geomCheck 1.1

Lots of Pixels / Steve Baines

www.lotsofpixels.com www.astrofish.com

Description

'geomCheck' is a **free** plugin for Cinema4D 6.3 and above.

Its purpose is to check the geometry of polygon objects and to detect and report defects in the mesh. It detects various defects including open edges / cracks, degenerate triangles, ill-conditioned quad winding, and reversed normals. Each of these is described in the 'usage' section of this document.

If you have any questions or suggestions, please email *steve@lotsofpixels.com*, with 'geomCheck' in the title (to make sure your message gets past the spam filters).

If you find this plugin useful, please consider purchasing something from 'Lots of Pixels' . 'spinPoly' is particularly useful for fixing several of the problems that 'geomCheck' detects...

Installation

To install, download the file from www.lotsofpixels.com, and unzip the contents into your 'Plugins' directory. Start Cinema4D, and in the Plugins menu you should find an entry for 'GeomCheck'.

Usage

To use, simply select the object that you wish to check, and select 'GeomCheck' from the plugins menu. There are two versions of the plugin that you can select from the submenu:

1. ***Full Checks – SLOW***

   This performs the full set of tests on the geometry. Be warned that for complex meshes this can take a *very* long time.

2. ***Polygon Checks only***

   This version of the plugin performs a smaller set of tests on the geometry, and runs very rapidly. This version does *not* check for open edges, or reversed normals, as

they are the time consuming tests, but it does perform *all* other tests.

Both versions check for:-

Degenerate triangles ('Slivers'), Degenerate Quads ('Slivers'), Badly wound Quads (which can cause shading artefacts), and Co-located vertices within a polygon.

In addition, the 'full checks' version checks for:-

Open Edges / Cracks, and Reversed normals.

A list of all defects found will be displayed in the console tab (Window->Console, if you can't see it), all defective polygons will be selected, and all vertices which are associated with open edges/cracks will be selected. Use convert selection vertices -> edges to see the edges directly. (Note: This is not done automatically because this plugin is written in Coffee, which cannot access edge selections).

By switching to vertex mode or polygon mode, you can immediately see where the problem areas on your mesh are (if any).

By looking at the console tab messages, you can see what exactly was wrong in each case.

Each of the messages that you may see in the console are now described.

1. Edge from x to y is open

There is an edge between vertex 'x' and vertex 'y', which is only used by a single polygon. If this is the edge of a sheet of polygons and is deliberately open then it is not necessarily a problem. Whenever it occurs within a mesh it indicates a crack in the mesh. This can cause shading problems and unexpected subdivision effects. These cracks can be very hard to spot manually. This can be repaired by creating extra polygons to seal up the gap. (But beware of creating ill-conditioned polygons, described below).

2. Poly x has reversed normal

This means that polygon 'x' has a normal which points in the opposite direction to its neighbours and is inconsistent. Repair this using the 'Reverse Normals' or 'Align Normals' tools in C4D.

3. Poly x is Sliver Tri (ill-conditioned normal)

Polygon 'x' is a co-linear triangle, i.e. All of it's vertices are in a straight line and it has no area. It also has no well-defined normal, which can cause shading artefacts. As you rotate the viewport you will typically see the normal marker for these triangles jumping around in random directions. This can be repaired by either merging with an adjacent triangle, using spinPoly to spin with any adjacent polygon, or moving one of the vertices so that they are no longer all in a line.

4. Poly x is Bad quad (ill-conditioned normal)

Polygon 'x' is a quad in which the first two edges are co-linear (in a straight line). Although the quad appears to have a well defined area and normal, Cinema4D will not calculate it correctly, and the displayed normal will 'twitch' as the view is rotated. Spinning the quad three times (to cycle through both possible triangulations, and then back to a quad) with

spinPoly will repair this, because when merging tris into quads, spinPoly always chooses the start vertex of the quad such that the quad is well-conditioned. Alternatively, you can delete and then recreate the polygon manually. If you do this then make the first vertex of the quad the one in the middle of the two straight edges.

5. Poly x is Bad quad (ill-conditioned secondary normal)

Polygon 'x' is a quad in which the last two edges are co-linear (in a straight line). This is similar to the previous case. In this case the normal marker will display correctly (it's calculated by C4D from the first two edges), but when C4D splits the quad into triangles internally for rendering, the second triangle will again be ill-conditioned. The solution in this case is the same as in case 4 above.

6. Poly x is Bad quad (inconsistent primary/secondary normals)

Polygon 'x' is a quad which when split into two triangles internally by C4D for rendering, is likely to have strong shading changes across its surface which are undesireable. Again, as in 4 and 5, the solution is to change the order of listing of the vertices which define the quad so that it starts with a different vertex.

