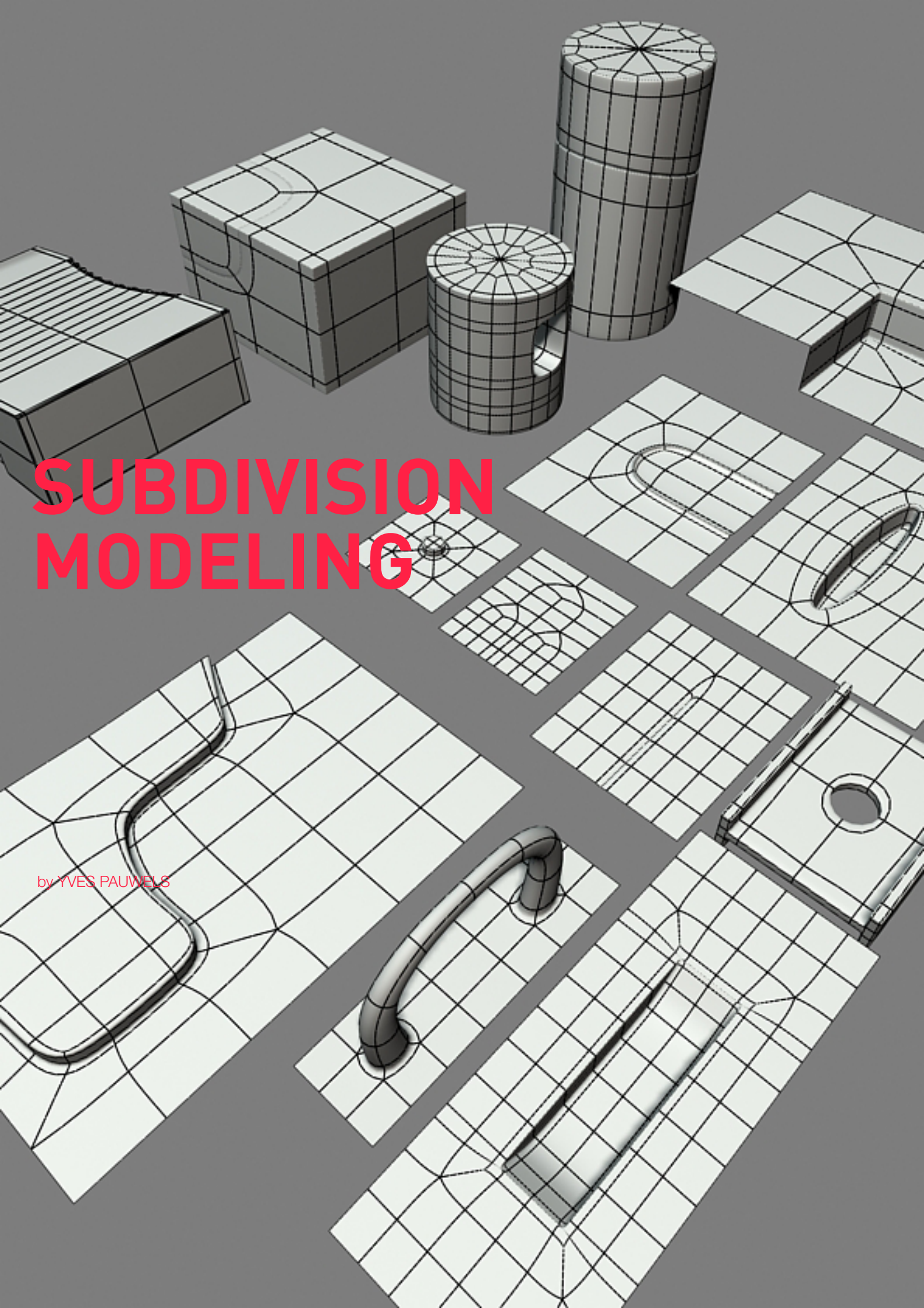


SUBDIVISION MODELING

by YVES PAUWELS

SUBDIVISION MODELING

Just as our course material, this document will not explain step by step how to make a specific object.

This document's purpose, however, is to enrich you with the insights and concepts with regard to Subdivision Modeling, so you can apply these to whatever model! Subdivision Modeling is also sometimes briefly called SubD.

The principle

The principle is as follows: a basic mesh – also called control mesh – is subdivided by an algorithm, which results in a rounded over form.
How many times one mesh is subdivided, is determined by the number of iterations. The algorithm is executed by a smooth modifier.

Subdivision Modeling allows the modeler to obtain a rounded over high resolution model by means of a lower resolution model. Consequently, the modeler has to edit less points and polygons to obtain a detailed object. The lower resolution model is also sometimes called the "base cage". For instance, one polygon is subdivided into 4 polygons (iterations 1), into 16 polygons (iterations 2), 64 polygons (iterations 3) … etc.

Polygons that form an angle are interpolated, whereby a somewhat rounder corner is formed. The more iterations, the more that corner will be rounded over.

MeshSmooth vs TurboSmooth

On a basic model, a MeshSmooth modifier or TurboSmooth modifier is placed. Both modifiers can subdivide your model. The MeshSmooth has more possibilities compared to the TurboSmooth modifier. However, the latter is faster at calculating and is memory-friendlier because it does not have a number of options that are present in the MeshSmooth.

