Program Flow documentation

Produced using Gemini AI 2.0 Flash

Here's a breakdown of the program flow across the files you've provided, focusing on how they interact to create the Leaflet map with the Upper East Side historical data:

1. `index.html`

- **Purpose:** This is the main HTML file that sets up the web page structure.
- **Flow:**
  - Loads CSS:
    - `leaflet.css`: Leaflet's core stylesheet for map display.
    - `ues_script.css`: Custom styles for the page (e.g., container, map dimensions).
  - Sets up the HTML:
    - A `div` with the id "map" where the Leaflet map will be rendered.
    - A `div` with the id "ues70nyc" (which seems unused in the provided code).
  - Loads JavaScript:
    - `leaflet.js`: The Leaflet library itself.
    - `L.Polyline.SnakeAnim.js`: A Leaflet plugin to animate polylines (though it's not directly used in the main map logic here).
    - `ues70Park2.js`: Contains the GeoJSON data (`ues70nyc`) for the Upper East Side historical district.
    - `ues_script.js`: The main script that initializes the map and adds the GeoJSON data.

2. `ues_script.js`

- **Purpose:** This JavaScript file contains the core logic for creating the Leaflet map and displaying the historical data.
- **Flow:**
  - Initializes the Map:
    - `const uesMap = L.map('map');`: Creates a Leaflet map object, attaching it to the `div` with the id "map" in `index.html`.
  - Adds Tile Layer:
    - `L.tileLayer(...).addTo(uesMap);`: Adds an OpenStreetMap tile layer, which provides the base map imagery.
  - `addGeoJSONLayer` Function:
    - This function takes GeoJSON data and a map object as input.
    - `L.geoJSON(geojsonData, { ... }).addTo(map);`: Creates a Leaflet layer from the GeoJSON data.
    - `onEachFeature`: This option is used to:
      - Create popups for each feature (historical site) with information from the GeoJSON properties (name, address, style, URL).

        - `style`: This option is used to style MultiPolygon features (likely the district boundaries) with specific colors and fill.
    - Conditional Execution:
        - `if (typeof ues70nyc !== 'undefined') { ... }`: Checks if the `ues70nyc` data is loaded (it should be, from `ues70Park2.js`).
        - Calls `addGeoJSONLayer` to add the historical site data to the map.
        - `uesMap.setView([40.7751175, -73.9607865], 17);`: Sets the initial map view to the Upper East Side.
        - Optionally opens the popup for the first feature.

3. `ues70Park2.js`

- **Purpose:** This file contains the GeoJSON data (`ues70nyc`) representing the historical sites in the Upper East Side.
- **Data Structure:**
    - The data is a FeatureCollection of GeoJSON objects.
    - Each Feature has:
        - `properties`: Contains information about the site (name, address, architectural style, etc.).
        - `geometry`: Defines the location of the site (MultiPolygon coordinates).

4. `L.Polyline.SnakeAnim.js`

- **Purpose:** This file adds a "snakeIn" animation to Leaflet polylines. While included, it's not used in the core map display logic of `ues_script.js` in the provided code. If you were to draw routes, this would animate them.
- **Functionality:**
    - It extends Leaflet's `L.Polyline` class to add `snakeIn` functionality.
    - The `snakeIn` method animates the drawing of a polyline, making it appear to "snake" across the map.

5. CSS Files (`ues.css`, `ues_script.css`)

- **Purpose:** These files provide styling for the map and page elements.
- `ues.css`: Seems to contain some general Leaflet styling, possibly overriding defaults.
- `ues_script.css`: Contains styles for the page layout (container, headings) and the map (`#map`).

6. `new.html`

- **Purpose:** This HTML file appears to be a separate page with information about the project and a link back to the main map. It also embeds a PDF.
- **Flow:**
    - Sets up a basic HTML page with navigation.
    - Includes CSS for styling a popup div.
    - Embeds the `UES district.pdf` document.

- Has JavaScript functions (`showPopup`, `hidePopup`) to control the display of a popup div.

7. **`UES district.pdf`**

- A PDF document, likely containing a map or other information about the Upper East Side Historic District.

Overall Program Flow

1. **Page Load (`index.html`):**
   - The browser loads `index.html`.
   - CSS files are loaded and applied.
   - JavaScript files are loaded.
2. **Map Initialization (`ues_script.js`):**
   - The Leaflet map is created.
   - The OpenStreetMap tile layer is added.
3. **Data Loading (`ues70Park2.js`):**
   - The `ues70nyc` GeoJSON data is loaded into memory.
4. **GeoJSON Layer Creation (`ues_script.js`):**
   - The `addGeoJSONLayer` function is called to create a Leaflet layer from the `ues70nyc` data.
   - Popups are created for each feature with data from the properties.
   - The map view is set to the Upper East Side.