

# QuadRemesher™ – User Documentation

QuadRemesher™ is an automatic Quad-remeshing algorithm.

It's available as plugins for 3D DCC softwares such as:

- Autodesk® 3ds Max® Software.
- Autodesk® Maya® Software.
- Foundry® Modo® Software
- Blender® Software
- MAXON Cinema4D® Software
- Side Effects Houdini® Software.

QuadRemesher™ allows to perform a quad-remeshing from any input mesh in a single-click.

It takes as input a mesh made of triangles or polygons and return a mesh mainly composed of quadrangles following natural directions.

You can find tutorial and video on [www.exoside.com](http://www.exoside.com)

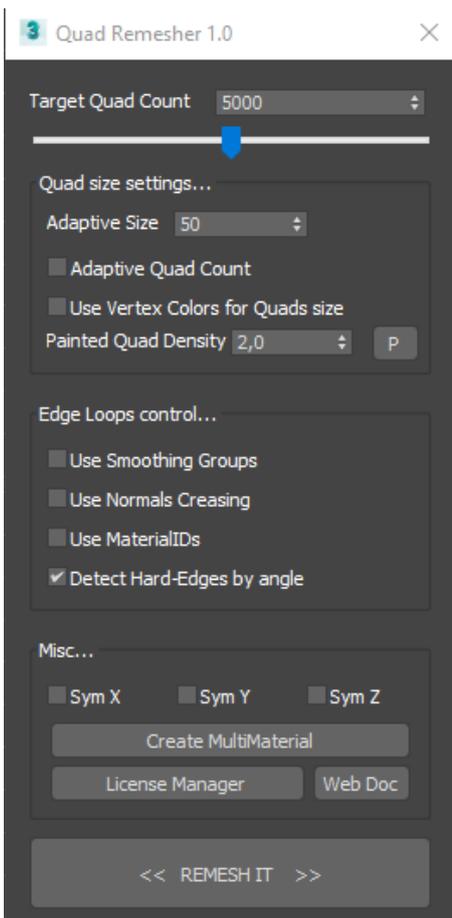
It's compatible with these versions:

- Modo : from modo 10, for Windows and macOS
- Maya : from Maya 2012 for Windows and macOS
- 3ds Max: from 3ds Max 2016
- Blender : from Blender 2.80
- Houdini : tested on Houdini 17.0 and higher (should work well on previous versions)
- Cinema4D: compatible with Cinema4D R14 and higher.

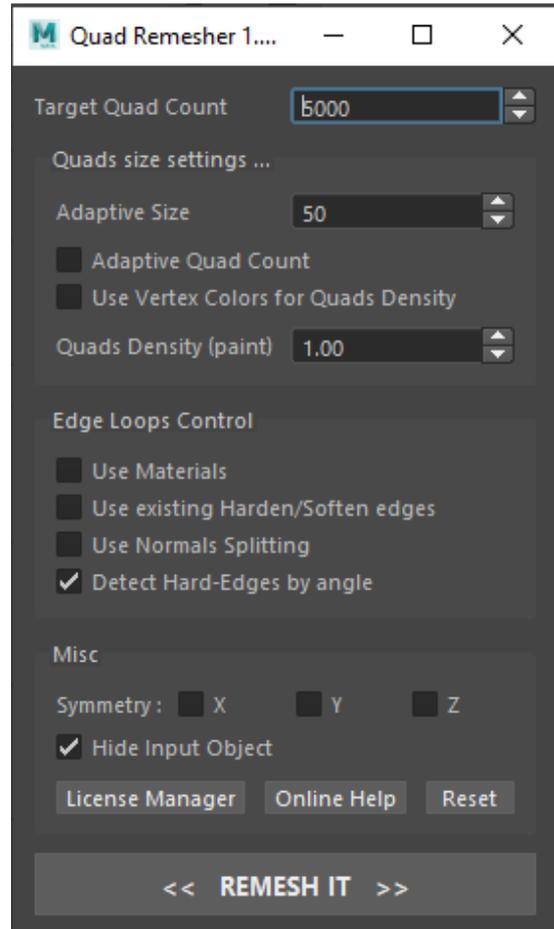
You can find the list of improvements in all version of QuadRemesher in this document:

[QuadRemesher WhatsNew.pdf](#)

