

---

# **Qgis2threejs plugin Documentation**

*Release 1.4*

**Minoru Akagi**

Oct 09, 2017



---

## Contents

---

<b>1</b>	<b>Examples</b>	<b>3</b>
<b>2</b>	<b>Tutorial</b>	<b>5</b>
<b>3</b>	<b>Export Settings</b>	<b>9</b>
<b>4</b>	<b>Object Types</b>	<b>15</b>
<b>5</b>	<b>3D Viewer</b>	<b>21</b>
<b>6</b>	<b>Plugin Settings</b>	<b>23</b>
<b>7</b>	<b>Optional Feature</b>	<b>25</b>
<b>8</b>	<b>Programmatic Export Using Python</b>	<b>27</b>
<b>9</b>	<b>API</b>	<b>29</b>
<b>10</b>	<b>Development</b>	<b>31</b>



Qgis2threejs plugin is a **QGIS** plugin. This plugin exports terrain data, map canvas image and vector data to your web browser. You can view 3D objects in the web browser (You need a web browser which supports WebGL).

Table Of Contents:



Click the images below to see live examples.

- **Mt. Fuji:** Shaded relief map made with SRTM data

[\[Live example\]](#)

Source: U.S. National Aeronautics and Space Administration (NASA) and National Geospatial-Intelligence Agency (NGA). [SRTM elevation data](#) (SRTM V2.1. Filled small voids using `gdal_fillnodata.py`)

- **Mt. Fuji (Large Area):** With surroundings option. Shaded relief map made with SRTM data.

[\[Live example\]](#)

- **Seto Inland Sea:** Custom plane template. Shaded relief map made with SRTM data.

[\[Live example\]](#)

- **Mt. Aso:** Polygon overlay, switching layer visibility and adjusting layer opacity

[\[Live example\]](#)

Sources: Geospatial Information Authority of Japan. [GSI Tiles](#) (standard map, shaded relief and elevation tile) / Geological Survey of Japan, AIST. [Seamless digital geological map of Japan 1: 200,000 \(CC BY-ND 2.1 JP\)](#)

- **Forest basins:** Polygon overlay with texture

[\[Live example\]](#)

Sources: Geospatial Information Authority of Japan. [GSI Tiles](#) (Orthophoto and elevation tile)





Let's start using Qgis2threejs plugin!

### Installing the plugin

Open the QGIS plugin dialog (Plugins > Manage and install plugins...), and then install Qgis2threejs plugin.

---

**Hint:** Need help? See the [10.1.2. Installing New Plugins](#) section of the QGIS training manual.

---

### Obtaining elevation data

If you already have raster DEM data, you can skip this step.

NASA released elevation data generated from NASA's [Shuttle Radar Topography Mission](#) digital topographic data. We can use the data freely. Elevation data version 2.1 can be downloaded from the [distribution site](#).

Download a zip file that contains elevation data of the area you are interested in from under the `version2_1/SRTM3` directory. The zip file contains a `.hgt` file, which is readable by the GDAL.

---

**Tip:** If the area extends over two or more files, you might want to create a virtual mosaic using [Build Virtual Raster \(Catalog\)](#) tool of GdalTools.

---

---

**Tip:** Do you have time to explore new high-resolution SRTM elevation data? You can download 1 arc-second SRTM data from the [EarthExplorer](#) (User registration and login are required).

---

## Loading DEM data

Drag & drop the downloaded zip file to QGIS window (or load the file using `Add Raster Layer` dialog). You don't have to unzip it.

## CRS setting

Horizontal unit of SRTM data is degree, whereas vertical unit is meter. For appropriate visualization, you need to transform the DEM data to a projected CRS. QGIS can perform the CRS transformation on the fly.

So, let's enable the `On The Fly CRS transformation` and change the map CRS to a projected CRS.

Click the CRS status icon in the bottom-right corner of the window to open the project properties dialog. Activate the `Enable 'on the fly' CRS transformation` checkbox and then select a suitable CRS for the DEM extent. If you don't know which CRS is best suited, select the Spherical Mercator projection (EPSG: 3857), which is adopted by many web maps.

---

**Note:** In the Spherical Mercator projection, size of every feature is horizontally larger than actual size except features on the equator. At latitude 40 degrees it is enlarged 1.3 times, at 60 degrees enlarged twice.

---

## Layer styling

Open the raster properties dialog for the DEM layer and colorize the DEM layer richly.

