BACK FORTY

MapBox & TileMill

An open-source-ish alternative to MapKit

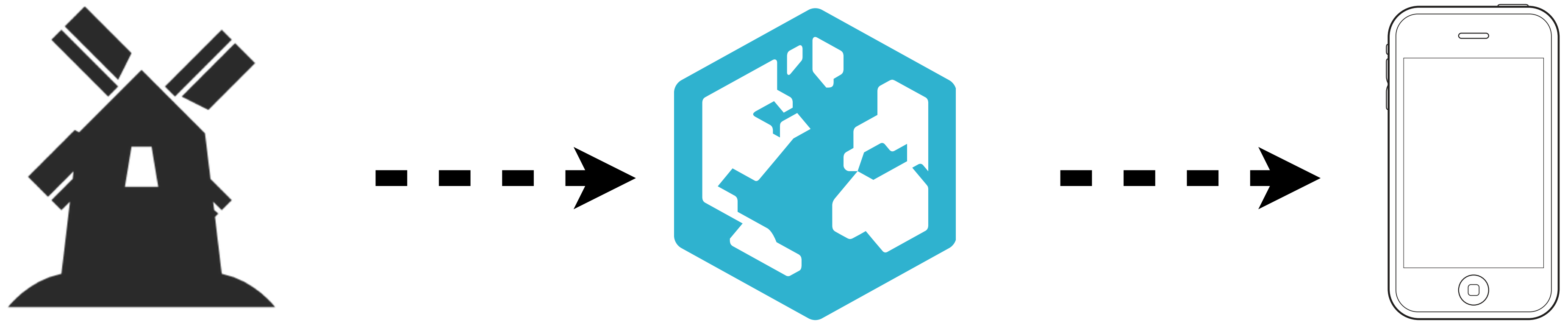
Flip Sasser
@flipsasser

inthebackforty.com
@InTheBackForty

I’m Flip

I’m just learning about MapBox but it’s kinda cool but kinda not so let me explain

The Fit



TileMill

(makes tiles)

MapBox

(makes maps)

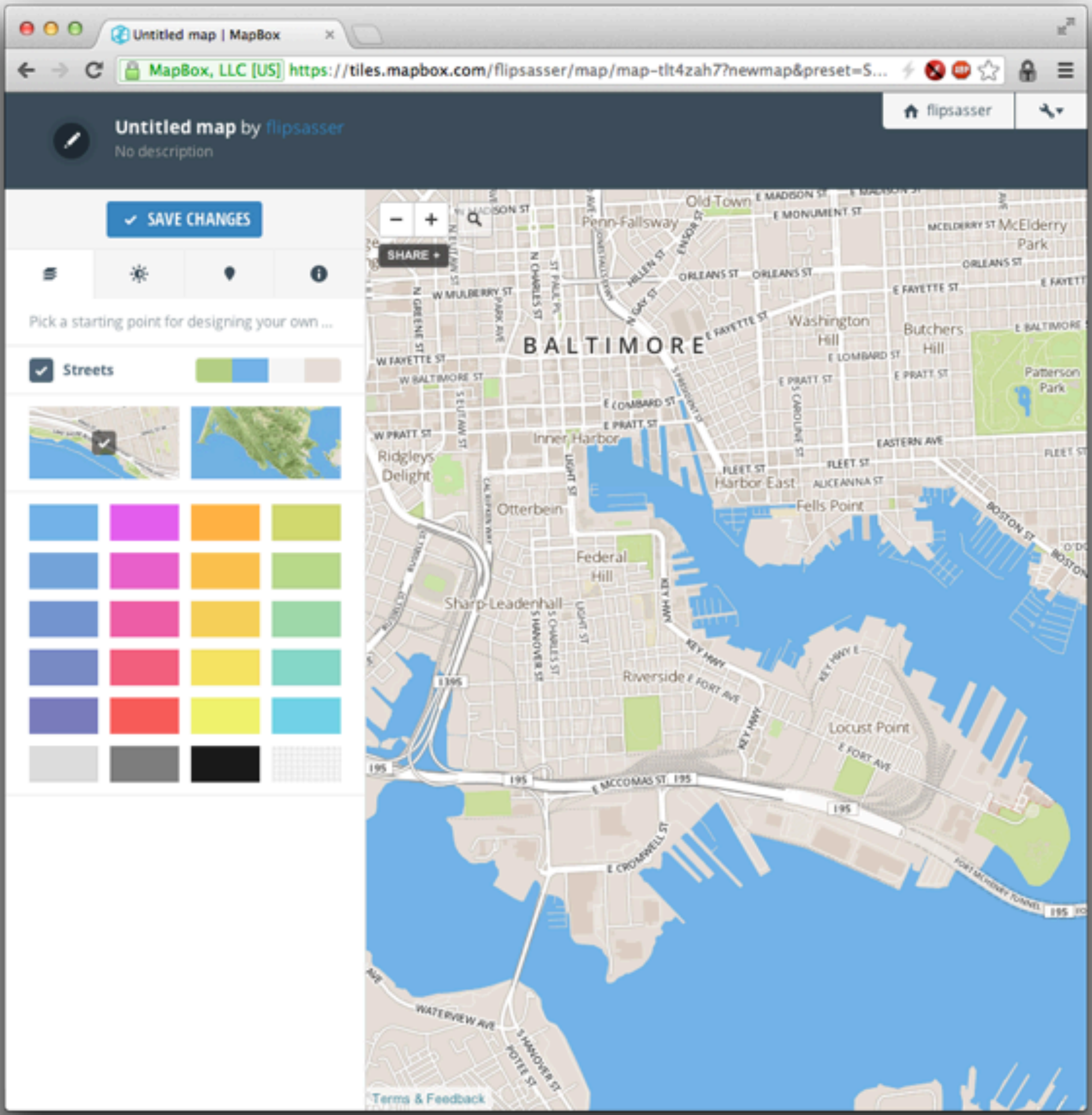
iOS

(renders maps)

Chapter 1

MapBox serves your tiles
(if you ever get them)

BACK ◆ FORTY


Untitled map | MapBox
MapBox, LLC [US] https://tiles.mapbox.com/flipsasser/map/map-tlt4zah7?newmap&preset=S...
flipsasser
Untitled map by flipsasser
No description
SAVE CHANGES
Pick a starting point for designing your own ...
Streets
SHARE +
BALTIMORE
Old Town
Penn-Fallsway
McElderry Park
Washington Hill
Butchers Hill
Patterson Park
Inner Harbor
Ridgleys Delight
Harbor East
Fells Point
Otterbein
Federal Hill
Sharp-Leadenhall
Riverside
Locust Point
Terms & Feedback

Creates a tile API endpoint for your map

```
<iframe width='500' height='300' frameBorder='0'
src='http://a.tiles.mapbox.com/v3/flipsasser.map-
tlt4zah7.html#14/39.274300000000004/-76.602'></iframe>
```

This is a *pay* service

But if you can get TileMill to export, you get a free, locally cached tileset!

