**Lab 5**

*This exercise uses OpenLayers 5.3* to introduce very basic interactive map design with the OpenLayers API. Subsequent exercises will engage with importing and using modules to conduct more complex functions and features. Although OpenLayers 6 was released at the end of 2019, you should be able to find plenty of working examples and online examples for this slightly-out-of-date version, and the two releases are highly compatible.

Given the recent news that classes will be canceled next week and moved to an online-only format for the remainder of the semester, this lab has been cut down substantially so that you can finish it within one class session and regroup.

**Stylize Layers**

Include the OpenLayers stylesheet in the <head> section of your HTML page

**<link** rel="stylesheet" href="https://cdn.rawgit.com/openlayers/openlayers.github.io/master/en/v5.3.0/css/ol.css" type="text/css"**>**

Include the OpenLayers JavaScript library at the end of the <body> section of your HTML page

**<script** src="https://cdn.rawgit.com/openlayers/openlayers.github.io/master/en/v5.3.0/build/ol.js"**></script>**

Put a <div id="map"> element in your code

**<div** id="map" style="width: 600px; height: 400px;"**></div>**

Now you can add a <script> section at the end of the <body> section (after the <script> that loads OpenLayers JavaScript library). All the following JavaScript code needs to go in that new <script> section.

Initialize the map with an OpenStreetMap baselayer. Note that if you borrow code from the Leaflet Extras page that you have to remove the s from the URL {s}.tile.openstreetmap.org/{z}/{y}/{x}.png. See below for the code:

**var** attribution = **new** ol.control.Attribution({

collapsible: **false**

});

**var** map = **new** ol.Map({

controls: ol.control.defaults({attribution: **false**}).extend([attribution]),

layers: [

**new** ol.layer.Tile({

source: **new** ol.source.OSM({

url: 'https://tile.openstreetmap.org/{z}/{x}/{y}.png,

attributions: [ ol.source.OSM.ATTRIBUTION, 'Put your name here' ],

maxZoom: 19

})

})

],

target: 'map',

view: **new** ol.View({

center: ol.proj.fromLonLat([-96.6607, 40.8073]),

maxZoom: 18,

zoom: 12

})

});

The map will be displayed in <div id="map"> element and will be centered on Lincoln (Latitude = 40.8073, Longitude = -96.6607) at zoom level 12.

You can add a marker at a specific location:

**var** layer = **new** ol.layer.Vector({

source: **new** ol.source.Vector({

features: [

**new** ol.Feature({

geometry: **new** ol.geom.Point(ol.proj.fromLonLat([-96.6607, 40.8073]))

})

]

})

});

map.addLayer(layer);

You can create a new <div id="popup"> element right after the <div id="map"> element

**<div** id="popup" class="ol-popup"**>**

**<a** href="#" id="popup-closer" class="ol-popup-closer"**></a>**

**<div** id="popup-content"**></div>**

**</div>**

Initialize the popup (the following JavaScript code needs to go in the <script> section)

**var** container = document.getElementById('popup');

**var** content = document.getElementById('popup-content');

**var** closer = document.getElementById('popup-closer');

**var** overlay = **new** ol.Overlay({

element: container,

autoPan: **true**,

autoPanAnimation: {

duration: 250

}

});

map.addOverlay(overlay);

closer.onclick = **function**() {

overlay.setPosition(**undefined**);

closer.blur();

**return** **false**;

};

Add the function to open the popup when you click on the marker,

map.on('singleclick', **function** (event) {

**if** (map.hasFeatureAtPixel(event.pixel) === **true**) {

**var** coordinate = event.coordinate;

content.innerHTML = '<b>Hello world!</b><br />I am a popup.';

overlay.setPosition(coordinate);

} **else** {

overlay.setPosition(**undefined**);

closer.blur();

}

});

By default, the popup will open if you click on the marker. If you want the popup to open automatically when the map is loaded, you can add

content.innerHTML = '<b>Hello world!</b><br />I am a popup.';

overlay.setPosition(ol.proj.fromLonLat([4.35247, 50.84673]));

Final steps and submitting your assignment:

* Zoom in by at least one step (10 pts)
* Move the circle to sit atop any landmark in Lincoln (10 pts)
* Change the pop-up text to identify the landmark (10 pts)
* Make the map at least 800 pixels wide by 600 pixels high (10 pts)

If you have any trouble as you compile your code, before making the final changes described above, confirm that your code is arranged correctly by comparing it to my example here: <http://rshepard2.github.io/lab7/>