

High map plug-in

Purpose

This plug-in is used for creation of polygonal (triangular) models from so called “height map” pictures. It could be very handy when modeling a terrain mesh (terrain relief), since the modeler defines the required surface structure via intuitive painting.

Input data

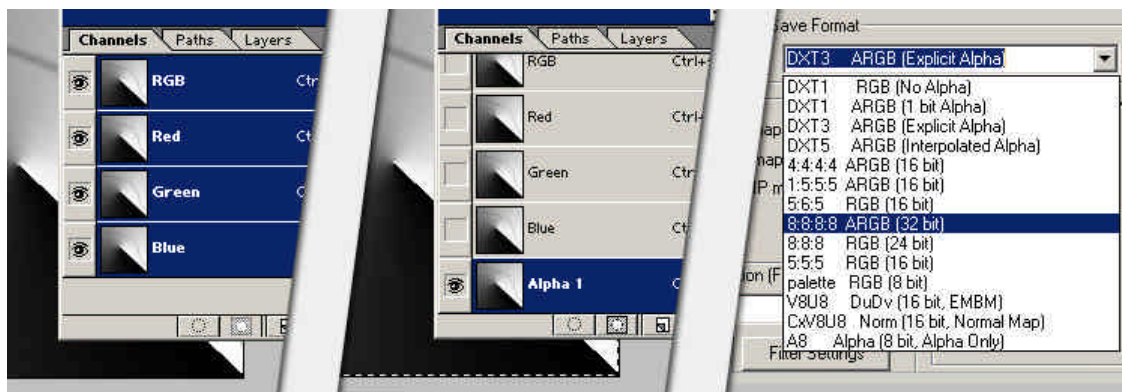
To successfully create height map based mesh, it necessary to prepare high (elevation) data, encoded into the texture. The data are read from alpha channel of the texture, so its height (or depth) range could be defined with 8-bit precession (0-255 units). The bright colors represent hills and darker ones stand for valleys. Texture must be of DDS format.

Texture preparation

It is advisable to use PHOTOSHOP accompanied with NVIDA DDS plug-in (the 5.45 version is most up to date one).

(For PintShopPro go <http://www.sportplanet.com/tigerwoods/proshop/dds.asp>)

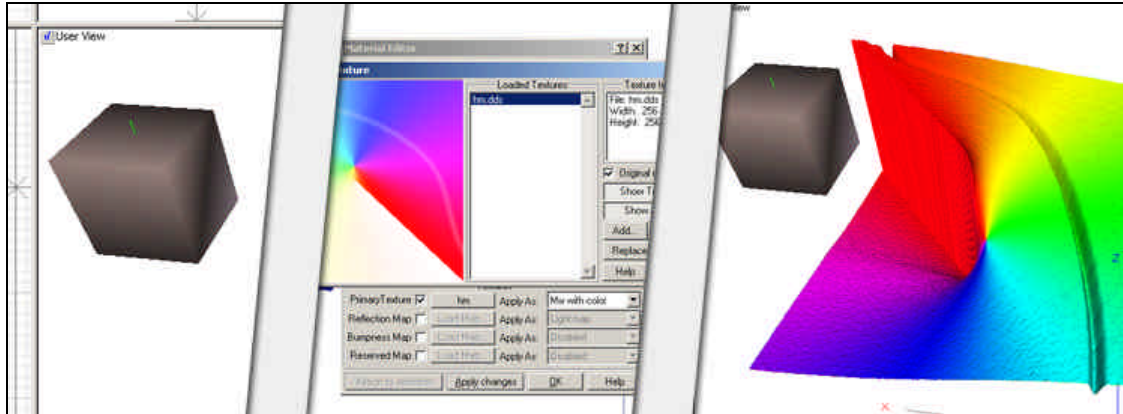
First create an empty canvas (it should be a square of 2n side). Choose gray color 50% brightness for background. Than use a gray-scale colored brushes to define a structure. Second step is to create new channel (let's say 'Alpha 1') and copy the painted picture onto it. Texture should be saved in DDS format, forcing the alpha component to be included. Choose some with alpha from a DDS format choice (it is recommended to use 8:8:8:8 uncompressed). The procedure is best to understand from the following picture.