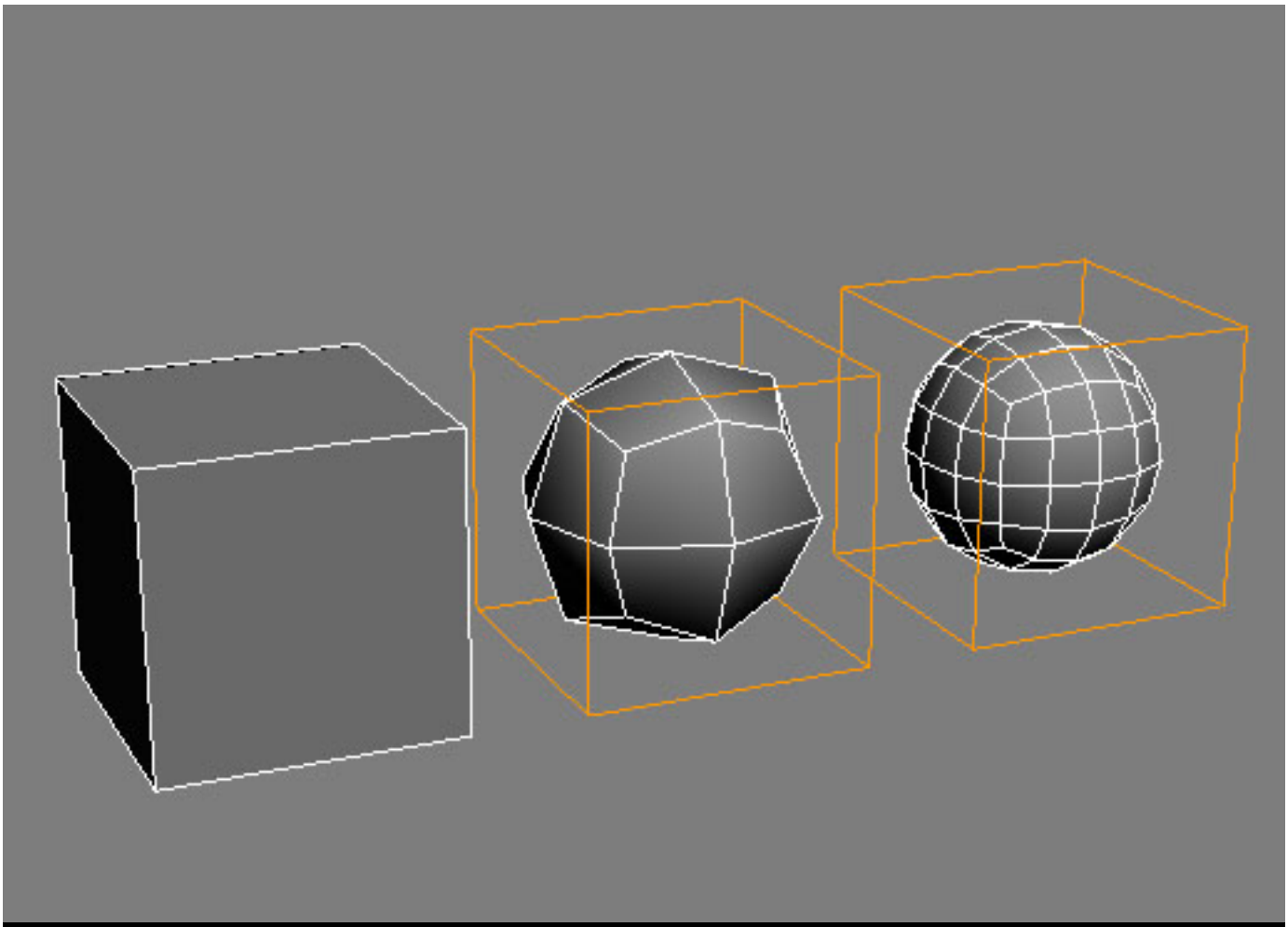
Working without one of these two smooth modifiers, however, is also possible. In your Edit Poly modifier, you can also set the Subdivision Surface Parameters. The disadvantage is that you are within your Edit Poly.

Via the Subdivision Surface rollout, you check "use NURMS subdivision". Your object is immediately subdivided. The advantages of using a smoothing modifier (Mesh- or TurboSmooth) is that it is in the stack and that you can quickly return to the original model and switch the SubD on or off.

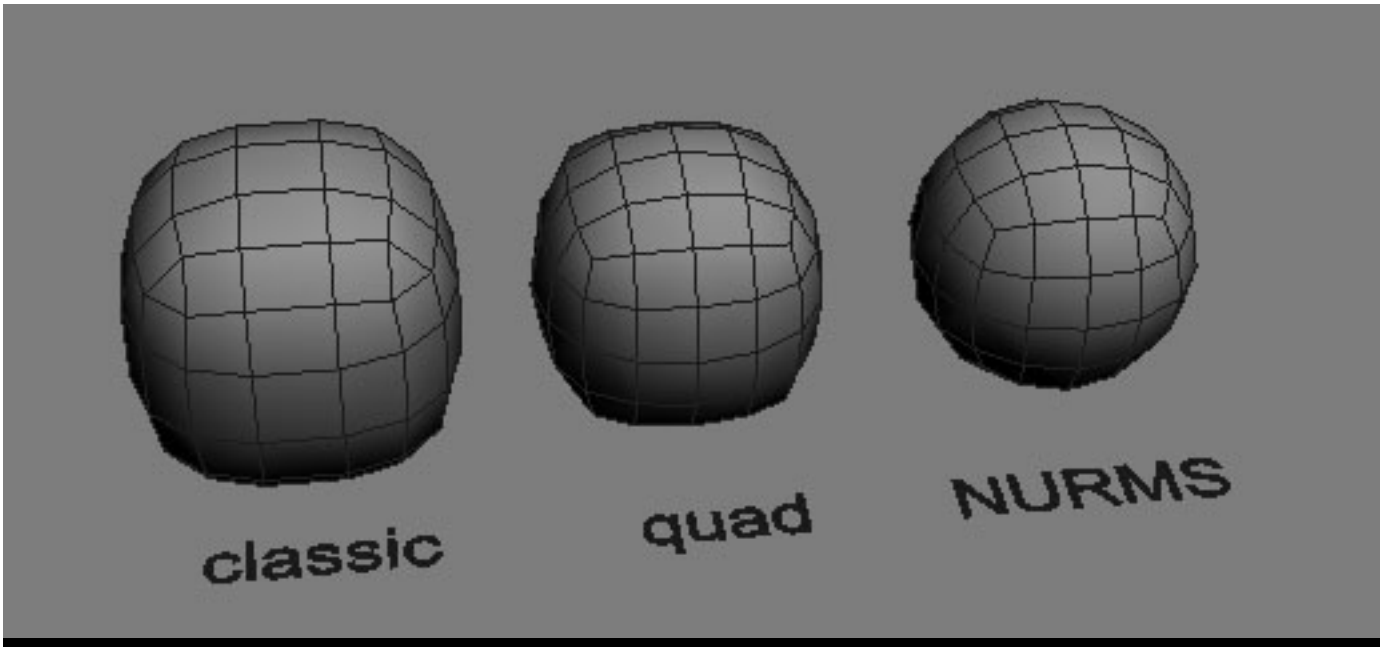
Instantiate a smoothing modifier

If you work with multiple objects, which will each separately receive a smoothing modifier, it is recommended to instantiate the smooth modifier so you do not have to run through all your objects to – for example – adjust the iterations; However, you have one modifier that manages all your objects.

Procedure: put the modifier on your object, right-click it and select "Copy". Select another object, go to its modifier



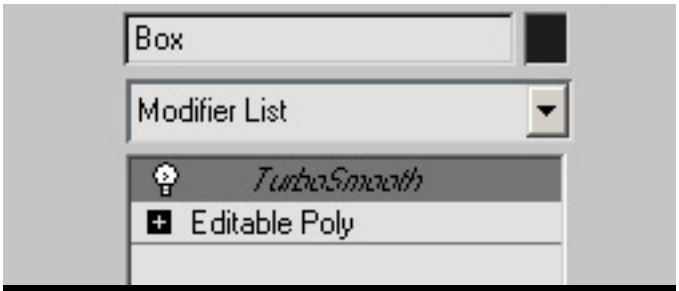
The cube is the basic form. Cube subdivided 1x (iterations 1). Cube subdivided 2x (iterations 2).



The 3 type algorithms (Meshsmooth). Classic results in triangles. Quad is not so round. NURMS gives the best result.

stack and right-click it; next you select “Paste Instanced”. Now, on both objects, the same modifier is placed. When you adjust the parameters of one object, then these will also be adjusted with the other object. This is convenient!

