Master Class OL3 Documentation

Version 0.1

Éric Lemoine

05 June 2013

Table des matières

Ce document contient les énoncés et corrections des exercices de la *master class* OpenLayers 3 donnée aux Rencontres SIG-la-lettre 2013. Il est disponible en ligne à http ://erilem.net/master-class-ol3/exercices/. Une version PDF peut être téléchargée à http ://erilem.net/master-class-ol3/exercices/exercices.pdf.

La partie *Exercices élémentaires* contient des exercices relativement simples visant à se familiariser avec OpenLayers 3. Chaque exercice élémentaire est indépendant.

La partie *Développement d'une application* propose le développement d'une (modeste) application web cartographique de A à Z. Les données géographiques utilisées pour cette application sont issues de la plate-forme GéoBretagne. C'est le service WMS mis à disposition par GéoBretagne qui sera utilisé.

CHAPITRE 1

Exercices élémentaires

Tous les exercices de cette partie se font à partir de la page HTML suivante :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no, width=device-width">
    <title>Exercice élémentaire</title>
    <link rel="stylesheet" href="http://ol3js.org/en/master/build/ol.css" type="text/css">
    <style>
        #map {
          width: 600px;
          height: 400px;
        }
    </style>
  </head>
  <body>
      <div id="map"></div>
      <script src="http://ol3js.org/en/master/build/ol.js"></script>
      <script>
      var map = new ol.Map({
        target: 'map',
        view: new ol.View2D({
          zoom: 0,
          center: [0, 0]
        }),
        layers: [
          new ol.layer.TileLayer({
            source: new ol.source.OSM()
          })
        ]
      });
      </script>
  </body>
</html>
```

Note : Pensez à lire les **indices** !

1.1 Exercice élémentaire 1

Afficher une carte simple

1. Copier le code HTML donné ci-dessus et le coller dans un fichier du nom de votre choix, `exercice-elementaire-1.html` par exemple.
2. Placer ce fichier dans un répertoire rendu accessible par HTTP par le serveur web.
3. Ouvrir la page web dans un navigateur, en utilisant `http` comme protocole d'accès à la page.
4. Vérifier que l'exemple fonctionne correctement.
5. Utiliser les *interactions* et les *controls* de la carte.

(Pas de correction pour cet exercice.)

1.2 Exercice élémentaire 2

Choisir le renderer de la carte

OL3 peut utiliser trois technologies web différentes pour l'affichage de la carte : WebGL, Canvas, et DOM. Si rien n'est précisé par l'utilisateur la librairie OL3 choisit WebGL si le navigateur prend en charge WebGL. Sinon OL3 choisit Canvas, et enfin, si Canvas n'est pas pris en charge, c'est le *renderer* DOM qui est utilisé.

1. Copier le code HTML de base dans un nouveau fichier, `exercice-elementaire-2.html` par exemple.
2. Faire en sorte que le *renderer* Canvas soit utilisé par la carte. Il faut pour ceci positionner la propriété `renderer` dans l'objet d'options passé au constructeur `ol.Map`. Et la valeur pour cette propriété doit être `ol.RendererHint.CANVAS`.
3. Faire maintenant en sorte que ce soit le *renderer* DOM qui soit utilisé.
4. Utiliser maintenant la valeur `ol.RendererHints.createFromQueryData()` pour que le *renderer* puisse être spécifié dans l'URL par le paramètre `renderer`. Exemple : `exercice-elementaire-2.html?renderer=canvas`.

Correction exercice élémentaire 2

Indice : Pour vérifier quel *renderer* est utilisé vous pouvez effectuer un clic droit sur la carte et sélectionner "Inspecter l'élément". Si c'est le *renderer* DOM qui est utilisé l'élément inspecté sera une balise `<img>`, sinon ce sera une balise `<canvas>`. Cette technique ne permet pas de distinguer Canvas de WebGL.

1.3 Exercice élémentaire 3

Initialiser la vue de la carte

1. Copier le code HTML de base dans un nouveau fichier, `exercice-elementaire-3.html` par exemple.
2. Modifier l'objet d'options passés au constructeur `ol.View2D` pour que la vue soit initialisée avec un niveau de zoom égal à 17.
3. Modifier à nouveau l'objet d'options pour que la vue soit initialisée avec un centre égal à `[288074.8449901076, 6247982.515792289]`.
4. Modifier à nouveau l'objet d'options pour que la vue soit pivotée (rotation) d'un angle de 45 dans le sens des aiguilles d'une montre.

Correction exercice élémentaire 3

1.4 Exercice élémentaire 4

Changer la vue de la carte

1. Copier le code HTML de base dans un nouveau fichier, `exercice-elementaire-4.html` par exemple.
2. Ouvrir la nouvelle page HTML et ouvrir les outils de développement du navigateur. Vous pouvez utiliser la touche `F12` comme raccourci.
3. Saisir dans la console des outils de développement la commande JavaScript permettant d'obtenir le centre actuel de la vue.
4. Toujours dans la console, agir sur la vue pour changer son centre à `[288074.8449901076, 6247982.515792289]`.
5. Obtenir dans la console la résolution actuelle de la vue.
6. Agir sur la vue pour passer à une résolution 131072.0 fois plus petite que la résolution actuelle.
7. Agir sur la vue pour faire pivoter la carte de 45 degré à l'est.
8. Ajouter les lignes JavaScript saisies précédemment dans la console dans le code JavaScript de la page HTML, juste après la création de la carte. Recharger la page dans le navigateur et vérifier que la vue est correcte.

Indice :
– La variable `map` définie dans le code JavaScript de la page est globale. Elle est donc directement accessible dans la console.
– Pour obtenir l'objet *vue* (`ol.View2D`) avec lequel la carte a été configurée il faut utiliser `map.getView()`.
– Les objets de type `ol.View2D` fournissent des fonctions *getter* pour accéder aux états de la vue. Exemple : `view.getCenter()`.
– De la même façon ils fournissent des *setters* pour changer les états de la vue. Exemple : `view.setRotation(45)`.

Correction exercice élémentaire 4

1.5 Exercice élémentaire 5

Aller un peu plus loin avec la vue