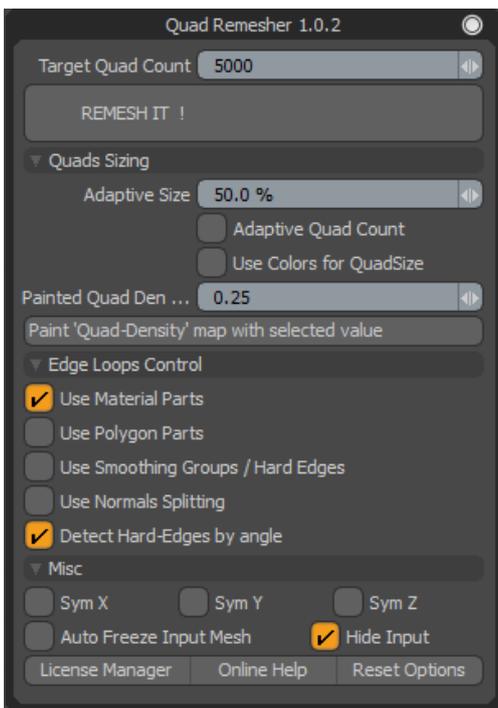
Here are some examples of the user-interfaces of the plugins:



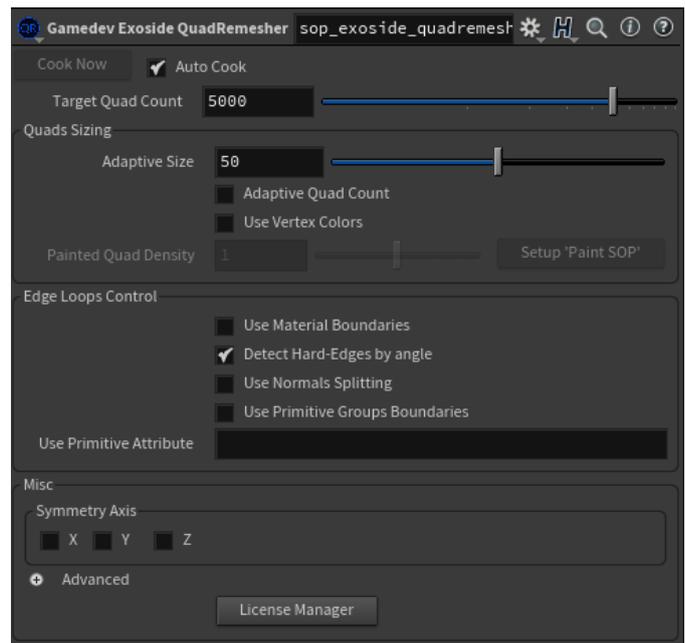
QuadRemesher for 3ds Max



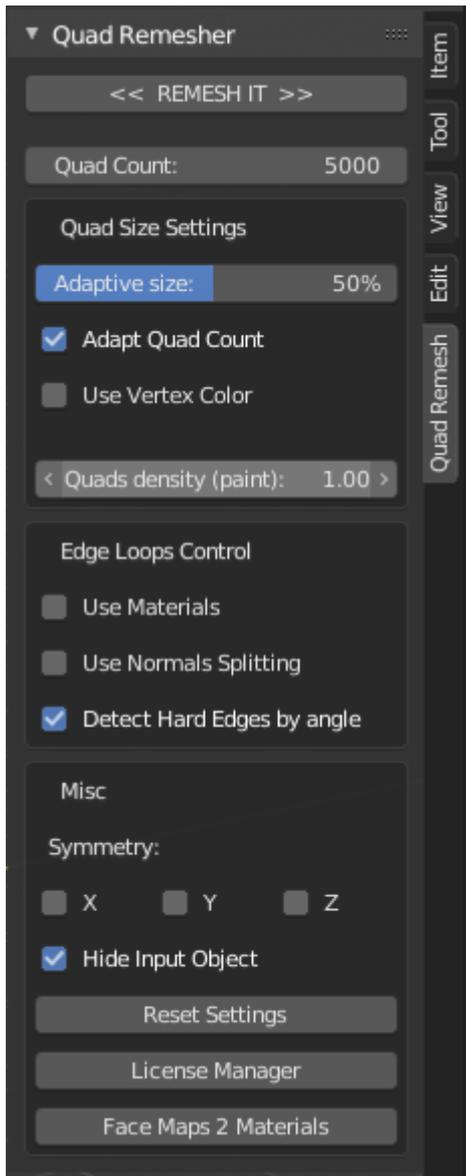
QuadRemesher for Maya



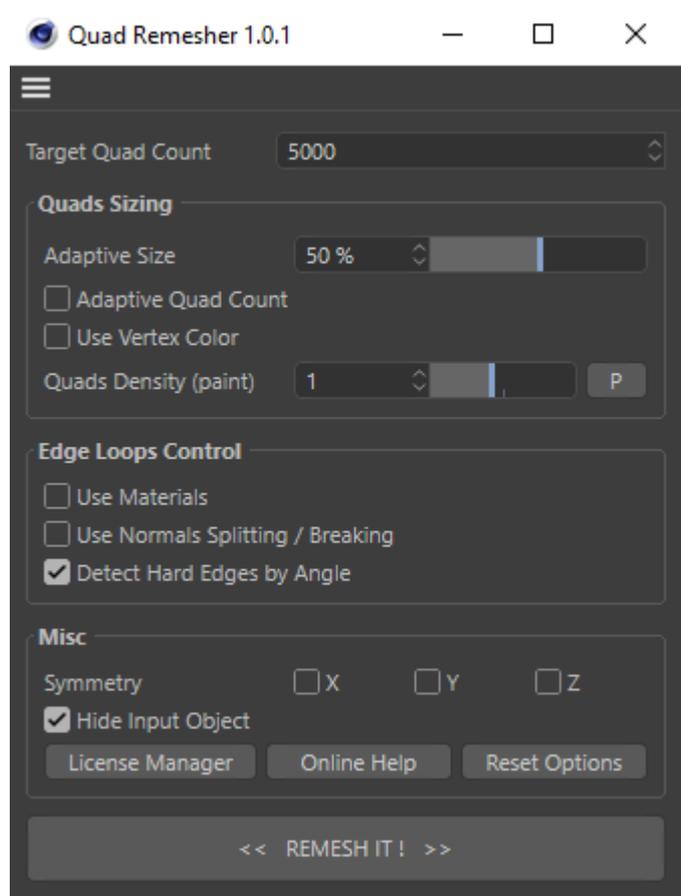
QuadRemesher for modo.



QuadRemesher for Houdini



QuadRemesher for Blender



QuadRemesher for C4D

## A – Installation and Activation

### A.1 – Installation

The installation process is different for each DCC software.

QuadRemesher is using the usual way for installing plugins depending on each software.

For instance:

QuadRemesher for modo provides a .lpk file that you simply drag-and-drop in modo.

QuadRemesher for 3ds Max provides a .mzp file that you simply drag-and-drop in 3DS Max.

QuadRemesher for Maya provides a .zip file that you can open and put content in Maya's plugin path (C:\ProgramData\Autodesk\ApplicationPlugins on Windows)

QuadRemesher for Cinema4D needs you to copy the zip content in the folder dedicated to C4D plugins.

QuadRemesher for Blender provides a zip that you can drag and drop in "Prefs > Addons > Install" window.

A dedicated video can be found for each software on <http://www.exoside.com/quadremesher-tuto>

It's advised, to follow the video tutorials for the installation, activation and first use.

NB: for upgrades (install a newest version of QuadRemesher), please see A.5 below.

If you have any issue while installing or running the plugin, please take a look at the troubleshooting page:

<https://exoside.com/quadremesher/quadremesher-troubleshooting/>

## **A.2 – Activation**

Before using the plugin, you must activate it (even if you are using it in trial mode).

The 1<sup>st</sup> time you click on <<REMESH IT >> you will be asked to activate the plugin.

The activation window will open.

- If you have purchased a license,
  - Enter the email used during purchase,
  - The license-key you received by email just after purchase,
  - Enter a name to identify the computer on which you are working. It can be anything.
  - Click on [<< ACTIVATE >> ] button.

Then you are ready to use QuadRemesher.

- If you have not purchased a license, you can use the plugin in TRIAL mode during one month.
  - Enter the email used when downloading the plugin.
  - Enter a “computer name” to identify the computer on which you are working. It can be anything.