You can recognize an instantiated modifier by the fact that it is in italics in the modifier stack.



Instantiating the turbosmooth modifier

Edwin Catmull: the father of SubD and president of Pixar

Subdivision algorithms

The MeshSmooth modifier has 3 type algorithms that can be used:
- Classic

Produces quads & tris when subdividing. The advantage of this algorithm is that the UVs are maintained. By means of a “Strength” and “Relax” value, you can set the rounding. However, as the name says: an older and less used algorithm.

- Quad

Only produces quads when subdividing. By means of a “Strength” and “Relax” value, you can set the rounding. Advantage: no more tris.

- NURMS: Non-Uniform Rational MeshSmooth

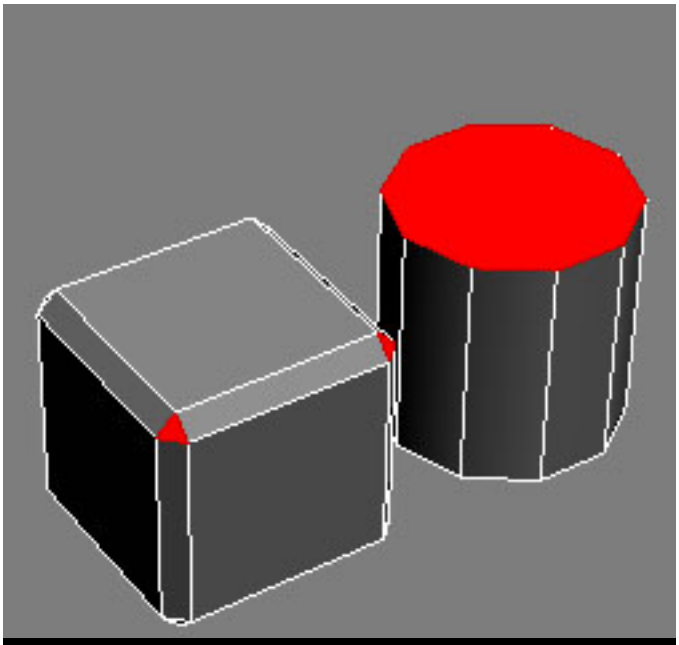
With this algorithm, you can set a different weight for each control vertex, just as for the edges. We will mainly use the NURMS algorithm; however, we will not set the separate weighting.

The TurboSmooth does not have these choice options and uses the NURMS algorithm anyway.

Everything Quad

Although most software packs offer the possibility to allow 3- or 5-sided polygons in SubD geometry, it is advised, however, to limit this to only 4-sided polygons or Quads.

The simple reason is that most software packs use the Catmull-Clark algorithm, which is based on 4-sided polygons. There have already been many debates on many fora on this subject; however, we are advocates to have everything Quad. Of course, it is more difficult because you cannot have any tri or Ngon, and therefore you are obligated to solve all these “problem polygons”. Tris and Ngons create “pinches” in your mesh, which causes a deformation in the surface.



Tris and Ngons must be avoided in SubD!

Isoline display

If you place a smooth modifier on your object, its mesh is subdivided according to the number of iterations you have set (do not go to high). You will also see this visually: you get more polygons. However, if you only want to see the edges of the base mesh, you can set the smooth modifier in such a way that it only shows these lines. By checking “Isoline Display”, the additional created lines are filtered away and your mesh is clearer. Note: nothing changes structurally about your mesh when you check this; it is purely a visual setting. Note: do not collapse a mesh when you are in isoline mode, because this gives no correct result!

Procedure

Create an object and convert it to an Editable Poly. Place a MeshSmooth or TurboSmooth on the object. Set the number of iterations. Do not set these higher than 2 or 3. Return to your editable poly and manipulate it. By means of the “show end result” toggle, you can see the result with the smooth modifier applied on it.

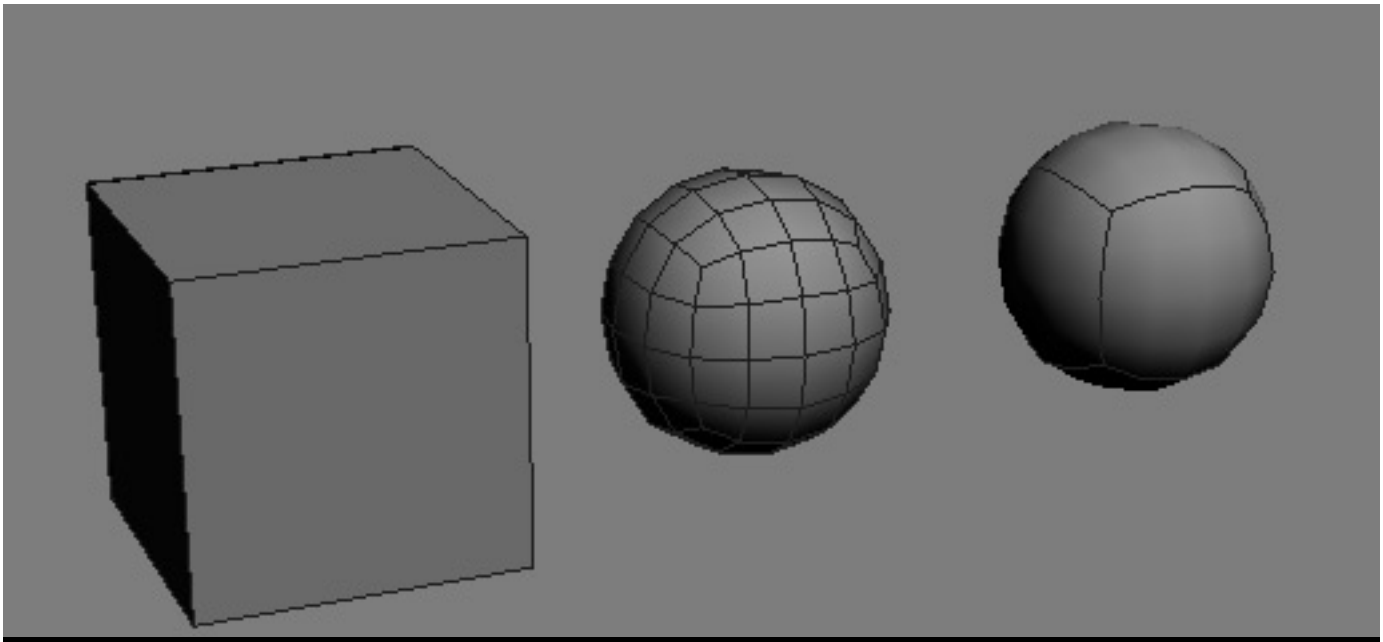
Symmetry modifier

Many products are symmetrical or – construction-wise – to a certain extent symmetrical. This is why we will often be using the Symmetry modifier. The latter allows you to mirror a part of your mesh so you only have to create half. Thus, this modifier can save you plenty of time. From a certain point, it is possible that your product becomes asymmetrical: a certain button is only on the left hand side, the current input is on the right hand side, or a diagonal chamfer runs through the object… it is all possible. However, certainly start – if the basic form of the product is like this – symmetrical.

