

Export XYZ 1.0.0 for SketchUp 2014 and higher

© D. Bur - January 2020

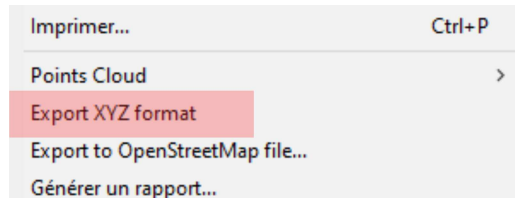
Export XYZ is the resulting plugin following experiments to export terrains from SketchUp to TwinMotion (free version 2019).

The import terrain feature of TM is not documented and somewhat strange, so this document explains what you can achieve with this plugin and the workarounds to avoid drawbacks and inconvenience of the TM import process.

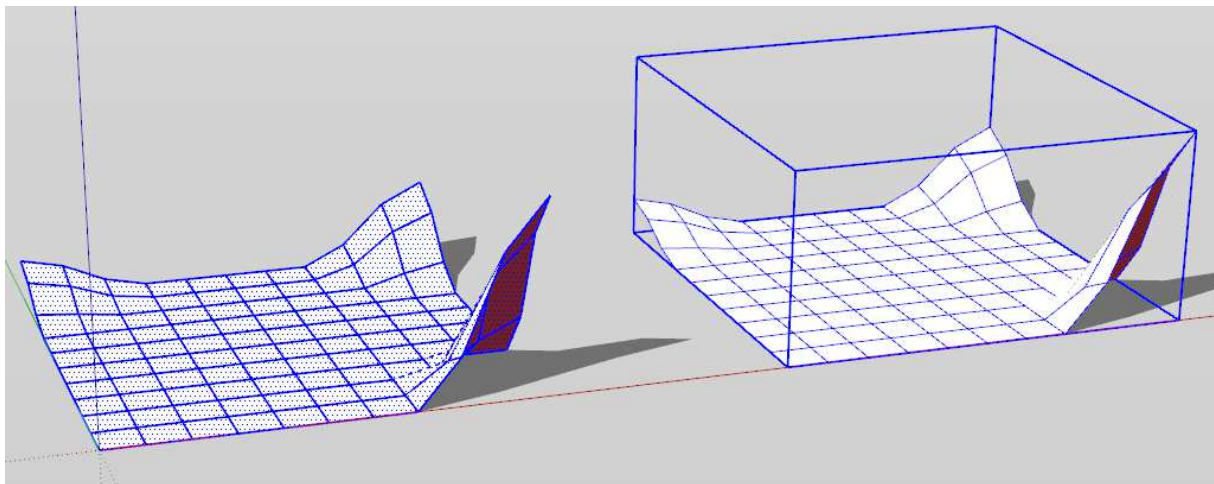
Installation: just put the RBS file in your Plugins folder and that's it. Re-start SketchUp.

Footprint: just a RBS file in your Plugins folder.

Usage: select "Export XYZ format" from the "File" menu.



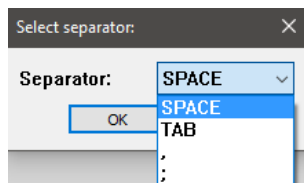
You have to select either a group of faces or faces in the model prior to use the command. Objects other than faces will be ignored.



2 valid selections

Set the name of the exported file in the save dialog box.

Select separator:



A report message is displayed as the process ends.



The XYZ format:

In fact, there are many variants of this format, which is mainly used out of LIDAR scanners and such.

Each line of the file represents a point in 3D space, through its 3 coordinates X, Y, and Z separated by a separator (space, comma, colon, semicolon), other properties can be added to each line.

A header can be present on the first lines of the file, or not.

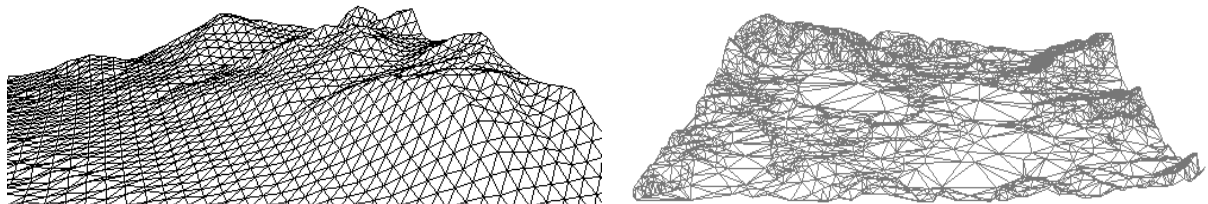
4 lines describing 4 points set a quad (or cell) of the resulting mesh.