  - Click on [<< Start Trial >>] button

## **A.3 - Offline activation**

For the activation process described in A.2, QuadRemesher needs to have access to internet.

If not, QuadRemesher can be activated by “Offline activation” process.

There are 2 different cases:

- The computer has no access to internet
- The computer has access to internet but the firewall has blocked QuadRemesher

For both this cases, follow the instructions provided by the plugin.

## **A.4 – License Manager**

In the QuadRemesher window, one can find a [License Manager] button.

It allows to:

- Deactivate your license on the computer you are working on.  
It's useful if you want move your license from one computer to another computer. Moving a license is equivalent to uninstall/deactivate QuadRemesher from one computer and install and activate it on another computer.  
In this case, you have to, first, deactivate the license, and then activate it on the new computer.
- Check the license used on the computer.

## A.5 – How to upgrade:

(From v1.0 to v1.1 for instance)

In general, it's advised to remove the previous version before installing the new one.

But it depends on the host software.

- **3dsMax:** simply drag and drop the mzp in 3dsMax as for a standard install.  
NB: you can't use multiple version of QR n 3dsMax in the same time. The last installed will be used.
- **Maya:** Copy the zip content in Addins folder as for a normal install.  
NB: you can't use multiple versions of QR in Maya in the same time. The newer version should be used automatically by Maya, but it's advised to remove the previous version before.
- **Blender :**  
Drag and Drop the zip as for a normal installation (see tuto).  
In Blender, it's possible to have multiple versions of the plugin (you should be able to see them in the Manager) but as there is only one "QuadRemesh" panel, it's advised to enable only the latest version of the plugin in Blender's Preferences > Addons tab.
- **Cinema 4D:**
  - Open the folder where QR 1.0 is installed
  - Remove the old version of QR : "QuadRemesher" folder for the 1.0
  - Copy the new QuadRemesher folder from the zip into this folder : "QuadRemesher1.1"
  - NB: it's not possible to have 2 different versions of QuadRemesher running in C4D.
- **Houdini:**  
You just need to replace the previous .hda by the new one.