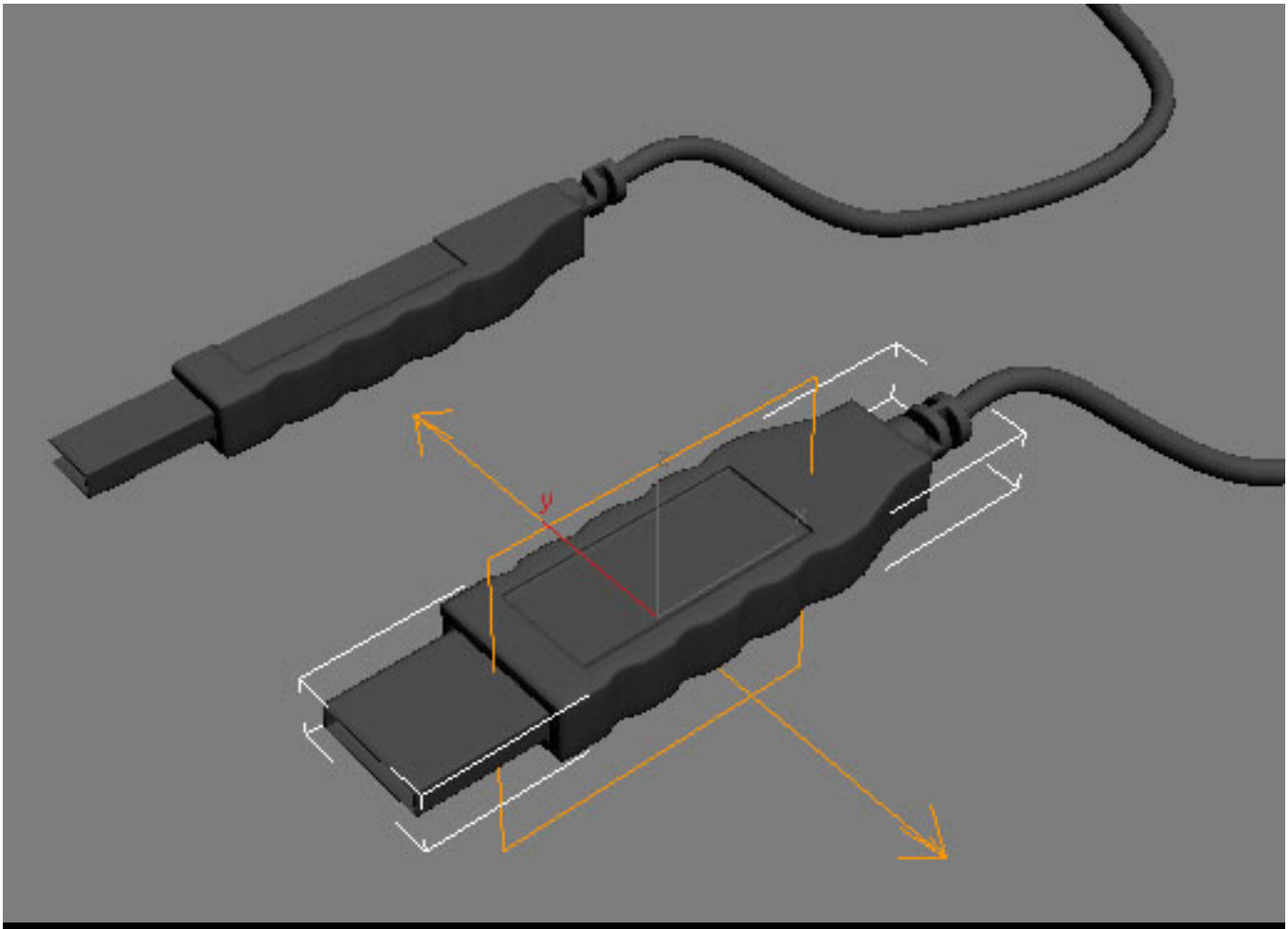
At a certain moment, you can collapse the Symmetry modifier and turn it into an Editable Poly, and then you can model something different on both halves. (Make sure you save the step for the collapsing separately, you never know you will have to repeat it!)

Shell modifier

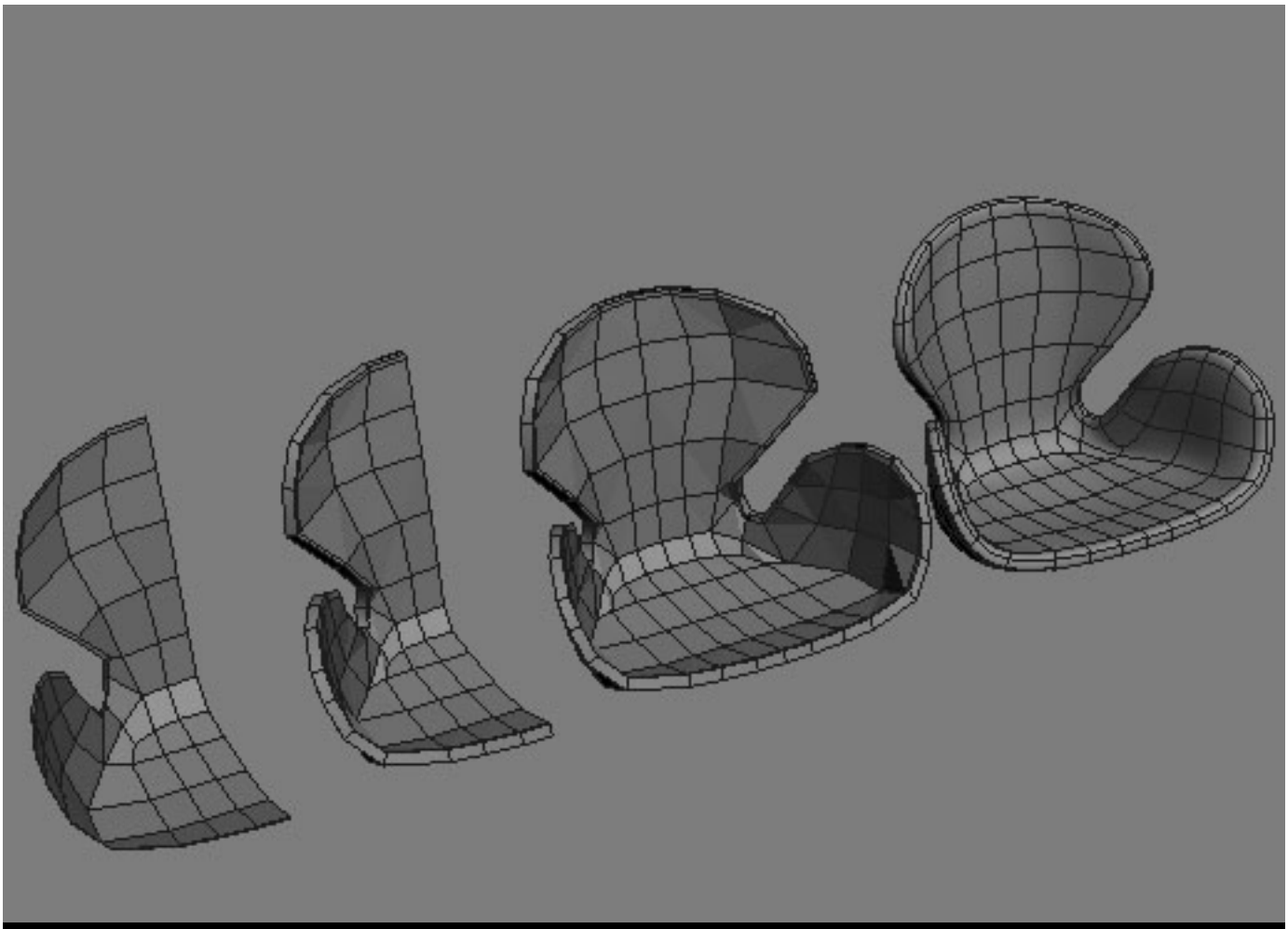
To give a thickness to the mesh, it is sometimes difficult to model everything (though sometimes you have to!). There exists a modifier, which can give your mesh a thickness by a simple setting. You can choose whether the thickness should be outwards or inwards. Note: sometimes you can get overlaps. You can also set how many segments the thickness should have.