1. Copier le code HTML de base dans un nouveau fichier, `exercice-elementaire-5.html` par exemple.
2. Modifier le code JavaScript de la page afin que l'objet `ol.View2D` soit créé à l'extérieur de la définition de l'objet des options passé à la carte. L'objet `ol.View2D` créé sera référencé par une variable nommé `view`. Et c'est cette référence qui devra être passé à la carte par l'intermédiaire de l'option `view`.
3. Après la création de la carte, agir sur la vue (référencée par la variable `view`) pour recentrer celle-ci sur l'étendue `[287716.5464200208, 288433.14356019435, 6247743.650078897, 6248221.38150568]`.
4. Changer le code pour que la vue n'ait pas d'état initial, et pour qu'elle ne soit rééllement définie que lorsque qu'elle est recentrée sur l'étendue spécifiée.
5. Tenter de comprendre pourquoi la fonction `fitExtent` a besoin des dimensions de la carte en pixels pour faire ce qui lui est demandé.

Indice :
– C'est la fonction `fitExtent` de `ol.View2D` qui doit être utilisée pour ça.
– La fonction `getSize` de `ol.Map` doit aussi être utilisée pour cette exercice.
– Pensez à jetter un oeil à la doc de l'API : http ://ol3js.org/en/master/apidoc/.

Correction exercice élémentaire 5

1.6 Exercice élémentaire 6

Utiliser une autre source de donnée de type OpenStreetMap

1. Copier le code HTML de base dans un nouveau fichier, `exercice-elementaire-6.html` par exemple.
2. Changer les options de la vue pour centrer la carte sur l'ENSG. (Options utilisées dans *Exercice élémentaire 3* par exemple.)
3. Passer au constructeur `ol.source.OSM` un objet d'options contenant une propriété `url` dont la valeur est `http://{a-c}.tile3.opencyclemap.org/landscape/{z}/{x}/{y}.png`. Voir http ://www.thunderforest.com/landscape/ pour obtenir des informations sur cette source de donnée. Et voir `OSMOptions` dans la doc de l'API pour connaître toutes les options qui peuvent être passées à `ol.source.OSM`.
4. Ajouter une attribution à la source pour que les conditions d'utilisation des tuiles OpenCycleMap soient respectées. Une attribution du type `<a href="http://www.opencyclemap.org">OpenCycleMap</a>` est en accord avec ces conditions.

Indice : Regarder l'exemple http ://ol3js.org/en/master/examples/localized-openstreetmap.html pour inspiration.

Correction exercice élémentaire 6

1.7 Exercice élémentaire 7

Changer les paramètres d'affichage d'une couche

1. Copier le code HTML de base dans un nouveau fichier, `exercice-elementaire-7.html` par exemple.
2. Dans l'objet d'options passé au constructor `ol.layer.TileLayer` ajouter des propriétés `opacity`, `hue`, `saturation`, `brightness`, et `contrast`.
3. Dans la console, récupérer une référence sur la couche et changer sa visibilité.

Indice :
– `ol.Map` fournit une fonction `getLayers`. Cette fonction retourne un object de type `ol.Collection`, qui lui même fournit des méthodes pour accéder aux différents objets de la collection. Voir la doc de l'API pour plus d'informations.

Correction exercice élémentaire 7

1.8 Exercice élémentaire 8

Manipuler les projections

1. Copier le code HTML de base dans un nouveau fichier, `exercice-elementaire-8.html` par exemple.
2. Dans une console, afficher le centre de la vue.
3. Le système de coordonnée (projection) de ce centre est "EPSG :3857" (connu sous le nom de Spherical Mercator). Convertir les coordonnées du centre de "EPSG :3857" à "EPSG :4326" (longitude/latitude WGS84).
4. Les coordonnées GPS de l'ENSG sont latitude : 48,8413379, longitude : 2,5878203. Modifier le code JavaScript de la page pour que la vue soit centrée sur ces coordonnées à l'état initial. Changer aussi le zoom à 17 pour un meilleur résultat.

Indice :
– La fonction `ol.proj.transform` est à utiliser pour transformer des coordonnées d'un système de coordonnée à un autre. Voir la doc de l'API.

Correction exercice élémentaire 8

1.9 Exercice élémentaire 9

Manipuler les controls

1. Copier le code HTML de base dans un nouveau fichier, `exercice-elementaire-9.html` par exemple.
2. Modifier le code JavaScript de la page de telle façon qu'une échelle graphique soit ajoutée à la carte.
3. Modifier la configuration du *control* pour que les unités de mesure anglo-saxonnes (inches, feets, miles) soient utilisées plutôt que les unités métriques.

Indice :
– C'est le *control* `ol.control.ScaleLine` qui permet d'ajouter une échelle graphique sur la carte.
– La fonction `ol.control.defaults` permet d'obtenir un tableau contenant les *controls* par défaut, et d'autres choisis de manière spécifique.

Correction exercice élémentaire 9

1.10 Exercice élémentaire 10

Manipuler les interactions

1. Copier le code HTML de base dans un nouveau fichier, `exercice-elementaire-10.html` par exemple.
2. Modifier le code JavaScript de la page pour ajouter une interaction de type `ol.interaction.DragRotateAndZoom` à la carte. (C'est la touche `SHIFT` qui active cette interaction.)

Correction exercice élémentaire 10

CHAPITRE 2

Corrections des exercices élémentaires

2.1 Correction exercice élémentaire 1

Exercice élémentaire 1

Aucune correction pour cet exercice.

2.2 Correction exercice élémentaire 2

Exercice élémentaire 2

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no, width=device-width">
    <title>Exercice élémentaire</title>
    <link rel="stylesheet" href="http://ol3js.org/en/master/build/ol.css" type="text/css">
    <style>
        #map {
          width: 600px;
          height: 400px;
        }
    </style>
  </head>
  <body>
      <div id="map"></div>
      <script src="http://ol3js.org/en/master/build/ol.js"></script>
      <script>
      var map = new ol.Map({
        target: 'map',
        view: new ol.View2D({
          zoom: 0,
          center: [0, 0]
        }),
        layers: [
          new ol.layer.TileLayer({
            source: new ol.source.OSM()
          })
        ],
        renderer: ol.RendererHint.CANVAS
```

```
      });
      </script>
  </body>
</html>
```

