**Geography 572**

**Lab #3: Advanced CartoCSS Styling**

**Lab Objectives:**

* Create an aesthetic tileset based off of the pastiche lecture material
* Serve your own set of tiles using Mapbox Studio and CartoCSS
* Continue to build your knowledge of Illustrator, Photoshop, and HTML5/CSS

**Evaluation:**

This lab is worth **40 points** toward the Lab Assignments evaluation item, which is worth 25% of your overall course grade. A grading rubric is provided at the end of the lab to inform your work.

**Schedule of Deliverables:**

* **October 30th:** Lab #3 Assigned //client contract begins
* **November 6th:** Inspiration Board Due (2pts) //design & feedback from client
* **November 13th:** POIs list Due (6pts) //input & feedback from client
* **November 20th:** Lab #3 Due (32pts) //contract deadline

**Challenge Description**

The director of a local art museum wants to incorporate artistic principles into a tourism slippy map they’ll display on their website. They want to promote the City’s natural, cultural, historical, and/or economic points of interest, or POIs(i.e., places that tourists may wish to visit during their stay). Since the museum is renowned for its forward thinking, the map should incorporate an aesthetic time period or movement in cartography. The Director has made clear that a trained cartographer must be hired to design a map tile set that is innovative and aesthetically pleasing.

CartoCSS is a styling language specific to mapping that uses existing CSS principles and enhances them to be used for designing vector or raster tile sets online. Professional cartographers have found a way of styling web tiles via the CartoCSS framework to push the limits of what people expect from slippy maps. These maps are highly designed and are innovating the possibilities of the field.

**Notes from the Director of the Chazen**

Your inspiration board serves as a deliverable, to show that you understand how to interpret advanced map stylings that fit with a particular aesthetic look. Your tileset will be built to incorporate all zoom levels, from a view of the world (zoom 1) to street level view (zoom 15) and must be designed for all appropriate scenarios.

The tilesets will be embedded in a webpage, with basic HTML/CSS and Javascript provided to guide you with basic slippy map interaction. Your tilesets must include **at least** 30 POIs icons on the map (i.e., point markers with a basic description). POIs can include a variety of locations, like restaurants, parks, etc. that might be useful in this scenario. You may choose any city for your scenario, as long as you follow the basic requirements for the POIs. It is important to keep in mind that the contract includes ***both*** the design of the tileset at every zoom level AND the insertion of this map with markers denoting POIs. Finally, you must make use of Mapbox Studio for Lab #3, an open source product for serving custom vector tiles provided by the company MapBox that is introduced in the lab.

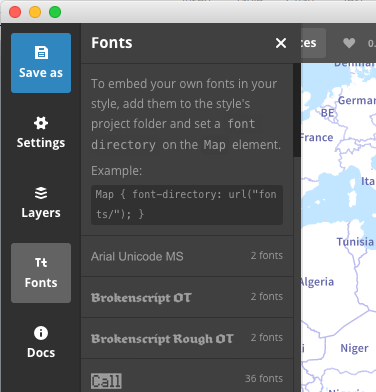
**1. Creating an Inspiration Board**

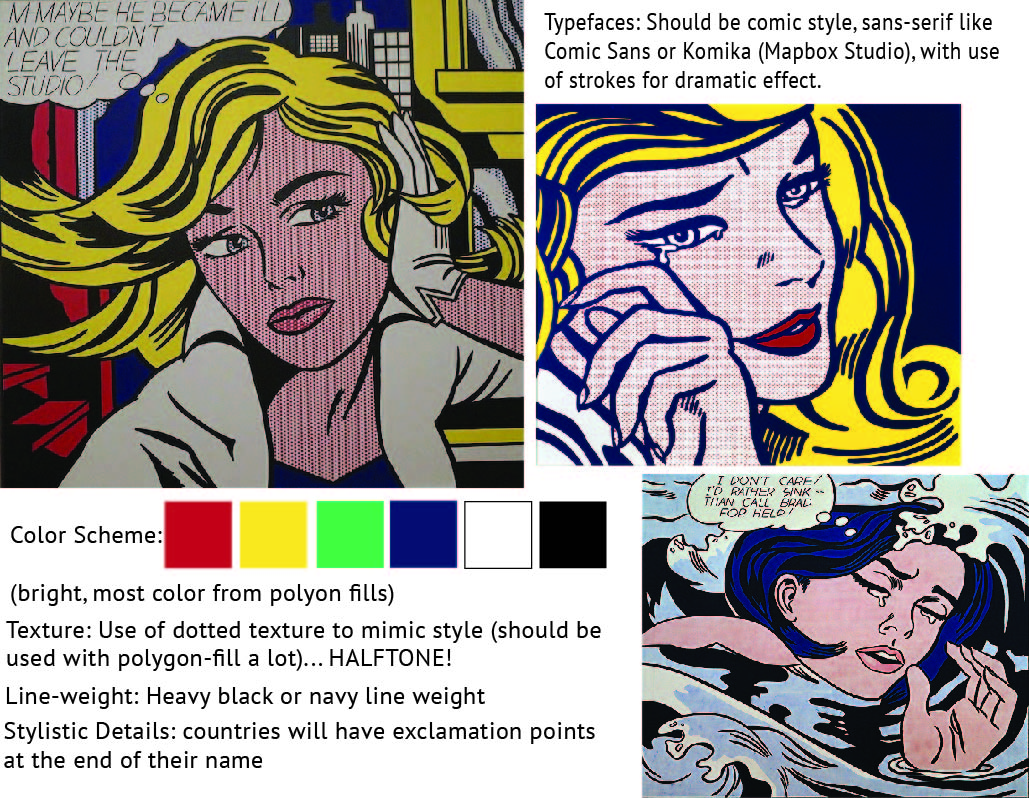
The first step in Lab #3 is to demonstrate the inspiration for your tileset, based off of two scenarios. Scenario One is incorporating one of the aesthetic styles of mapping as described in Lecture X, and providing appropriate examples from maps and art that reference specific details you plan on including in the map that you find appropriate for the map style. Scenario Two is using a specific map or media (such as a painting or video game) to determine the tileset style. Scenario One will be the example for this lab, creating a ‘Pop Art’ tileset based off of the work of Roy Lichtenstein, a famous 1960’s artist who is famous for his comic-like paintings of people.

Some visual variables to consider for your inspiration board include:

1. **Color:** What colors are you going to use? Make sure they are either relevant to the aesthetic style you are mimicking or included in your sample images.
2. **Texture:** Are there specific textures common to the style?
3. **Typefaces:** Mapbox Studio provides over 300 fonts, but you can (and definitely should) included your own to be used, which typeface best fit your map?
4. **Stylistic effects:** Are waterlines or certain effects like drop shadow common? Should your design be minimalistic? Should your line weight be heavy or light? Colors dark or transparent?

Your inspiration board should have sufficient images to describe what you will attempt to accomplish in your tileset and your artistic/cartographic vision. You should have at least 5 various specific pieces of inspiration for your tileset described in short sentences to guide you through the tileset design process. The sky is the limit, and if you have questions- look at the example gallery or ask your TA/Professor if you have questions. You are required to have a PDF of the inspiration board assembled and ready for mapping by **November 6th**.