An example (`Singleband pseudocolor` render type and inverted `BrBG` color map):

## Exporting

Zoom to a part of the DEM layer extent as the map canvas is filled by the colorized DEM.

Click the plugin icon in the web toolbar.

Then, click `Run` button in the dialog.

3D terrain appears in your web browser!

If you want to hide the dat-gui controls box at the top-right corner, press `H` key.

## Conclusion

Tutorial is over. Now, you know 3D visualization with QGIS is very easy. If you can use high-quality data, you can create beautiful 3D scenes!

---

**Tip:** Next, how about adding a background map layer to the map canvas. You can do it easily with [QuickMapServices plugin](#). Also, how about adding vector data to the scene. [Object Types](#) page has example images of various object types. See [Export Settings](#) for the detail.

---

---

**Tip:** You can publish the exported scene just by uploading the exported files to a web server.

---

---

**Note:** Please do not forget to ensure that you comply with the Terms and use for the data before publishing the scene to the web.

---



### Plugin Dialog

In order from the top:

- Combo box to select a template
  - Select one from templates with different functions. See *Template* section.
- Tree widget on the left side and panel with widgets on the right side
  - Items with check box in the tree widget are optional. When the current item is optional and not checked, widgets on the right side are grayed out.
- Output HTML file path edit box
  - Select output HTML file path. Usually, a js file with the same file title that contains whole data of geometries and images is output into the same directory, and some JavaScript library files are copied into the directory. Leave this empty to output into temporary directory. Temporary files are removed when you close the QGIS application.
- Settings button
  - Pop-up menu with the following menu items is shown:
    - Load Settings
      - Loads export settings from a settings file.
    - Save Settings As
      - Saves export settings to a settings file. Default file extension is `.qto3settings`.
    - Clear Settings
      - Clears current export settings.
    - Plugin Settings
      - Shows *Plugin Settings* dialog.

- Run, Close and Help buttons

Exporting starts when you press the Run button. When the exporting has been done, the exported page will be opened in web browser. At this time, export settings are automatically saved to a file under the same directory as the project file if you are working with a project file. Later the export settings of the project will be automatically loaded into the plugin.

Pressing the Help button will open the local document with default web browser.

## General Settings

### Template

Available templates:

- 3DViewer.html

This template is a 3D viewer without any additional UI library.

- 3DViewer(dat-gui).html

This template has a `dat-gui` panel, which makes it possible to toggle layer visibility, adjust layer opacity and add a horizontal plane movable in the vertical direction.

- FileExport.html

This template builds 3D models on the web browser, but doesn't render them. Instead, it has some buttons to save 3D models in [STL format](#), [Wavrefront OBJ format](#) or [COLLADA format](#). It also has ability to save the texture image(s).

These formats are widely supported by 3DCG softwares such as [Blender](#).

### World

- Base size

Enter a size in 3D world that corresponds to the map canvas width. The default value is 100.

- Vertical exaggeration

Vertical exaggeration factor. This value affects terrain shape and z positions of all vector 3D objects. This also affects 3D object height of some object types with volume. Object types to be affected:

Point : Cylinder, Cube, Cone

Polygon : Extruded

3D objects of the following types have volume, but their heights aren't affected by this factor:

Point : Sphere, JSON model, COLLADA model

Line : Pipe, Cone, Box

The default value is 1.5.

- Vertical shift

Vertical shift for all objects. If you want to export high altitude and narrow area, you should adjust the object positions to be displayed at the center of browser by changing this value. If you set the value to -1000, all objects are shifted down by 1000 in the unit of map CRS.

- Background

Select either sky-like gradient or a solid color for the background of scene. Default is Sky.

- Display of coordinates

If the `Latitude and longitude (WGS84)` option is selected, coordinates of clicked position on a 3D object are displayed in longitude and latitude (WGS84). If `Proj4js` doesn't support current map CRS, this option is disabled.

## Controls

Two available controls: `OrbitControls`, `TrackballControls`.

The usage of each control is displayed below the combo box.

## Layer Settings

### DEM

You can select a DEM layer from 1-band rasters loaded in QGIS using `Add Raster Layer (GDAL provider)`. Selected DEM layer is used to calculate z positions of vector objects. You can also select a flat plane at zero altitude.

### Resampling

- Simple

Select a DEM resolution from several levels. This resolution is used to resample the DEM, but is not for texture.