Pour que *renderer* DOM soit utilisé il faut passer la valeur `ol.RendererHint.DOM` pour l'option `renderer`. Et pour que le renderer puisse être spécifié dans l'URL c'est la valeur retournée par la fonction `ol.RendererHints.createFromQueryData` qu'il faut passer.

2.3 Correction exercice élémentaire 3

Exercice élémentaire 3

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no, width=device-width">
    <title>Exercice élémentaire</title>
    <link rel="stylesheet" href="http://ol3js.org/en/master/build/ol.css" type="text/css">
    <style>
        #map {
          width: 600px;
          height: 400px;
        }
    </style>
  </head>
  <body>
      <div id="map"></div>
      <script src="http://ol3js.org/en/master/build/ol.js"></script>
      <script>
      var map = new ol.Map({
        target: 'map',
        view: new ol.View2D({
          zoom: 17,
          center: [288074.8449901076, 6247982.515792289],
          rotation: 45
        }),
        layers: [
          new ol.layer.TileLayer({
            source: new ol.source.OSM()
          })
        ]
      });
      </script>
  </body>
</html>
```

2.4 Correction exercice élémentaire 4

Exercice élémentaire 4

```
> view = map.getView()
> view.getCenter()
> view.setCenter([288074.8449901076, 6247982.515792289])
> view.getResolution()
> view.setResolution(view.getResolution() / 131072.0)
> view.setRotation(45)
```

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no, width=device-width">
    <title>Exercice élémentaire</title>
    <link rel="stylesheet" href="http://ol3js.org/en/master/build/ol.css" type="text/css">
    <style>
        #map {
          width: 600px;
          height: 400px;
        }
    </style>
  </head>
  <body>
      <div id="map"></div>
      <script src="http://ol3js.org/en/master/build/ol.js"></script>
      <script>
      var map = new ol.Map({
        target: 'map',
        view: new ol.View2D({
          zoom: 0,
          center: [0, 0]
        }),
        layers: [
          new ol.layer.TileLayer({
            source: new ol.source.OSM()
          })
        ]
      });
      view = map.getView()
      view.setCenter([288074.8449901076, 6247982.515792289])
      view.setResolution(view.getResolution() / 131072.0)
      view.setRotation(45)
      </script>
  </body>
</html>
```

2.5 Correction exercice élémentaire 5

Exercice élémentaire 5

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no, width=device-width">
    <title>Exercice élémentaire</title>
    <link rel="stylesheet" href="http://ol3js.org/en/master/build/ol.css" type="text/css">
```

```
    <style>
        #map {
          width: 600px;
          height: 400px;
        }
    </style>
  </head>
  <body>
      <div id="map"></div>
      <script src="http://ol3js.org/en/master/build/ol.js"></script>
      <script>
      var view = new ol.View2D();
      var map = new ol.Map({
        target: 'map',
        view: view,
        layers: [
          new ol.layer.TileLayer({
            source: new ol.source.OSM()
          })
        ]
      });

      var extent = [287716.5464200208, 288433.14356019435,
                    6247743.650078897, 6248221.38150568];
      view.fitExtent(extent, map.getSize());
      </script>
  </body>
</html>
```

La fonction `fitExtent` a besoin de connaître les dimensions de la carte en pixels pour être capable de recentrer la vue sur une étendue donnée. La vue est en effet complétement déterminée par un centre, une résolution et une rotation, et elle n'a aucune connaissance des dimensions du rectangle d'affichange dans la page. Dans ces conditions, si on ne lui passe pas des dimensions (largeur et hauteur en pixels), il ne lui est pas possible de recentrer la vue sur une étendue géographique.

2.6 Correction exercice élémentaire 6

Exercice élémentaire 6

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no, width=device-width">
    <title>Exercice élémentaire</title>
    <link rel="stylesheet" href="http://ol3js.org/en/master/build/ol.css" type="text/css">
    <style>
        #map {
          width: 600px;
          height: 400px;
        }
    </style>
  </head>
  <body>
      <div id="map"></div>
      <script src="http://ol3js.org/en/master/build/ol.js"></script>
```

```
      <script>
      var map = new ol.Map({
        target: 'map',
        view: new ol.View2D({
          zoom: 17,
          center: [288074.8449901076, 6247982.515792289]
        }),
        layers: [
          new ol.layer.TileLayer({
            source: new ol.source.OSM({
              attributions: [
                new ol.Attribution(
                  '<a href="http://www.opencyclemap.org/">OpenCycleMap</a>'),
                ol.source.OSM.DATA_ATTRIBUTION
              ],
              url: 'http://{a-c}.tile3.opencyclemap.org/landscape/{z}/{x}/{y}.png',
            })
          })
        ],
        renderer: ol.RendererHint.CANVAS
      });
      </script>
  </body>
</html>
```