BACK ◆ FORTY

Chapter 2

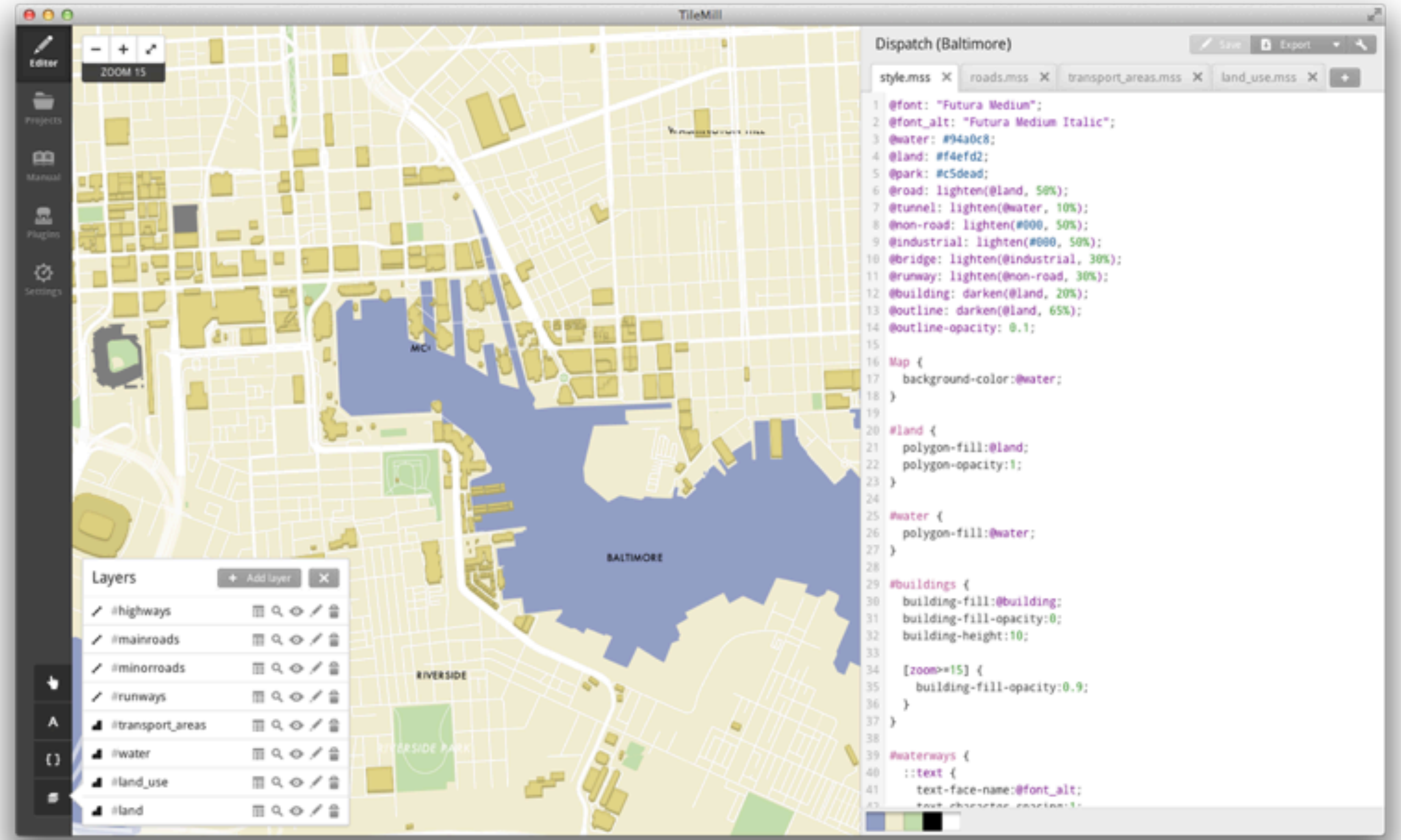
TileMill in all its ~~misery~~ ~~glory~~ misery

BACK FORTY

Node.js-backed HTML UI

It’s “cross platform”

BACK ◆ FORTY



```
@font: "Futura Medium";
@font_alt: "Futura Medium Italic";
@water: #94a0c8;
@land: #f4efd2;
@park: #c5dead;
@road: lighten(@land, 50%);
@tunnel: lighten(@water, 10%);
@non-road: lighten(#000, 50%);
@industrial: lighten(#000, 50%);
@bridge: lighten(@industrial, 30%);
@runway: lighten(@non-road, 30%);
@building: darken(@land, 20%);
@outline: darken(@land, 65%);
@outline-opacity: 0.1;

Map {
  background-color:@water;
}

#land {
  polygon-fill:@land;
  polygon-opacity:1;
}

#water {
  polygon-fill:@water;
}

#buildings {
  building-fill:@building;
  building-fill-opacity:0;
  building-height:10;

  [zoom>=15] {
    building-fill-opacity:0.9;
  }
}

#waterways {
  ::text {
    text-face-name:@font_alt;
```

BACK ◆ FORTY

Draws tile layers from various data sources

What data sources?

- **Open Street Maps**
- **Open ... Street Maps**
- **Open, well, Street Maps**

BACK FORTY

Ways to get OSM data

Because there's a lot of it

BACK FORTY

The firehose

planet.openstreetmap.org/
25GB of data

Landmasses (landmassi̇?)

download.geofabrik.de/openstreetmap/

Large maps or maps of specific territories

Coastlines

openstreetmapdata.com/data/land-polygons

These make a *huge* difference

Coastlines w/OSM base data

BACK ◆ FORTY

*that's Baltimore, yo!