Usage

First of all it is necessary to create arbitrary object which the terrain will be derived from. Second, assign the height-map to the object as a primary texture. Chose Terrain tool under Create menu placed on the main toolbar. Than click the object and set up height

coefficient into pop-up dialog. A new object with the same material is created (with suffix '_Terrain'). The whole procedure is shown on above picture.



UV coordinates of the terrain mesh are projected from $-Y$ direction. Several high quality LODs can be obtained if the height-map is scaled down using a bi-cubic down sampling.

Constrains

Since Zmodeler has certain problems with local paths, it is necessary to place your height texture into the actual directory (that one where *.z3d was saved in/loaded from). Only one texture format is now supported, because it is hard to access already loaded texture data within Zmodeler and plug-in has to have its own texture loader/decoder. The plug-in is capable to create only so called 'regular grid' based surfaces, ranging from 4x4 to 512x512 pixels resolution.

Cylindrical coordinates mapping

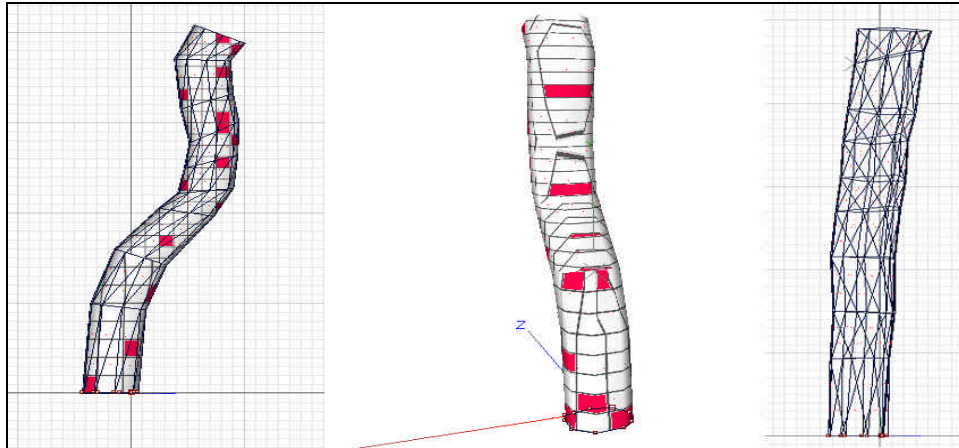
Purpose

This plug-in is used for mapping the texture on the object like on the cylindrical envelope.

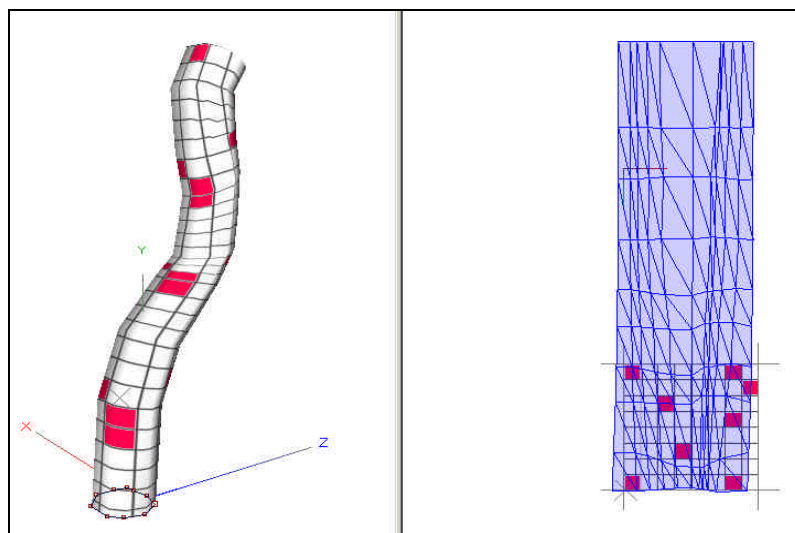
Usage

It does not push changes into the UNDO stack. This could sound like a fault, but it has been done in such a way intentionally. The thing is that cylindrical coordinates are computed with respect to the axis which goes through arithmetical center of the object. It is usual in common models that the center does not correspond with the intended place for texture projection (it is pretty common that it lays outside the object). If it is used

without any preparation, the result is not what we expected. (The next picture shows the situation, on the right hand side there is a straightened trunk.)