2.7 Correction exercice élémentaire 7

Exercice élémentaire 7

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no, width=device-width">
    <title>Exercice élémentaire</title>
    <link rel="stylesheet" href="http://ol3js.org/en/master/build/ol.css" type="text/css">
    <style>
        #map {
          width: 600px;
          height: 400px;
        }
    </style>
  </head>
  <body>
      <div id="map"></div>
      <script src="http://ol3js.org/en/master/build/ol.js"></script>
      <script>
      var map = new ol.Map({
        target: 'map',
        view: new ol.View2D({
          zoom: 0,
          center: [0, 0]
        }),
        layers: [
          new ol.layer.TileLayer({
            source: new ol.source.OSM(),
```

```
            brightness: 0.1,
            contrast: 1.625,
            hue: -1.25,
            opacity: 0.5,
            saturation: 5
          })
        ]
      });
      </script>
  </body>
</html>
```

Il est important de noter ici que les paramètres d'affichage (`opacity`, etc.) sont des paramètres de l'objet `Layer`, et non de l'objet `Source`. La source ne concerne que les données elles mêmes, et pas la façon dont ces données sont affichées.

Pour changer la visibilité de la couche dans la console :

```
> layerCollection = map.getLayers()
> osmLayer = layerCollection.getAt(0)
> osmLayer.setVisible(!osmLayer.getVisible())
> osmLayer.setVisible(!osmLayer.getVisible())
```

`osmLayer.getVisible()` retourne un booléen (`true` ou `false`). `osmLayer.setVisible(!osmLayer.getVisible())` permet donc d'inverser la visibilité.

2.8 Correction exercice élémentaire 8

Exercice élémentaire 8

Dans la console :

```
> ol.proj.transform(map.getView().getCenter(), 'EPSG:3857', 'EPSG:4326')
```

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no, width=device-width">
    <title>Exercice élémentaire</title>
    <link rel="stylesheet" href="http://ol3js.org/en/master/build/ol.css" type="text/css">
    <style>
        #map {
          width: 600px;
          height: 400px;
        }
    </style>
  </head>
  <body>
      <div id="map"></div>
      <script src="http://ol3js.org/en/master/build/ol.js"></script>
      <script>
      var center = ol.proj.transform(
        [2.5878203, 48.8413379], 'EPSG:4326', 'EPSG:3857');
      var map = new ol.Map({
        target: 'map',
        view: new ol.View2D({
```

```
          zoom: 17,
          center: center
        }),
        layers: [
          new ol.layer.TileLayer({
            source: new ol.source.OSM()
          })
        ]
      });
      </script>
  </body>
</html>
```

2.9 Correction exercice élémentaire 9

Exercice élémentaire 9

Si rien concernant les *controls* n'est spécifié dans l'objet d'options passé au constructeur `ol.Map` alors `ol.Map` crèe trois *controls* (les *controls* par défaut) : `ol.control.Attribution`, `ol.control.Logo`, et `ol.control.Zoom`.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no, width=device-width">
    <title>Exercice élémentaire</title>
    <link rel="stylesheet" href="http://ol3js.org/en/master/build/ol.css" type="text/css">
    <style>
        #map {
          width: 600px;
          height: 400px;
        }
    </style>
  </head>
  <body>
      <div id="map"></div>
      <script src="http://ol3js.org/en/master/build/ol.js"></script>
      <script>
      var map = new ol.Map({
        target: 'map',
        view: new ol.View2D({
          zoom: 0,
          center: [0, 0]
        }),
        layers: [
          new ol.layer.TileLayer({
            source: new ol.source.OSM()
          })
        ],
        controls: ol.control.defaults({}, [
          new ol.control.ScaleLine({
            units: ol.control.ScaleLineUnits.IMPERIAL
          })
        ])
      });
```

```
      </script>
  </body>
</html>
```

2.10 Correction exercice élémentaire 10

Exercice élémentaire 10

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no, width=device-width">
    <title>Exercice élémentaire</title>
    <link rel="stylesheet" href="http://ol3js.org/en/master/build/ol.css" type="text/css">
    <style>
        #map {
          width: 600px;
          height: 400px;
        }
    </style>
  </head>
  <body>
      <div id="map"></div>
      <script src="http://ol3js.org/en/master/build/ol.js"></script>
      <script>
      var map = new ol.Map({
        target: 'map',
        view: new ol.View2D({
          zoom: 0,
          center: [0, 0]
        }),
        layers: [
          new ol.layer.TileLayer({
            source: new ol.source.OSM()
          })
        ],
        interactions: ol.interaction.defaults({}, [
          new ol.interaction.DragRotateAndZoom()
        ])
      });
      </script>
  </body>
</html>
```

CHAPITRE 3

Développement d'une application

Vous allez ici créer une carte occupant tout l'écran et compatible avec des appareils mobile. Cette carte affichera des données issues de la plate-forme GéoBretagne, selon la projection Lambert93 (projection officielle pour les cartes de France métropolitaine).

3.1 Étape 1

Créer une page web avec une carte occupant tout l'écran

Votre objectif à cette étape est d'obtenir une carte quelconque occupant tout l'écran. La carte sera la plus simple possible : projection globale EPSG :3857 (projection par défaut dans OL3), et une couche OSM.

Vous pouvez, si vous le souhaitez, d'ores-et-déja centrer la carte sur la Bretagne. Voici une étendue dans le système de coordonnée EPSG :3857 pour la Bretagne : `[-40679590.16384748, -40093165.28284361, 5967723.758526416, 6264605.176386041]`.

Indice :

1. Pour créer une carte occupant tout l'écran vous pouvez vous inspirez de l'exemple http ://ol3js.org/en/master/examples/mobile-full-screen.html.
2. Pour créer une carte OSM la plus simple possible vous pouvez vous inspirez de l'exemple http ://ol3js.org/en/master/examples/simple.html.

Correction étape 1

3.2 Étape 2

Charger proj4js dans la page et ajouter la projection Lambert93

Cette étape consiste à ajouter la bibliothèque proj4js et à la configurer pour qu'elle puisse fonctionner avec la projection Lambert93 (EPSG :2154).