To examine different fonts included in Mapbox Studio, click on the ‘Fonts’ button on the sidebar of Studio when open:

**Table 1**. **An example inspiration board, based off a Roy Lichtenstein painting.**

**2. Assembling Your POIs**

The next step in Lab #3 is assembling some data for your places of interest to be added to your map. The information set for Lab #3 emphasizes features at the point dimensionality. While you can include linear or areal features in your map (e.g., a scenic drive, a park), you will need to represent them as points for Lab #3; while this may seem like a limitation, the conceptualization of features as points is common on web maps due to the provision of integrated map designs at multiple scales, the smallest of which require collapsing to points.

It is up to you to decide which features of interest will be included in your map; if you choose a large city, consider giving your map a specific theme rather than including all possible POIs that may be found on a tourism map. Regardless of which features you ultimately decide to map, you will need to collect five pieces of information about each feature:

1. **Category:** The category representing the type of POI (i.e., the iconic point symbol used to represent the POI); this is not the higher-level category further organizing groups of POIs mentioned above.
2. **Latitude:** In decimal degrees, using negative numbers for latitudes in the Southern Hemisphere. There are many places to get the coordinates of places, whether through Google Maps or a free geo-referencing service.
3. **Longitude:** In decimal degrees, using negative numbers for longitudes in the Western Hemisphere.
4. **Name:** The specific name of the instance of the feature. This is different from the category, as several mapped features may receive the same iconic point symbol.
5. **Description:** A brief (100 words or less) description of the feature, perhaps with a link to additional information. You should compose the descriptions yourself such that the set of descriptions are at a common length and contain a common set of summary information. Both the name and description columns will be used to populate an information window upon clicking the icon.

Your information must be stored in a *comma separated file*, or ***.csv***, for loading into Mapbox Studio. As its name implies, a *.csv* file is a flat text file that delimits rows using the comma (“,”) character. To create your *.csv* file, you can assemble your spreadsheet in Microsoft Excel (as shown in **Table 1**) and export using *Save As->Save As Type: CSV.* You are required to have your *.csv* file assembled and ready for mapping by **November 13th**.



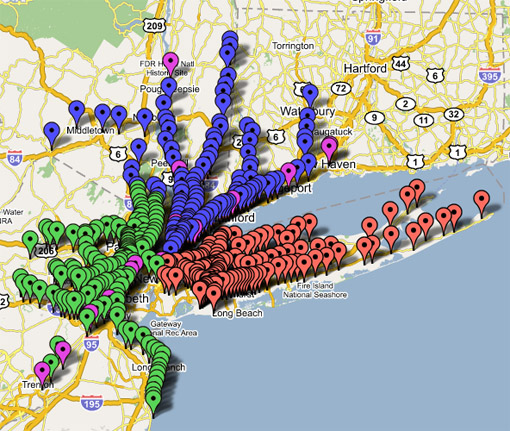
**Table 1** **shows example POI information for Madison, WI**

**3. Intro to Mapbox Studio**

Cartography has forever been altered with the influx of mapping on the web, using interactivity to animate and create more functional maps than ever before. Popularized by Google Maps and Mapquest for their navigation-oriented basemaps, the term ***slippy map*** describes a mashup that draws upon web mapping services to provide integrated, multiscale map designs, typically using a tile-based raster system.

A slippy map also exhibits basic interactive functionality, such as, panning, zooming (together where the ‘slippy’ term came from), overlay of different basemaps, and retrieval of details through information windows (e.g., your point icons). The advantage of slippy map mashups is that their creation requires very little custom code by combining existing services.

While originally used as reference maps for Google Maps or Mapquest, slippy maps have undergone a revolution with CartoCSS. As original slippy maps were considered poorly designed and under-developed, the cartographic world has worked to alter the stigma around slippy maps. Companies like Mapbox incorporate CartoCSS into their business model, creating a platform to make custom map tiles that are best for what a cartographer is creating for a client. These range from basic transportation maps, to minimalist designs, to the highly designed (the latter of which is the concentration of this lab).



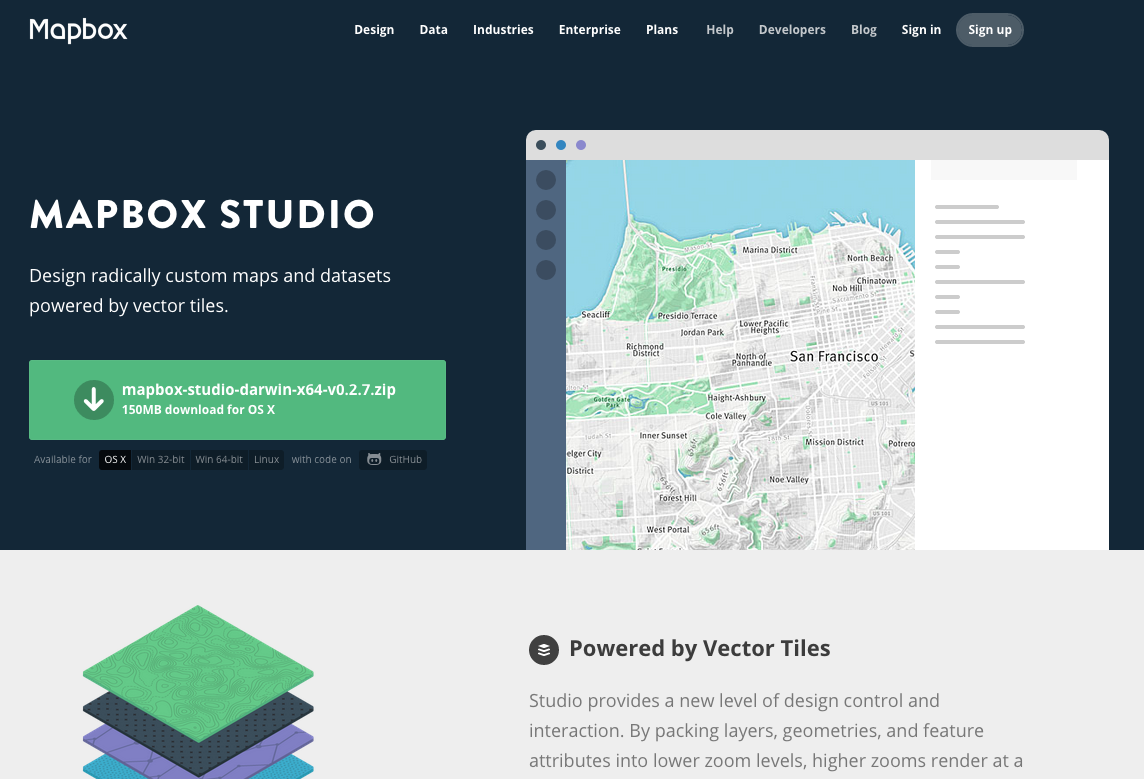
**Figure 8:** The transformation of tilesets, from the original Google Maps ‘Pushpin’ raster tileset to the highly designed Mapbox Studio vector tileset

**a. Getting Started with Mapbox Studio:**

Despite initial cartographic frustrations with the limitations in designing map tiles, the ability to customize the cartographic display has improved substantially in recent years through expansion and refinement of tools for styling and rendering your own basemap tiles.