NB: Except for first 1.0 version, the version numbers are displayed in the QuadRemesher window, so that you can check which version you are using.

## A.6 – How to uninstall:

- **3dsMax:**  
Remove 'QuadRemesher' folder from  
"C:\Users\YOU\AppData\Local\Autodesk\3dsMax\202? - 64bit\ENU\scripts"  
Remove 'QuadRemesher.mcr' from  
"C:\Users\YOU\AppData\Local\Autodesk\3dsMax\202? - 64bit\ENU\temp"
- **Maya, Cinema4D and Houdini :**  
Simply remove the files That you have created during installation.
- **Modo:**  
Open the Content folder (from modo's System menu) and remove the 'QuadRemesher' folder in the 'Kits' folder
- **Blender:**  
Remove "QuadRemesher" folder from  
"C:\Users\YOU\AppData\Roaming\Blender Foundation\Blender\2.8?\scripts\addons"

## B - Using QuadRemesher:

- **B.1 - The main button :** [ << *REMESH IT* >> ].

It triggers the quad-remeshing.

When you press "< REMESH IT >",

- The remeshing is computing....
- the selected mesh becomes hidden
- and a new object is added with the quad-remeshing result.

You must select one and only-one mesh before pressing "<< *REMESH IT* >>".

- **B.2 - Settings:**

Before pressing "<< *REMESH IT* >>", you will certainly have to define some settings depending on what you want as remeshing.

"Target Quad Count":

This is the main setting. It allows to set the number of quad you want in the output mesh.

NB: In most cases, QuadRemesher won't return exactly the number of quad you have set. QuadRemesher will return a mesh approximating the asked number of quads the best it can do.

(See "*Adaptive Quad Count*")

- **B.3 - Quads sizing settings:**

"Adaptive Size":

This setting allows to control how the remesher locally adapts quads size to the curvature of the input mesh. In general, big quads can be used on flat areas of the input mesh, whereas small quads needs to be used on highly curved areas of the input mesh.

This settings allows to handle this idea.

"Adaptive Size" at 0 means that you want equally sized quad all over you mesh.

The higher "*Adaptive Size*" is, the smaller will be the quads on curved areas.

The default value is 50% and is well suited for many cases.

Except if you specifically wants equally sized quads, it's advised to let it at 50 % and then try other values to fine tune the remeshing.

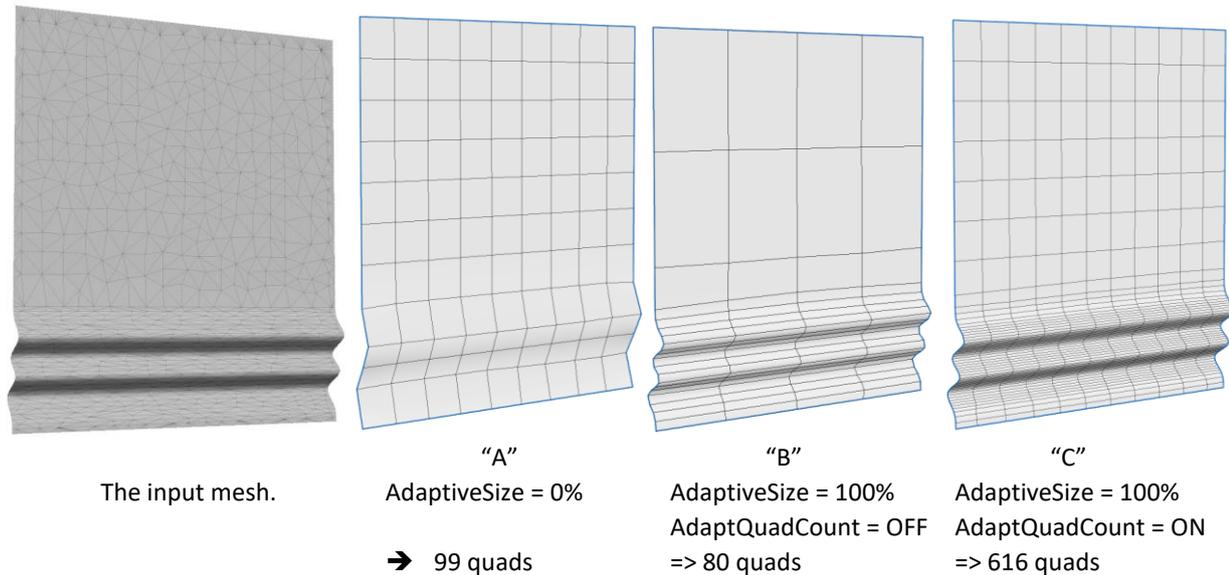
### “Adaptive Quad Count”

This settings is an On/Off switch.