7. Poly x has bad vertex structure (duplicated vertices)

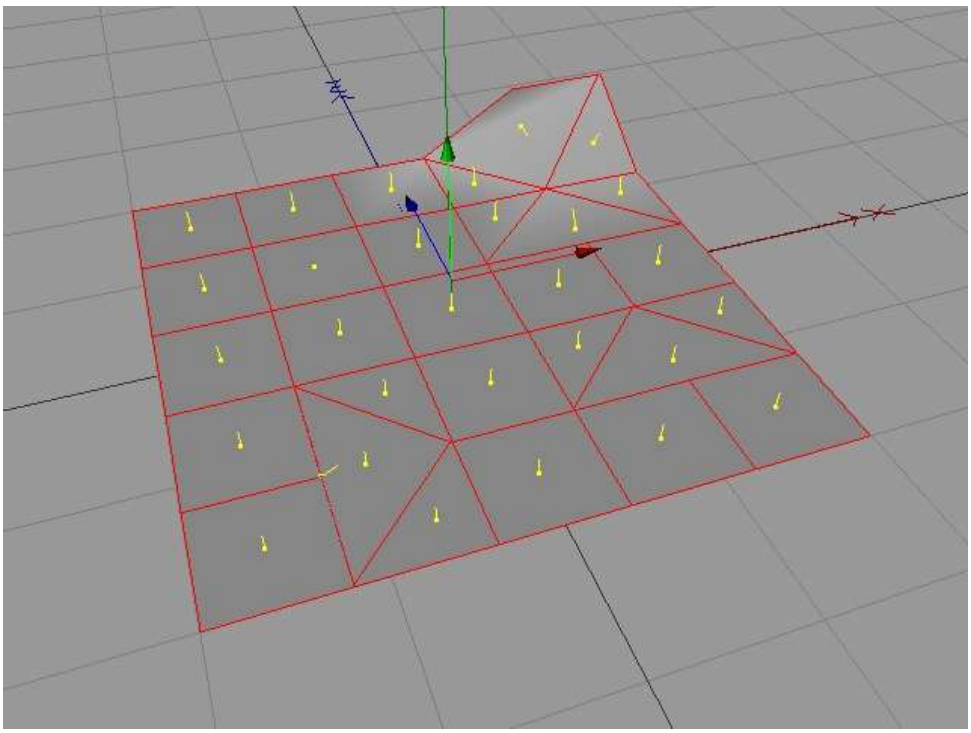
Polygon 'x' has the same vertex index used more than once, (and is not a valid triangle). Normally, every vertex index used in a polygon should be different, with the one exception being that C4D represents triangles as quads with the last two vertices the same. This warning indicates that some other pair of vertices are identical, which doesn't represent a valid polygon.

8. Poly x has co-located vertices

At least two vertices from Polygon 'x' are in exactly the same position (though they have different vertex numbers). Polygons with this problem will also report problems with normals.

Example

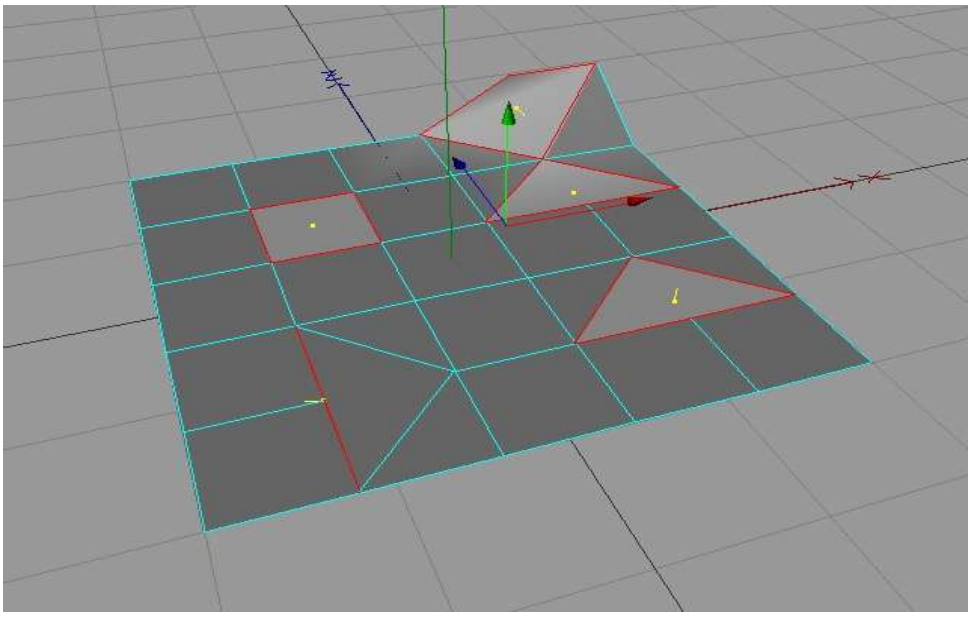
Load the file 'example.c4d' file. The viewport should look like this:-



Try rotating the view and note that some of the normals jitter about, which indicates problems with the mesh. Also note that the quad at top right has unexpected shading and displayed normal direction.