The solution is in straightening the object in the Y direction and mapping the object in this position. Then it should be deformed back into its original shape – best with ‘undo’ function. If there was the ‘undo’ for cylindrical coordinate tool implemented, the texture UV would be released or mapped like before. This is just overcome by relaxing of ‘undo’ function. The geometry is turned back but the texture UVs remain. See next picture where such procedure has been done on the trunk (that one straightened on the previous picture). It is also possible to scale UV to get appropriate look corresponding to the texture size.



Constraints

The coordinates are computed in the following way; the cylinder bases lay on XZ plane, meanwhile the height is mapped on Y axis. Thus the mapped object must head upwards.

Crease Tool Plug-In

Purpose

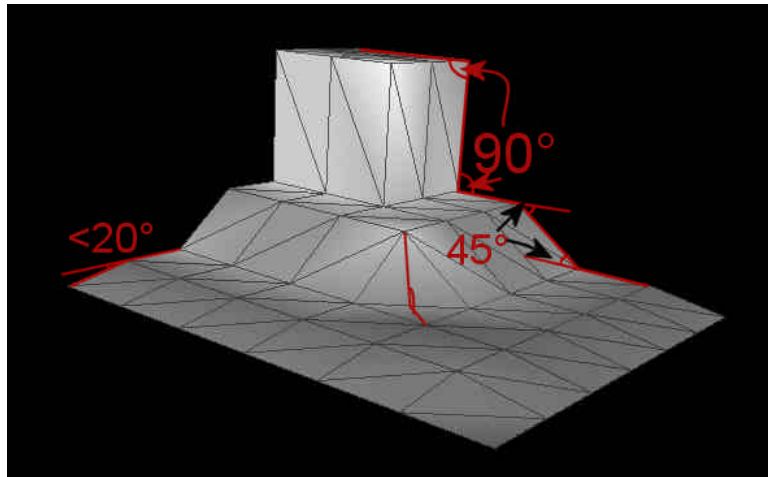
The Crease tool automatically creates detached hard edges in a mesh based on the angle of one poly to another. By specifying a cutoff angle, a user can automatically “find” all the creases in his mesh. The appropriate faces will be detached and re-united in one step. The tool works only on an entire object. There are no options available with the crease tool.

Use

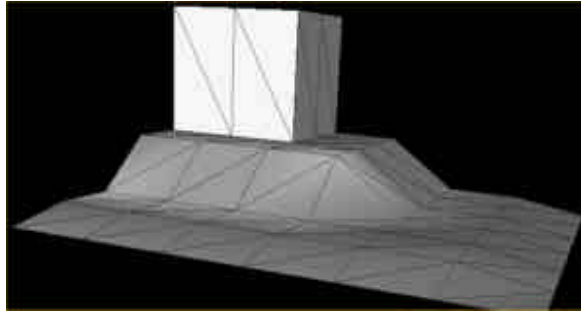
To use the Crease tool, simply click on an object. A box will appear and ask you for the Crease angle. This is the maximum angle between polies that should be ignored by the plug-in. For example, if you enter 89° , only polies with an angle of 90° or more will be creased. After using it on an object, the normals will be distorted, so it is advisable to re-calculate them.

Example

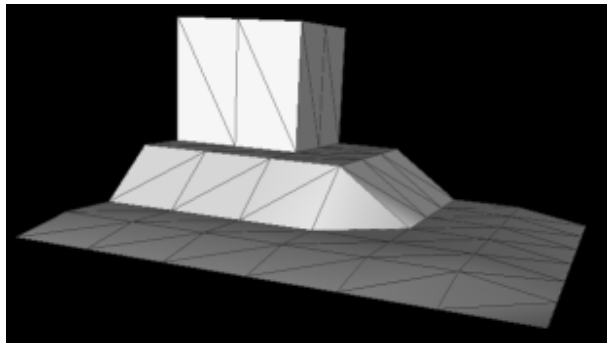
Consider this mesh as an example of how the crease tool works. In this mesh we have many different discernable angles. We have two approximate 45° angles, two 90° angles, and one less than 20° angle.



