OpenLayers

Contrôles

Les contrôles sont des classes OpenLayers qui affectent l'état de la carte ou affichent des informations supplémentaires pour l'utilisateur. Les contrôles sont l'interface primaire pour les interactions cartographique.

Contrôleurs par défaut

Les contrôles suivant sont les contrôles par défaut sur la carte :

- *ArgParser*
- *Attribution*
- *Navigation*
- *PanZoom*

Panels

Les panels de contrôles permettent de réunir une collections de plusieurs contrôles, comme il arrive souvent dans les applications.

Style des panels

Les panels sont stylés par CSS. "ItemActive" et "ItemInactive" sont ajoutés au displayClass du contrôleur.

Tous les contrôles ont une propriété 'displayClass' écrasable qui correspond à leur nom de classe CSS de base. Ce nom est calculé en changeant le nom de la classe en supprimant tous les caractères '.' et changeant "OpenLayers" par "ol". Donc OpenLayers.Control.ZoomBox est changé en olControlZoomBox.

Les objets du Panel sont stylés en combinant le style du Panel avec le style d'un contrôleur à

l'intérieur. En utilisant le Panel NavToolbar comme exemple :

```
.olControlNavToolbar div {
  display:block;
  width:  28px;
  height: 28px;
  top: 300px;
  left: 6px;
  position: relative;
}
.olControlNavToolbar .olControlNavigationItemActive {
  background-image: url("img/panning-hand-on.png");
  background-repeat: no-repeat;
}
.olControlNavToolbar .olControlNavigationItemInactive {
  background-image: url("img/panning-hand-off.png");
  background-repeat: no-repeat;
}
```

Voici ce que nous disons :

- les éléments Div affichés dans la barre d'outils ont 28 px de largeur et 28 px de hauteur. Le haut du div doit être à 300px, le côté gauche à 6px.
- Puis, pour le contrôleur, nous fournissons deux images d'arrière-plans : une pour le bouton actif, l'autre pour le bouton inactif.

Pour que la barre d'outils s'empile de gauche à droite, vous pouvez également les contrôler en CSS :

```
.olControlEditingToolbar div {
    float:right;
    right: 0px;
    height: 30px;
    width: 200px;
}
```

Définissez simplement le paramètre 'float: right' et donnez à l'élément parent XXXXXX

Dans le but d'améliorer le travail de l'utilisateur, les paneaux existant comme *EditingToolbar* utilise une seule image d'arrière-plan, et contrôle l'icône à afficher via les paramètres 'top' et 'left',

décalant et découpant l'image d'arrière-plan. Ce n'est pas obligatoire, mais comme cela vous n'avez pas à attendre l'image 'inactive' à afficher lors de la sélection de l'outil avant de continuer.

Les contôleurs à utiliser avec les Panels

Les Panels peuvent contenir des contrôles de plusieurs 'types'. Chaque outils dans un panel doit avoir un attribut 'type' qui est parmi :

- OpenLayers.Control.TYPE_TOOL (celui par défaut)
- OpenLayers.Control.TYPE_BUTTON
- OpenLayers.Control.TYPE_TOGGLE

Personnaliser un Panel existant

Les différents panels existant – comme *EditingToolbar* ou *PanPanel* – ont plusieurs contrôles combinés, mais il n'est pas toujours souhaitable d'utiliser tous ces contrôles. Cependant, il est relativement simple de créer un contrôleur qui copie le comportement de ces contrôles. Par exemple, si vous désirez créer un contrôleur d'édition qui a seulement la possibilité de dessiner des lignes, vous pouvez faire cela avec le code suivant :

```
var layer = new OpenLayers.Layer.Vector();
var panelControls = [
 new OpenLayers.Control.Navigation(),
 new OpenLayers.Control.DrawFeature(layer,
     OpenLayers.Handler.Path,
     {'displayClass': 'olControlDrawFeaturePath'})
];
var toolbar = new OpenLayers.Control.Panel({
   displayClass: 'olControlEditingToolbar',
   defaultControl: panelControls[0]
});
toolbar.addControls(panelControls);
map.addControl(toolbar);
```

Il y a deux choses à noter ici :

- Nous reétulisons le style de EditingToolbar en appelant ca classe 'displayClass'. Cela signifie que nous prendrons les icônes par défaut et les autres définitions du CSS pour

cette barre d'outils (pour plus de détails, lisez *Style des panels*.)
- Nous définission le contrôleur par défaut sur le contrôleur navigation, mais nous pouvons changer cela facilement.

De cette manière, vous pouvez utiliser n'importe quel contrôleur qui fonctionne dans un panel – incluant par exemple, les contrôles SelectFeature, ZoomToMaxExtent et d'autres – simplement en changeant les contrôles présents dans la liste.

Contrôleurs de la carte

ArgParser

Prend des arguments sous formes d'URL et met à jour la carte.

