**Walking Directions and Bicycle Layer**

**Goals of Contribution:**

The goal of my contribution to the campus map was to have a functioning walking directions feature and to add and style a bicycle rack layer. Originally, I wanted to be able to have directions from current location or chosen building origin to another building destination. The main audience for the directions would be incoming students overwhelmed by the size of campus. For the bicycle rack layer, I originally wanted to style the shape files for three different scales: small scale – point, midscale – bicycle icon, large scale – polygons representing the size of the bicycle rack.

**Workflow:**

Walking Directions:

* First, I researched existing walking directions platforms. Aaron pointed me in the correct direction by showing me an example of walking directions done by mapbox. The code was similar to what I wanted to be featured on the map, but it was still necessary for me to filter out much of the existing code

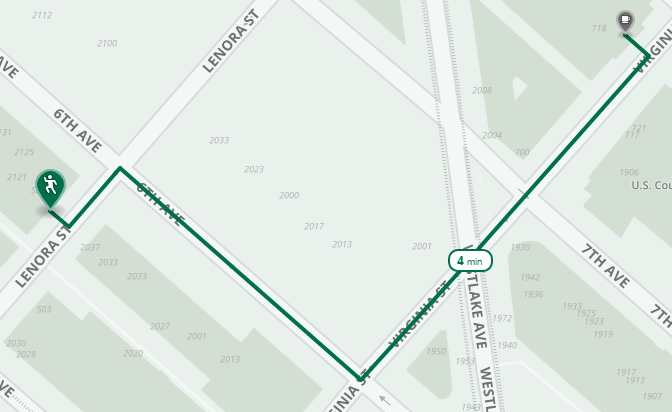


Figure 1: Original mapbox walking directions API: https://www.mapbox.com/bites/00087/

* In order to better understand the page source code of the walking directions API sent to me by Aaron, I went through the majority of JavaScript Code Academy.

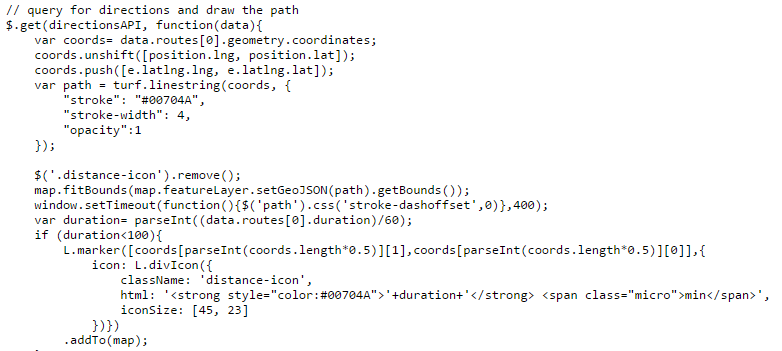


Figure 2: Raw code from Mapbox directions API page source

* With a better, but still amateur, understanding of JavaScript I was able to identify which lines of code I needed from the page source.



Figure 3: The code taken from the page source with coordinates and some styling.

* After having the walking directions code isolated, I could then add it to the jsfiddle Aaron created for the campus map. The challenge here came in understanding how the API works. I had to understand how to send the origin and destination coordinates to the API and then style to resulting geojson. I also needed to understand what the API was using as paths (highway:footpath in OSM).



Figure 4: The structure of the geojson returned by the mapbox directions API

* In order to better understand the parts of a geojson and how to style them, I consulted the geojson and leaflet documentation. Below are links that would be helpful for future work on styling and working with geojsons and the mapbox directions API.
  + <https://www.mapbox.com/bites/00087/>
    - Mapbox directions API example
  + <https://www.mapbox.com/developers/api/directions/>
    - Documentation for mapbox directions API
  + <http://geojson.org/>
    - Understanding geojsons generally
  + <http://leafletjs.com/examples/geojson.html>
    - Understanding how to style geojsons
  + <https://www.mapbox.com/mapbox.js/example/v1.0.0/mapbox-directions/>
    - Mapbox driving directions API with code
  + <https://jsfiddle.net/spatrick467/q0qdpdwp/>
    - My jsfiddle containing the directions API

Bicycle Layer:

* Research to see if data exists for Penn State’s bicycle racks. The data is available in OSM.
* Download the OSM extent map and convert the bicycle racks to shape files using OSMIQ (<http://osm-iq.de/de/OSM-IQ-EN>).

**Challenges**

Walking Directions:

The major challenges I faced were in trying to understand Javascript. This was expected – as I have never before used Javascript, Leaflet, or HTML. The mapbox directions API is rather new so there is little documentation available. The majority of my understanding of it came from fiddling and digging through endless forum threads. Also, geojsons are not the easiest file types to style. The origin and destination returned by the API must undergo some processing before being able to add a custom style.

Bicycle Layer:

Due to the walking directions, I was unable to commit time to styling the bicycle racks or add them to mapbox and the campus map.

**Improvement**

Walking Directions:

The directions feature still needs functionality for current location or chosen building of origin. Currently, it is hard coded to use Walker (of course) as the initial destination. Also, it would be useful to be able to drag origins or destinations as pinned points. Styling definitely needs to be done, especially for the pop up containing estimated time and distance. I only converted the units from meters and seconds to miles and minutes. The pop-up itself could be much nicer and it may be useful to ground check these estimates.

Bicycle Layer:

The bicycle rack shape files need to be added to mapbox and styled for different scales.