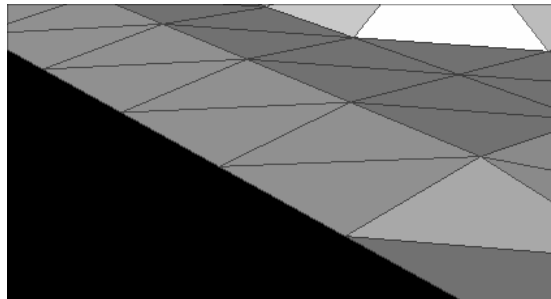
To demonstrate the crease, notice the changes that are made when it is applied with different settings. First, we apply crease with a smoothing angle of 89° :



As you can see, only the 90° angles have been creased. Now we will apply the Crease tool with an angle of 35:



At this point, the 45° angles have also been detached. Now let us try it with an angle of 10:



Now, all the angles have been detached. Note: If we had used the angle of 10 in the first step, all the angles would have been creased at one time.

Constraints

At times, it seems that the crease tool “finds” angles that aren’t really there. After applying the crease tool to an object, recalculate the normals and examine it carefully to make sure that no inappropriate creasing has occurred. Also, be sure that you want to apply crease over the ENTIRE object. Make sure no parts of the object are hidden, etc. As with all global object tools, use with caution.

Unification Plug-In

Purpose

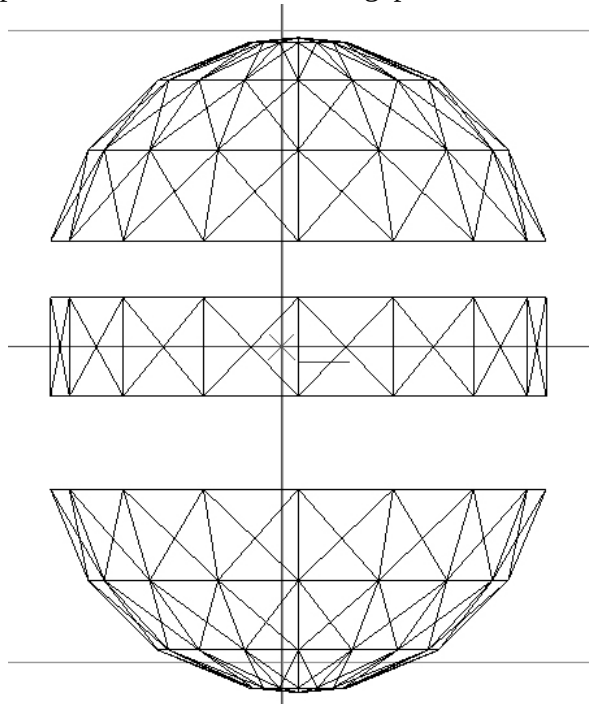
The Unification Plug-In is meant to automatically unite Vertices, from a selection, that are close to one another. The normals of the United vertices are also fused. There are two adjustable settings for the Unificator Plug-In. The user can disable normals fusing, and the user can specify the Unification radius.

Use

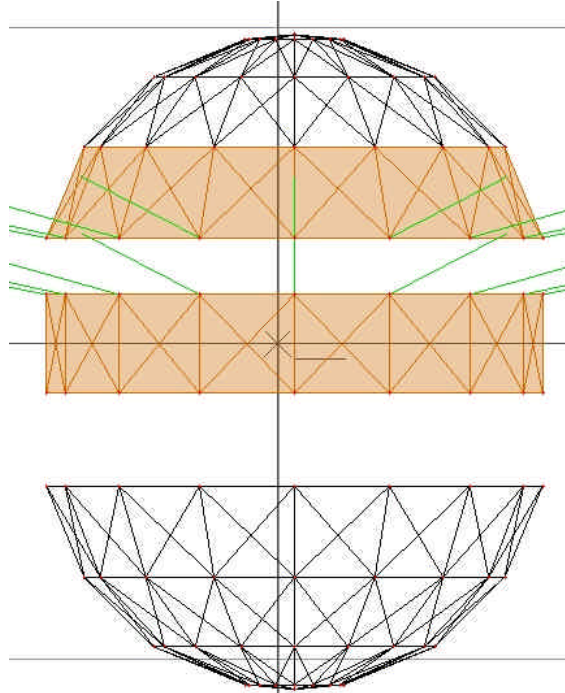
To use the Unificator Plug-In, First select the vertices that you wish to unite. Next, find the largest gap in the group of vertices that you have selected. Position the mouse over one of the two vertices with the largest gap, press Shift, and left-click the mouse. Release Shift, and drag the mouse until the blue circle encompasses the other vertex. Make sure that this radius is not so large that it will cause unwanted vertices to be united. (If you think this may happen, try moving some of the vertices closer together.) Once you have the correct radius specified, left-click to accept it. Next, just left-click again. All the verts that you have selected, that were within the specified radius of each other, will have been fused.