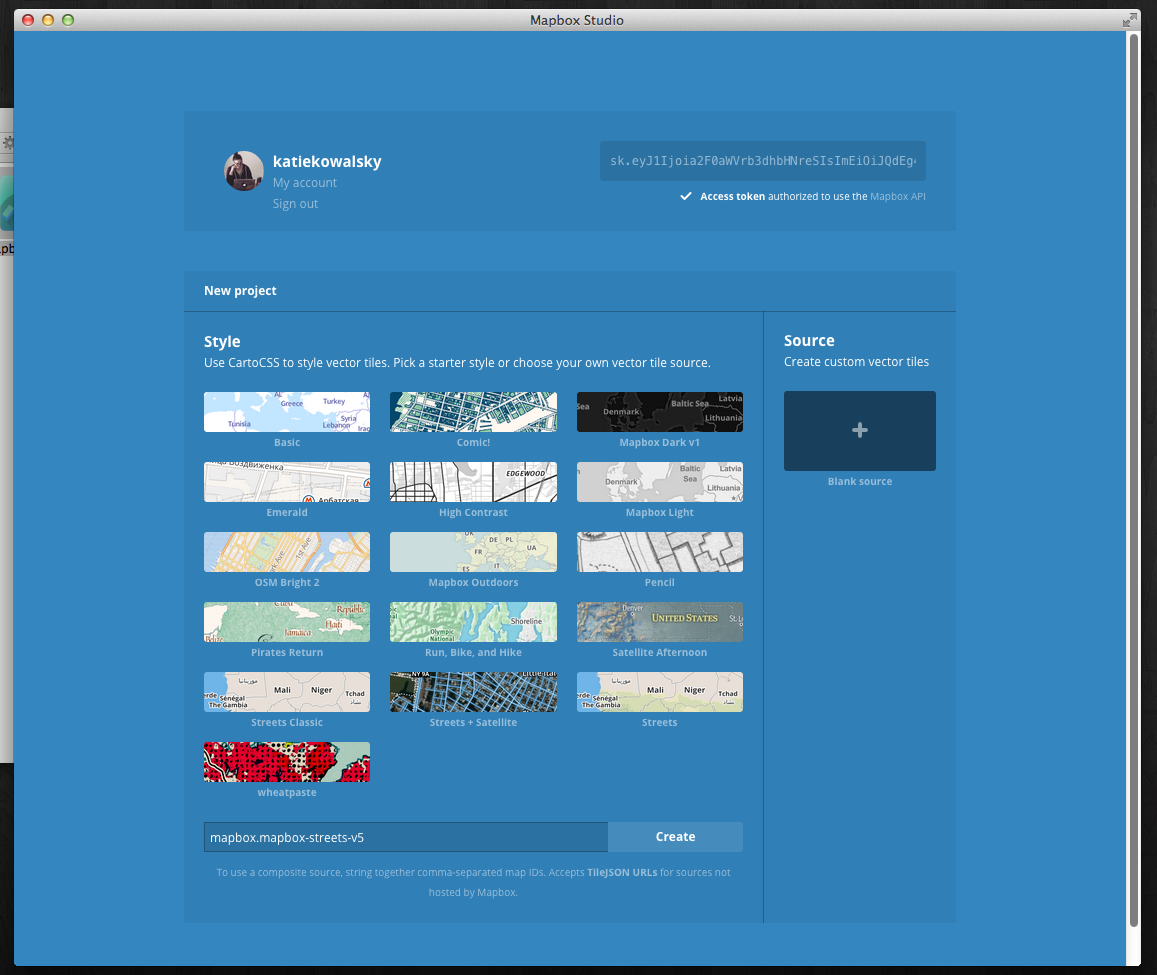
Your Lab #3 slippy map mashup will make use of Mapbox Studio*,* an application developed by a company called Mapbox that supports custom styling and rendering of vector basemap tiles. MapBox is quickly becoming a leader in the geospatial industry, particularly in the area of web mapping. Mapbox Studio allows you load and style your own datasets, such as your set of POIs, or to draw from existing data services, such as OpenStreetMap or Natural Earth.

It is important to note that Mapbox Studio can be used for free, using their educational pricing that allows you a basic plan for free with a .edu email address: <https://www.mapbox.com/education/>. If you exceed the limit, which for the purposes of Lab #3 challenge you definitely will not, you would need to pay. Finally, we will be serving raster tiles for Lab #3, given the commonality of this solution. MapBox has been highly innovative in the development of vector-based tiling, a promising solution for responsive web cartography in the future.

