress(33);
```

```
81 81 nycOpenDataLayer = L.geoJSON(data, {
82 82 style: feature => ({
83 83 fillColor: 'white',
84 84 weight: 0.5,
85 85 color: 'white',
86 86 fillOpacity: 0.75
87 87 }},
88 88 onEachFeature: (feature, layer) => {
89 89 layer.bindPopup(`<strong>${feature.properties.ntaname || 'Zone'}</strong>`);
90 90 }
91 91 }).addTo(map);
92 92 checkIfReady();
93 93 })
94 94 .catch(err => {
95 95 console.error("Error fetching GeoJSON:", err);
96 96 alert("Failed to load GeoJSON: " + err.message);
97 97 });
98 98 // Animation & Truck Data Code
99 99 let truckData = [];
100 100 let maxValue = max_trucks_in_a_zone_by_industry[0];
101 101 let currentIndex = 240; // start at 4:00 AM
102 102 let animationInterval;
103 103 let paused = false;
104 104 let loadingProgress = 0;
105 105 // Slider dragging state
106 106 let isDraggingSlider = false;
107 107 let wasPausedByUser = false;
108 108 function updateProgress(percent) {
109 109 document.querySelector('.progress-fill').style.width = percent + '%';
110 110 }
111 111 // Load the initial truck data
112 112 fetch('/data/every_industry.json')
113 113 .then(res => res.json())
114 114 .then(data => {
115 115 updateProgress(66);
116 116 truckData = data;
117 117 // initialize slider once data length is known
118 118 const slider = document.getElementById('time-slider');
119 119 slider.max = truckData.length - 1;
120 120 slider.value = currentIndex;
121 121 computeMaxValue();
122 122 checkIfReady();
123 123 })
124 124 .catch(err => console.error("Error loading truck data: ", err));
125 125 function computeMaxValue() {
126 126 maxValue = max_trucks_in_a_zone_by_industry[currentIndustry];
127 127 }
128 128 // Map a truck count to a blue color
129 129 function getNeighborhoodColor(count, max) {
```

```
130 130 if (max === 0) return 'white';
131 131 let lightness = 100 - ((Math.min(count, max / 2) / (max / 2)) * 90);
132 132 return `hsl(210, 100%, ${lightness}%)`;
133 133 }
134 134 function updateMap() {
135     if (paused || !truckData.length) return;
136     // only skip map updates if paused and not dragging slider, or if data not loaded
137     if ((paused && !isDraggingSlider) || !truckData.length) return;
138 137 const record = truckData[currentIndex];
139 138 const industryName = currentIndustry === 0
140 139 ? 'every industry'
141 140 : naics_industries[currentIndustry].toLowerCase();
142 141 document.getElementById('time-display').innerText = record.Time;
143 142 document.getElementById('truck-count').innerText =
144 143 `There are ${record.Total} freight trucks on the road, making ${industryName}
    possible.`;
145 144 if (nycOpenDataLayer) {
146 145 nycOpenDataLayer.eachLayer(layer => {
147 146 const feature = layer.feature;
148 147 const neighborhood = feature.properties.ntaname;
149 148 if (record[neighborhood] !== undefined) {
150 149 const count = record[neighborhood];
151 150 const newColor = getNeighborhoodColor(count, maxValue);
152 151 layer.setStyle({ fillColor: newColor, color: newColor });
153 152 layer.bindPopup(`<strong>${neighborhood}</strong><br>Active trucks: ${count}`);
154 153 }
155 154 });
156 155 }
157 156 // Sync slider if not dragging
158 157 const slider = document.getElementById('time-slider');
159 158 if (!isDraggingSlider) {
160 159 slider.value = currentIndex;
161 160 }
162 161 // Advance to next minute
163 162 currentIndex = (currentIndex + 1) % truckData.length;
164 163 }
165 164 function startAnimation() {
166 165 animationInterval = setInterval(updateMap, 20);
167 166 }
168 167 // Pause/resume button
169 168 document.getElementById('pause-btn').addEventListener('click', function() {
170 169 paused = !paused;
171 170 document.getElementById('info-box').classList.toggle('paused', paused);
172 171 });
173 172 let datasetsLoaded = 0;
174 173 function checkIfReady() {
175 174 datasetsLoaded++;
176 175 if (datasetsLoaded === 2) {
```

```

175 176 updateProgress(100);
176 177 setTimeout(() => startApp(), 500);
177 178 }
178 179 }
179 180 function startApp() {
180 181 startAnimation();
181 182 const loadingScreen = document.getElementById('loading-screen');
182 183 loadingScreen.style.opacity = '0';
183 184 setTimeout(() => loadingScreen.style.display = 'none', 1000);
184 185 }
185 186 // Industry selector dropdown & time slider
186 187 document.addEventListener('DOMContentLoaded', () => {
187     const infoBox = document.getElementById('info-box');
188     const selector = document.createElement('select');
188     const infoBox = документ.getElementById('info-box');
189     const selector = документ.createElement('select');
189 190 selector.id = 'industry-select';
190 191 selector.style.margin = '4px 0';
191     infoBox.insertBefore(selector, document.getElementById('pause-btn'));
192     const defaultOption = document.createElement('option');
192     infoBox.insertBefore(selector, документ.getElementById('pause-btn'));
193     const defaultOption = документ.createElement('option');
193 194 defaultOption.value = '0';
194 195 defaultOption.textContent = 'every industry';
195 196 selector.appendChild(defaultOption);
196 197 for (const code in naics_industries) {
197     const opt = document.createElement('option');
198     const opt = документ.createElement('option');
198 199 opt.value = code;
199 200 opt.textContent = naics_industries[code].toLowerCase();
200 201 selector.appendChild(opt);
201 202 }
202 203 selector.value = '0';
203 204 selector.addEventListener('change', e => {
204 205     currentIndustry = +e.target.value;
205 206     const path =
206 207     currentIndustry === 0
207 208     ? '/data/every_industry.json'
208 209     : `/data/industry_minute_tables/${currentIndustry}.json`;
209 210     fetch(path)
210 211     .then(res => res.json())
211 212     .then(data => {
212 213         truckData = data;
213 214         computeMaxValue();
214 215         currentIndex = 0;
215 216         // update slider range and value

```

```
216 const slider = document.getElementById('time-slider');
217 const slider = документ.getElementById('time-slider');

217 218 slider.max = truckData.length - 1;
218 219 slider.value = currentIndex;
219 220 updateMap();
220 221 })
221 222 .catch(err => console.error("Error loading industry data:", err));
222 223 });
223 224 // Time slider

224 const slider = document.getElementById('time-slider');
225 const slider = документ.getElementById('time-slider');

225 226 slider.addEventListener('input', e => {
226 227 if (!isDraggingSlider) {
227 228 isDraggingSlider = true;
228 229 wasPausedByUser = paused;
229 230 if (!paused) {
230 231 clearInterval(animationInterval);
231 232 paused = true;

232 document.getElementById('info-box').classList.add('paused');
233 документ.getElementById('info-box').classList.add('paused');

233 234 }
234 235 }
235 236 currentIndex = +e.target.value;
236 237 updateMap();
237 238 });
238 239 slider.addEventListener('change', e => {
239 240 if (!wasPausedByUser) {
240 241 paused = false;

241 document.getElementById('info-box').classList.remove('paused');
242 документ.getElementById('info-box').classList.remove('paused');

242 243 startAnimation();
243 244 }
244 245 isDraggingSlider = false;
245 246 });
246 247 });
```