Example

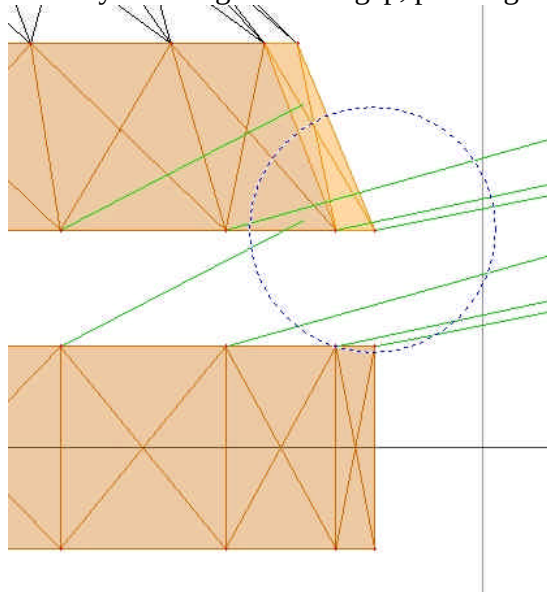
Consider an example of using the Unificator tool. To begin, assume we have an object in Z-Modeler with a gap in it. We wish to close that gap:



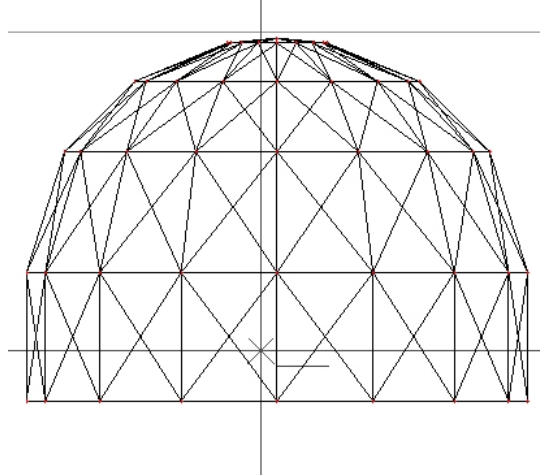
We switch to Vertex mode, and select the vertices of the top gap:



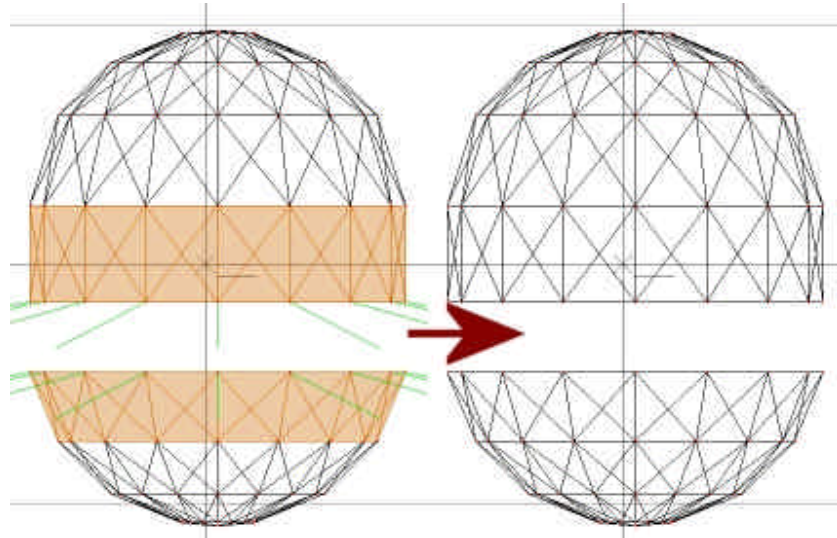
Next, we set the unite radius by zooming in on the gap, pressing shift, and left-clicking:



Finally, we perform the uniting by left clicking:



If we try to unite a larger gap, it will not work because we specifically set the unite radius:



Constraints

You should be very careful setting the unite radius. When uniting a group of vertices, if the gap is larger than the distance between vertices that should not be united, many vertices that should not be united will be united. It is best to use this tool with very small gaps, or to smooth hard edges created by detaching.