Pour ajouter la bibliothèque proj4js ajouter à la page une balise `<script>` dont l'attribut `src` est positionné à `http://cdnjs.cloudflare.com/ajax/libs/proj4js/1.1.0/proj4js-compressed.js`.

Et pour que proj4js puisse être utilisé pour la projection Lamber93 il faut déclarer cette projection auprès de proj4js. Comme ceci :

```
Proj4js.defs['EPSG:2154'] = '+proj=lcc +lat_1=49 +lat_2=44 +lat_0=46.5 ' +
    '+lon_0=3 +x_0=700000 +y_0=6600000 +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 '+
    '+units=m +no_defs';
```

Cette ligne doit donc être ajoutée dans le code JavaScript de la page, avant la création de la carte.

Le chargement de proj4js et la déclaration de la projection EPSG :2154 ne devrait avoir aucun impact sur la carte créée à l'étape précédente. Recharger la page dans le navigateur pour vous en assurer.

Correction étape 2

3.3 Étape 3

Ajouter la couche "carte" de la plate-forme GéoBretagne

À cette étape vous aller modifier la carte pour que celle-ci la couche WMS "carte" de la plate-forme GéoBretagne, et que l'affichage se fasse selon la projection "EPSG :2154". Vous allez pour ceci modifier la configuration de la vue (`View2D`), et remplacer la source `ol.source.OSM` par une source `ol.source.TiledWMS` configurée correctement.

1. Le service WMS de la plate-forme GéoBretagne que nous utilisons ici prend en charge l'extension WMS-C pour fournir des images (tuiles) selon une grille donnée. Cette grille fixe les résolutions d'affichage et l'étendue des tuiles.

   Dans un but de préparation créer dans le code JavaScript de la page, après la définition de la projection EPSG :2154, deux tableaux JavaScript : un contenant les résolutions prises en charge par le service WMS-C, et un contenant l'étendue des tuiles.

   ```
   Proj4js.defs['EPSG:2154'] = '+proj=lcc +lat_1=49 +lat_2=44 +lat_0=46.5 ' +
       '+lon_0=3 +x_0=700000 +y_0=6600000 +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 '+
       '+units=m +no_defs';

   var resolutions = [156543.0339, 78271.51695, 39135.758475,
       19567.8792375, 9783.93961875, 4891.969809375,
       2445.9849046875, 1222.99245234375, 611.4962261718748,
       305.7481130859374, 152.87405654296887, 76.43702827148444,
       38.21851413574208, 19.10925706787104, 9.55462853393552,
       4.77731426696776, 2.38865713348388, 1.1943285667420798,
       0.5971642833710399, 0.29858214168551994, 0.14929107084275997,
       0.07464553542123999];

   var extent = [-357823.2365, 1313632.3628, 6037008.6939, 7230727.3772];
   ```

   Le service de la plate-forme GéoBretagne utilisé ici est décrit à cette page : http ://tile.geobretagne.fr/gwc02/home. Les résolutions et l'étendue sont définies dans le document WMS GetCapabilities à cette URL : http ://tile.geobretagne.fr/gwc02/service/wms ?SERVICE=WMS&VERSION=1.1.1&REQUEST=getcapabilities&TILED=true

2. Il s'agit maintenant de configurer la vue (`View2D`) pour que la carte soit représentée selon la projection EPSG :2154 et que les résolutions d'affichage soient celles prises en charge par le service WMS-C.

   ```
   var view = new ol.View2D({
     projection: 'EPSG:2154',
     resolutions: resolutions
   });
   ```

3. Vous allez maintenant créer un objet de type `ol.layer.TileLayer` permettant d'afficher les tuiles de la couche WMS-C "carte" du service http ://tile.geobretagne.fr/gwc02/service/wms. Voici le code à écrire pour créer cet objet :

   ```
   var carteLayer = new ol.layer.TileLayer({
     source: new ol.source.TiledWMS({
       url: 'http://tile.geobretagne.fr/gwc02/service/wms',
   ```

```
    params: {
      'LAYERS': 'carte',
      'VERSION': '1.1.1'
    },
    tileGrid: new ol.tilegrid.TileGrid({
      resolutions: resolutions,
      origin: [-357823.2365, 1313632.3628]
    }),
    extent: extent
  })
});
```

Ce bloc de code peut être ajouté après la création de la vue. Prenez un peu de temps pour le comprendre. (Et poser des questions !)

4. Maintenant que la vue et la couche sont créées il faut adapter l'objet d'options passé à `ol.Map` :

```
var map = new ol.Map({
  target: 'map',
  renderer: ol.RendererHint.CANVAS,
  view: view,
  layers: [carteLayer]
});
```

À noter que le *renderer* Canvas est utilisé ici. Dans ce cas précis, à cause d'une limitation au niveau du service WMS (en-tête CORS non positionné par le serveur), il n'est pas possible d'utiliser le *renderer* WebGL.

5. La dernière étape consiste à centrer la vue sur la Bretagne :

```
var initialExtent = [117427.53782167949, 410639.9782710938,
    6731783.8687657695, 6880224.577668993];
view.fitExtent(initialExtent, map.getSize());
```

Correction étape 3

3.4 Étape 4

Ajouter une deuxième couche tuilée

Ajoutez à cette étape une deuxième couche WMS tuilée, exactement du même type que la première mais reposant sur la couche WMS "satellite".

Les deux couches étant opaques vous observerez que la couche "satellite" recouvre complètement la couche "carte".

Correction étape 4

3.5 Étape 5

Ajouter un outil de sélection de couche

À cette étape vous allez ajouter un outil de type `<select>` permettant de sélectionner, parmi "carte" et "satellite", quelle couche est visible.

