Thanks for taking the time to check out my slides. This presentation covers my experiences with working with open street map data for the first time, in particular the trials and tribulations of trying to stream massive OSM data sets in 3D on the web with Cesium.
Cesium is an open-source browser-based geospatial visualization engine for 3D globes and maps. It uses the Apache 2 license and because it's built with browser-based technology can run almost anywhere. It’s grown into a large open source community project, with over 80 contributors and a vibrant forum with over 900 members.

Cesium provides 3D, 2D, and 2.5D (what we call Columbus View), all through a single API. It’s not just open source, it was also built for and with open standards in mind. It’s highly customizable with a flexible API. Finally, in addition to 3D, one of the things that sets it apart is our handling of time-dynamic data, almost everything in Cesium can be animated over time without losing interactivity. Cesium was originally built with aerospace use cases in mind, so it's incredibly precise and accurate in its visualization and can go from ground level to outer space from within the same map.
When we started Cesium, we had no interest in trying to invent new data formats. We just wanted to leverage existing open standards as much as possible. However, what we found is that most data and formats out there are 2D-only. Visualizing data in 3D is very different than traditional 2D maps and 3D is a lot more than 2D + 1. On top of that, most 3D data that is available is either not web-friendly or not designed for 3D visualization. Even geospatial standards like GML are built more for data interoperability than direct visualization.
For these reasons the Cesium team had a new goal and has gotten more and more involved in the content-generation pipeline.

- New Goal
  - Help create open 3D formats and standards
  - Make it easy to process existing open data into visualization-friendly formats
I'm proud to say we've published several open formats, starting with CZML and then quantized-mesh for terrain. We got serious when we were part of the team that helped create glTF (you can tell because it has a fancy logo). glTF is an official 3D model standard published by Khronos. For those that may not be familiar with them, Khronos is the governing body behind many graphics standards, such as OpenGL, Collada, and WebGL, so it was incredibly exciting for us to be a part of it and glTF is gaining tons of traction across many domains.

But there was still some big-data sized holes in our needs, and while we could represent single buildings, trees, and small point clouds; our goal was to render truly massive datasets.
Our most recent initiative is 3D Tiles, a new open format whose goal is to enable massive heterogeneous streaming 3D visualization and styling. Common use cases for this is point clouds, large scale 3D vector data, terrain, and of course cities.

3D Tiles are the missing link between massive heterogeneous 3D geospatial datasets and 3D mapping engines. Bringing techniques from graphics research, the movie industry, and the game industry to geospatial.

Once 3D Tiles was far enough along in development, it became time to really look for good datasets to help flex its muscle. I'm a big fan of open data; but before I started on this app, I never worked with open street map before. I heard they provided height information for some of their buildings and I thought that OSM could potentially be an awesome source of data to show off 3D Tiles and Cesium. Then I saw on Twitter that NYC had just imported all of it's data into OSM and I knew the time had come.
The end result was this demo: https://cesiumjs.org/NewYork

We took the New York City OpenStreetMap extract from Mapzen and turned it into 3D Tiles.

Not only are we visualizing over 1.1 million buildings, but you can mouse over each building to see what it is and click to get more information. Keeping this level of interactivity in tact was very important to us. We didn’t just want to turn the data into a pretty picture, we wanted to avoid losing any information.

Here are some highlights:

1. You can select various points of interest in NYC using the combo box in the upper left
2. Clicking on a building gets you the OSM metadata, including links back to the same feature on OSM
3. You can color buildings by their height (this styling is computed on the fly and doesn’t have to be “baked in” to the dataset.) You can also select different color palletees.
4. You can also use the slider to hide buildings below a certain height.
5. The “Show broken OSM relations” checkbox will be explained in a future slide.
6. Flyover mode is just a “demo” mode that disables mouse control and takes you on a tour of the data set.
Now that you’ve seen the demo, I want to talk about how I built it.

The first step was getting the data. I knew that OSM made data available for download, but I figured trying to start with the 70 gig global dataset was probably not the best idea. I then found an extracts page that made OSM pbf files for individual cites and metro areas available. A few different sites generate these extracts, but lately I've been using this page from Mapzen to play with different data sets.