Choose 'Plugins->GeomCheck -> Full Checks' to run the plugin.

The display now looks like this. There are five seperate polygons which have problems, which have all been selected.



If you click on the Console tab, then you will see what the id numbers of the affected polygons are, and what particular problem they have. There is one example of each of the first five types of polygon problems from the above list.

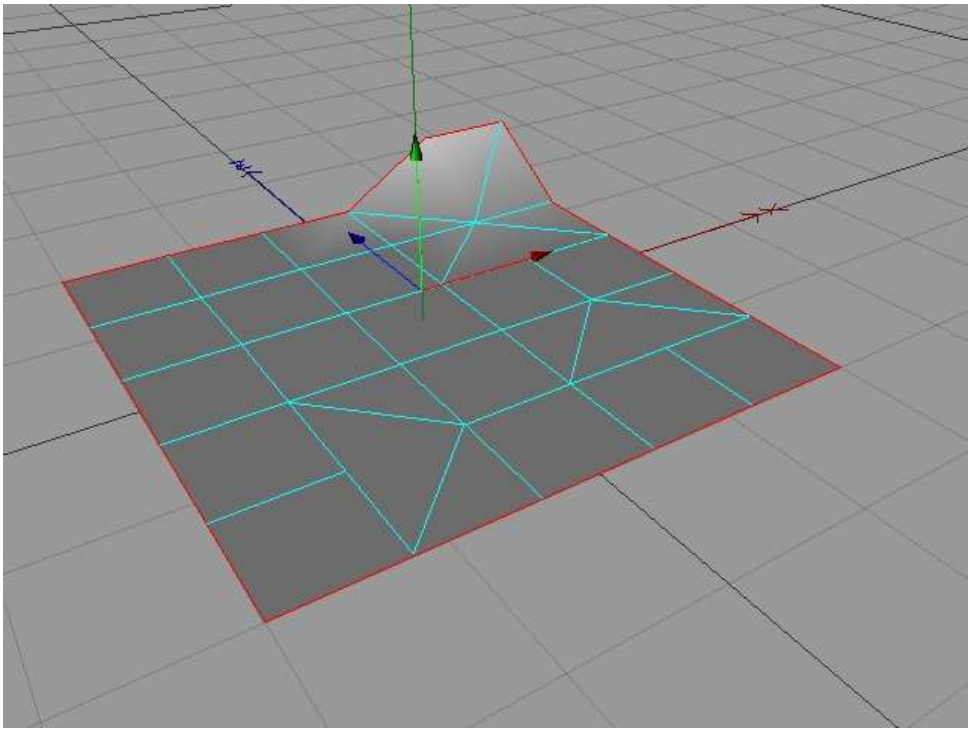
The three quads on the right can all be repaired by using spinPoly on each of them three

times. The 'sliver' triangle can be repaired by selecting it and the adjacent tri, and spinning them into a single quad. If you don't have spinPoly then the easiest solution is to delete those polygons and create new ones. The square polygon with the reversed normals can be repaired with the 'reverse Normals' tool.

If you run 'geomCheck' again, none of the polygons should be selected. Select all, so that you can see the normals, and rotate the view again. This time none of the normals should twitch. The shading on the top-right quad should be much smoother also.

Finally, use 'Convert Selection' to convert the vertex into an edge selection, to see if we have any cracks.

The view now looks like this:



Note that the only edges selected are the perimiter edges. These are intentionally disconnected, so they are not a problem. There are no other edges selected, so this means that there are no cracks in the mesh. We have finished!

Note: In certain circumstances conversion from vertices to edges will result in some edges being selected that do not actually have a problem. The plugin cannot select edges directly to prevent this, because of limitations in the Coffee scripting interface to Cinema4D. In practice these situations are not particularly likely, however, unless you have a lot of open edges / cracks. The vertex selection will always be correct, the problem is that knowing which vertices are affected is not always enogh information to know which edges are affected.

I hope you find this plugin useful.

Please have a look around at www.lotsofpixels.com and www.astrofish.com

This version of the plugin (1.1) originally released: 8th February 2004.
This documentation updated 12 June 2004.

End of document.