

OL3

<http://erilem.net/talks/frog2013/ol3.html>

Éric Lemoine



<http://www.camptocamp.com>



eric.lemoine@camptocamp.com



[@elemoine](https://twitter.com/elemoine)



[@elemoine](https://github.com/elemoine)

Liens



<http://ol3js.org>



<http://github.com/openlayers/ol3>



<https://groups.google.com/forum/#>

Vision OL3

→ Où veut-on aller avec OL3 ?

2D et 3D

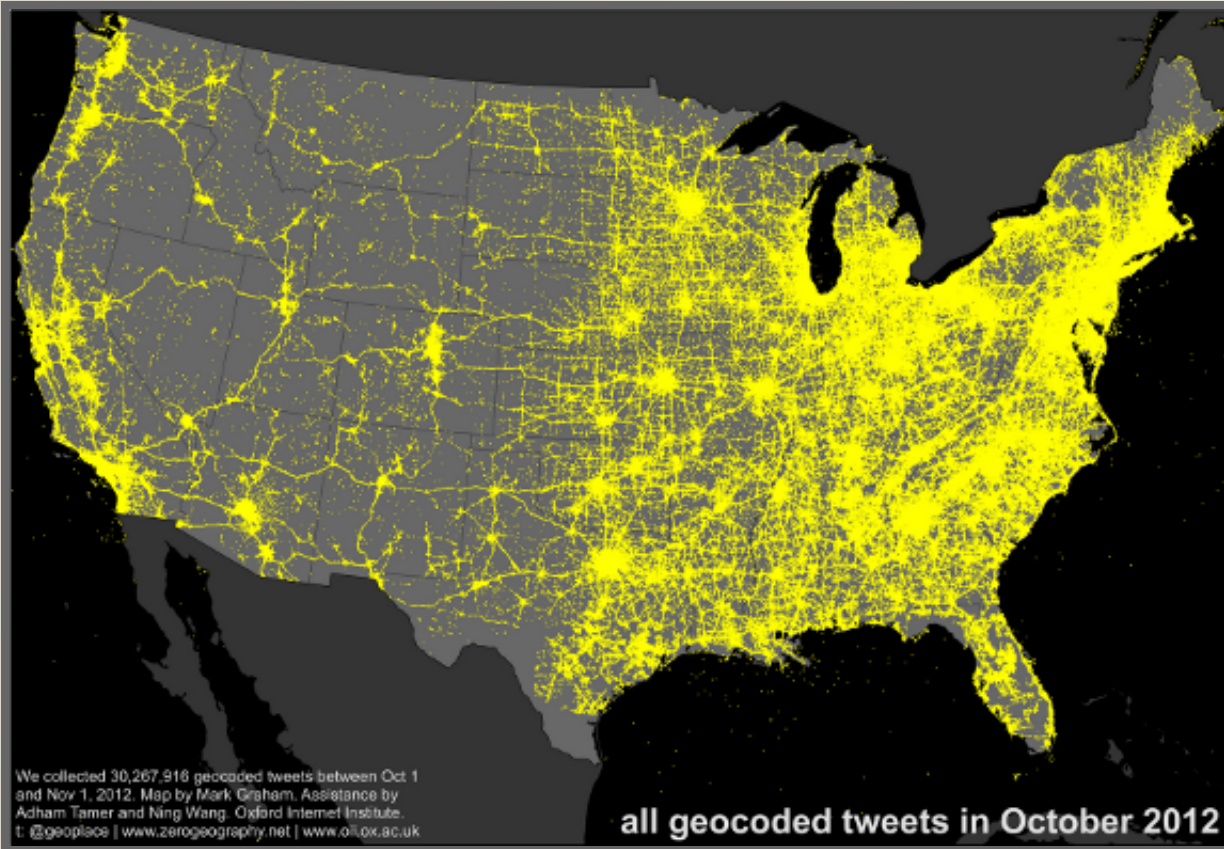


2D et 3D

Un de objectifs : afficher des données 3D de type "bâtiment" et "terrain".

Vecteur

Des millions?!



Vecteur

Objectif : afficher vecteur "complexes", autant au niveau des styles que des nombres d'objets à afficher.

Cartes = Graphiques



Cartes = Graphiques

Objectif : afficher des données 2D et 3D complexes avec de bonnes performances.