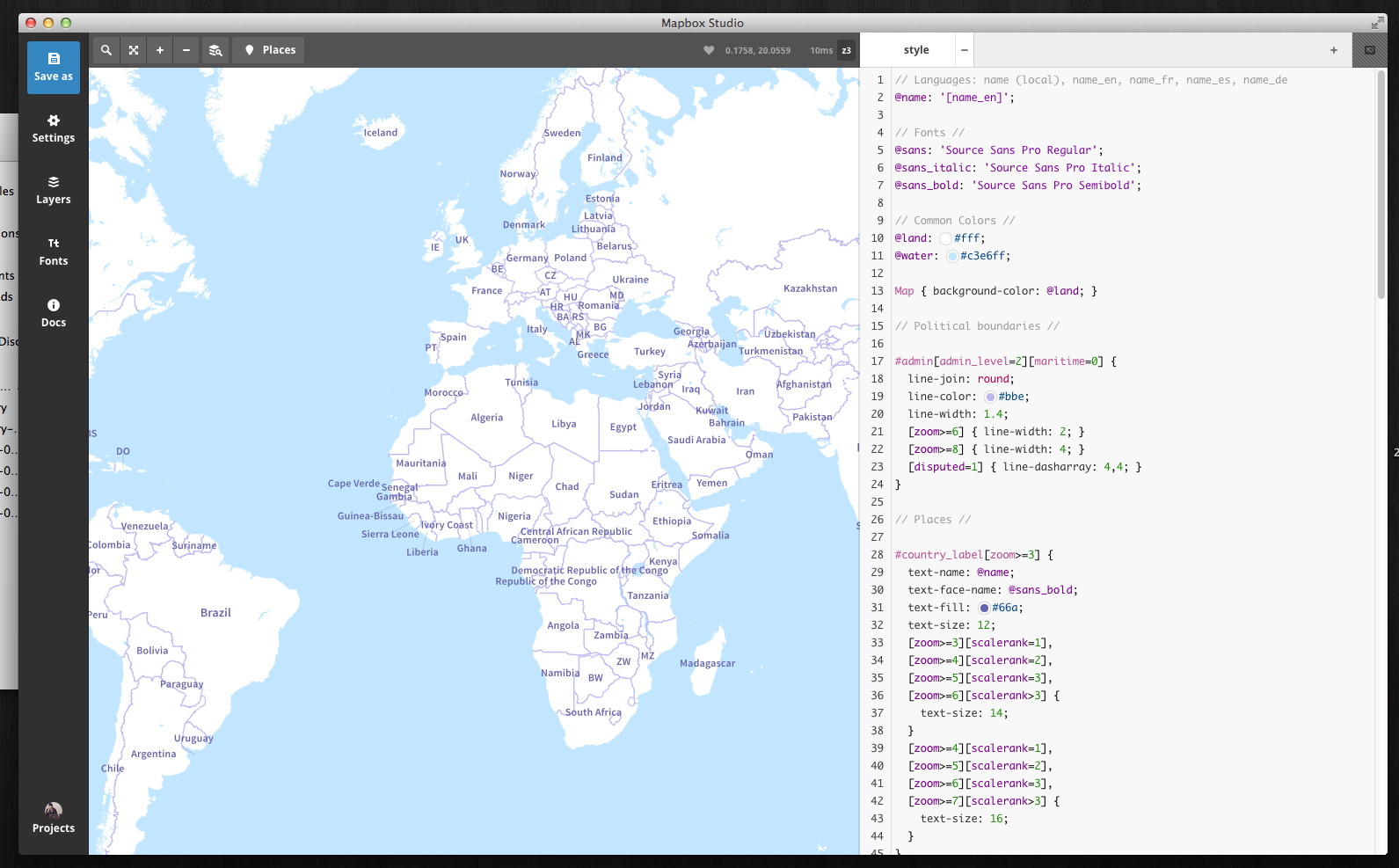


**Figure 9:** The Mapbox Website for Mapbox Studio

Get started with Mapbox Studio after registering for the Education account (you should be all set a minute or two after sending, but double check your SPAM folder) and go to the Mapbox website for Mapbox Studio: [https://www.mapbox.com/mapbox-studio/](https://www.mapbox.com/mapbox-studio/#darwin) (**Figure 9**). If you prefer to work on your own machine, you then can install Mapbox Studio by selecting the appropriate operating system from the download options. Studio is also available on all machines in Science Hall 380 and M376. If you are working on a Science Hall machine, open Mapbox Studio to begin (**Figure 10**). It’s important that you create an account and login; you can save everything to Mapbox’s cloud and can be accessed on any computer.



**Figure 10: The Mapbox Studio Application.**



**Settings**

**Layers**

**Fonts**

**Docs**

**Projects**

**Save as**

**Figure 11: The Mapbox Studio Interface**

Once open, click on a sample template to get a feel of where things are in Mapbox Studio. Under the ‘Docs’ tab, there is an **Interface Tour that is highly recommended** to get a sense of where things are in the application.

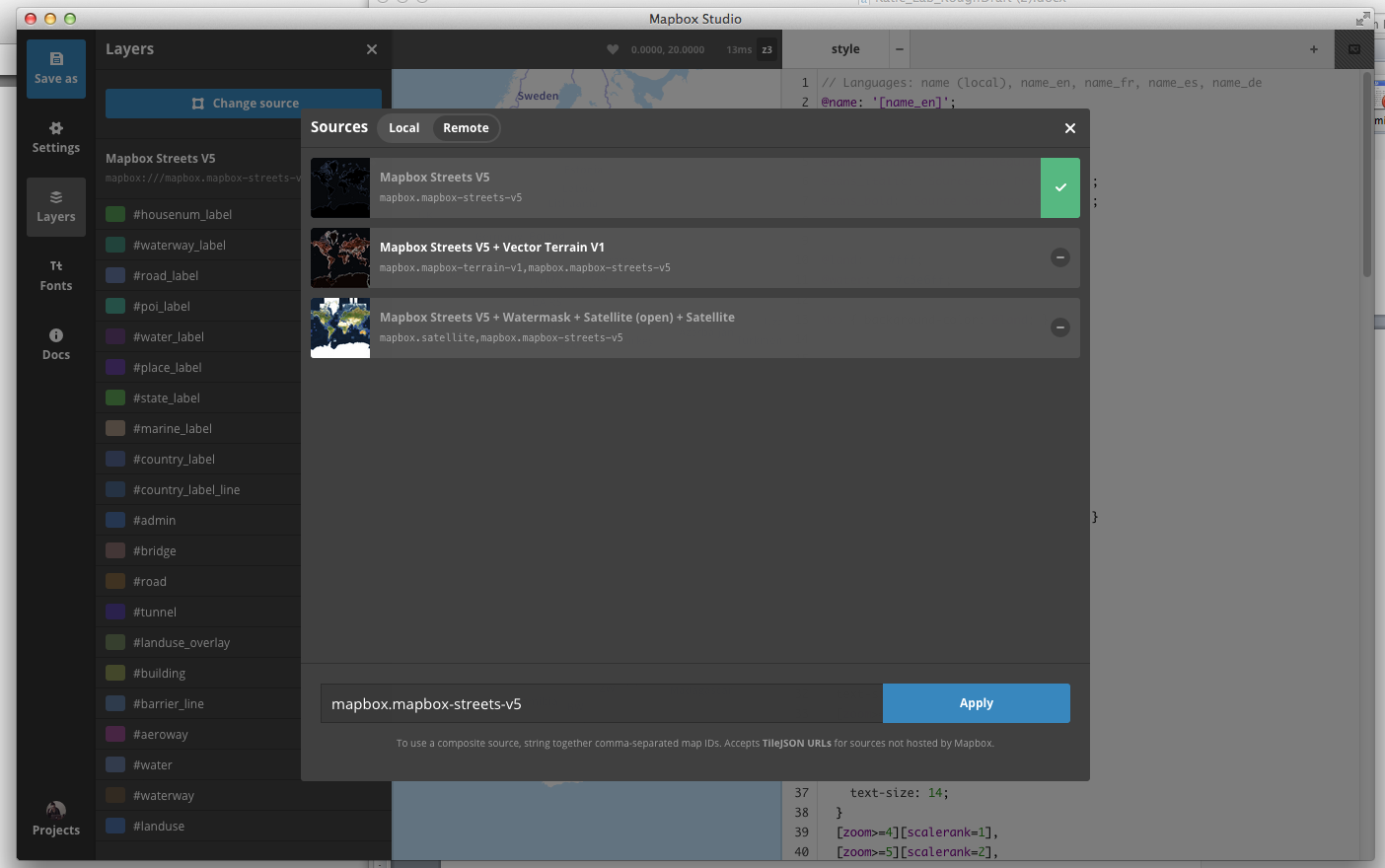
The application includes five tabs along the left bar (**Figure 11**):

* The ***Save As*** allows you to constantly save any changes made to your tileset (make sure you choose either the Desktop or your flash drive if working in 380).
* The ***Settings***tab allows you to upload your style to Mapbox when finished to be viewed/hosted on their website, to name and describe it, and various settings like altering the center position, changing the min/max zoom and so on.
* The ***Layer*** tab is extremely important for CartoCSS, allowing you to add/change sources and sees the layer order for your source. You can also click on each source layer to see the data properties (that can be used as selectors in CartoCSS)
* The ***Fonts*** tab shows the variety of typefaces available and built into Mapbox Studio. Clicking on each will allow you to see each and every font possible. You can also see the documentation for adding your own fonts as well.
* The ***Docs***tab gives you access to all of the acceptable CartoCSS documentation and has the interface tour, which is highly recommended that you use.
* The ***Projects*** tab is where you can create a new source or style project, log in and out and browse existing projects you might have. *Be sure to logout of your MapBox account at the end of your work session*.