Cette étape nécessite d'ajouter des éléments HTML dans la page, et de manipuler ces éléments en JavaScript. Pour faciliter la manipulation d'éléments DOM en JavaScript nous allons utiliser la célèbre bibliothèque jQuery.

Tout d'abord, pour charger jQuery dans la page ajouter la balise `<script>` suivante :

```
<script src="http://code.jquery.com/jquery-2.0.0.min.js"></script>
```

Ensuite, ajouter une balise `<select>` dans le code HTML, à l'intérieur du `div` de la carte :

```
<div id="map" class="map">
  <select id="background-selector">
    <option value="carte" selected>Carte</option>
    <option value="satellite">Satellite</option>
  </select>
</div>
```

Afin de placer le sélecteur dans la carte, un peu de CSS est nécessaire :

```
#background-selector {
  position: absolute;
  top: 2px;
  right: 2px;
  z-index: 100;
}
```

Il ne reste alors plus qu'à écrire le code JavaScript permettant de changer la visibilité des couches quand la sélection change :

```
$('#background-selector').change(function() {
  var selected = $(this).find(':selected').val();
  if (selected == 'carte') {
    carteLayer.setVisible(true);
    satelliteLayer.setVisible(false);
  } else if (selected == 'satellite') {
    carteLayer.setVisible(false);
    satelliteLayer.setVisible(true);
  }
});
$('#background-selector').trigger('change');
```

À titre d'exercice, déterminer pourquoi cette dernière ligne est nécessaire.

Toujours à titre d'exerice vous pouve écrire ce code JavaScript sans utiliser jQuery. Il est en effet un peu idiot de charger jQuery pour si peu.

Correction étape 5

3.6 Étape 6

Ajouter une couche image

Pour cette étape vous devez ajouter une couche non-tuilée (de type image) pour la couche WMS "paimpol_zone_plu_ccpg" du service WMS http ://geobretagne.fr/geoserver/id22/wms.

Pour bien visualiser la couche vous pouvez changer la valeur de `initialExtent` pour se rapprocher de la commune Paimpol :

```
var initialExtent = [246462.7961724792, 264788.57370056753,
    6864884.621557758, 6874162.16586421];
view.fitExtent(initialExtent, map.getSize());
```

Indice :

1. Vous utiliserez ici `ol.layer.ImageLayer` plutôt que `ol.layer.TileLayer`.
2. Et vous utiliserez `ol.source.SingleImageWMS` plutôt que `ol.source.TiledWMS`.

Correction étape 6

3.7 Étape 7

Ajouter un bouton de géo-localisation

Il s'agit à cette étape d'ajouter à la carte un bouton permettant à l'utilisateur de centrer la carte sur sa position actuelle.

La classe `ol.Geolocation` d'OpenLayers doit être utilisée pour ça.

En exercice *bonus* : animer le recentrage de la carte pour chaque géo-localisation. (La solution de cette exercice bonus n'est pas donnée dans la correction.)

Correction étape 7

CHAPITRE 4

Correction application

4.1 Correction étape 1

Étape 1

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="chrome=1">
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no, width=device-width">
    <link rel="stylesheet" href="http://ol3js.org/en/master/build/ol.css" type="text/css">
    <title>Application</title>
    <style type="text/css">
      html, body, .map {
        margin: 0;
        padding: 0;
        width: 100%;
        height: 100%;
      }
    </style>
  </head>
  <body>
    <div id="map" class="map"></div>

    <script src="http://ol3js.org/en/master/build/ol.js"></script>
    <script>
    var view = new ol.View2D();
    var map = new ol.Map({
      target: 'map',
      view: view,
      layers: [
        new ol.layer.TileLayer({
          source: new ol.source.OSM()
        })
      ]
    });
    var extent = [-40679590.16384748, -40093165.28284361,
        5967723.758526416, 6264605.176386041];
    view.fitExtent(extent, map.getSize());
    </script>
```

```
  </body>
</html>
```

4.2 Correction étape 2

Étape 2

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="chrome=1">
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no, width=device-width">
    <link rel="stylesheet" href="http://ol3js.org/en/master/build/ol.css" type="text/css">
    <title>Application</title>
    <style type="text/css">
      html, body, .map {
        margin: 0;
        padding: 0;
        width: 100%;
        height: 100%;
      }
    </style>
  </head>
  <body>
    <div id="map" class="map"></div>

    <script src="http://cdnjs.cloudflare.com/ajax/libs/proj4js/1.1.0/proj4js-compressed.js"></script>
    <script src="http://ol3js.org/en/master/build/ol.js"></script>
    <script>
    Proj4js.defs['EPSG:2154'] = '+proj=lcc +lat_1=49 +lat_2=44 +lat_0=46.5 ' +
        '+lon_0=3 +x_0=700000 +y_0=6600000 +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 '+
        '+units=m +no_defs';

    var view = new ol.View2D();
    var map = new ol.Map({
      target: 'map',
      view: view,
      layers: [
        new ol.layer.TileLayer({
          source: new ol.source.OSM()
        })
      ]
    });
    var extent = [-40679590.16384748, -40093165.28284361,
        5967723.758526416, 6264605.176386041];
    view.fitExtent(extent, map.getSize());
    </script>
  </body>
</html>
```

4.3 Correction étape 3

Étape 3

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="chrome=1">
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no, width=device-width">
    <link rel="stylesheet" href="http://ol3js.org/en/master/build/ol.css" type="text/css">
    <title>Application</title>
    <style type="text/css">
      html, body, .map {
        margin: 0;
        padding: 0;
        width: 100%;
        height: 100%;
      }
    </style>
  </head>
  <body>
    <div id="map" class="map"></div>