Both rounded cubes have as many polygons, however, the right one is rendered in Isoline Display.



To save yourself some work, use the symmetry modifier – if possible – and you will only have to create half.



The Shell modifier was applied to the basic form to give it a thickness. Then it was mirrored and smoothed.

SUBD TOOLS

Now you know how a Subdivision algorithm works in theory, you still cannot model correctly. It is completely different compared to Low Poly modeling, because you have to bear in mind the extra rounding that will be on your mesh. Not everything is allowed or you get an ugly mesh. Subdivision modeling is nothing more than constantly solving problems and looking for the correct solutions.

Idem as low poly tools

First a positive note: you already know the tools we need to start SubD modeling! We usually start from a box (subdivided according to a number of segments) and turn it into an Editable Poly. Then, you get to see the familiar tools, which you already needed for Low Poly modeling.

So the tools are the same, however, the techniques are not always. For you need to bear in mind that everything should be quad, and that a smooth modifier comes on top.

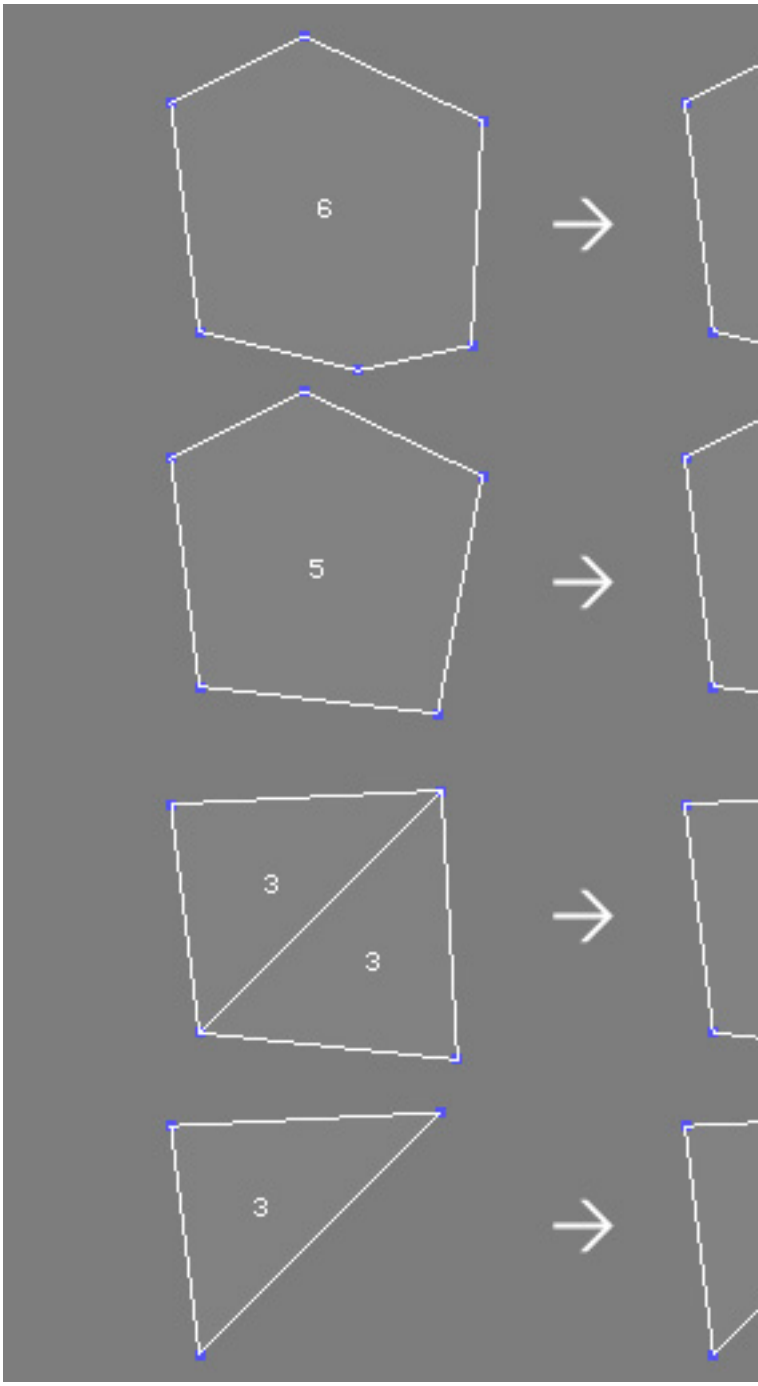
EVERYTHING QUAD: HOW?