Coastlines w/detailed data*

BACK FORTY

Streets, railways, and buildings

metro.teczno.com/

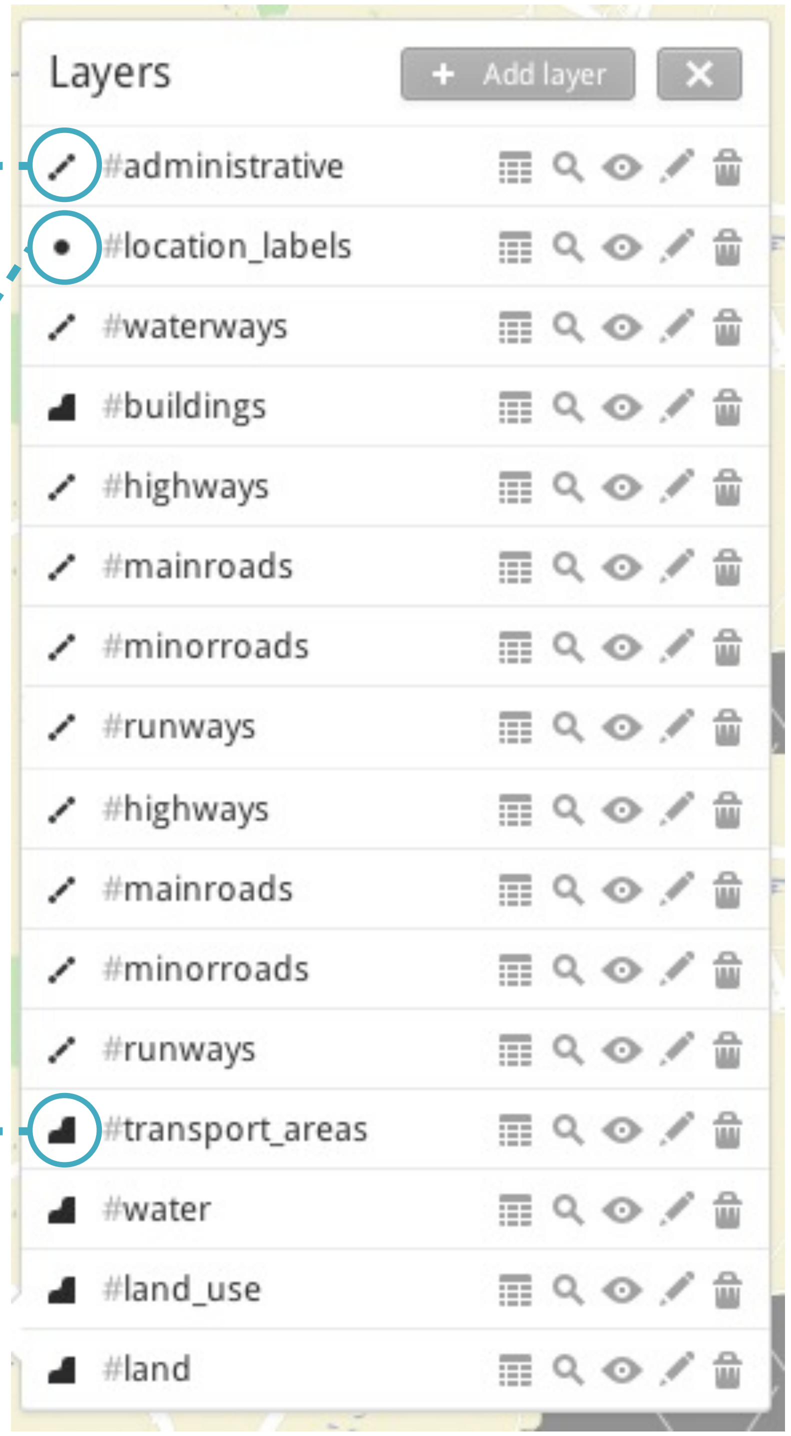
Look for your specific metro area

BACK ◆ FORTY

Path layers
Style as lines

Point layers
Style as markers

Polygon layers
Style as shapes



#administrative
#location_labels
#waterways
#buildings
#highways
#mainroads
#minorroads
#runways
#highways
#mainroads
#minorroads
#runways
#transport_areas
#water
#land_use
#land