    <script src="http://cdnjs.cloudflare.com/ajax/libs/proj4js/1.1.0/proj4js-compressed.js"></script>
    <script src="http://ol3js.org/en/master/build/ol.js"></script>
    <script>
    Proj4js.defs['EPSG:2154'] = '+proj=lcc +lat_1=49 +lat_2=44 +lat_0=46.5 ' +
        '+lon_0=3 +x_0=700000 +y_0=6600000 +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 '+
        '+units=m +no_defs';

    var resolutions = [156543.0339, 78271.51695, 39135.758475,
        19567.8792375, 9783.93961875, 4891.969809375,
        2445.9849046875, 1222.99245234375, 611.4962261718748,
        305.7481130859374, 152.87405654296887, 76.43702827148444,
        38.21851413574208, 19.10925706787104, 9.55462853393552,
        4.77731426696776, 2.38865713348388, 1.1943285667420798,
        0.5971642833710399, 0.29858214168551994, 0.14929107084275997,
        0.07464553542123999];
    var extent = [-357823.2365, 1313632.3628, 6037008.6939, 7230727.3772];

    var view = new ol.View2D({
      projection: 'EPSG:2154',
      resolutions: resolutions
    });

    var carteLayer = new ol.layer.TileLayer({
      source: new ol.source.TiledWMS({
        url: 'http://tile.geobretagne.fr/gwc02/service/wms',
        params: {
          'LAYERS': 'carte',
          'VERSION': '1.1.1'
        },
        tileGrid: new ol.tilegrid.TileGrid({
          resolutions: resolutions,
          origin: [-357823.2365, 1313632.3628]
        }),
        extent: extent
```

```
      })
    });

    var map = new ol.Map({
      target: 'map',
      renderer: ol.RendererHint.CANVAS,
      view: view,
      layers: [carteLayer]
    });

    var initialExtent = [117427.53782167949, 410639.9782710938,
        6731783.8687657695, 6880224.577668993];
    view.fitExtent(initialExtent, map.getSize());
    </script>
  </body>
</html>
```

4.4 Correction étape 4

Étape 4

Pour créer la deuxième couche :

```
var satelliteLayer = new ol.layer.TileLayer({
  source: new ol.source.TiledWMS({
    url: 'http://tile.geobretagne.fr/gwc02/service/wms',
    params: {
      'LAYERS': 'satellite',
      'VERSION': '1.1.1'
    },
    tileGrid: new ol.tilegrid.TileGrid({
      resolutions: resolutions,
      origin: [-357823.2365, 1313632.3628]
    }),
    extent: extent
  })
});
```

La nouvelle configuration de la carte :

```
var map = new ol.Map({
  target: 'map',
  renderer: ol.RendererHint.CANVAS,
  view: view,
  layers: [carteLayer, satelliteLayer]
});
```

4.5 Correction étape 5

Étape 5

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="chrome=1">
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no, width=device-width">
    <link rel="stylesheet" href="http://ol3js.org/en/master/build/ol.css" type="text/css">
    <title>Application</title>
    <style type="text/css">
      html, body, .map {
        margin: 0;
        padding: 0;
        width: 100%;
        height: 100%;
      }
      #background-selector {
        position: absolute;
        top: 2px;
        right: 2px;
        z-index: 100;
      }
    </style>
  </head>
  <body>
    <div id="map" class="map">
      <select id="background-selector">
        <option value="carte" selected>Carte</option>
        <option value="satellite">Satellite</option>
      </select>
    </div>

    <script src="http://code.jquery.com/jquery-2.0.0.min.js"></script>
    <script src="http://cdnjs.cloudflare.com/ajax/libs/proj4js/1.1.0/proj4js-compressed.js"></script>
    <script src="http://ol3js.org/en/master/build/ol.js"></script>
    <script>
    Proj4js.defs['EPSG:2154'] = '+proj=lcc +lat_1=49 +lat_2=44 +lat_0=46.5 ' +
        '+lon_0=3 +x_0=700000 +y_0=6600000 +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 '+
        '+units=m +no_defs';

    var resolutions = [156543.0339, 78271.51695, 39135.758475,
        19567.8792375, 9783.93961875, 4891.969809375,
        2445.9849046875, 1222.99245234375, 611.4962261718748,
        305.7481130859374, 152.87405654296887, 76.43702827148444,
        38.21851413574208, 19.10925706787104, 9.55462853393552,
        4.77731426696776, 2.38865713348388, 1.1943285667420798,
        0.5971642833710399, 0.29858214168551994, 0.14929107084275997,
        0.07464553542123999];
    var extent = [-357823.2365, 1313632.3628, 6037008.6939, 7230727.3772];

    var view = new ol.View2D({
      projection: 'EPSG:2154',
      resolutions: resolutions
    });

    var carteLayer = new ol.layer.TileLayer({
      source: new ol.source.TiledWMS({
        url: 'http://tile.geobretagne.fr/gwc02/service/wms',
        params: {
```

```
        'LAYERS': 'carte',
        'VERSION': '1.1.1'
      },
      tileGrid: new ol.tilegrid.TileGrid({
        resolutions: resolutions,
        origin: [-357823.2365, 1313632.3628]
      }),
      extent: extent
    })
  });

  var satelliteLayer = new ol.layer.TileLayer({
    source: new ol.source.TiledWMS({
      url: 'http://tile.geobretagne.fr/gwc02/service/wms',
      params: {
        'LAYERS': 'satellite',
        'VERSION': '1.1.1'
      },
      tileGrid: new ol.tilegrid.TileGrid({
        resolutions: resolutions,
        origin: [-357823.2365, 1313632.3628]
      }),
      extent: extent
    })
  });


  var map = new ol.Map({
    target: 'map',
    renderer: ol.RendererHint.CANVAS,
    view: view,
    layers: [carteLayer, satelliteLayer]
  });

  var initialExtent = [117427.53782167949, 410639.9782710938,
      6731783.8687657695, 6880224.577668993];
  view.fitExtent(initialExtent, map.getSize());

  $('#background-selector').change(function() {
    var selected = $(this).find(':selected').val();
    if (selected == 'carte') {
      carteLayer.setVisible(true);
      satelliteLayer.setVisible(false);
    } else if (selected == 'satellite') {
      carteLayer.setVisible(false);
      satelliteLayer.setVisible(true);
    }
  });
  $('#background-selector').trigger('change');

  </script>
 </body>
</html>
```