**b. Style vs. Source in Mapbox Studio**

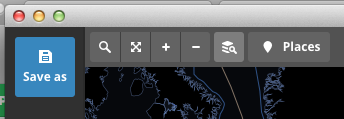
When you open Mapbox Studio, you’ll find two options: Style or Source. These two options are kept separate in Studio, a division not implemented in Mapbox’s earlier product: Tilemill. As you can see, the various style examples that come built in Studio (**Figure 10**) are on the Style side. That part of the app is where we’ll do all of our CartoCSS styling.

So what’s Source for? Mapbox has its tiles, Mapbox-Streets that is going to be used in most cases for this lab. However, say you need to want to use bathymetry data from Natural Earth and Mapbox doesn’t have it included in their own vector tile source files. You can easily import any geographic file (Shapefile, KML, geojson, CSV, etc.) and upload it to be used in your tileset. However, Mapbox provides three different in-house datasets you’ll probably use: Mapbox-Streets, Mapbox-Terrain, and Mapbox-Satellite. You can also combine multiple data layers under the Layer Tab by pressing ‘Change Source’ and stringing the names together with commas.

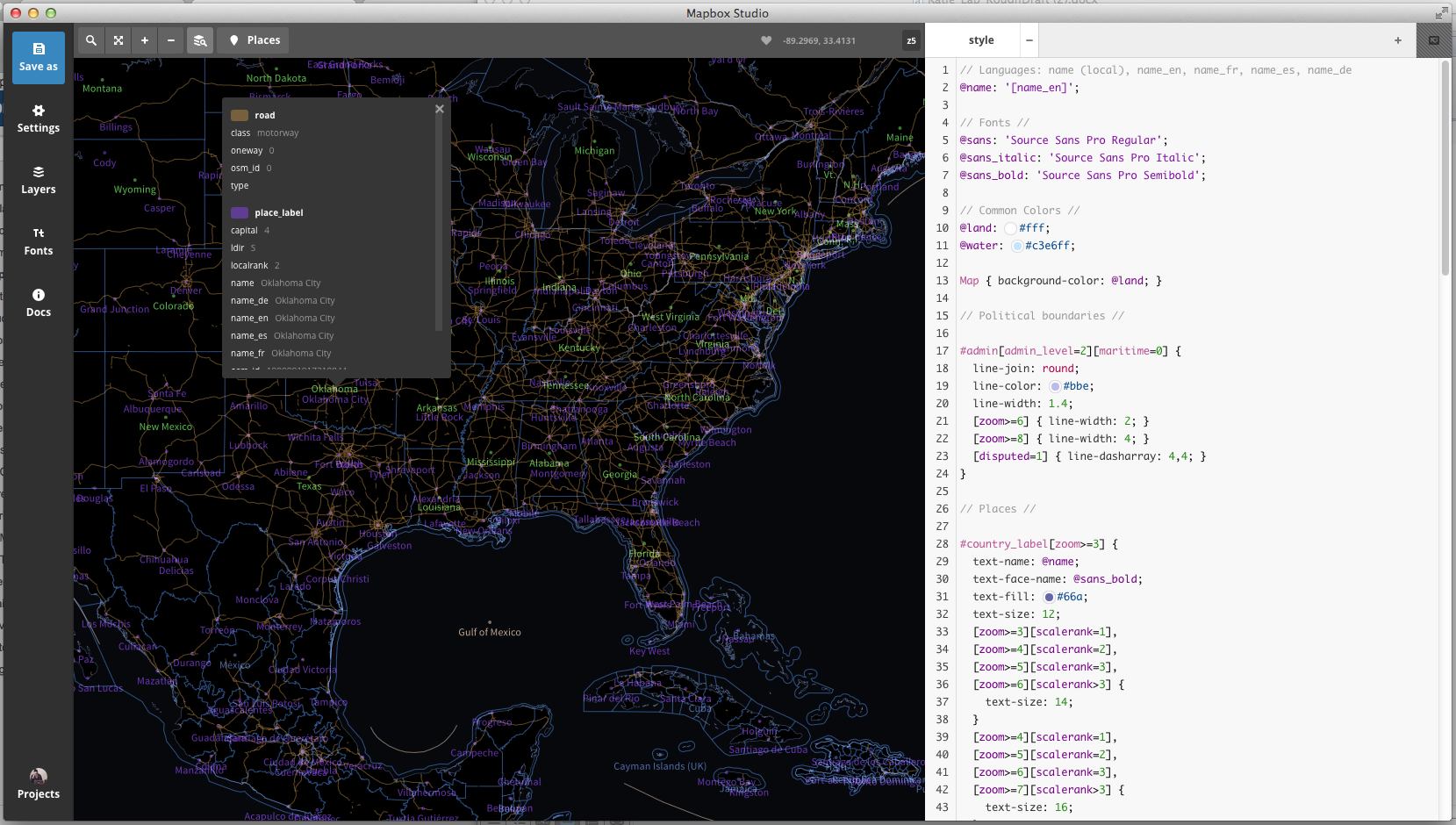


**Figure 12: Mapbox Remote Data Sources**

Why is this important? The layers tab shows what layers you have available, and what attributes you can use for selectors in CartoCSS. Sources control what you have available. Another way to view what the sources look like under the surface of your tileset is by pressing the layers view next to zoom in/out and the Places marker on the toolbar for Studio (**Figure 13**)