Put ‘em together

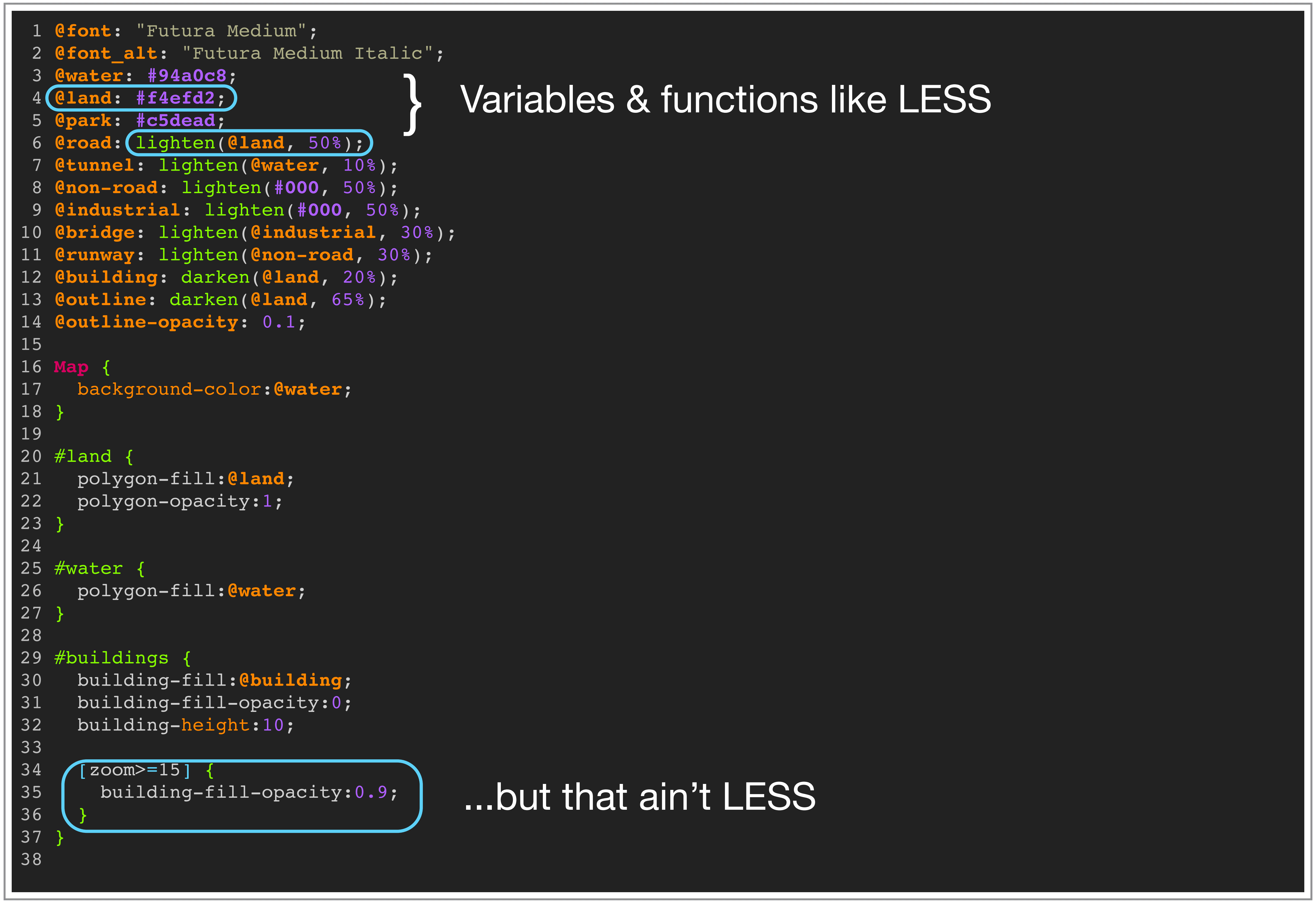
BACK FORTY

CartoCSS

for to style your maps with

It’s LESS CSS, but insane

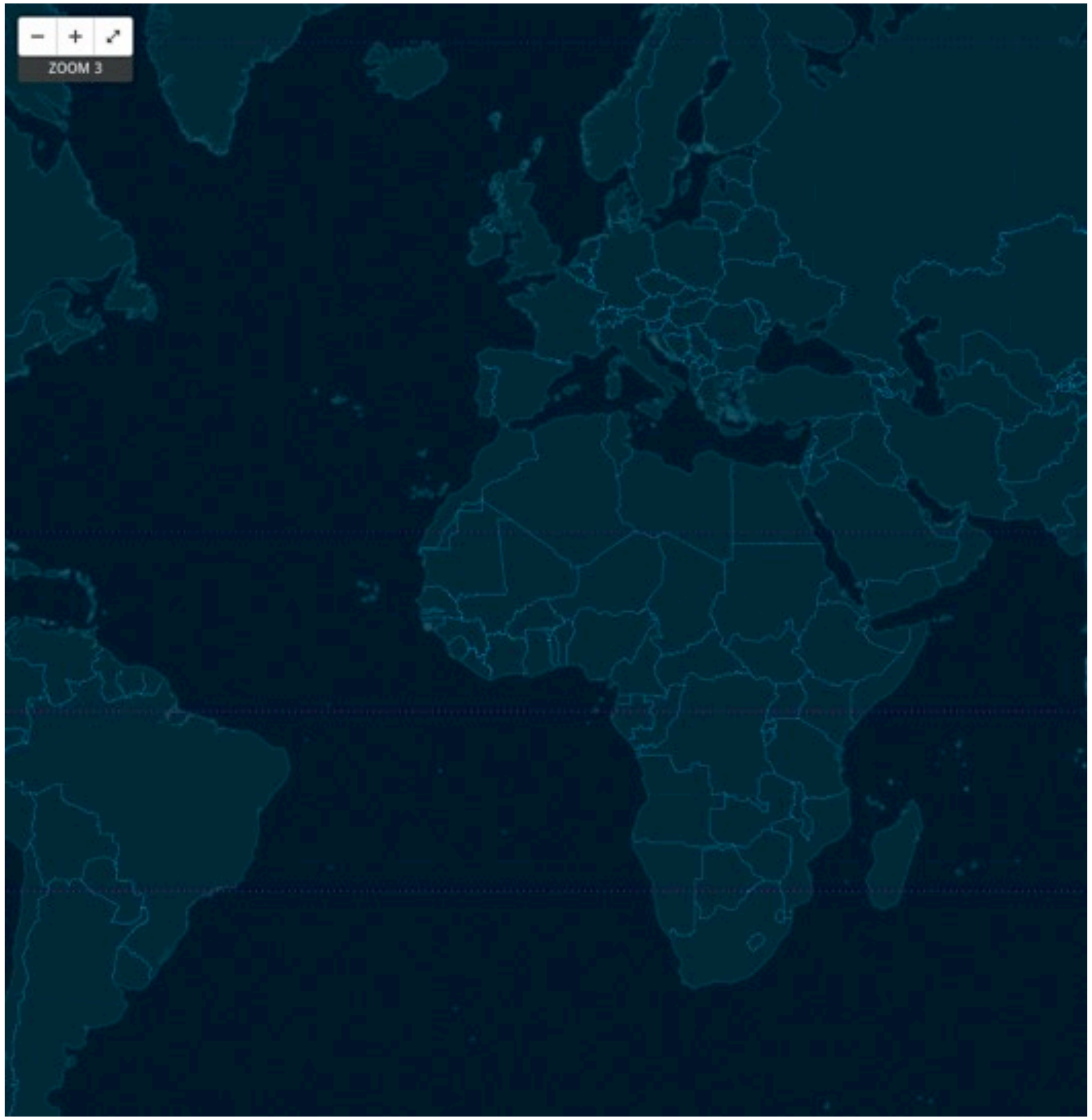
BACK ◆ FORTY



```
@font: "Futura Medium";
@font_alt: "Futura Medium Italic";
@water: #94a0c8;
@land: #f4efd2;
@park: #c5dead;
@road: lighten(@land, 50%);
@tunnel: lighten(@water, 10%);
@non-road: lighten(#000, 50%);
@industrial: lighten(#000, 50%);
@bridge: lighten(@industrial, 30%);
@runway: lighten(@non-road, 30%);
@building: darken(@land, 20%);
@outline: darken(@land, 65%);
@outline-opacity: 0.1;

Map {
  background-color:@water;
}

#land {
  polygon-fill:@land;
  polygon-opacity:1;
}

#water {
  polygon-fill:@water;
}

#buildings {
  building-fill:@building;
  building-fill-opacity:0;
  building-height:10;

  [zoom>=15] {
    building-fill-opacity:0.9;
  }
}
```

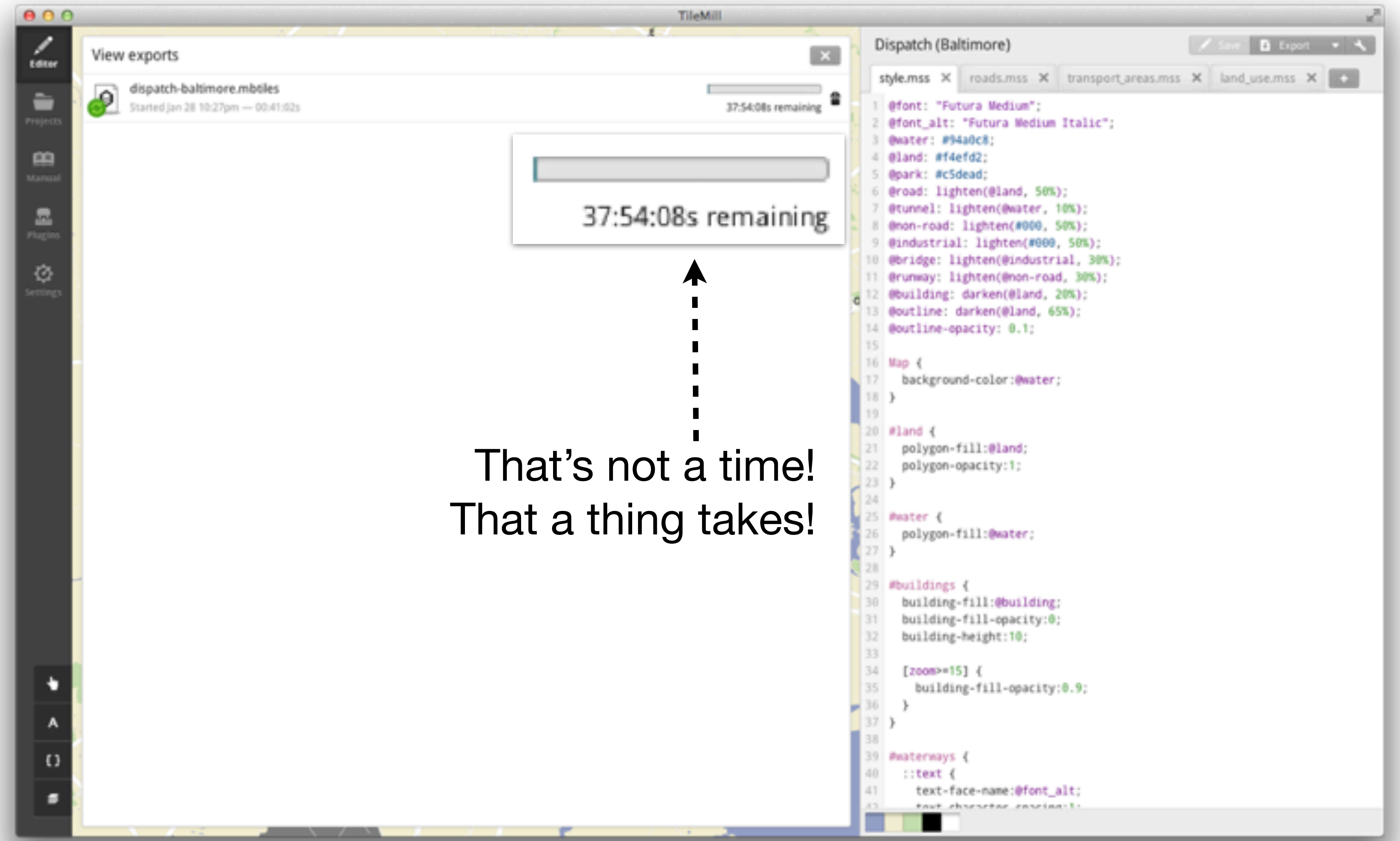
} Variables & functions like LESS

...but that ain't LESS



Still, you can make pretty maps...