As said here: <https://gdal.org/drivers/raster/xyz.html> :

The datasets are ASCII files with (at least) 3 columns, each line containing the X and Y coordinates of the center of the cell and the value of the cell.

The spacing between each cell must be constant and no missing value is supported. Cells with same Y coordinates must be placed on consecutive lines. For a same Y coordinate value, the lines in the dataset must be organized by increasing X values. The value of the Y coordinate can increase or decrease however.

We could think that TwinMotion would only require a TRN (triangular regular network) mesh, but TIN (triangular irregular network) will be accepted as well.



Valid TRN and TIN

So terrains made with the sandbox tool or manually drawn will be OK. Terrain shape seen from a top-down view can be regular (square, rectangle) or irregular.

If the shape is irregular, resulting terrain shape will be convex hull (as in SketchUp).

If there are holes in the mesh, resulting terrain will be continuous (no holes).

Unit:

You can use every unit to draw your terrain, the plugin will convert them to meters.

Grid size (or cell size, or quad size): any size can be used. The smaller the size, the highest accuracy of the resulting terrain.

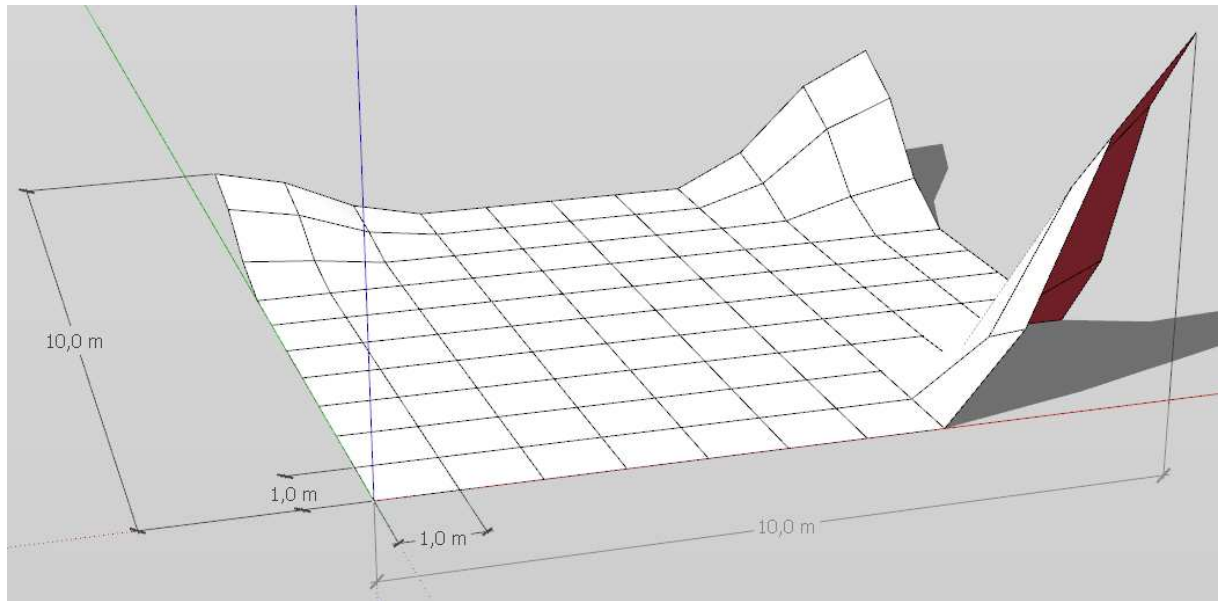
Mesh size: any total length or width of the mesh is accepted. Be warned: a mesh of say 10x10 quads of 1 meter x 1 meter will generate a TwinMotion terrain of XXXX "polygons".