- Surroundings option

This option enlarges output DEM by placing DEM blocks around the main block of the map canvas extent. Size can be selected from odd numbers in the range of 3 to 9. If you select 3, total 9 (=3x3) blocks (a center block and 8 surrounding blocks) are output. Roughening can be selected from powers of 2 in the range of 1 to 64. If you select 2, grid point spacing is doubled. It means that the number of grid points in the same area becomes 1/4. If map canvas image is selected as the display type, texture image size for each block is maximum 256 x 256.

- Advanced (quad tree)

Multiple resolution DEM export. Area you want to focus is output in high resolution and the surroundings are output in low resolution. Draw a rectangle on the map canvas to set focus area. Specifying a point is also possible. The higher QuadTree height, the higher resolution of the focus area. Grid size of each block is 64 x 64.

### Display type

You can choose from map canvas image, layer image, a image file or a solid color.

- Map canvas image

Map canvas image is used to texture the main DEM block in simple resampling mode. Each block of surroundings (in simple resampling mode) and quads (in advanced resampling mode) is textured with image rendered with the current map settings.

- Layer image

Each block is textured with image rendered with the selected layer(s).

- Image file

Texture with existing image file such as PNG and JPEG file. TIFF is not supported by some browser. See [Image format support](#) for details.

- Solid color

To select a color, press the button on the right side.

### Options

- Resolution

Increases the size of image applied to each DEM block. This option is enabled when either `Map canvas image` or `Layer image` is selected. You can select a ratio to map canvas size from 100, 200 and 400 (%). Image size in pixels follows the percent.

- Transparency

Sets transparency of the DEM. 0 is opaque, and 100 is transparent.

- Transparent background (With map canvas image or layer image)

Makes background of the image to be rendered transparent.

- Enable transparency (With image file)

Enables the image transparency.

- Enable shading

Adds a shading effect to the DEM.

### Clip

Clips the DEM with a polygon layer. If you have a polygon layer that represents the area that elevation data exist or represents drainage basins, you might want to use this option.

### Sides and frame

- Build sides

This option adds sides and bottom to the DEM. The z position of bottom in the 3D world is fixed. You can adjust the height of sides by changing the value of vertical shift option in the World panel. If you want to change color, please edit the output JS file directly.

- Build frame

This option adds frame to the DEM. If you want to change color, please edit the output JS file directly.

### Additional DEM

If you want to export more than one DEM, check the checkbox on the left of child item you want. For example of usage, it may be possible to cover the terrain with supposed terrain surface of a summit level map, or make a 3D heat map.

Some options that are available in main DEM panel cannot be used. Resampling mode is limited to simple. Surroundings, sides and frame options are not available.



## Vector

Vector layers are grouped into three types: Point, Line and Polygon. Common settings for all vector layers:

- Z coordinate

Mode combo box has these items:

- Z value

This item can be selected when the layer geometries have z coordinates and the layer type is point or line.

- Relative to DEM

$z = \textit{Elevation at vertex} + \textit{addend}$

- +”field name”

$z = \textit{Elevation at vertex} + \textit{field value} + \textit{addend}$

Only numeric fields are listed in the combo box.

- Absolute value

$z = \textit{value}$

- “field name”

$z = \textit{field value} + \textit{addend}$

Only numeric fields are listed in the combo box.

The unit of the value is that of the map CRS.

- Style

Usually, there are options to set object color and transparency. Refer to the links below for each object type specific settings. The unit of value for object size is that of the map CRS.

- Feature

Select the features to be output.

- All features

All features of the layer are exported.

- Features that intersect with map canvas extent

Features on the map canvas are exported.

- \* Clip geometries

This option is available with Line/Polygon layer. If checked, geometries are clipped by the extent of map canvas.

- Attribute and label

If the export attributes option is checked, attributes are exported with feature geometries. Attributes are displayed when you click an object on web browser.

If a field is selected in the label combobox, a label is displayed above each object and is connected to the object with a line. This combo box is not available when layer type is line.

## **Point**

Point layers in the project are listed as the child items. The following object types are available:

Sphere, Cylinder, Cone, Box, Disk, Icon, JSON model, COLLADA model

See *Point Layer* section in *Object Types* page for each object type specific settings.

## **Line**

Line layers in the project are listed as the child items. The following object types are available:

Line, Pipe, Cone, Box, Profile

See *Line Layer* section in *Object Types* page for each object type specific settings.

## **Polygon**

Polygon layers in the project are listed as the child items. The following object types are available:

Extruded, Overlay

See *Polygon Layer* section in *Object Types* page for each object type specific settings.

