

---

# **xSGE TMX Library Documentation**

***Release 1.1***

**onpon4**

April 10, 2017



## CONTENTS

<b>1</b>	<b>xsge_tmx Classes</b>	<b>3</b>
<b>2</b>	<b>xsge_tmx Functions</b>	<b>5</b>
	<b>Python Module Index</b>	<b>7</b>
	<b>Index</b>	<b>9</b>



## Contents

- *xSGE TMX Library*
  - *xsgc\_tmx Classes*
  - *xsgc\_tmx Functions*

xSGE is a collection of extensions for the SGE licensed under the GNU General Public License. They are designed to give additional features to free/libre software games which aren't necessary, but are nice to have.

xSGE extensions are not dependent on any particular SGE implementation. They should work with any implementation that follows the specification.

This extension provides support for loading the [Tiled](#) TMX format. This allows you to use Tiled to edit your game's world (e.g. levels), rather than building a level editor yourself.

To load a TMX map, simply use `xsgc_tmx.load()`. See the documentation for this function for more information.



## XSGE\_TMX CLASSES

```
class xsge_tmx.Decoration (x, y, z=0, sprite=None, visible=True, active=False,
    checks_collisions=False, tangible=False, bbox_x=None, bbox_y=None,
    bbox_width=None, bbox_height=None, regulate_origin=False, collision_ellipse=False,
    collision_precise=False, xvelocity=0, yvelocity=0, xacceleration=0, yacceleration=0,
    xdeceleration=0, ydeceleration=0, image_index=0, image_origin_x=None, image_origin_y=None,
    image_fps=None, image_xscale=1, image_yscale=1, image_rotation=0,
    image_alpha=255, image_blend=None, image_blend_mode=None)
```

Default class for tiles and image layers. Identical to `sge.dsp.Object`, except that it is intangible and doesn't check for collisions by default.

```
class xsge_tmx.Rectangle (x, y, z=0, sprite=None, visible=False, active=True, checks_collisions=True,
    tangible=True, bbox_x=None, bbox_y=None, bbox_width=None,
    bbox_height=None, regulate_origin=False, collision_ellipse=False,
    collision_precise=False, xvelocity=0, yvelocity=0, xacceleration=0,
    yacceleration=0, xdeceleration=0, ydeceleration=0, image_index=0,
    image_origin_x=None, image_origin_y=None, image_fps=None,
    image_xscale=1, image_yscale=1, image_rotation=0, image_alpha=255,
    image_blend=None, image_blend_mode=None)
```

Default class for rectangle objects. Identical to `sge.dsp.Object`, except that it is invisible by default.

```
class xsge_tmx.Ellipse (x, y, z=0, sprite=None, visible=True, active=True, checks_collisions=True,
    tangible=True, bbox_x=None, bbox_y=None, bbox_width=None,
    bbox_height=None, regulate_origin=False, collision_ellipse=True,
    collision_precise=False, xvelocity=0, yvelocity=0, xacceleration=0,
    yacceleration=0, xdeceleration=0, ydeceleration=0, image_index=0,
    image_origin_x=None, image_origin_y=None, image_fps=None,
    image_xscale=1, image_yscale=1, image_rotation=0, image_alpha=255,
    image_blend=None, image_blend_mode=None)
```

Default class for ellipse objects. Identical to `sge.dsp.Object`, except that it is invisible and uses ellipse collision detection by default.

```
class xsge_tmx.Polygon (x, y, points=(), z=0, visible=False, tangible=False, **kwargs)
```

Default class for polygon objects. Identical to `xsge_path.Path`.

```
class xsge_tmx.Polyline (x, y, points=(), z=0, visible=False, tangible=False, **kwargs)
```

Default class for polyline objects. Identical to `xsge_path.Path`.





## XSGE\_TMX FUNCTIONS

`xsgt_mtm.load(fname, cls=<class 'sge.dsp.Room'>, types=None, z=0)`

Load the TMX file `fname` and return a room of the class `cls`.

The way the map generates the room, in general, is to convert all tiles, objects, and image layers into `sge.dsp.Object` objects. As a special exception, the object layer with the name “views” defines the views in the room; these objects are converted into `sge.dsp.View` objects.

Objects are given Z-axis positions based on the ordering of the layers in the TMX file: `z` is the Z-axis position of the first layer, and each subsequent layer’s Z-axis position is the Z-axis position of the previous layer plus one.

Except for views, all tiles, objects, and image layers can be defined to be converted into any class derived from `sge.dsp.Object` via the `types` argument, which should be a dictionary matching strings to corresponding `sge.dsp.Object` classes, or `None`, which is equivalent to `{ }`. Classes are determined in the following ways:

- Tiles are converted to the class connected to, in order of preference, the name of the tileset or the name of the tile layer. If neither of these strings are valid keys in `types`, `xsgt_mtm.Decoration` is used.
- Objects are converted to the class connected to, in order of preference, the name of the object, the type of the object, the appropriate class for the respective tile if applicable (see above), or the name of the object group. If none of these strings are valid keys in `types`, the class used depends on what kind of object it is:
  - Rectangle objects default to `xsgt_mtm.Rectangle`.
  - Ellipse objects default to `xsgt_mtm.Ellipse`.
  - Polygon objects default to `xsgt_mtm.Polygon`.
  - Polyline objects default to `xsgt_mtm.Polyline`.
  - Tile objects default to `xsgt_mtm.Decoration`.
- Image layers are converted to the class connected to the image layer’s name. If the image layer’s name is not a valid key in `types`, `xsgt_mtm.Decoration` is used.

Property lists, converted to integers or floats if possible, are passed to objects as keyword arguments in the following ways:

- Tiles have their properties, the properties of their tilesets, and the properties of their layers applied to them. Tileset properties override layer properties, and tile properties override tileset properties.
- Tile objects have their properties, the properties of their tiles, the properties of their tiles’ tilesets, and the properties of their object groups applied to them. Object properties override tile properties, tile properties override tileset properties, and tileset properties override object group properties.
- Other objects have their properties and the properties of their object groups applied to them. Object properties override object group properties.

- Image layers have their properties applied to them.

**X**

`xsgc_tmx`, 1



## D

Decoration (class in xsge\_tmx), 3

## E

Ellipse (class in xsge\_tmx), 3

## L

load() (in module xsge\_tmx), 5

## P

Polygon (class in xsge\_tmx), 3

Polyline (class in xsge\_tmx), 3

## R

Rectangle (class in xsge\_tmx), 3

## X

xsge\_tmx (module), 1