****

**Figure 13: Layer View**

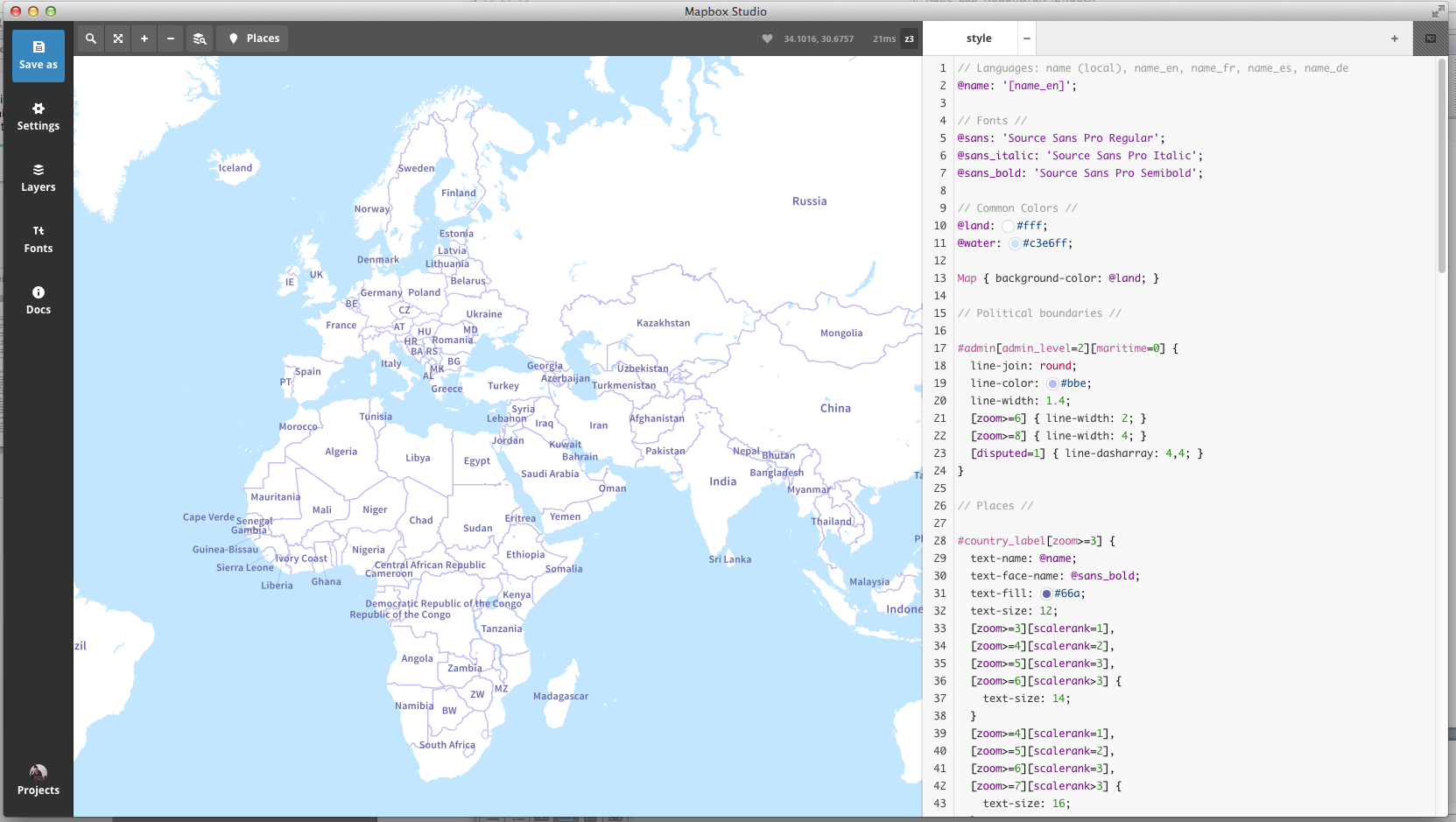
****

**Figure 14: Layer View with attributes shown**

The layer view allows you to see all possible vector tile layers. Clicking on an area (**Figure 14**) allows you to see all the layers ‘on’ that certain area and what attributes will be shown. This feature is a bit clumsy, but necessary for understanding certain features of Mapbox-Streets, Terrain, and Satellite. Using this can help you understand what values you’ll use in selecting statements for CartoCSS.

**c. Styling with CartoCSS**

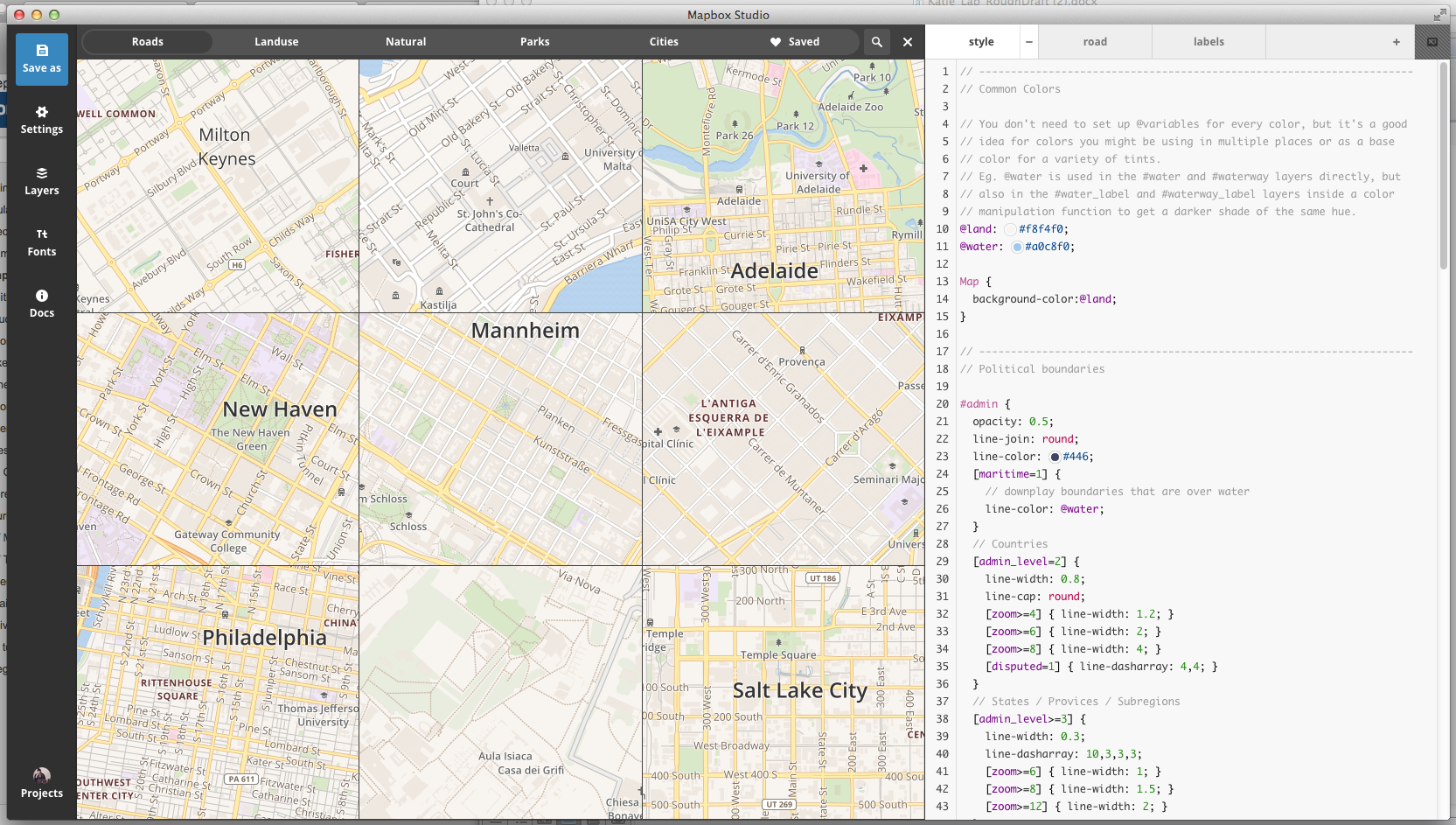
The best way to start styling a tileset is to choose the ‘basic’ tileset example and open that up. Take a look at the way CartoCSS operates by changing some of the functions and selectors.



**Style Editor**

**Figure 15: Mapbox Studio Style Editor**

The Style Editor is the heart of Mapbox Studio. As you can see, there’s already some CartoCSS styling put into the style sheet, which is why you can see all of the various country names, outlines, water, etc. You can also examine everything at various zoom levels by using the zoom in/out buttons or the Places button. The Places view allows you to see various areas that are ‘tagged’ by Mapbox Studio to see how your styles affect certain areas.