4.6 Correction étape 6

Étape 6

Le code pour créer la couche “paimpol_zone_plu_ccpg” :

```
var paimpollayer = new ol.layer.imagelayer({
  source: new ol.source.singleimagewms({
    url: 'http://geobretagne.fr/geoserver/id22/wms',
    params: {
      'layers': 'paimpol_zone_plu_ccpg',
      'version': '1.1.1',
      'transparent': 'true'
    },
    extent: extent
  })
});
```

La nouvelle configuration de la carte, ainsi que le nouvel appel à `fitExtent` pour centrer la vue sur Paimpol :

```
var map = new ol.Map({
  target: 'map',
  renderer: ol.RendererHint.CANVAS,
  view: view,
  layers: [carteLayer, satelliteLayer, paimpolLayer]
});

var initialExtent = [246462.7961724792, 264788.57370056753,
    6864884.621557758, 6874162.16586421];
view.fitExtent(initialExtent, map.getSize());
```

4.7 Correction étape 7

Étape 7

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="chrome=1">
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no, width=device-width">
    <link rel="stylesheet" href="http://ol3js.org/en/master/build/ol.css" type="text/css">
    <title>Application</title>
    <style type="text/css">
      html, body, .map {
        margin: 0;
        padding: 0;
        width: 100%;
        height: 100%;
      }
      #background-selector {
        position: absolute;
        top: 2px;
        right: 2px;
        z-index: 100;
```

```
    }
    #geolocation {
      position: absolute;
      top: 2px;
      right: 76px;
      z-index: 100;
    }
  </style>
</head>
<body>
  <div id="map" class="map">
    <select id="background-selector">
      <option value="carte" selected>Carte</option>
      <option value="satellite">Satellite</option>
    </select>
    <button id="geolocation">Localise moi!</button>
  </div>

  <script src="http://code.jquery.com/jquery-2.0.0.min.js"></script>
  <script src="http://cdnjs.cloudflare.com/ajax/libs/proj4js/1.1.0/proj4js-compressed.js"></script>
  <script src="http://ol3js.org/en/master/build/ol.js"></script>
  <script>
  Proj4js.defs['EPSG:2154'] = '+proj=lcc +lat_1=49 +lat_2=44 +lat_0=46.5 ' +
      '+lon_0=3 +x_0=700000 +y_0=6600000 +ellps=GRS80 +towgs84=0,0,0,0,0,0,0 '+
      '+units=m +no_defs';

  var resolutions = [156543.0339, 78271.51695, 39135.758475,
      19567.8792375, 9783.93961875, 4891.969809375,
      2445.9849046875, 1222.99245234375, 611.4962261718748,
      305.7481130859374, 152.87405654296887, 76.43702827148444,
      38.21851413574208, 19.10925706787104, 9.55462853393552,
      4.77731426696776, 2.38865713348388, 1.1943285667420798,
      0.5971642833710399, 0.29858214168551994, 0.14929107084275997,
      0.07464553542123999];
  var extent = [-357823.2365, 1313632.3628, 6037008.6939, 7230727.3772];

  var view = new ol.View2D({
    projection: 'EPSG:2154',
    resolutions: resolutions
  });

  var carteLayer = new ol.layer.TileLayer({
    source: new ol.source.TiledWMS({
      url: 'http://tile.geobretagne.fr/gwc02/service/wms',
      params: {
        'LAYERS': 'carte',
        'VERSION': '1.1.1'
      },
      tileGrid: new ol.tilegrid.TileGrid({
        resolutions: resolutions,
        origin: [-357823.2365, 1313632.3628]
      }),
      extent: extent
    })
  });

  var satelliteLayer = new ol.layer.TileLayer({
    source: new ol.source.TiledWMS({
```

```
    url: 'http://tile.geobretagne.fr/gwc02/service/wms',
    params: {
      'LAYERS': 'satellite',
      'VERSION': '1.1.1'
    },
    tileGrid: new ol.tilegrid.TileGrid({
      resolutions: resolutions,
      origin: [-357823.2365, 1313632.3628]
    }),
    extent: extent
  })
});

var paimpolLayer = new ol.layer.ImageLayer({
  source: new ol.source.SingleImageWMS({
    url: 'http://geobretagne.fr/geoserver/id22/wms',
    params: {
      'LAYERS': 'paimpol_zone_plu_ccpg',
      'VERSION': '1.1.1',
      'TRANSPARENT': 'TRUE'
    },
    extent: extent
  })
});


var map = new ol.Map({
  target: 'map',
  renderer: ol.RendererHint.CANVAS,
  view: view,
  layers: [carteLayer, satelliteLayer, paimpolLayer]
});

var initialExtent = [246462.7961724792, 264788.57370056753,
    6864884.621557758, 6874162.16586421];
view.fitExtent(initialExtent, map.getSize());


$('#background-selector').change(function() {
  var selected = $(this).find(':selected').val();
  if (selected == 'carte') {
    carteLayer.setVisible(true);
    satelliteLayer.setVisible(false);
  } else if (selected == 'satellite') {
    carteLayer.setVisible(false);
    satelliteLayer.setVisible(true);
  }
});
$('#background-selector').trigger('change');

var geoloc = new ol.Geolocation({projection: 'EPSG:2154'});
geoloc.on('change:position', function() {
    view.setCenter(geoloc.getPosition());
    geoloc.setTracking(false);
});
$('#geolocation').click(function() {
  geoloc.setTracking(true);
});
```

```
    </script>
  </body>
</html>
```