It says how “*Target Quad Count*” is understood by the algorithm.

To understand how it works, let remesh this simple plane with some waves at the bottom which represent some curved area that will be detected by QuadRemesher:

For this example, the “*Target Quad Count*” is set at 100.



A: **Adaptive Size = 0%** means that the remesher will not detect the curvatures in order to created smaller quads. It will just remesh the curved areas as flat areas. This leads to equally sized edges.

B+C: **Adaptive Size=100%, Adaptive Quad Count = OFF/ON**

**Adaptive Size = 100 %** means the remesher will detect the curvatures and will create really smaller edges on curved areas than on flat areas.

There are 2 ways to handle this quad/edge size variation:

B: **Adaptive Quad Count = OFF** : this means that your priority is to respect the number of quads you have set (the *Target Quad Count*). To achieve this goal, it creates smaller edges on curved areas and, as a consequence, it has to create bigger edges on flat areas.

In this example, the mesh B has a little bit less than 100 quads.

**So, use “Adaptive Quad Count = OFF” if you want your Target Quad Count to be respected the best it can be.**

C: **Adaptive Quad Count = ON**: This means that your *Target Quad Count* is not the priority. Your priority is more the quality of the remeshing in term of respecting the input shape at the price of adding more quads. In this case, the flat areas are receiving quads which have the same size than in case “A” (remeshing not detecting curvatures), and create smaller quads on curved areas.

In this example, the mesh “C” has around 600 quads.

**So, in this case, the “Target Quad Count” is used has something which defines the quads size on flat areas!**

### “Use Vertex Colors”

You can paint colors on each vertices of the input mesh to control locally the quad size.

Red colors means that you want 4 times smaller quads edges.

Cyan color means that you want 4 time bigger quads edges.

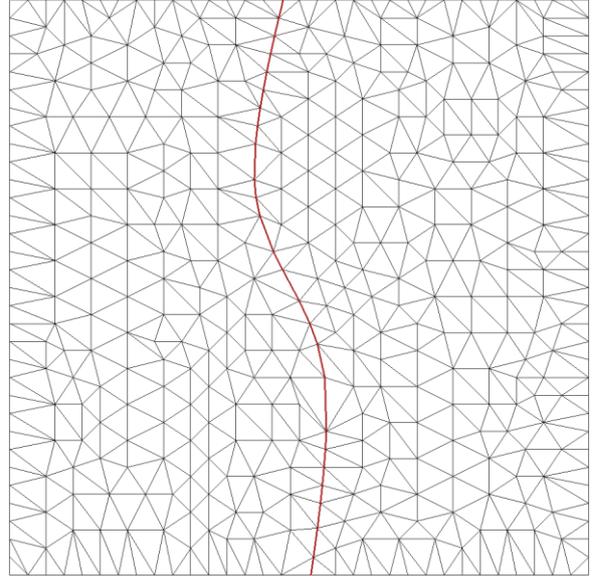
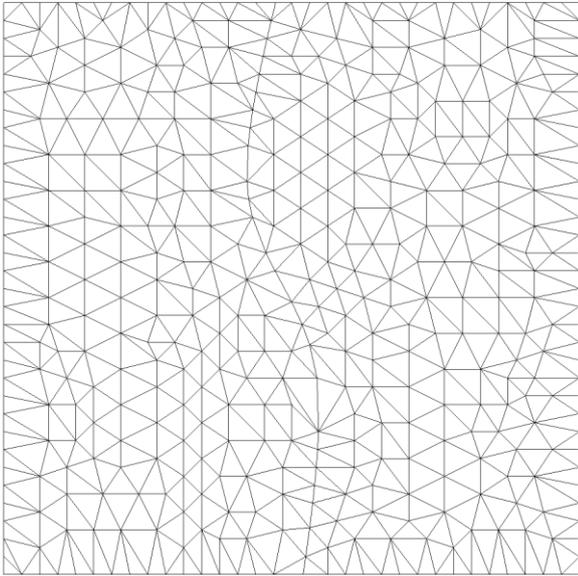
That said, you don't have to set the painted color manually, there is a slider from 0.25 to 4 which automatically set the proper color depending on the slider value (the desired density that you want to paint)

You should use this feature like this:

- Set the “**Quad density paint**” slider at the value you want (from 0.25 to 4)
- Depending on the software, This will set the painted color automatically or you will have to press the [P] button;
- 3ds Max: [P] button allows to create and select the appropriate modifier for painting. Then the color is automatically set when you are changing the quad density slider.
- Blender:  
You just need to select the ‘Vertex Paint’ mode (instead of Object Mode/Edit Mode/....) and choose the ‘Paint’ tool.  
Then each time you change the “Quad Density” slider, the painting color is adjusted automatically.
- Modo : click on [Paint quad-density map with selected value] is needed to set the painting color.  
C4D: hit [P] button is needed to set the painting color.
- Houdini : The plugin is looking for a Paint SOP node that you have to place just before (parent) the QuadRemesher node. If such Paint SOP node is found, the painted color is automatically set
- **NB**: Don't forget to enable “Use Vertex Colors” to tell the remesher that it must take the vertex colors as input to drive the quads sizes.