**Figure 16: Places View in Mapbox Studio**

Mapbox Studio supports the styling of basemap tiles through their ***CartoCSS*** styling specification. In Lab #1, you learned about CSS and the components of style rules (e.g., selectors, properties, values, ids, classes, etc.); return to the second part of Lab #1 if you are not yet comfortable with these concepts. The CartoCSS specification leverages the CSS syntax to allow you to style map features, rather than page elements. Hopefully you now see the value of Mapbox Studio: it makes map design much more like web design. The skills learned to build your web portfolio in Lab #1 are now directly applicable to making elegant and aesthetically pleasing tilesets!

Like CSS, CartoCSS apply style rules to map layers by referencing unique IDs (e.g., #waterway\_label). You can see in **Figure 15** that default style rules were added to the Map itself, for which there are a small subset of styling rules, as well as each uploaded layer using the unique id you gave it during upload. CartoCSS then is organized into ten different groups of styles based on map features; each group is described as a *symbolizer*, a term derived from the [Mapnik](http://mapnik.org/) rendering engine atop which Mapbox Studio is built. Symbolizers include:

* ***Line***: styles for line features and the strokes of polygons;
* ***Polygon***: styles for the fill of polygons;
* ***Point***: styles for point features;
* ***Text***: styles for labeling point, lines, and polygons;
* ***Shield***: styles for symbol annotation for points and lines;
* ***Line Pattern***: styles for applying textures (e.g., dashing) to lines;
* ***Polygon Pattern***: styles for applying textures to polygons;
* ***Markers***: styles for manipulating iconic point features

For Lab #3, you will style your tilesets at every zoom level, which means you will be designing for small and large scale simultaneously. This unique feature of slippy maps means that you have to incorporate your design scheme at the country, state, and road scale. Mapbox Studio’s use of CartoCSS is most often used with constricting certain design considerations at different zoom levels. For examples beside this lab, look at the built in example tilesets (BUT DO NOT COPY!).

Create a new tileset, and save it. You’ll notice that if you open the .tm2 file, there are a few different things in there. Any file with an .mss is a style sheet; you always start out with one called style. The next two files are an XML and YML. These files control your layer order, the spatial reference system being used, and all the various parameters involved in your tileset.

Let’s set up the essentials of the pop-art tileset (**Code Bank 1**). The first things I like to do are create global variables so I don’t have to use hex codes every time. I also want to declare my fonts so I can use nicknames to call them when needed. To set up your map, you must use the Map{} function, which creates a map object to be added to the tileset. There are several things you can do with Map{}, but declaring your background-color (or if you were to have a background-pattern) is typical for most CartoCSS tilesets.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1 //colors used a lot

2 @red: #BC243B;

3 @blue: #1D4E89;

4 @yellow: #FBD82B;

5 @black: #000;

6 @white: #fff;

7 @green: #019875;

8

9 @sans: ‘Komika Hand Regular’;

10 @sans\_italic: ‘Komika Hand Italic’;

11 @sans\_bold: ‘Komika Hand Bold’;

12

13 Map{

14 background-color: #fff;

15 }

16

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Code Bank 1: Global variables in Mapbox Studio**

After declaring global variables, it’s time to actually add things to the map! I always start with water because it rarely is different for zoom levels. It’s worth mentioning that Mapbox-Streets separates the labels for the layers into a different layer. You won’t want to use text functions for non-label layers. To color the water, you just use the selector for water and polygon-fill. You can also change the opacity or use change the polygon-gamma number. Changing the polygon-gamma affects anti-aliasing for polygon features, prevents gaps in the water coloring.

In order to add a border, I have to add a double colon and name this interior statement and can now style the water layer’s line properties. From Mapbox’s manual: “Within a layer, styles can be broken up into ‘attachments’ with the :: syntax. Think of attachments like sub-layers. Add attachment to data layers with heavy filtering to keep file sizes low.”

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1 #water {

2 opacity: .75;

3 polygon-fill: @blue;

4 polygon-gamma: .2;

5 ::line{

6 line-color: @blue;

7 line-width: 1.5;

8 line-join: round;

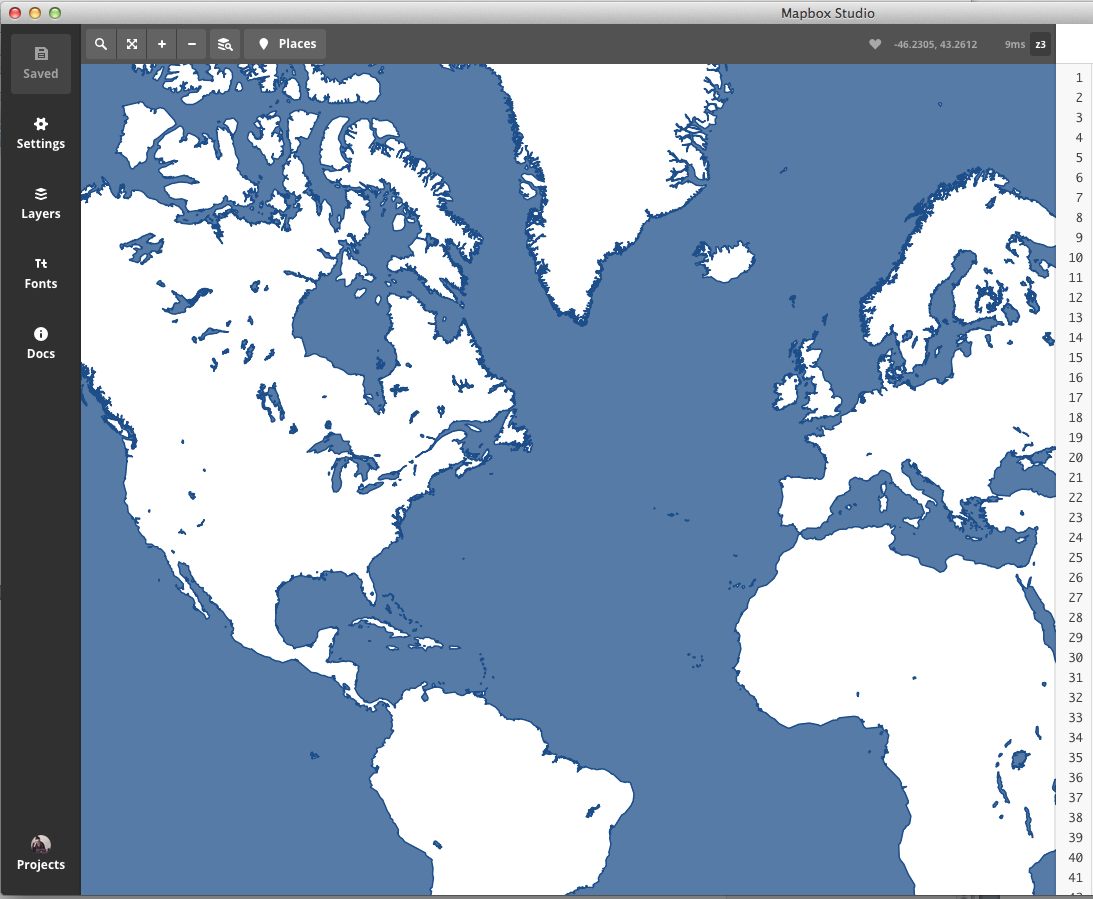
9 line-cap: round;

10 }

11 }

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Code Bank 2: styling #water in Mapbox Studio**



**Figure 17: #water drawn in Mapbox Studio**

The next thing to style are the country lines. The #admin selector has various attributes that can be used to specify my styling. If you look at the layer tab **(Fig 11)**, you’ll see that the admin\_level can be used to specify whether the administrative lines are for country, state, region, county, etc. Having it less than or equal to 2 guarantees that only country lines are showing. I also don’t want the borders on the water showing, so I can use the maritime attribute to only show the borders that don’t touch water.