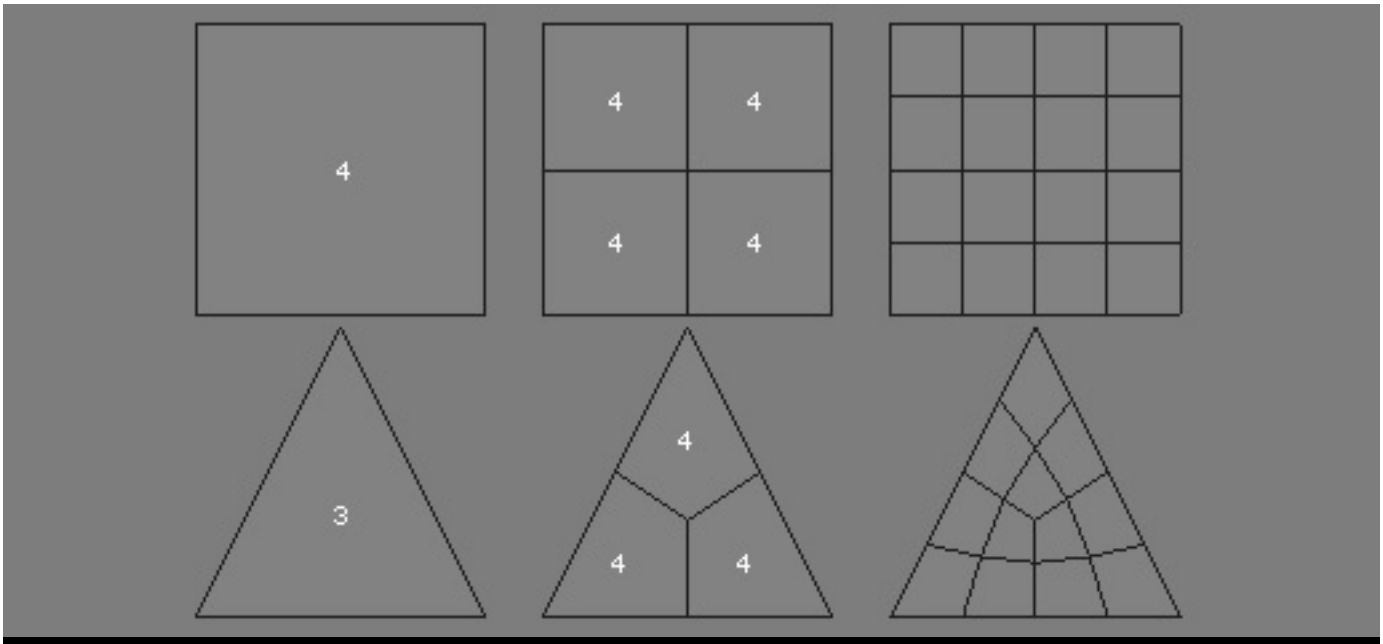
Of course, the first question is: why? That is easy: as already mentioned, on the one hand, the subdivision algorithm is based on it and on the other hand, working with quads is easier, more logical, and clearer.

If it encounters a triangle, the algorithm will subdivide it into 3 quads, so in theory this is no problem, only, it is for the modeler not immediately clear how the mesh will react.

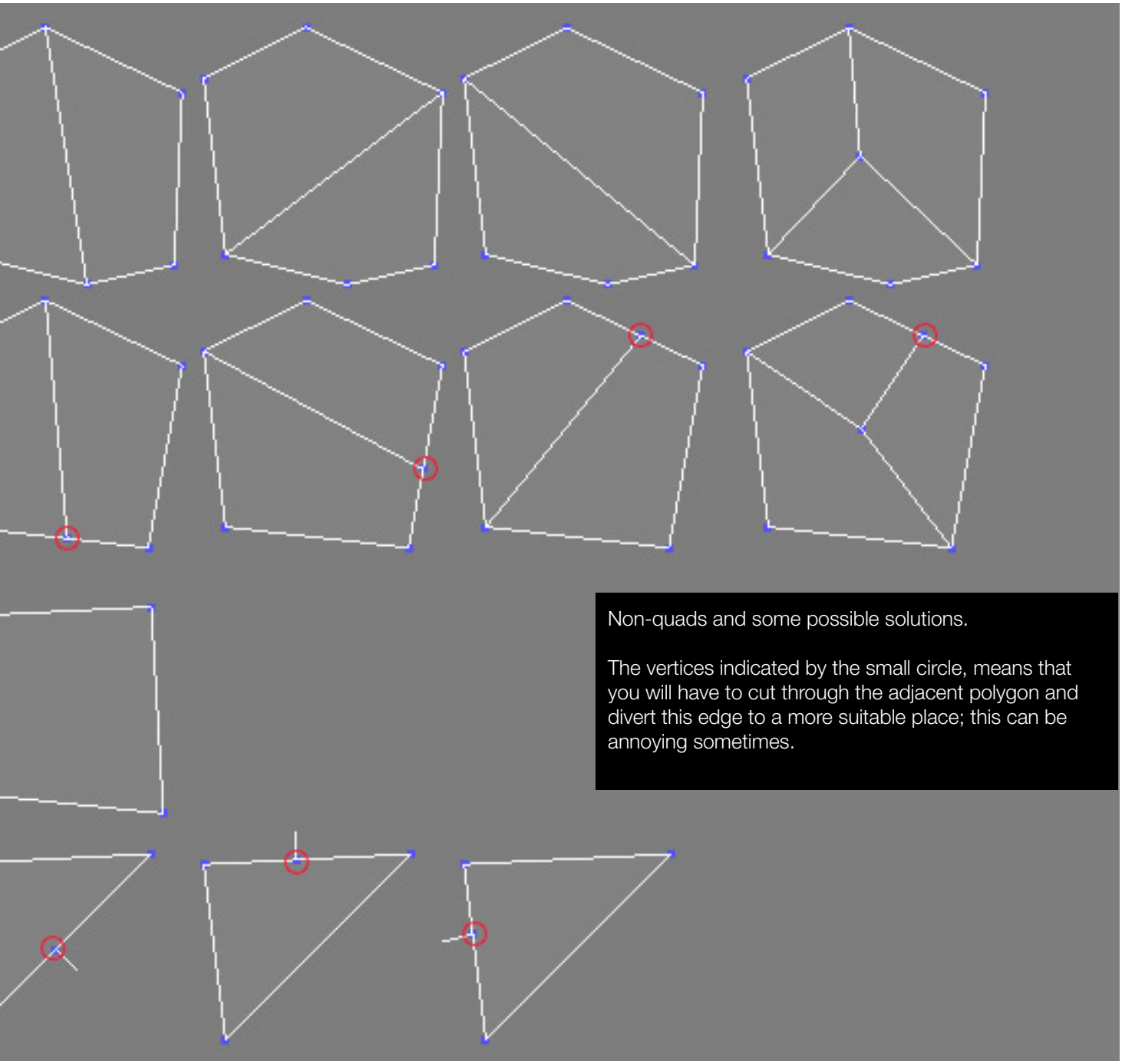
So, compared to low poly modeling, this is the big difference: in low poly modeling, we calculate in triangles, whereas with SubD modeling, we look at the quads.

In the adjoining figure, you see some possible solutions to remove non-quads.





A quad and a triangle after 1 and 2 iterations. The algorithm sees to it that everything becomes quad.



Non-quads and some possible solutions.

The vertices indicated by the small circle, means that you will have to cut through the adjacent polygon and divert this edge to a more suitable place; this can be annoying sometimes.

Hexagonal polygon

A hexagonal polygon can be easily cut in half from one vertex to the vertex across. In this way, you obtain two quads. Dependent on with which vertex you start, you can obtain different solutions.