- **B.4 - “Edge Loops Control...” : guiding the remeshing**

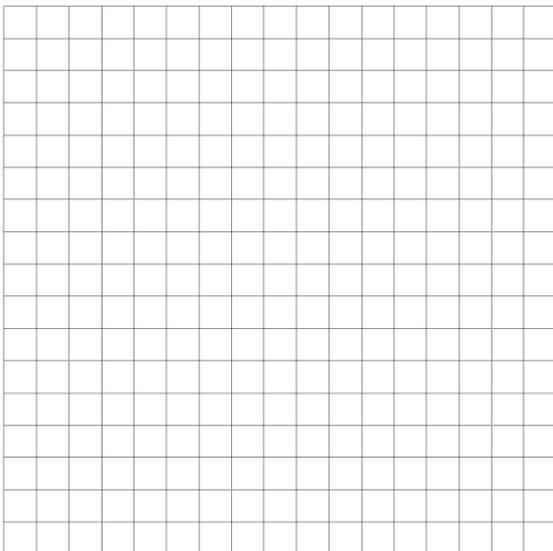
Let’s imagine that you want to remesh this simple plane, but not as a regular grid:  
For some reasons, you want the remeshing to follow the edge path/loop in red.



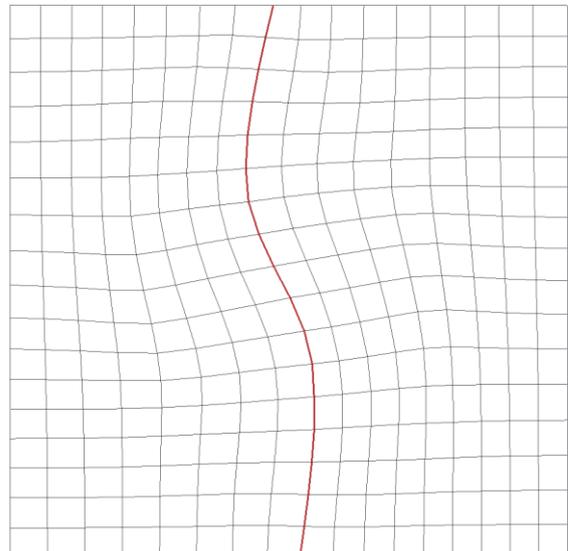
In this case, you will have to “**guide**” the quad-remesher.

The red path/curve is called a **guide**. The guide will ask to the remesher to put an edge loop following the guide in the output mesh. This way, the edge flow of the remeshing can be controlled by using appropriate guides all over the mesh.

Here is what you get **without** the guide (left)



and **with** the guide (right):



**Guides** can be defined by multiple ways:

- The boundaries of **Materials**. (3dsMax, Maya, modo, Blender, C4D)  
In this case, the *"Use Materials"* option must be enabled
- The boundaries of **Smoothing Groups** (3ds Max, modo)  
In this case, the *"Use Smoothing Groups"* option must be enable.
- The user defined **Hard Edges** (modo, Maya).  
This is using the edges tagged with the Harden/Soften edges commands.  
In this case, the *"Use existing Harden/Soften edges"* option must be enable.
- The **Normals Splitting (also named Normals Breaks)** :  
When normals are defined on your mesh, you can define normals so that the mesh looks smooth locally on some areas, and some edges look sharp by defining 2 different normals on both sides of this edge.  
The edges where normals are sharp/creased/hard often define great edge loops/path to guide the quad-remesher.  
In this case, the *"Use Normals Creasing"* option must be enable.  
**NB:** If the normals are defined so that the mesh looks faceted in your renders, it's certainly not a good idea to enable this setting.  
**NB:** This option is equivalent to *"Use HardEdges"* or *"Use SmoothingGroups"* if the normals are computed from Harden Edges or Smoothing Groups.
- ***"Detect Hard Edge by angles"*** :  
The ***"Detect Hard Edge by angles"*** setting allows to ask to QuadRemesher to detect automatically hard edges from the geometry of the input mesh and to create guides on these hard edges so that you will get an edge loop on the output remeshing. With this option, the guides are computed automatically. They are not defined by the user.  
The hard edges are detected from the angles of each edges, and from other geometrical considerations.
- The boundaries of **Polygon Parts** (modo only) or **Primitive Groups** (Houdini)  
In this case, the *"Use Polygon Parts"* option must be enabled
- **Curves (WIP)** : User drawn curves could be a good guide input as well. It's not yet implemented yet in QuadRemesher 1.0. This will probably be added in next versions of QuadRemesher.
- **Selection Sets (WIP)**: Edge's Selection Sets could be a good guide input as well. It's not yet implemented yet in QuadRemesher 1.0. This will probably be added in next versions of QuadRemesher.