BACK ◆ FORTY



dispatch-baltimore.mbtiles
Started Jan 28 10:27pm — 00:41:02s
37:54:08s remaining

Dispatch (Baltimore)

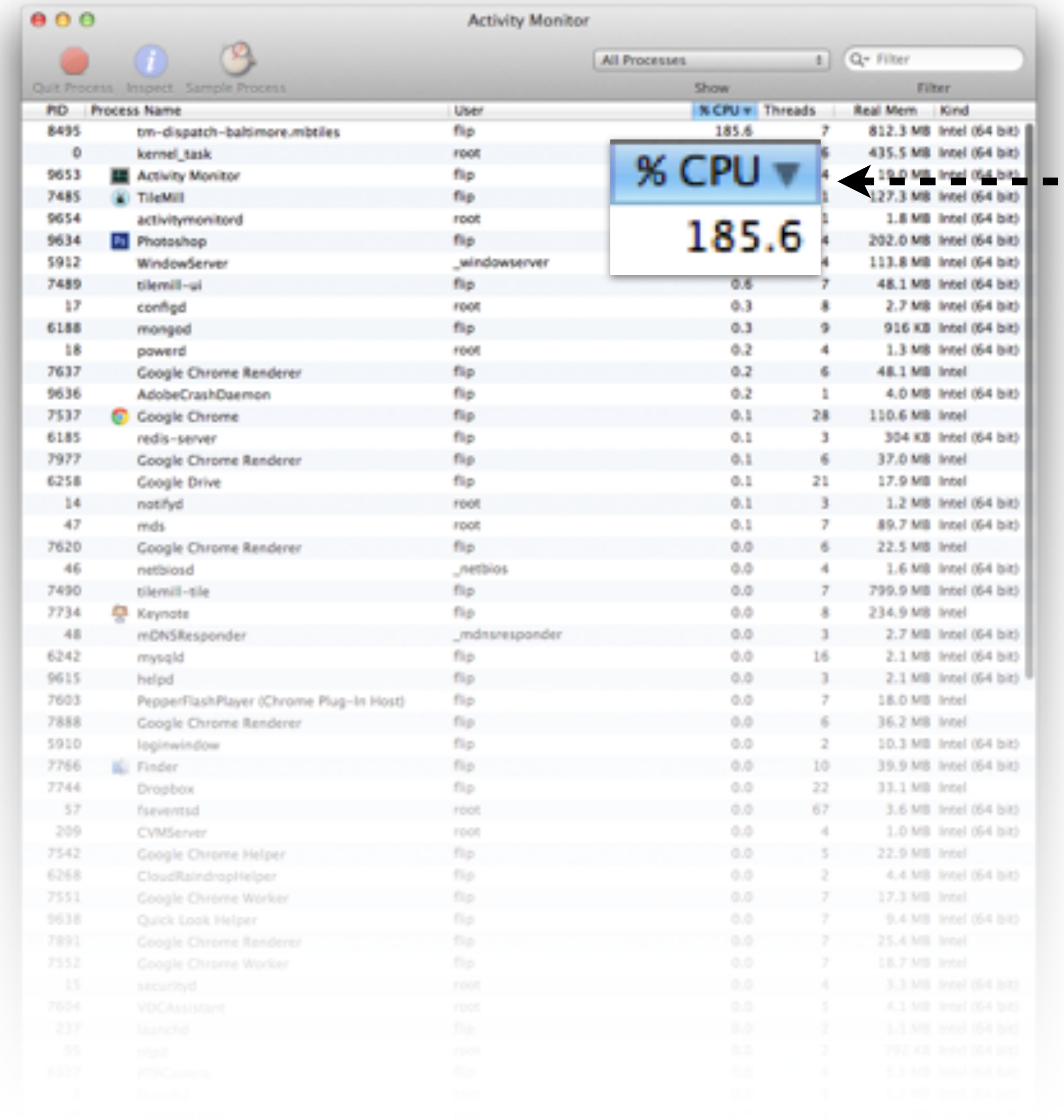
```
@font: "Futura Medium";
@font_alt: "Futura Medium Italic";
@water: #94a0c8;
@land: #f4efd2;
@park: #c5dead;
@road: lighten(@land, 50%);
@tunnel: lighten(@water, 10%);
@non-road: lighten(#000, 50%);
@industrial: lighten(#000, 50%);
@bridge: lighten(@industrial, 30%);
@runway: lighten(@non-road, 30%);
@building: darken(@land, 20%);
@outline: darken(@land, 65%);
@outline-opacity: 0.1;

Map {
  background-color:@water;
}

#land {
  polygon-fill:@land;
  polygon-opacity:1;
}

#water {
  polygon-fill:@water;
}

#buildings {
  building-fill:@building;
  building-fill-opacity:0;
  building-height:10;

  [zoom>=15] {
    building-fill-opacity:0.9;
  }
}

#waterways {
  ::text {
    text-face-name:@font_alt;
```

37:54:08s remaining

That's not a time!
That a thing takes!

Unless they're too complex.