- *Point Layer*
  - *Line Layer*
  - *Polygon Layer*
- 

## Point Layer

*Sphere | Cylinder | Cone | Box | Disk | Icon | JSON model | COLLADA model*

### Sphere

Sphere with specified radius, color and transparency

[No image]

**Specific settings :**

- **Radius** Numerical value.

**Origin :**

center of sphere

**three.js geometry class:**

`SphereGeoemtry`

## Cylinder

Cylinder with specified radius, height, color and transparency

[No image]

**Specific settings :**

- **Radius** Numerical value.
- **Height** Numerical value.

**Origin :**

center of bottom (if height > 0)

**three.js geometry class:**

CylinderGeometry

## Cone

Cone with specified radius, height, color and transparency

[No image]

**Specific settings :**

- **Radius** Numerical value.
- **Height** Numerical value.

**Origin :**

center of bottom (if height > 0)

**three.js geometry class:**

CylinderGeometry

## Box

Box with specified width, depth, height, color and transparency

[No image]

**Specific settings :**

- **Width** Numerical value.
- **Depth** Numerical value.
- **Height** Numerical value.

**Origin :**

center of bottom (if height > 0)

**three.js geometry class:**

BoxGeometry

## Disk

Disk with specified radius, orientation, color and transparency

[No image]

**Specific settings :**

- **Radius** Numerical value.
- **Dip** In degrees. See [Strike and dip - Wikipedia](#).
- **Dip direction** In degrees.

**Origin :**

center of disk

**three.js geometry class:**

[CylinderGeometry](#)

## Icon

Image which always faces towards the camera

[No image]

**Specific settings :**

- **Image file** File path.
- **Scale** Numerical value.

**Origin :**

center of image

**three.js object class:**

[Sprite](#)

## JSON model

[No image]

**Specific settings :**

- **JSON file** File path.
- **Scale** Numerical value.
- **Rotation (x)** In degrees.
- **Rotation (y)** In degrees.
- **Rotation (z)** In degrees.

**Origin :**

origin of model

## COLLADA model

[No image]

### Specific settings :

- **COLLADA file** File path (.dae). If the model has texture images, they need to be copied to the destination directory manually.
- **Scale** Numerical value.
- **Rotation (x)** In degrees.
- **Rotation (y)** In degrees.
- **Rotation (z)** In degrees.

### Origin :

origin of model

---

## Line Layer

*Line | Pipe | Cone | Box | Profile*

### Line

[No image]

### Specific settings :

no specific settings

### three.js object class:

*Line*

Image was created with [GSI Tiles](#) (ort, dem).

### Pipe

Places a cylinder to each line segment and a sphere to each point.

[No image]

### Specific settings :

- **Radius** Numerical value.

### three.js geometry classes:

*CylinderGeometry* and *SphereGeoemtry*

Image was created with [GSI Tiles](#) (airphoto, dem).

## Cone

Places a cone to each line segment. Heading of cone is forward direction.

[No image]

**Specific settings :**

- **Radius** Numerical value.

**three.js geometry class:**

[CylinderGeometry](#)

Image was created with [GSI Tiles](#) (ort, dem) and [National Land Numerical Information](#) (Rivers. MILT of Japan).

## Box

Places a box to each line segment.

[No image]

**Specific settings :**

- **Width** Numerical value.
- **Height** Numerical value.

**three.js geometry class:**

[BoxGeometry](#) and [Geometry](#)

Image was created with [GSI Tiles](#) (airphoto, dem).

## Profile

Makes a vertical plane under each line segment. When the altitude mode of z coordinate is `Relative to DEM` or `"field name"`, each linestring is split into segments using a triangle mesh generated from DEM and every upper edge is located at the relative height from the mesh surface.

[No image]

**Specific settings :**

- **Lower Z** Z coordinate of lower edge.

**three.js geometry class:**

[PlaneGeometry](#)

Image was created with SRTM3 elevation data.

---

## Polygon Layer

*Extruded | Overlay*

## Extruded

Extruded polygon with specified height, color and transparency

[No image]

**Specific settings :**

- **Height** Numerical value.

**three.js geometry class:**

[ExtrudeGeometry](#)

Image was created with [GSI Tiles](#) (ort, dem) and [OpenStreetMap](#) (© [OpenStreetMap contributors](#), [License](#)).

## Overlay

Overlay polygon draped on the main DEM with specified color, border color and transparency. When the altitude mode of z coordinate is `Relative to DEM` or `+ "field name"`, each polygon is split into triangles using a triangle mesh generated from DEM, and is located at the relative height from the mesh surface. Otherwise, creates a flat polygon at a specified altitude. You can add side to each polygon if you want.

