OL3
A unique mapping library
Introduction

- New version of OL
- Version 2's been around since 2006
- Development began in 2013
- Complete rewrite!
Goals

- Feature-complete library
- More consistent API
- Smaller builds
- Push the limits of vector rendering
- Use Computer Graphics technologies (WebGL & Canvas)
« Maps as Graphics »
Positionning

- OpenLayers 2
- Leaflet
- OpenLayers 3
- Cesium
- OpenWebGlobe
Features

- Map rotations
- Animations
- Local/custom projections
- Arbitrary tile grids (WMTS)
- Various data providers and formats (OSM, BingMaps, WMS, GeoJSON, KML...)
- Image effect/pixel manipulation
- « Complex vector »
- ...
Example #1 (Rotation, Animation, Vector)
« Draw early, Draw often »
For good rendering quality:

- Vectors redrawn at each animation frame
- While interacting and animating
- (Hopefully) At 60 fps!
Performance challenge!

Let's look at the techniques used...
#1 Batching

Minimize data processing and manipulation:

- Style calculations
- Geometry simplifications
- R-tree lookups
- Object creations

⇒ Replay batch during animations/interactions
#2 Geometry Simplification

- Douglas Peuker (lines)
- Quantization – to maintain topology (polygons)

(Also allows for better rendering quality)
#3 Over-simplification

Over-simplification and clipping for the parts that are outside the viewport.
Example #2 (Complex and large features)
Hit Detection

Principle: redraw a subset of the scene in a small and off-screen canvas, and test if there's a color.
Example #3 (Hit Detection)
Support for Hit Detection for raster layers is about to be merged...
Roadmap

- Draw lines, polygons and text with WebGL
- Support tilted/perspective views
- Finish online build tool
- Improve our CommonJS/distribution story
- Improve the doc
- Continue with the Cesium integration (ol3-cesium)

⇔ Code sprint in April...
Tilted view prototype

Example #4 (Tilt)