BACK ◆ FORTY



PID	Process Name	User	% CPU	Threads	Real Mem	Kind
8495	tm-dispatch-baltimore.mbtiles	flip	185.6	7	812.3 MB	Intel (64 bit)
0	kernel_task	root	[illegible]	6	435.5 MB	Intel (64 bit)
9653	Activity Monitor	flip	[illegible]	4	19.0 MB	Intel (64 bit)
7485	TileMill	flip	[illegible]	1	127.3 MB	Intel (64 bit)
9654	activitymonitord	root	[illegible]	1	1.8 MB	Intel (64 bit)
9634	Photoshop	flip	[illegible]	4	202.0 MB	Intel (64 bit)
5912	WindowServer	_windowserver	[illegible]	4	113.8 MB	Intel (64 bit)
7489	tilemill-ui	flip	0.6	7	48.1 MB	Intel (64 bit)
17	configd	root	0.3	8	2.7 MB	Intel (64 bit)
6188	mongod	flip	0.3	9	916 KB	Intel (64 bit)
18	powerd	root	0.2	4	1.3 MB	Intel (64 bit)
7637	Google Chrome Renderer	flip	0.2	6	48.1 MB	Intel
9636	AdobeCrashDaemon	flip	0.2	1	4.0 MB	Intel (64 bit)
7537	Google Chrome	flip	0.1	28	110.6 MB	Intel
6185	redis-server	flip	0.1	3	304 KB	Intel (64 bit)
7977	Google Chrome Renderer	flip	0.1	6	37.0 MB	Intel
6258	Google Drive	flip	0.1	21	17.9 MB	Intel
14	notifyd	root	0.1	3	1.2 MB	Intel (64 bit)
47	mds	root	0.1	7	89.7 MB	Intel (64 bit)
7620	Google Chrome Renderer	flip	0.0	6	22.5 MB	Intel
46	netbiosd	_netbios	0.0	4	1.6 MB	Intel (64 bit)
7490	tilemill-tile	flip	0.0	7	799.9 MB	Intel (64 bit)
7734	Keynote	flip	0.0	8	234.9 MB	Intel
48	mDNSResponder	_mdnsresponder	0.0	3	2.7 MB	Intel (64 bit)
6242	mysqld	flip	0.0	16	2.1 MB	Intel (64 bit)
9615	helpd	flip	0.0	3	2.1 MB	Intel (64 bit)
7603	PepperFlashPlayer (Chrome Plug-In Host)	flip	0.0	7	18.0 MB	Intel
7888	Google Chrome Renderer	flip	0.0	6	36.2 MB	Intel
5910	loginwindow	flip	0.0	2	10.3 MB	Intel (64 bit)
7766	Finder	flip	0.0	10	39.9 MB	Intel (64 bit)
7744	Dropbox	flip	0.0	22	33.1 MB	Intel
57	fseventsd	root	0.0	67	3.6 MB	Intel (64 bit)
209	CVMServer	root	0.0	4	1.0 MB	Intel (64 bit)
7542	Google Chrome Helper	flip	0.0	5	22.9 MB	Intel
6268	CloudRaindropHelper	flip	0.0	2	4.4 MB	Intel (64 bit)
7551	Google Chrome Worker	flip	0.0	7	17.3 MB	Intel
9638	Quick Look Helper	flip	0.0	7	9.4 MB	Intel (64 bit)
7891	Google Chrome Renderer	flip	0.0	7	25.4 MB	Intel
7552	Google Chrome Worker	flip	0.0	7	18.7 MB	Intel
15	securityd	root	0.0	4	3.3 MB	Intel (64 bit)
7604	VDCAssistant	root	0.0	5	4.1 MB	Intel (64 bit)
237	launchd	flip	0.0	2	1.1 MB	Intel (64 bit)
95	ntpd	root	0.0	2	792 KB	Intel (64 bit)
6327	PTPCamera	flip	0.0	4	5.5 MB	Intel (64 bit)
1	launchd	root	[illegible]	[illegible]	[illegible]	[illegible]

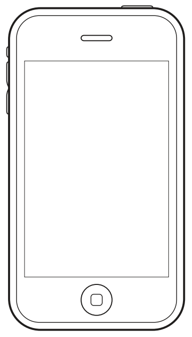
That's not an amount of CPU!
That a thing takes!

Unless they're too complex.

BACK FORTY

My map of Baltimore wouldn’t export.

It’s *just* of Baltimore.



Chapter 3: iOS

Cause you’re all like, “WTF THIS IS BMORE COCOA NOT BMORE MAPPING”

BACK ◆ FORTY

3.1: Installing MapBox

I prefer git submodules. YMMV, but this is how I got it working.

```
$ git submodule add git://github.com/mapbox/mapbox-ios-sdk.git
```

Add MapBox's submodules

```
$ git submodule update --init --recursive
```

This is the *most important* part of getting MapBox running!

Add MapBox to your target

`Demo/mapbox-ios-sdk/MapView/MapView.xcodeproj`

drag to your Frameworks folder

Add libraries to your target

- CoreLocation
- QuartzCore
- libsqlite3
- libz
- libMapBox

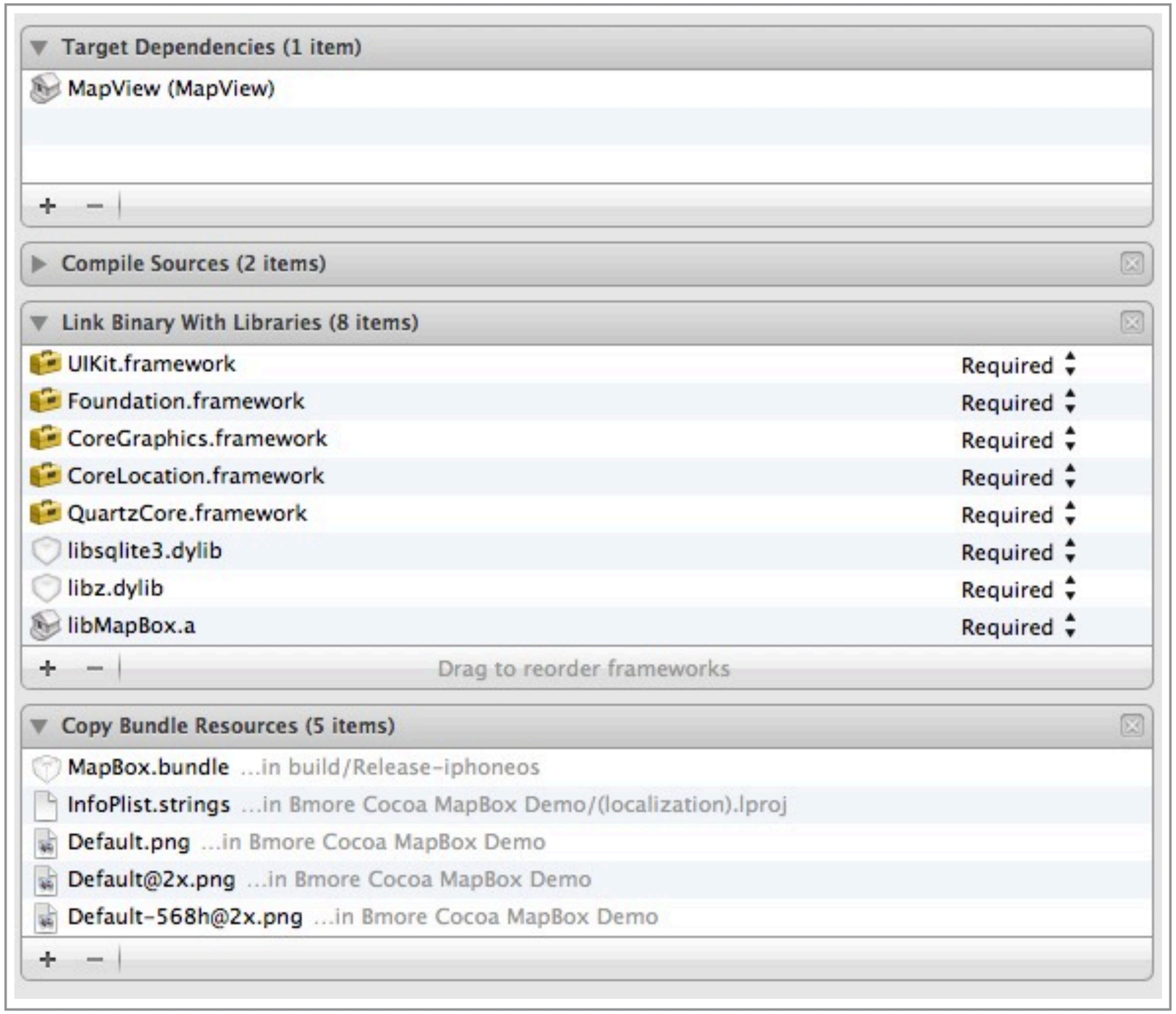
BACK FORTY

Add to your header search path

`$(SRCROOT)/mapbox-ios-sdk/MapView/`

Add to Header Search Paths for your target

Check "recursive"