So at the same time I was learning about OSM data, I was also diving headfirst into NodeJS development, so most of the pipeline is written in JavaScript. Turns out there’s an OSM pbf reader that will give you JSON representations of nodes, ways, and relations from OSM, osm-pbf-parser. OSM also has a pretty large set of wiki pages that discuss where the data is supposed to look like and how data is supposed to be defined; and that was a valuable resource in having some sense of what to look for in the data; but ultimately I ended up with logging lots of output to the screen in order to figure out how the objects worked.
Remember how I mentioned that most data was more for interop than visualization, OSM is no different. They don't actually provide any data that is any way ready to be visualized. Instead, I had to reconstruct a 3D model representation from the object description. This obviously isn't even the full description, it's just a table of cross-references that refer to other parts other OSM data. Everything is laid out in an order that's excellent for archival and interop purposes, but a long way away from visualization.
Once I could generate Collada files, the next step was taking those files and turn them into something streamable on the web. From the single 75 megabyte pbf file; I ended up generating 7.67 GB gigabytes of model data, that's over 1.1 million (1,160,188) individual model files. Collada files in general aren't very web-friendly, and not only that but making a new web request for every building isn't scalable and will take forever to load, even if you're only loading buildings in view (since as we saw from the demo, a single view can still have thousands of buildings in it). Plus, just the locations and metadata for the buildings was an 800mb JSON file on top of the models themselves.
So the next step was to leverage both the glTF and 3D Tiles formats I mentioned earlier to enable me to visualize the data in Cesium. I won't go into the details of exactly how that works, but basically we tile up the models into a 3D geospatial data structure and then create 3D tiles, which has combined geometry from multiple models. All of the data is stored in the glTF model format, which is less verbose than collada and is much more performant to load and render at runtime. Here's what a single tile looks like when visualized.

The complete dataset was reduced from 7.65 gigs down to just 301 megabytes, on top of that, the number of files went from over a million down to 3620 files.

A detailed break down of how 3D Tiles works is available at http://cesiumjs.org/publications.html
Once I started visualizing that data, it was pretty easy to waste a bunch of time just exploring the city. As I showed you early, you can mouse around and identify different buildings, or click on them and get additional information from OSM.

I started to notice some problems, as I showed you earlier, as you mouse of each building it highlights. However, some buildings wouldn't be fully highlighted and instead only a piece of it was. I beat my head against the wall for a couple of hours until I realized that this is actually a problem in the OSM data itself. It turns out that a lot of building parts don't actually refer back to their parent relation like they are supposed to, and these broken relations are littered throughout the NYC data set.

It's really easy to detect that a relation is broken, basically an OSM way is marked as a building:part but no relations actually ever refer to that part.

The problem is that while it's easy to detect that these orphaned parts exist, it's very difficult (at least from what I can tell) to determine which relation they belong to. But by looking at the data visually, we can actually highlight these broken relations and it becomes obvious which pieces belong to which
buildings. Now I didn't go as far as starting to write a 3D OSM editor, but hopefully you can see from this demo that there are many benefits to such possibilities.

Simply visualizing data in 3D provides new insights into that data even without trying to take advantage of 3D specific geo-analysis.

Of course visualizing the data is only the beginning. A crucial piece of the 3D Tiles format is being able to declaratively style things with meta-data, which you already saw in these red broken relations. I can also do things like color code the buildings based on their height, or zoning, or any other metadata you might have, such as energy usage or average temperature. I also added a little slider that shows and hides buildings above a certain height to make it easier to navigate around large cities like NY. It was really important for us to maintain this level of interactivity in 3D Tiles and it allows for a ton of use cases for working with large data.
Of course styling the data is just one use case. Seeing how those buildings influence the world around them is another. For example; in city planning you may want to do a shadow study to see how a new structure will affect sunlight. Now this is still a working in progress so it’s behind the "shadows" query parameter on the demo URL to enable shadow casting. We accurately model the sun in Cesium and so when implementing shadows we also want them to be as accurate as possible. Here I’m going to hit play and what you are looking at now is 4:00pm, every day for a whole year. Every frame that gets rendered is advancing at a 24 hour clip and you can see as the seasons in NY change, shadows at 4pm get longer or shorter throughout the year. You can pick whatever date and time you want and see what it will look like, but I though this particular demo was kind of cool.
Using the New York open street map data worked out better than I could have possibly imagined, and having such a large data set available has actually helped foster a lot of the development that is taking place in Cesium. The work we did with OSM data is also easily leveraged for other data sets, so for example we are not that far away from being able to take CityGML or KML datasets and turn them into 3D Tiles as well. If anyone has datasets they would like to see visualized, I'd love to hear about them. I think it's safe to say that availability of open data directly drives open source development.

If there's one major downside I found it's that NYC data on open street map is the exception rather than the rule. Only a very small percentage of OSM data actually has 3D building metadata. Even major cities like Philadelphia have less than a dozen 3D buildings. Of course it's not all doom and gloom, for example I just discovered last week that an import of San Francisco buildings with height information is in progress. I'm hoping that as the open data movement continues to thrive, more and more high-quality sources become available.
We’re hiring

- [https://cesiumjs.org/jobs](https://cesiumjs.org/jobs)
  - Geospatial data lead
  - 3D developers
  - Full stack developers
  - Internships and co-ops
- Contact Me: [mamato@agi.com](mailto:mamato@agi.com)
If you saw this talk in person at FOSS4GNA 2016, please evaluate it the session (and all other sessions) to help drive the conference next year: https://2016.foss4g-na.org/node/add/session-evaluation?nid=2124&destination=conference/schedule/session/2016-05-04
So I'm just about out of time, but I hope I've peaked some of your interest in what modern 3D visualization on the web can provide, particular when it comes to large datasets. If you want to know more about Cesium, definitely head on over to cesiumjs.org and check us out. I'll also be around all today and tomorrow if anyone wants to chat.