To have a line show up, you have to declare the line-color and line-width. Because borders can appear jagged sometimes, I set the line-cap and line-joins to be round and add a line-smooth element. Also, to increase the line width at a higher zoom, I just need to use the zoom selector and set the line-width to a higher number at different zooms.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1 #admin[maritime=0][admin\_level<=2]{

2 line-color: @red;

3 line-cap: round;

4 line-join: round;

5 line-smooth: .75;

6 [zoom>=3]{

7 line-width: 1.15;

8 }

9 [zoom>=4]{

10 line-width: 1.5;

11 }

12 }

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Code Bank 3: Styling country lines in Mapbox Studio**



**Figure 18: country lines added to Mapbox Studio**

Now that we’ve explored how to style lines and polygons, it’s time to learn how to label. As mentioned earlier, layers are separated in Mapbox-Streets by the geometry and the labels. This makes it much easier to manage and avoid having to constantly use those attachments. If you look in the documentation tab, you’ll notice that under the text selectors, there are two different key words: text or shield.

**e. Uploading to Mapbox**

Now that you have designed your tiles, it is time to practice your understanding of TileMill, the TileMill documentation, and HTML/CSS generally by adding interactive functionality for the POIs using tooltips and by adding a custom legend. Make use of the ‘name’ and ‘description’ columns in the *.csv* file when creating the tooltip. For discussion about basic tooltip and legend templates, see: <https://www.mapbox.com/tilemill/docs/crashcourse/tooltips/#legends>. For discussion about advanced legends, see: <https://www.mapbox.com/tilemill/docs/guides/advanced-legends/>. Solutions to both interactive tooltips and a legend will be discussed the week before Lab #3 is due.

**f. Mapbox Editor**

After finishing the design of your tiles, and adding tooltips and a legend, it is time to export and serve them for public viewing on the web. In the *Main toolbar*, click the *Export* button and choose the *MBTiles* option. This will open a dialog window for creating your tileset (**Figure 16**). Provide a logical *Name*, *Description*, *Version*, and *Filename* for your tiles; also list OpenStreetMap under *Attribution*. Importantly, you need to set the *Zoom*, *Center* (click map)*,* and *Bounds* (shift+drag map)of your tileset. Keep in mind that you need to restrict the number of tiles you are producing to 50MB, otherwise you will need to pay for tile service from MapBox or implement the open source [TileStache](http://tilestache.org/doc/) solution; you can inspect a preview of the number of tiles and the tileset filesize under the *Zoom* interface. Once you are satisfied with your export parameters, click the *Export* button. This will activate the rendering and export process; depending on the number of tiles you are exporting, the export process can vary from minutes to hours to even days. Be patient, it is well worth the wait!

Once the tile rendering is complete, press the *Save* button, which will then navigate you to a preview page of your tiles. On the preview page, press the *Save MBTiles* button and be sure to take note of the file location at which your MBTiles are stored (**Figure 17**); *navigate to that location and save a copy of your MBTiles to your USB drive.* One saved and stored, navigate back to the MapBox website and login to your MapBox account.

Once logged in, click on the wrench icon in the top-right corner of your MapBox profile page and select the *Upload Layer* option. When prompted, select your stored MBTiles file and click *Upload File*; give your tileset a logical name once uploaded. You then will be redirected to a preview of your map tiles. Click on the pencil symbol to edit the settings of your tileset (**Figure 18**). Once the dialog window is activated, click on the eye icon beneath *Privacy* so that the tiles become *Public: Visible to everyone*. Save your changes and then click the *Publish* button.

Congratulations! You now are serving your own custom slippy map mashup. Finish the lab by embedding the map into your portfolio page using the provided HTML (i.e., the *Easy Embed*) from the *Publish* dialog window.

**Evaluation Rubric: Slippy Map Iconicity Challenge (40pts)**

**Inspiration Board Check-In (6)**

**5.5-6.0pts:** The icon library is ready for use in TileMill. All four of the above considerations for icon design were taken into account. All icons hold up when rasterized into the *.png* format.

**4.5-5.0pts**: The icon library needs a few small tweaks before it is ready for use in TileMill. One of the four above considerations for icon design was not fully taken into account during design. Several icons break down when rasterized into the *.png* format.

**3.5-4.0pts**: We need to rethink aspects of the icon library before it is ready for use in TileMill. Several of the four design guidelines were violated systematically. Multiple icons break down when rasterized into the *.png* format.

**3.0 pts or less:** The submitted icon library did not meet the expectations of the assigned challenge, or was not submitted at all.

**Due November 6th**

**POIs Check-In (6)**

**2.0pts:** You have assembled your *.csv* file.

**0pts:** You have not assembled your dataset

**Due November 13th**

**Mapbox Studio Tileset (32): Due Online November 20th**

**30-32pts:** The slippy map mashup is attractive, informative, and engaging; the map is a vast improvement over most web map mashups. The icon library unambiguously signifies the POIs and their higher-level categorization. The icons themselves are at the highest level of the visual hierarchy and hold up at all map scales. Your tileset is loading properly and reinforces the aesthetic style of your icon library. The tooltip and legend solutions are working properly and are well-designed. The aesthetic style of the tileset is consistent across zoom scales; what a stunning example of multiscale map design!

**26-29pts:** The slippy map mashup overall is successful as a web map. The icon library mostly works in representing the POIs and their higher-level categorization, but there are one or two icons that remain problematic. The icons work against the basemap and hold up at most, but not all, map scales. Your tileset is loading properly, as are the tooltip and legend solutions, although these solutions could be improved with further refinement. The aesthetic style of the tileset is mostly consistent across zoom scales, but there are unrefined transitions in linework styling between one or two zoom level changes.

**22-25pts:** The slippy map mashup only marginally improves upon typical web maps. Significant problems remain in the icon library, both in the representation of the unique POIs and their higher-level categorization. Many of the icons do not work against the basemap and/or across map scales. Your tileset is loading properly, but the tooltip and/or legend solutions are non-functional or illogical. Little effort was taken to develop an aesthetic style that is consistent across scales. Much more work needs to go into your linework generalization as you go from a large to a small cartographic scale (e.g., elimination of unneeded detail, simplification of lines and polygons, restyling the sizes or colors of features to maintain a proper visual hierarchy).

**Below 21pts:** The submitted slippy map mashup did not meet the expectations of the assigned challenge in multiple and critical ways. Please speak with Rob and Rashauna about strategies to improve the design.