Target Dependencies (1 item)

MapView (MapView)

Compile Sources (2 items)

Link Binary With Libraries (8 items)

UIKit.framework	Required
Foundation.framework	Required
CoreGraphics.framework	Required
CoreLocation.framework	Required
QuartzCore.framework	Required
libsqlite3.dylib	Required
libz.dylib	Required
libMapBox.a	Required

Drag to reorder frameworks

Copy Bundle Resources (5 items)

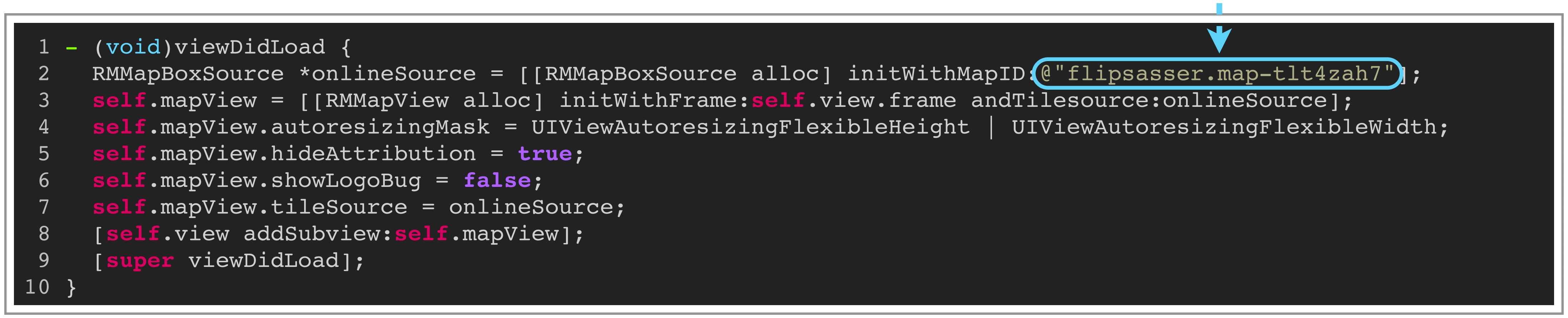
MapBox.bundle ...in build/Release-iphoneos
InfoPlist.strings ...in Bmore Cocoa MapBox Demo/(localization).lproj
Default.png ...in Bmore Cocoa MapBox Demo
Default@2x.png ...in Bmore Cocoa MapBox Demo
Default-568h@2x.png ...in Bmore Cocoa MapBox Demo

Target dependencies & resources

BACK FORTY

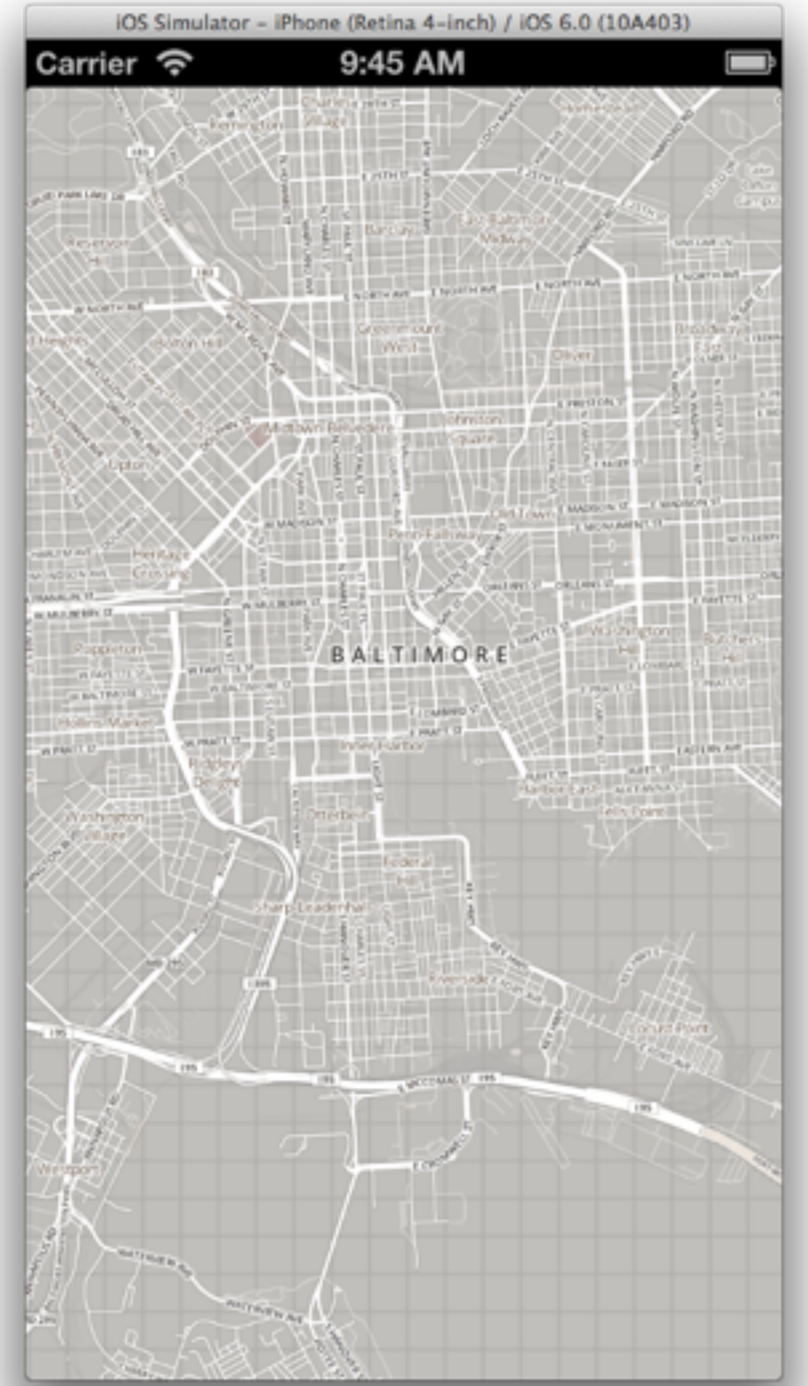
...back to the demo

MapBox ID



```
- (void)viewDidLoad {
  RMMapBoxSource *onlineSource = [[RMMapBoxSource alloc] initWithMapID:@"flipsasser.map-tlt4zah7"];
  self.mapView = [[RMMapView alloc] initWithFrame:self.view.frame andTilesource:onlineSource];
  self.mapView.autoresizingMask = UIViewAutoresizingFlexibleHeight | UIViewAutoresizingFlexibleWidth;
  self.mapView.hideAttribution = true;
  self.mapView.showLogoBug = false;
  self.mapView.tileSource = onlineSource;
  [self.view addSubview:self.mapView];
  [super viewDidLoad];
}
```