[No image]

**Specific settings :**

- Border color
- **Side** Check this option to add side to each polygon.
- Side color
- **Side lower Z** Z coordinate of lower edge of side.

**three.js classes:**

[Geometry](#), [Line](#) and [PlaneGeometry](#)

Image was created with [GSI Tiles](#) (ort, dem) and [National Land Numerical Information](#) (Sediment Disaster Hazard Area. Provided by Okayama prefecture, Japan).



### Controls

Mouse and keyboard operation method depends on the control selected in the export settings. There is list of mouse buttons and keys in the about box. Press `I` key to show the box.

### Identifying Features

When you click on an object, layer name that the feature (object) belongs to and the clicked coordinates (in order of x, y, z) are shown. If `Latitude and longitude (WGS84)` option (in the `Display of coordinates` group of `World` page) is selected, longitude and latitude are shown in DMS format (degrees, minutes and seconds). If `Export attributes` option of each vector layer is selected, attribute list of the clicked feature follows them.

### Controls Box

This feature is available with **3DViewer(dat-gui) template**.

The controls box has:

- layer sub menus

Each sub menu has:

- a check box to toggle layer visibility
- a slider to adjust layer transparency
- sub menu to control a vertically movable plane
- help button to show the about box

## Rotate Animation

This feature is available with **OrbitControls**.

Pressing R key starts/stops rotate animation. Camera rotates around the camera target clockwise.

## Save Image

Press `Shift + S` to show save image dialog, then enter image size and click the OK button. In addition, with some web browsers, you need to click a link to save image. The image file format is PNG. To change label color and/or adjust label size, edit `Qgis2threejs.css` (`print-label` class).

---

**Note:** A known issue: Wrong image output if the size is too large ([issue #42](#))

---

## URL Parameters

You can get current view URL in the about box, and later restore the view by entering the URL in the URL box of web browser.

Parameters used in view URL:

- `cx, cy, cz`: camera position
- `tx, ty, tz`: camera target
- `ux, uy, uz`: camera up direction (`TrackballControls`)

e.g. `file:///D:/example.html#cx=-64.8428840144039&cy=-40.75234765087484&cz=24.603200058346065`

Other parameters:

- `width`: canvas width (pixels)
- `height`: canvas height (pixels)
- `popup`: pop up another window with specified width and height

---

### Plugin Settings

---

- **Browser path**

If you want to open exported page with a web browser other than default browser, use this option.

- **Optional Features**

There is list of optional features. You can enable optional feature you want to use.

For details, see *Optional Feature*.



To use this feature, you need to go to the settings dialog and activate the feature.

### **GSI Elevation Tile Provider (DEM Provider)**

This DEM provider downloads GSI Tiles (elevation) from the web server of Geospatial Information Authority of Japan, and provides elevation data to Qgis2threejs. The tiled elevation data is based on the popular map tile system known as “XYZ tile”, and each tile data is provided in the format of CSV. It covers the area of Japan in zoom levels from 0 to 14. Read and follow [Geospatial Information Authority of Japan Website Terms of Use](#).

Technical information (in Japanese):

- Specification of elevation tile: <http://maps.gsi.go.jp/development/demtile.html>



---

## Programmatic Export Using Python

---

Do you want to export many scenes to web? You can do it programmatically using Python!

### Step 1

You need to prepare an export settings file. The export settings contains various items, so you might want to create the settings file using the plugin dialog.

Procedure:

1. Open a project and click the Qgis2threejs icon in the web tool bar to open the plugin dialog.
2. Configure the export settings.
3. Click the Run button to see the export on the web browser and check that the the settings are good.
4. Open the plugin dialog again. Click the settings button at the bottom-left corner, and then save the export settings to a file (file extension is `.qto3settings`).