You can also make a vertex at the centre and in this way obtain 3 quads.

Pentagonal polygon

A pentagonal polygon is somewhat more annoying to solve. You cut from a vertex to its edge opposite. You cut the latter in half to create a vertex at that place, so you obtain a hexagonal polygon.

In theory, this is no problem, however, you do need to bear in mind the polygons lying around this pentagonal polygon (on the figure above, not depicted), so you will need to trace this new edge through the mesh and divert it to a place where it does not disturb as much (usually to the symmetry axis, or to an opening in the mesh).

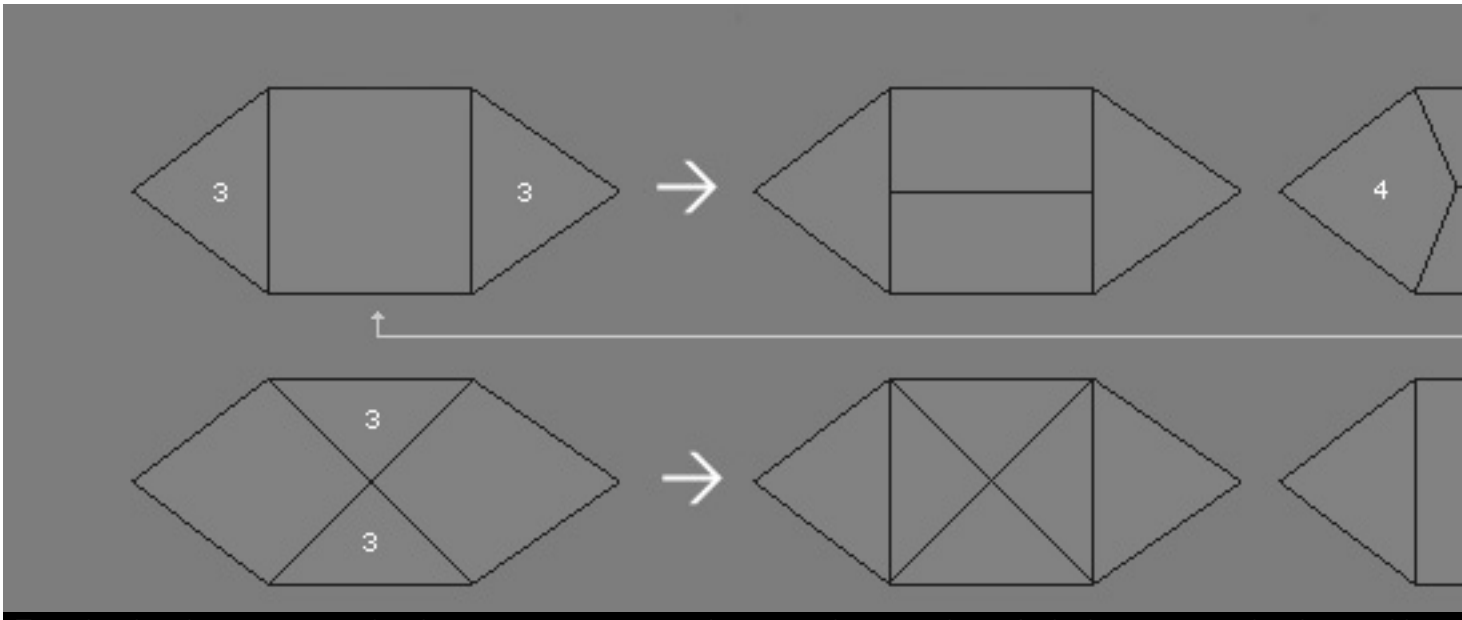
Another way is to create a vertex at the centre and in thus obtaining 3 quads, however, you are still stuck with a line you need to trace through your mesh.

Two triangular polygons sharing an edge

This is a piece of cake, you remove the edge between the two triangles via the backspace key and you obtain 1 quad.

One triangular polygon

Same idea as with the pentagonal polygon. You create an extra vertex by cutting through the adjacent polygon, with as a result the creation of an extra vertex and thus obtaining a quad. However, the disadvantage is that you need to trace the edge through the mesh and divert it to a more suitable place.



Two triangles close to one another, however, one or more polygons are between them. At the bottom: two triangles touching e
in common.

Two not adjacent triangles

Sometimes, you need to look somewhat further than that one “problem” polygon. In the situation above, two triangles are a bit further apart, and one quad stands in between (however, it could also be multiple quads).

You can solve this rather quickly by cutting an edge, which connects the two triangles. You do need to cut an edge in half to obtain that extra vertex and thus turning your triangle into a quad.

Two triangles having one vertex in common

If two triangles touch one another and thus have one vertex in common, you can – by somewhat rearranging the mesh – reduce this problem to the solution depicted above.

KEEP ROUNDINGS UNDER CONTROL

SubD modeling is very good for making organic shapes. The trick is to have control over those shapes and to be able to determine how the flow of the mesh should be. You need to be able to determine what should be rounded over and what should be sharp. By manipulating the low poly control mesh, you determine how the high poly mesh – by means of a smooth modifier – will look like.

What with sharp edges?

Because our smooth modifier rounds over everything, it is not so obvious to have something sharp in a SubD model. In this case, the following easy rule applies: the closer the edges are together, the sharper the smoothing is going to be.

In real life, each product is given a rounding; sharp edges are avoided because on the one hand they can be really painful, and on the other hand they can be more fragile. Take a good look around you and try to find a product that has really sharp edges, you will not find many.

E.g. rounding over an angle

There are different ways to create rounded angles. We apply this to a simple box (Editable Poly), of which we want to round off all sides in one time.

-Technique 1: Chamfer (not good)

You select all edges and you chamfer these. When we look at it with the TurboSmooth modifier (iterations 2) on it, you see that the box is indeed rounded over, however, let us look at it closely.

Disadvantage 1: at the corners, the triangles are disregarded; as already mentioned, algorithms demand of the smooth modifier that everything is quad.

Disadvantage 2: the original edge has disappeared (you have split it) and you notice that the sides of the box are rather bulged than really being flat.

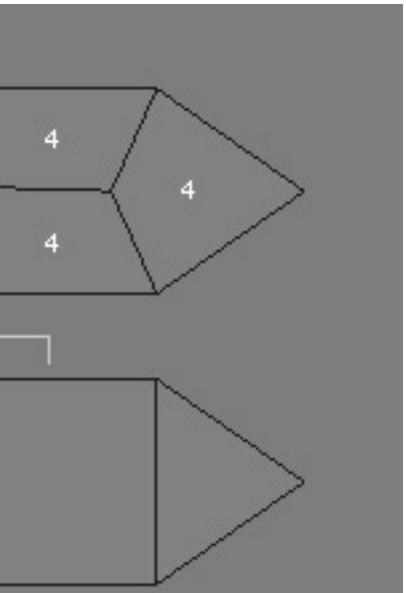
Disadvantage 3: not so easy to adjust the rounding.