You can combine all these ways to create guides for the Remesher.

**IMPORTANT:** Adding badly defined guides can confuse the algorithm!!!

**For this reason, it's important to enable/disable the "Edge Loops Control..." settings carefully.**

- If the normals are not well defined, do not enable *"Use Normals Creasing"*
- If the model is really noisy, you can try with *"Detect Hard Edges by angles"* enabled but,

**Material Ids**  
(Boundaries)

User defined  
**HARD Edges**

- Smoothing Groups
- or Harden/Soften

**Normals Splitting**

Edges where the normals are split  
*"Use Normals Splitting"*

**Hard Edges** computed  
by analyzing the geometry (angles)

**PolygonGroups**  
(Boundaries)  
(modo + houdini)

**Curves**  
(WIP, not in 1.0)

**SelectionSets**  
(WIP, not in 1.0)

## "GUIDES "

Guides are:

- Edge paths in the input mesh
- Or Curves lying on the input mesh (WIP)

Where we want to get an **EDGE LOOP** in the output remeshing .

**NB:** You can choose multiple sources of guides, they will be used, combined together (ex: Detect Hard-Edges + Material Ids)

**ADVICE 1:**

In most cases, you can enable "*Detect Hard Edge by angles*" and let the other settings off:

This works well as long as:

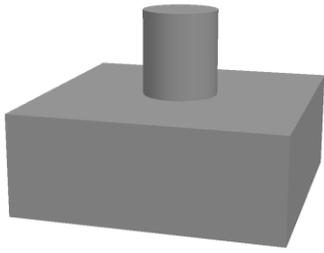
- There is no extreme noise in the input mesh. (QuadRemesher should work well even on noisy meshes, but this setting can sometimes fail with extreme noise)
- There is not too much degenerated triangles. I mean by degenerated triangle, a triangle which is so thin that it has no real normal, no real area. QuadRemesher should work well, most of the time, even if the input mesh contains some degenerated triangles, but few times not.
- Edges that you expect to be detected, have an angle greater than 30°.

**ADVICE 2:**

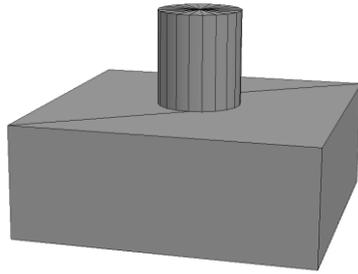
- 1<sup>st</sup> try with the settings you think are the best (use Materials, SmoothingGroups, Hard Edges...)
- If the result is not what you want, control visually where are the Smoothing Groups, Hard Edges, Materials boundaries or Normals creasing.

Let see some simple examples:

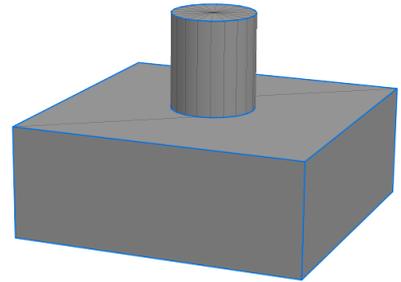
- **Example 1: a simple Boolean union with a cube and a cylinder:**



Input mesh



input mesh with wireframe



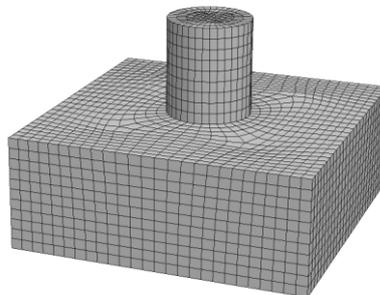
desired guides in blue

We want the blue edges in the right image as guides because we want an edge loop on those blue lines in the output mesh. We will need to enable the settings appropriately so that QuadRemesher will understand that the blue edges must be used as guides.