### Step 2

You are ready to play with Python. Let's open the QGIS Python console (you need QGIS version 2.4 or later).

```
from PyQt4.QtCore import QSize
from qgis.core import QgsCoordinateReferenceSystem, QgsCoordinateTransform, QgsPoint
from Qgis2threejs.api import Exporter

# Path to the prepared .qto3settings file
settingsPath = "D:/pref_offices.qto3settings"

# Places to export (in WGS 84)
places = [(u"Kyoto", QgsPoint(135.7555, 35.0210)),
          (u"Osaka", QgsPoint(135.5199, 34.6863)),
```

```
(u"Nara", QgsPoint(135.8329, 34.6852)),
(u"Ehime", QgsPoint(132.7657, 33.8416))]

# Output filename template
path_tmpl = "D:/output_scenes/{0}.html"

# Coordinate transformer: WGS 84 to JGD2000 / UTM zone 53N
wgs84 = QgsCoordinateReferenceSystem(4326, QgsCoordinateReferenceSystem.EpsgCrsId)
utm53 = QgsCoordinateReferenceSystem(3099, QgsCoordinateReferenceSystem.EpsgCrsId)
transform = QgsCoordinateTransform(wgs84, utm53)

# Make sure that map canvas CRS is EPSG:3099
canvas = iface.mapCanvas()
canvas.setCrsTransformEnabled(True)
canvas.setDestinationCrs(utm53)

# Get map settings from the map canvas
mapSettings = canvas.mapSettings()

# Canvas size (base image size)
canvasSize = QSize(600, 600)
mapSettings.setOutputSize(canvasSize)

# Size of extent, and rotation
width = 10000.
height = width * canvasSize.height() / canvasSize.width()
rotation = 0

# Create an exporter
exporter = Exporter(iface, settingsPath)
exporter.setMapSettings(mapSettings)

for name, point in places:
    # Coordinate transform
    center = transform.transform(point)
    # Set extent
    exporter.setExtent(center, width, height, rotation)
    # Output HTML file path
    filepath = path_tmpl.format(name)
    # Export
    err = exporter.export(filepath, openBrowser=False)
    if err == Exporter.NO_ERROR:
        print "{0} has been exported to {1}".format(name, filepath)
    else:
        print "Failed to export {0}: {1}".format(name, err)
```

Exported scene examples:

- Kyoto, Osaka, Nara, Ehime

Sources: Geospatial Information Authority of Japan. [GSI Tiles](#) (Orthophoto and elevation tile)



See *Programmatic Export Using Python* for an example of usage.

## Exporter class

**class** `api.Exporter` (*iface=None, settingsPath=None*)  
A convenient class to export the scenes to web programmatically

**\_\_init\_\_** (*iface=None, settingsPath=None*)  
Constructor.

### Parameters

- **iface** (*QgisInterface*) – If specified, mapSettings attribute is initialized with the map settings of the map canvas.
- **settingsPath** (*unicode*) – Path to an existing settings file (.qto3settings).

**export** (*htmlPath, openBrowser=False*)  
Do export.

### Parameters

- **htmlPath** (*unicode*) – Output HTML file path.
- **openBrowser** (*bool*) – If True, open the exported page using default web browser.

**Returns** `Exporter.NO_ERROR` if success. Otherwise returns error message.

**Return type** None or unicode.

**setExtent** (*center, width, height, rotation=0*)  
Set map extent to export settings.

This is a convenience method to set map extent to export settings. Map settings should be set before this method is called.

### Parameters

- **center** (*QgsPoint*) – Center of the map extent in the map CRS.
- **width** (*float*) – Width of the map extent in unit of the map CRS.
- **height** (*float*) – Height of the map extent in unit of the map CRS.
- **rotation** (*float*) – Rotation in degrees. Requires QGIS version 2.8 or later.

**setMapSettings** (*mapSettings*)

Set map settings to export settings.

Map settings is used to define base extent of the export and render a map canvas image.

**Parameters** **mapSettings** (*QgsMapSettings*) – Map settings to be set.

## CHAPTER 10

---

### Development

---

This plugin is now being developed personally by [Minoru Akagi](#), and received many contributions.

Many thanks to all the [contributors](#)! Thanks also to bug reporters and everyone who gave me constructive suggestions!

Source Code: <https://github.com/minorua/Qgis2threejs/>

Issue Tracker: <https://github.com/minorua/Qgis2threejs/issues>



## Symbols

`__init__()` (api.Exporter method), 29

## B

Box (Line Layer), 19

Box (Point Layer), 16

## C

COLLADA model, 17

Cone (Line Layer), 18

Cone (Point Layer), 16

Cylinder, 15

## D

Disk, 16

## E

`export()` (api.Exporter method), 29

Exporter (class in api), 29

Extruded, 19

## I

Icon, 17

## J

JSON model, 17

## L

Line, 18

## O

Overlay, 20

## P

Pipe, 18

Profile, 19

## S

`setExtent()` (api.Exporter method), 29

`setMapSettings()` (api.Exporter method), 30

Sphere, 15