Compared to a Google Earth terrain capture within SketchUp, the advantages of importing XYZ terrains in TwinMotion are:

- easy "painting" or mixed texturing (4 materials per terrain available)
- many sculpting features to adjust the terrain shape.

- more comfortable vegetalization
- better seasonal variations of the textures applied.
- more realistic relief (SketchUp captures are not so accurate, depending on the location of your project and the corresponding Google Earth DEM)

1st experiment:



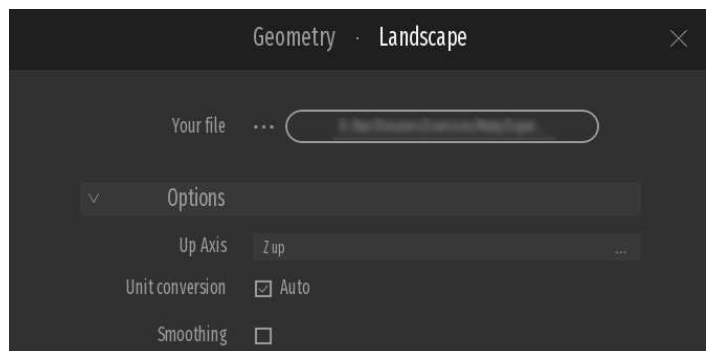
Test terrain: a regular mesh (made of pairs of 2 triangles), 10x10 squares, square size 1m.

Import XYZ format in TwinMotion:

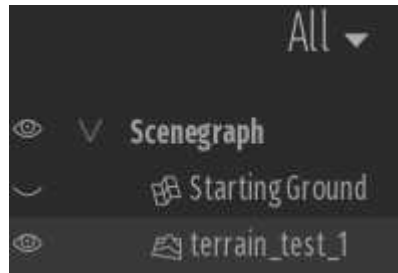
Click on the **Import** icon:



Select your XYZ file and options (leave the **Up Axis** option set to **Z up**, the **Unit conversion** option set to **Auto**, and **Smoothing** unticked).



Resulting hierarchy:

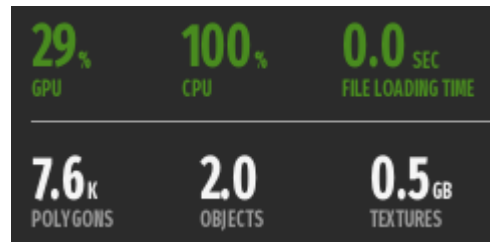


- Hide or delete the **Starting Ground** object.
- Select the terrain in the scene graph and type hit the **F** key (shortcut to focus on the selection)
- Click on **Nature Localization** to geolocalize the terrain, if needed.



Statistics:

A 10x10 quads mesh (200 triangles in SketchUp) gives 7600 so-called polygons (triangles I guess). Surprising.
TM file size: 18k



Apart from that, as you can see there are some strange things happening:

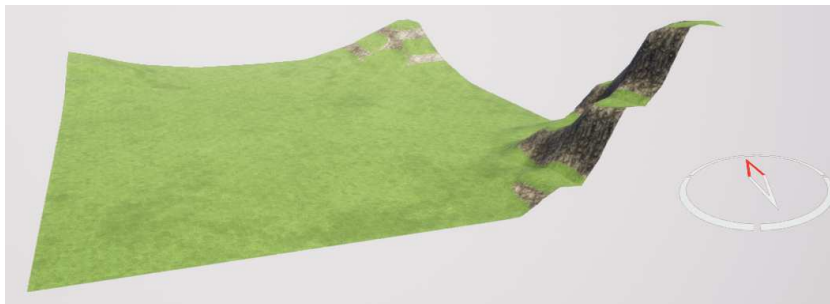
- SketchUp geolocation (if any) is lost,
- hills and valleys modify as you move back and forth (this strange behavior decreases as the mesh accuracy increases).
- the **Move** gizmo isn't at its original location in SketchUp (lower-left corner) but at the center of the terrain bounding box, no matter the way points are exported.
- there are less triangles than in SketchUp, although the terrain is made of 7600 triangles.
- once imported, such terrains aren't scalable anymore.

