

## How to use the "LineOfSight" operator

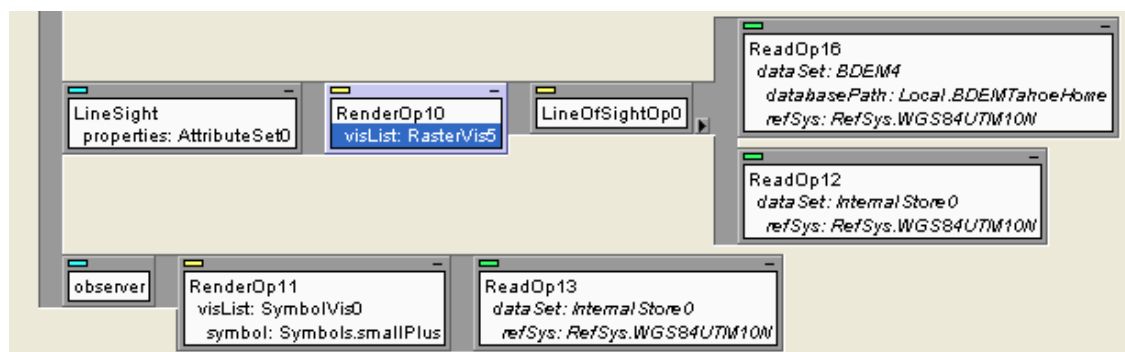
The LineOfSightOp uses elevation data to compute the regions that have a free line of sight to a given position.

Open the map configuration in Space Lab. You need a map configuration that contains height data.

Create a new **OrdinaryLayer** and give him a good name then connect a **RenderOp**, a **LineOfSightOp** and connect **two ReadOp**. One of them connected to the input "inputElevation", that reads the height data and one ReadOp with an InternalStore dataset and connect that one to the input "inputObservers". Set the reference system on the InternalStore to the same reference system as on the height data. Add a **RasterVis** into the RenderOp and create a nice gradient in the color editor to set the colors which appear when the user clicks on the map. Further more set the alpha property (this is a global alpha for all colors) to a nice semi-transparent value (like 127) and set filter to #bicubic.

RasterVis	
(Name)	<b>RasterVis5</b>
(Modifiers)	<input type="checkbox"/> Public <input type="checkbox"/> Global
alpha	255
alphaMethod	#correct
color	[[ 0, [ 0, 255, 0, 0 ]], [ 1, [ 255, 204...
condition	[ ]
filter	<b>#bicubic</b>
pickingSupport	#default
properties	[ ]

That you can see where the observer is placed add a new ordinary layer connect a RenderOp and a ReadOp. Add a SymbolVis into the RenderOp that draws a small dot. The ReadOp should read from the same internal store like the ReadOp in the LineOfSight.



Add to the attributeSet of the the main view and the two ordinary layer the attribute "spaceWeb" (Boolean → true) and to the "LineOfSight" layer additional the attribute "rwcType" (String → individuallImage).

Go to the LineOfSightOp and set a default value in maxDistance and Sensorheight.

Next step is to create a simple **Styled Layer Descriptor** with only one UserLayer that contains a **gml point** and store this in a string. That the coordinates are set when the user defines an observation point create a **placeholder** for the coordinates. Do the same for the sensor height and the distance, that the user can choose these parameters himself.

#### Example:

```
var sldString = "<?xml version='1.0' encoding='UTF-8'><slid:StyledLayerDescriptor xmlns:se='http://www.opengis.net/se' xmlns:sld='http://www.opengis.net/sld' xmlns:xlink='http://www.w3.org/1999/xlink' xmlns:ogc='http://www.opengis.net/ogc' xmlns:gml='http://www.opengis.net/gml' xmlns:cab='http://www.carmenta.com/cab' version='1.1.0'><slid:UserLayer><se:Name>MyLayer</se:Name><slid:InlineFeature><gml:FeatureCollection><gml:featureMember><gml:PointProperty><gml:Point srsName='EPSG:32610'><gml:pos>XPlaceholder, YPlaceholder</gml:pos></gml:Point><Distance type='int'>distancePlaceholder</Distance><Height type='double'>heightPlaceholder</Height></gml:PointProperty></gml:featureMember></gml:FeatureCollection></slid:InlineFeature><slid:UserStyle></slid:UserStyle></slid:UserLayer></slid:StyledLayerDescriptor>"
```

After these settings are done go to the index file of your page. Implement a **ClickTool** in the userInit function and get with the Coord.prototype.click = **function** (X, Y) the coordinates of the position the user clicked on the map. After the user clicks you need to replace the placeholder in the SLD file with these coordinates. Also define two **selectboxes** in the body of the html element one for the distance and one for the sensorheight. Take these values the user chose and send them to SLD file too. After you send all these values to the SLD file **load the SLD with the "addAttribute"** function to the map.

```
function userInit() {
    map.setClickTool('Coord');
    map.config.viewSettings.useRWMapRequests = false;
    map.backgroundColor = "0xA9D0F5";
}

var firstCoord = null;
function Coord(map, prefix) {

}

Coord.prototype.click = function (X, Y) {
    this.firstCoord = null;
    this.firstCoord = map.getPosition(X, Y);
    this.firstX = X;
    this.firstY = Y;
}
```

```
var sldRealString = sldString;
sldRealString = sldRealString.replace(/XPI aceholder/g, this.firstCoord.x);
sldRealString = sldRealString.replace(/YPI aceholder/g, this.firstCoord.y);

var di stselect = document.getElementById("di st");
var val di st = di stselect[di stselect.selectedIndex].text;
sldRealString = sldRealString.replace(/di stPI aceholder/g, val di st);

var hei ghtselect = document.getElementById("sensorhei ght");
var val hei ght = hei ghtselect[hei ghtselect.selectedIndex].text;
sldRealString = sldRealString.replace(/hei ghtPI aceholder/g, val hei ght);

var layer = map.getAllLayers();
layer[1].setAttribute("SLD_BODY", sldRealString);
layer[1].refresh(true);
}
```

</script>