Module 5: Time Slider

One exciting thing we can do with modern interactive web maps is add a time component! Why is this exciting? Because! Adding “time” to traditional maps means plotting an entire new map for every change in time, which can fill up a piece of paper quickly! Lucky for us there are many ways we can add a time component to our maps. In this module we will explore the use of a time slider, which as luck would have it exists as a plug-in (or at least a copy-and-paste) for LeafLet (https://github.com/dwilhelm89/LeafletSlider)!

Since we are all now a bit more advanced in our skills, instead of just copy and pasting some code into the index.html file lets take a look below at what you will need, and then you can use your knowledge and skills to add them to the index.html file correctly!

You will need to add:

access to the Leaflet Stylesheet: <http://cdn.leafletjs.com/leaflet-0.7.2/leaflet.css>

access to the jquery-ui Stylesheet: <http://code.jquery.com/ui/1.9.2/themes/base/jquery-ui.css>

access to the Leaflet Library: <http://cdn.leafletjs.com/leaflet-0.7.2/leaflet.js>

access to the JQuery library: <http://code.jquery.com/jquery-1.9.1.min.js>

access to the Jquery-UI library: <http://code.jquery.com/ui/1.9.2/jquery-ui.js>

access to the scipts for mobile touch: <http://cdnjs.cloudflare.com/ajax/libs/jqueryui-touch-punch/0.2.2/jquery.ui.touch-punch.min.js>

access to the local TimeSlider code: SliderControl.js

access to your local javascript code: javascript.js

Great! Now that you know what you need, add these components to your index.html file! Of course while you have your html open, you are going to want to add a new div for your soon-to-be map!

Before we go any further, we need to crack open our data set (points.json) to get an understanding of what is taking place. You will notice that this is a json data file which contains some features, and with each feature we get a property called “time”, and we also get some some geometry. The “time” property is what we will be using to run the slider, and of course the geometry we need to plot the points!

Ok, now lets crack open our javascript (note, our file is called javascript.js) file and get to work. The first thing we need to do is create a new variable for our time slider:

var sliderControl = null;

Next we need to go ahead and set up our map (note that depending on what you named your “map” div, you may need to edit this!):

var myMap = L.map('map').setView([52.06, 7.40], 10);

L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png', {

attribution: '&copy; <a href="http://osm.org/copyright">OpenStreetMap</a> contributors'

}).addTo(myMap);

With our map taken care of it is time to get down to business. Lets point to our data set and pass it into a new function:

$.getJSON("points.json", function(json) {

Then we can read it in:

var testlayer = L.geoJson(json),

Now we will set up place for our time slider to live in the top right side of our map:

sliderControl = L.control.sliderControl({

position: "topright",

layer: testlayer

});

The code above uses the built in function “L.control.sliderControl” to create the time slider which needs a location , in our case the top right, and it also needs the layer of data for which it will execute upon. The default settings of this function will recognize the “time” property in the data, so we do not need to define it here.

Great! So we added our data and created a slider, whats next? You guessed it! We have to add it to the map!

myMap.addControl(sliderControl);

And with that we can initialize our time slider (so it is active o page load) and close up the function!

sliderControl.startSlider();

});

Thats it! Fire up your new map and check it out!