Side note:

- the first code of the script was exporting the points in an ordered fashion (lines of increasing X's and columns of increasing Y's, TwinMotion crashed almost every time. More than that, when it didn't crash, the terrain was rotated 180° and symmetrized East-West.

2nd experiment:

Same XYZ file, same options except imported at a scale of 10:



89, GPU	95, CPU	0.0 SEC FILE LOADING TIME
32.8K POLYGONS	1.0 OBJECTS	0.5 GB TEXTURES

Same XYZ file, same options except imported at a scale of 10 and **Smoothing On**:



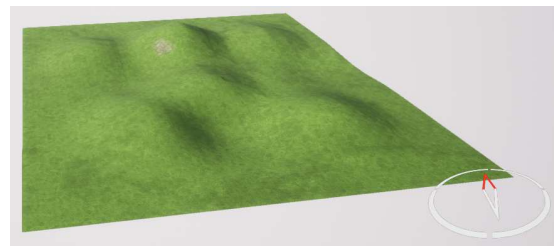
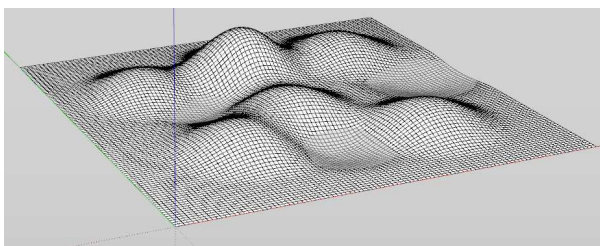
89, GPU	95, CPU	0.0 SEC FILE LOADING TIME
32.8K POLYGONS	1.0 OBJECTS	0.5 GB TEXTURES

- terrain is more detailed.
- the number of polygons is 4 times the number at scale 1, whether smoothing is On or Off.
- TM file size: 309k

Side notes:

- 2 default textures are applied: grass for the flat or nearly flat polygons and mossy rocks as the slope increases.
- if imported with a scale factor of 100, the number of polygons increases to 2,1 millions of polygons, equivalent to 64 times the number of polygons at scale 10 or 256 times more than the number of polygons at scale 1. TM file size would then be 103268k.

3rd experiment:

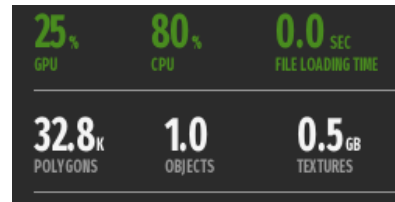


Test terrain: regular mesh (made of pairs of 2 triangles), 100x100 squares, square size 1m, imported at scale 1 ([Unit conversion](#) 1), smoothing On.



Statistics:

- Exactly the same numbers of polygons than a 10x10 mesh imported at scale 10.



So the amount of points doesn't seem to be of importance in regard to the resulting number of polygons in the TwinMotion terrain, but the overall dimensions of the mesh instead.

Terrain materials:

As said before, one of the advantages of using a true TwinMotion terrain object (instead of a SketchUp terrain) is the painting ability: mix up to 4 textures (materials) in a single map that will be applied on the terrain.

If you want to use your own textures (e.g. a Google Earth capture) to paint the terrain, you'll have to prepare them first.

Use of a specific terrain texture:

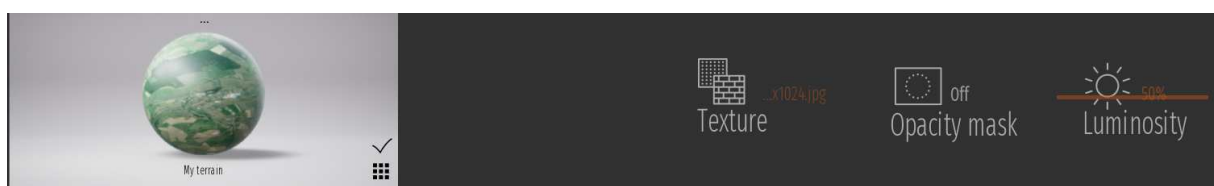
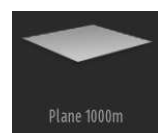
Typically this is a scanned geographic map, a screen capture in Google Earth or Google maps. Proceed as follows:



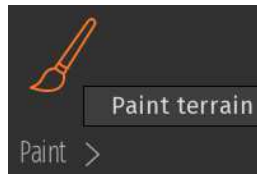
Terrain image sample, 1024x1024

- resize your screenshot to a square image with a number of pixels power of 2 (512, 1024...).