BACK ◆ FORTY

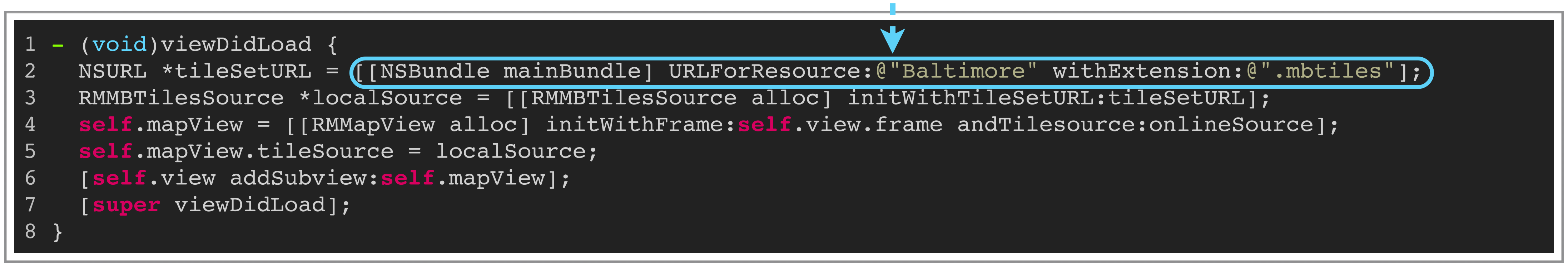


Voilà!

RMMBTilesSource

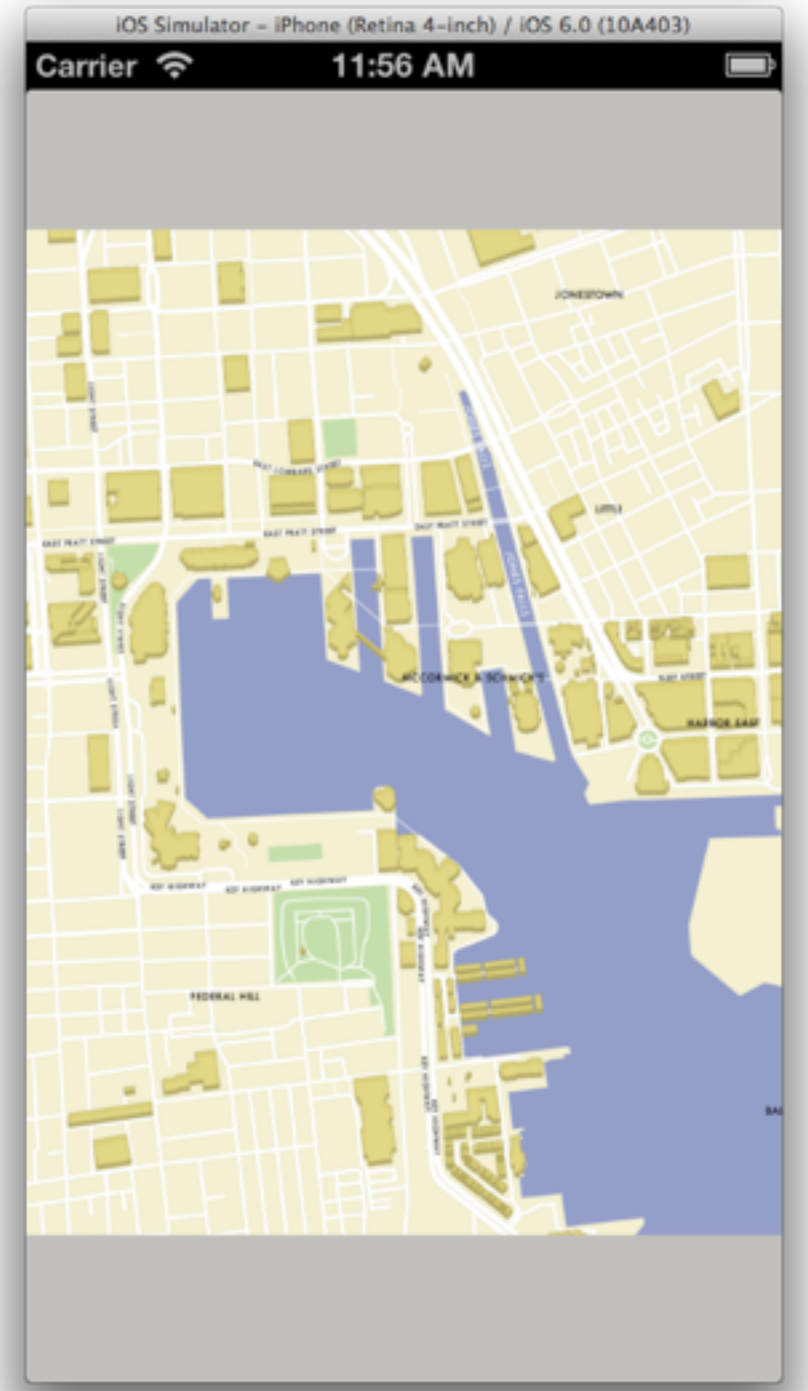
For storing tiles locally

Local Source



```objc
- (void)viewDidLoad {
  NSURL *tileSetURL = [[NSBundle mainBundle] URLForResource:@"Baltimore" withExtension:@".mbtiles"];
  RMMBTilesSource *localSource = [[RMMBTilesSource alloc] initWithTileSetURL:tileSetURL];
  self.mapView = [[RMMapView alloc] initWithFrame:self.view.frame andTilesource:onlineSource];
  self.mapView.tileSource = localSource;
  [self.view addSubview:self.mapView];
  [super viewDidLoad];
}
```

BACK ◆ FORTY



Voilàier!*

*this is a tiny subset of my original map

RMMapViewDelegate

For adding markers, shapes, layers!
For responding to boundary changes!
For handling taps and gestures!
RTFM!

BACK ◆ FORTY

Other awesome stuff

- REAL shape drawing
- Custom tile systems (for the adventurous!)
- Caching of remote tiles
- Animated zooming (looks AWESOME)

BACK FORTY

Drawbacks

TileMill

The worst or the worst?

BACK FORTY

Raster vs. Vector

Tiles are old technology

BACK FORTY

Pay-to-play

You pay for the API, or you pay to remove the logo from the UI, or you pay for both

BACK FORTY

Conclusions

MapBox is right if you need...

- Custom map styles
- Complicated drawing
- Beautiful animation
- Public APIs for drawing, tiling, and mercator projections
- Accurate data (thanks anyway, Apple)

BACK FORTY

MapBox is wrong if you need...

- Simple or quick maps
- Vector maps
- Money

BACK FORTY

Thnaks!

github.com/BackForty/map_box_demo

BACK FORTY

Check out demo the source and this presentation:

github.com/BackForty/map_box_demo