Pour faire fonctionner le contrôleur ArgParser, vous devez vérifier que 'getCenter()' renvoie null avant le centrage de votre carte pour la première fois. La plupart des applications utilise un appel setCenter ou zoomToMaxExtent : cet appel doit être évité si le centre est déjà défini :

```
var map = new OpenLayers.Map('map');
var layer = new OpenLayers.Layer();
map.addLayer(layer);

// Ensure that center is not set
if (!map.getCenter()) {
    map.setCenter(new OpenLayers.LonLat(-71, 42), 4);
}
```

Le contrôleur ArgParser est activé par défaut.

Attribution

Le contrôleur attribution affichera les propriétés d'attribution définies sur chaque couche dans la carte en bas à droite de la carte, par défaut. Le style et la localisation de ce contrôleur peuvent être modifié en écrasant la classe CSS 'olControlAttribution'.

L'utilisation du contrôleur d'attribution est présenté dans l'exemple Attribution. Pour des

informations sur l'API, lisez la documentation de l'API Attribution.

DragFeature

DragPan

Le contrôleur DragPan implémente les interactions de déplacement de la carte.

DrawFeature

EditingToolbar

Affiche un contrôleur *Navigation*, avec trois outils d'édition : Point, Chemin, et Polygone. Si cela ne vous convient pas, lisez *Personnaliser un Panel existant* ci-dessus.

KeyboardDefaults

LayerSwitcher

Measure

un outil de mesure de distance planaire.

ModifyFeature

Le contrôleur ModifyFeature peut être utilisé pour éditer un objet vecteur existant.

Ce contrôleur propose trois types différents d'événements pour se déclencher sur la couche : * beforefeaturemodified - déclenché lorsqu'un utilisateur sélectionne un objet pour commencer à l'éditer. * featuremodified - déclenché lorsqu'un utilisateur change quelque chose de l'objet. * afterfeaturemodified - déclenché après que l'utilsateur déselectionne l'objet.

Pour enregistrer l'un de ces événements, enregistrez le sur la couche :

```
var layer = new OpenLayers.Layer.Vector("");
layer.events.on({
  'beforefeaturemodified': function(evt) {
      console.log("Selected " + evt.feature.id  + " for modification");
  },
  'afterfeaturemodified': function(evt) {
      console.log("Finished with " + evt.feature.id);
  }
});
```

Il y a plusieurs modes différents sur lequel le contrôleur ModifyFeature peut fonctionner. Ceux-ci peuvent être combinés pour travailler ensemble.

- RESHAPE – par défaut. Permet de changer les sommets d'un objet en déplaçant les sommets existant, en créant de nouveaux sommets en déplaçant des 'sommets virtuels' ou en effaçant des sommets en se déplaçant au dessus d'eux et en pressant la touche delete.
- RESIZE – permet de modifier la taille de la géométrie ;
- ROTATE – change l'orientation de la géométrie ;
- DRAG – change la position de la géométrie.

Lors de la création du contrôleur, vous pouvez utiliser l'opérateur logique OU pour combiner les deux :

```
var modifyFeature = new OpenLayers.Control.ModifyFeature(layer, {
  mode: OpenLayers.Control.ModifyFeature.RESIZE | OpenLayers.Control.ModifyFeature.DRAG
});
```

Pour un exemple d'utilisation du contrôleur ModifyFeature, voyez l'exemple de ModifyFeature. Pour les informations sur l'API, lisez la documentation de l'API sur ModifyFeature.

Le contrôleur ModifyFeature peut seulement s'utiliser avec une seule couche à un moment données. Pour modifier plusieurs couches, utiliser plusieurs contrôles ModifyFeature.

Avertissement sur l'obsolescence

Dans la version 2.6 d'OpenLayers les fonctions onModificationStart, onModification et onModificationEnd sur ce contrôleur ne sont plus une manière recommandée pour recevoir les

événements de modifications. À la place, utilisez les événements beforefeaturemodified, featuremodified, et afterfeaturemodified pour prendre en charge ces cas.

MousePosition

NavToolbar

Navigation

Le contrôleur de remplacement pour l'ancien contrôleur *MouseDefaults*. Ce contrôleur est la combinaison de :

- *DragPan*
- *ZoomBox*
- Handler.Click, pour le zoom par double clique
- Handler.Wheel, pour le zoom avec la molette

La requête la plus commune pour le contrôleur Navigation est de désactiver la molette de la souris pour zommer lors de l'utilisation du contrôleur. Pour cela, assurez vous qu'aucun autre contrôleur de navigation n'ait été ajouté à votre carte – par exemple, par *EditingToolbar* – et appellez disableWheelNavigation sur le contrôleur de navigation.