Utilisation de Canvas et WebGL.

"Mais d'autres libs sont déjà disponibles, non ?"

→ Où se situe OL3 ?

Où ?



OL3



- OL3 doit savoir faire beaucoup de choses !
- "Un projet ambitieux, pour des cartes ambitieuses"

API,
et exemples

ol.Map / ol.View

```
var view = new ol.View2D({  
  center: [0, 0],  
  zoom: 0  
});  
var map = new ol.Map({  
  target: 'map',  
  layers: [layer],  
  view: view  
});
```

Exemple Hello World

ol.layer / ol.source

```
var osm = new ol.layer.TileLayer({
  source: new ol.source.OSM()
});

var bing = new ol.layer.TileLayer({
  source: new ol.source.BingMaps({
    // your key here
    key: 'AlQLZ0-5yk301_ESrmN...',
    style: 'AerialWithLabels'
  })
});

var mapbox = new ol.layer.TileLayer({
  source: new ol.source.TileJSON({
    url: '//api.tiles.mapbox.com/v3/mapbox.world-dark.jsonp'
  })
});
```

Exemple couches tuilées

ol.interaction

ol.interaction.Drag

ol.interaction.DragRotate

ol.interaction.KeyboardPan

ol.interaction.TouchZoom

// etc.

ol.control

```
ol.control.Zoom  
ol.control.ScaleLine  
ol.control.MousePosition  
ol.control.Attribution  
// etc.
```

Example interaction et control

What? Pas de popups !?!

ol.Overlay

```
var marker = new ol.Overlay({  
  map: map,  
  position: buriedTreasure,  
  element: document.createElement('X marks the spot')  
});
```

ol.Overlay

```
var el = document.getElementById('popup');

var popup = new ol.Overlay({
  map: map,
  element: el
});

map.on('click', function(evt) {
  $(el).popover({
    'placement': 'top',
    'html': true,
    'content': 'Roll your own popup!'
  });
  $(el).popover('show');
  popup.setPosition(evt.getCoordinate());
});
```

Example overlay

ol.animation

```
function fly(map, home, duration) {  
  var view = map.getView().getView2D();  
  var start = +new Date();  
  var pan = ol.animation.pan({  
    duration: duration,  
    source: view.getCenter(),  
    start: start  
  });  
  var bounce = ol.animation.bounce({  
    duration: duration,  
    resolution: 4 * view.getResolution(),  
    start: start  
  });  
  map.addPreRenderFunctions([pan, bounce]);  
  view.setCenter(home);  
}
```

Example animation

Vecteur



ol.layer.Vector

```
var vector = new ol.layer.Vector({  
  source: new ol.source.Vector({  
    projection: ol.proj.get('EPSG:4326')  
  })  
});
```

Vecteur avec Canvas 2D

Code pour l'interaction avec les vecteurs:

```
map.getFeatureInfo({
  pixel: evt.getPixel(),
  layers: [countries], // optional
  success: function(features) {
    $('#info').html('');
    for (var i = 0, ii = features.length; i < ii; ++i) {
      $('#info').append(features[i].get('name') + '<br>');
    }
  }
});
```

Exemple vecteur canvas

Un mot sur la compilation

Hein ? Oui oui.

Closure Compiler

- Renommage des variables
- Élimination du code non utilisé
- Applatissage des propriétés
- Dévirtualisation des méthodes
- "Inlining"

Closure Compiler Example

```
goog.provide('ANamespace.ASubNamespace.AClass');
ANamespace.ASubNamespace.AClass = function() {
  this.aProperty = 'prop1';
};
ANamespace.ASubNamespace.AClass.prototype.aMethod =
  function() {
    this.aProperty = 'change';
  };
var anInstance = new ANamespace.ASubNamespace.AClass();
anInstance.aMethod();
```

compiles to:

```
window.b=new function(){this.a="prop1"};window.b.a="change";
```

État actuel

Alpha 1 released

Limité, mais déjà utile

L'API change...

Vers 3.0.0 final

Plus d'alphas et betas à venir

Le renderer WebGL pour le vecteur

Stabilisation des API

Cesium Integration



Thanks!



Mis à disposition selon les termes de la license **Creative Commons Attribution-ShareAlike 3.0 Unported License**.