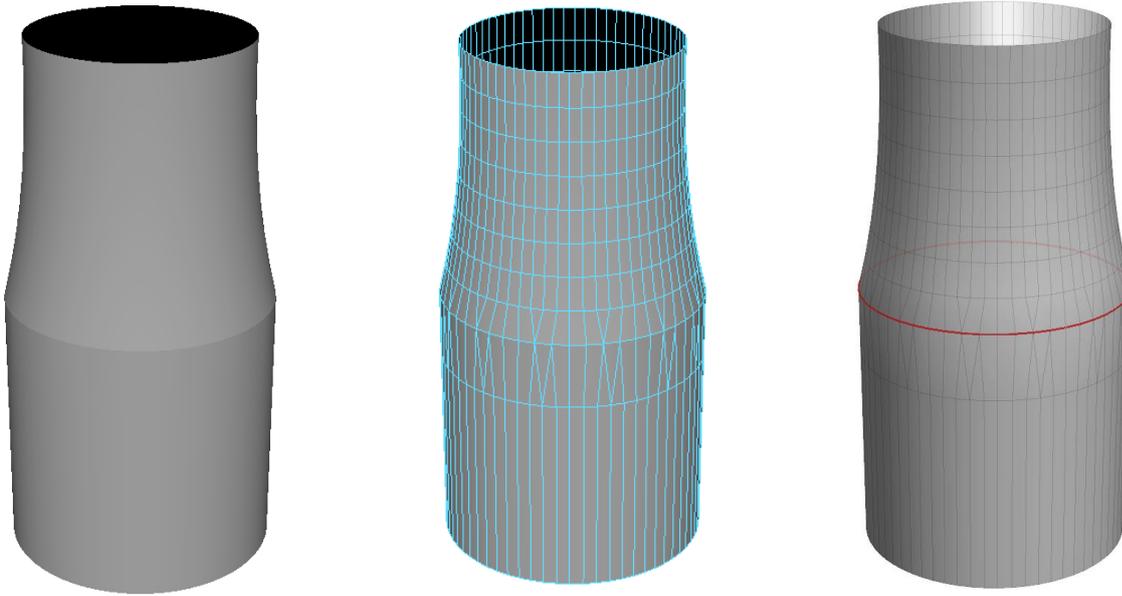
In this case we can enable one of the following options to get the desired Retopo. All of them works well for this mesh, you only need to enable one of them.

- *“Detect Hard Edges by angles”*: this work well because all angles are around 90°. The Hard Edge detection works well in this case.
- *“Use Normals Creasing”*: We can see in the left image that the Normals (the rendering normal) are well defined on this mesh. They are “Hard” on the blue edges in the right image and “Smooth” elsewhere.
- *“Use Smoothing Groups”* or *“Use Harden/Soften Edges”*:  
If this model has been made in 3DS Max or Maya, the Boolean tool has automatically set the Smoothing Groups and/or Hard Edges. So you can equivalently use this options instead of *“Use Normals Creasing”*.  
If you have made this model in modo, the Boolean tool has automatically set the appropriate normals but has not harden the blue edges automatically. For this reason *“Use Smoothing Groups / Hard Edges”* will not work. The solution is to use *“Use Normals Creasing”*.

The resulting quad-remeshing:



- **Example2: a simple cylinder made in CAD software.**



Here we have a simple mesh created in a CAD software. We can see that normals are explicitly defined on this mesh, and that these explicit normals are revealing a creasing of the normals in the mesh (left image). We want the remeshing to follow this creasing. So, for this purpose *“Use Normals Splitting”* must be enabled.

In this case, the angle made by the creasing, is too small to be detected by the *“Detect Hard Edges by angles”*. And there is no Smoothing Groups nor Harden Edges because the mesh has been made in a DCC software that does not create those information.

The only solution is to use the explicit normals with *“Use Normals Creasing”* enabled!

- **Advices about “Edge Loops control...”**
- About “Use Normals Creasing” and “Use Hard Edges / Smoothing Groups”

With 3dsMax and Maya, usually, the Normal’s Creasing are built/deduced from the Harden Edges (Maya) or from the Smoothing Groups (3DS Max), then using “Use Normals Creasing” is equivalent to using “Use Hard Edges / Smoothing Groups”.

But, in some cases, you can end up with a mesh with normals explicitly defined whereas neither Smoothing Groups nor Hardens-Edges are specifically defined. In this case you will have to enable “Use Normals Creasing” if you want the remeshing to follow the hard-looking edges.

This case happens:

- In modo: modo directly works with explicit normals, (Primitive creation, Boolean operation...), and doesn’t create Hard Edges tags or Smoothing Groups.
- When you imports a mesh in 3DS Max or Maya which has been created in modo or in some CAD softwares. In this case, the explicit normals are transmitted to 3DS Max or Maya, but not the Hard Edges tags nor the Smoothing Groups.
- When you work with locked normals in Maya. (normals explicitly defined on the mesh, instead of computed from the user defined Harden Edges or Smoothing Groups)

## LEGAL

Modo® is a registered trademark of Foundry LTD.

Autodesk, the Autodesk logo, 3dsMax®, Maya® are registered trademarks or trademarks of Autodesk, Inc., and/or its subsidiaries and/or affiliates in the USA and/or other countries.

Cinema4D® is a registered trademark of MAXON Computer GmbH.

Houdini® is a registered trademark of Side Effects Software Inc.

Blender® is a registered trademark of Blender Foundation.