- insert a Plane 1000m primitive (from the [Library/Furnitures/Primitives](#) catalog,
- click on it with the eyedropper, duplicate the base material, rename it,
- click on [Color More Texture](#) and select the [Open](#) option.
- select your image

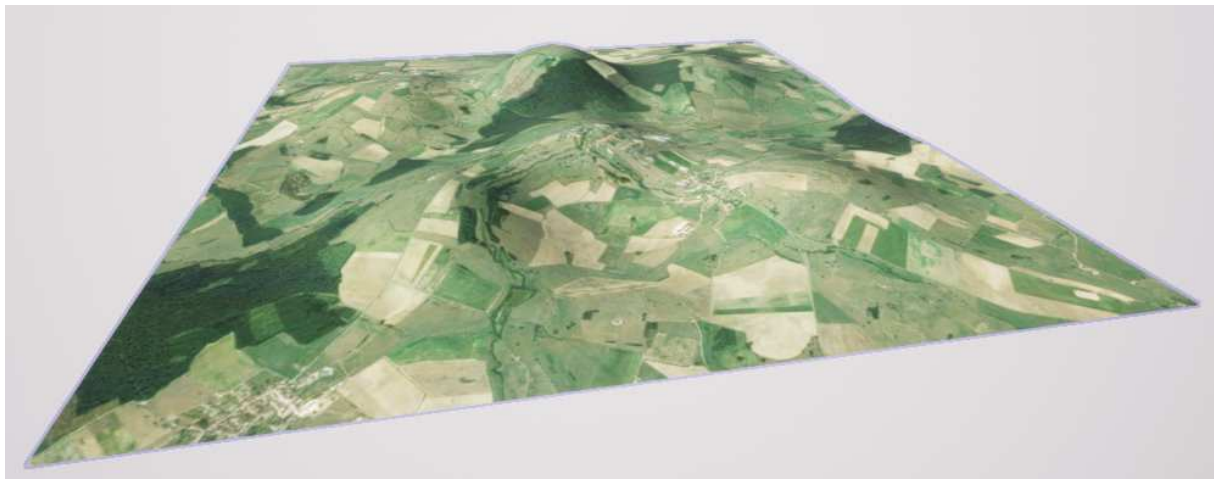
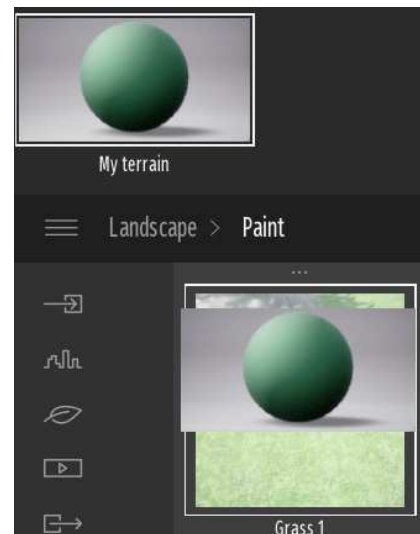


- set the reflection material property to 0% and adjust the scale so it fits the 1000m plane.
- click on the snowballs menu on top of the material thumbnail and save it to your user library.
- delete the plane object
- select your terrain and click on **Paint**



- Set the scale of the future replaced material to 100% BEFORE replacing it.
- browse your user library and drag-drop the material you just created on one of the 4 textures available, preferably **Grass1**. One image sample covers now the entire terrain.

- adjust the texture scale. Scale in TwinMotion is an integer percentage and it is increasing the texture size exponentially. For instance, a texture of 1024x1024 mapped on a 1000x1000m plane at a scale of 99% will cover the entire terrain, while at scale 100% only half the texture will be mapped. Dont ask me why :-)



Use of generic textures:

- use highres seamless textures like this one



Generic seamless terrain map, 3072px3072p

- set texture scale to 400%
- applied to a 1000mx1000m terrain: 1 texture sample covers the entire plane, same goes when applied to a XYZ terrain.