NavigationHistory

OverviewMap

PanPanel

Un ensemble de boutons visuels pour contrôler la localisation de la carte. Sous-classe de Control.Panel, il est facilement contrôlé par les styles via CSS. La classe `.olControlPanPanel` et ses divs internes contrôlent le style du PanPanel. Si vous désirez personnaliser le style des contrôles dans le coin gauche supérieur de la carte, ce contrôleur est celui qu'il vous faut.

Ce contrôleur est conçu pourfonctionner avec le contrôleur *ZoomPanel* pour répliquer la fonctionnalité du contrôleur *PanZoom*.

PanZoom

PanZoomBar

Permalink

Le contrôleur Permalink, avec le contrôleur ArgParser, sont conçus pour faciliter la création de lien vers une carte existante. En ajoutant le contrôleur permalink à votre carte, vous afficherez un texte dans la carte qui agit comme un lien pour vos utilisateurs.

Pour que le permalink fonctionne, vous devez vous assurez que vous avez vérifié si le centre a déjà été défini par le contrôleur ArgParser avant de définir le centre de votre carte. Pour cela, vérifiez simplement map.getCenter() d'abord :

```
if (!map.getCenter()) {
  map.setCenter(new OpenLayers.LonLat(0,0),0);
}
```

Scale

ScaleLine

SelectFeature

Snapping

Permet de contrôler la fonction de saisie par magnétisme (snapping) depuis les sommets des éléments d'une couche vers les noeuds, sommets ou arêtes des éléments d'autres couches. La saisie par magnétisme est possible lorsque le contrôle est actif, alors que lorsqu'il est désactivé,

l'édition se fait selon le comportement habituel. Le contrôle peut être configuré pour permettre la saisie par magnétisme sur les noeuds, sommets et/ou arêtes d'un nombre indéterminé de couches (les éléments de ces couches étant chargés côté client). La tolérance, le type de saisie par magnétisme, et un filtre optionnel peuvent être configurés pour chaque couche-cible.

Vous trouverez plus de détails sur la page *OpenLayers.Control.Snapping*.

Split

Permet de découper les éléments linéaires d'une couche vecteur en fonction des modifications/ éditions d'éléments linéaires de n'importe quelle autre couche vecteur, ou d'une couche dessin temporaire. Le contrôle fonctionne selon deux modes. Par défaut, le contrôle permet de dessiner les éléments temporaires sur une couche dessin temporaire (gérée par le contrôle), cette dernière étant ensuite utilisée pour découper les éléments choisis de la couche-cible. En mode auto-split, Le contrôle suit les éditions de la couche modifiable (ajouts de nouveaux éléments ou modification des éléments existants) et découpe les éléments choisis sur la couche-cible.

Comme tous les contrôles, celui-ci peut être ajouté à une carte. Il n'a pas de représentation visuelle distincte mais peut être connecté à un bouton, ou à tout autre outil, pour permettre son activation en cliquant. Aucune interface graphique n'est fournie pour la configuration du contrôle. La collecte des instructions de l'utilisateur pour la configuration des contrôles est une tâche spécifique de l'application.

Vous trouverez plus de détails sur la page *OpenLayers.Control.Split*.

ZoomBox

ZoomPanel

Un ensemble de bouttons pour contrôler le zoom de la carte. Il s'agit d'une sous-classe de Control.Panel, facilement contrôlable par CSS. La classe `.olControlZoomPanel` et ses div internes contrôlent le style de PanPanel. Si vous désirez personnaliser l'apparence du contrôleur dans le coin supérieur gauche, ce contrôleur est celui qu'il vous faut.

Ce contrôleur est conçu pour fonctionner avec le contrôleur *PanPanel*, en répliquant le fonctionnement du contrôleur *PanZoom*.

Classes button

Ces classes n'ont pas d'Interface Utilisateur et ont d'abord été conçues pour être utilisées dans un panneau de contrôleur.

Pan

Utilisé dans Panpanel ; lorsqu'il est déclenché, entraîne le déplacement de la carte dans une direction définie.

ZoomIn

Utilisé dans PanPanel ; lorsqu'il est déclenché, entraîne un zoom avant.

ZoomOut

Utilisé dans PanPanel ; lorsqu'il est déclenché, entraîne un zoom arrière.

ZoomToMaxExtent

Utilisé dans PanPanel ; lorsqu'il est déclenché, entraîne un zoom sur l'étendue maximale de la carte.

Classe de Base Générique

Les classes suivantes sont d'abord utilisées pour être étendues et n'ont pas pour objectif d'être utilisé directement.

Button

Utilisé dans les contrôles Panel.

Panel

Utilisé comme base pour les contrôles NavToolbar et EditingToolbar ainsi que d'autres. Réunis les boutons / outils pour être utilisés ensemble.

Contrôleurs dépréciés ¶

MouseDefaults

Remplacé par le contrôleur *Navigation*.

MouseToolbar

Remplacé par le contrôleur *NavToolbar*.

Documentation supplémentaire

- OpenLayers.Control.Snapping
- OpenLayers.Control.Split