-Technique 2: use Connect

You select an edge of your box, you make a ring selection, you cut all these edges with two edges by means of Connect, by which you set that you want two edges and you are going to push these edges apart by means of setting the push parameter on e.g. 90. Like this, in one direction detail was added, now also do this for the other two directions.

Advantage 1: the original edge remains and the sides of the box are flat.

Advantage 2: you can easily still adjust the amount of rounding by bringing the created edges closer to the centre again. (More information on edge loops further on in this course).



ach other and they have 1 vertex

Subdivision modeling is sometimes also called Box Modeling since you usually start from a box, although this is not really necessary.

Besides SubD modeling, other techniques exist, such as Patch Modeling and NURBS modeling; however, we will not discuss these here, because the first is a dated technique and the latter because 3dsMax is not so good at it. With both techniques, you need to draw splines to create polygons.

Poly-Modeling with 3ds Max: Thinking Outside of the Box

auteur: Todd Daniele
uitgeverij: Focal Press
ISBN: 978-0240810928

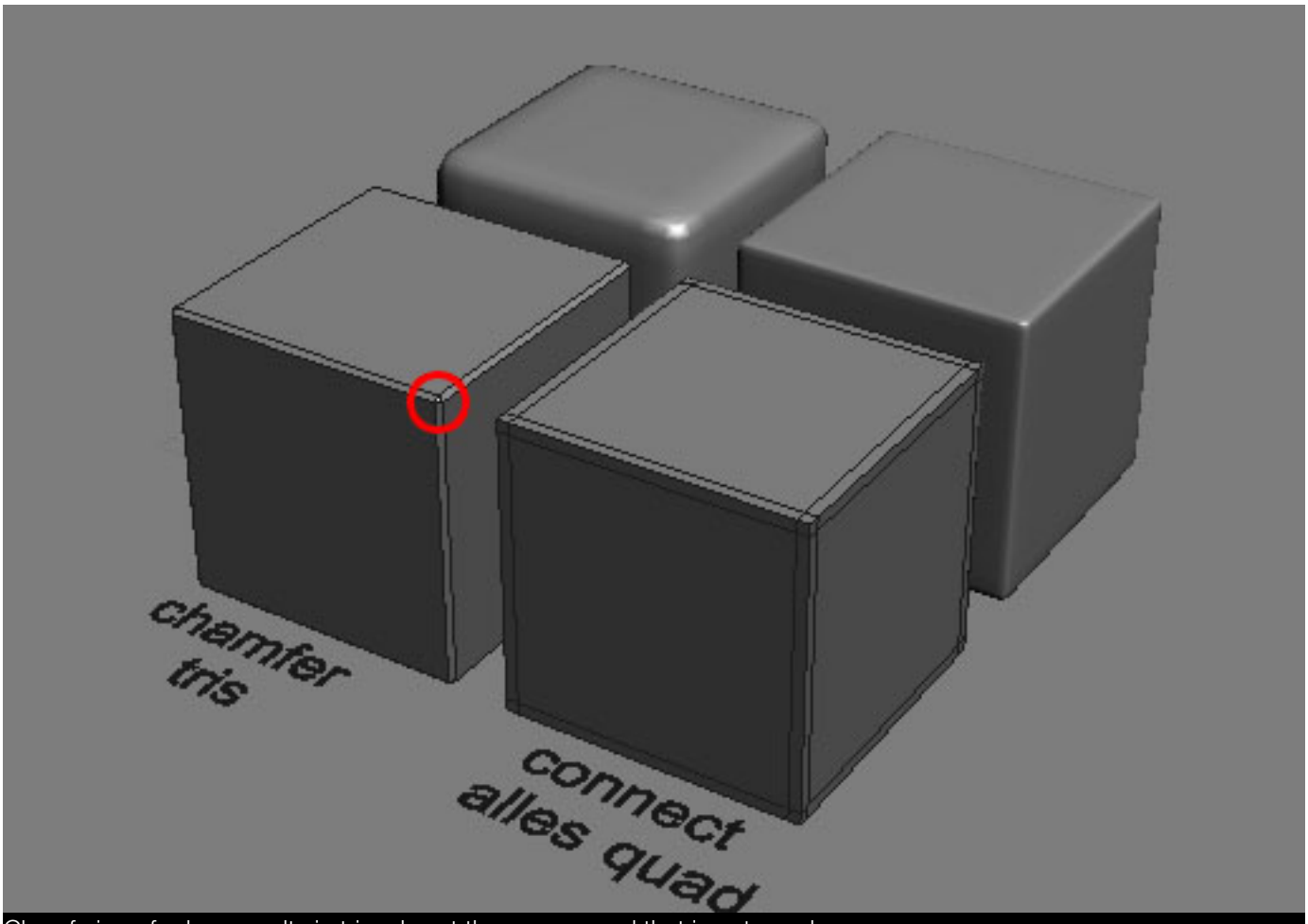
Advantage 3: all Quad

Disadvantage: a bit more work than the Chamfer technique, however, you obtain a much more correct result!

Conclusion: avoid the Chamfer tool when you are submodeling.

Turbosmooth will make it right ...NOT !

Too many students hope that the smooth modifier will make their mesh right again, however, it does not. You will obtain a blob mesh. You as the modeler need to have control over the mesh. You are the master; the smooth modifier is the slave! They do not call the low poly mesh “control mesh” for no reason. You need to be able to perfectly predict how the mesh will react after the smooth modifier is placed on it.



Chamfering of edges results in triangles at the corners and that is not good.
It is better to do it as on the right hand side; you can regulate the rounding better.

E.g. making a chamfer

Step 1

Select the edge where you want to make the chamfer (if it is not there yet, cut it in). Split the edges in two by using the Chamfer tool.

Step 2

Now extrude the newly obtained surface inwards. Note: at the ends, straight surfaces will arise, which you do not need immediately. Remove these.

Step 3

When we now look at this with a smooth modifier on it (iterations 2), the chamfer is more a wave in the mesh than a clear-cut chamfer. Select all edges that make a corner (4 in total, loop selection). Extrude these with the Edge Extrude tool, however, set the Extrusion Height on 0 and set the Extrusion Base Width to determine the sharpness of the rounding. As always: the closer the edges are together, the sharper the rounding is. However, make sure there still is a rounding.

Never set the Base Width on 0, since then the edges will overlap and this can cause mistakes.

In step 3, we have not used the chamfer tool because we would lose the original line.

EDGE LOOPS

Edge loops are edges, which are in line and which you can choose by means of a “LOOP” selection.

Then, you can move such an edge loop entirely, even rotate it. An edge loop is not necessarily at the edge of your mesh. An edge loop can occur anywhere in your mesh.

By moving the loops, you determine how rounded something must be. The closer the 3 edge loops are together, the sharper they are. The further they lie apart, the rounder the edge will be.

The art of SubD modeling

The art of SubD modeling is in the addition of edge loops in your mesh so you can quickly manipulate your mesh.

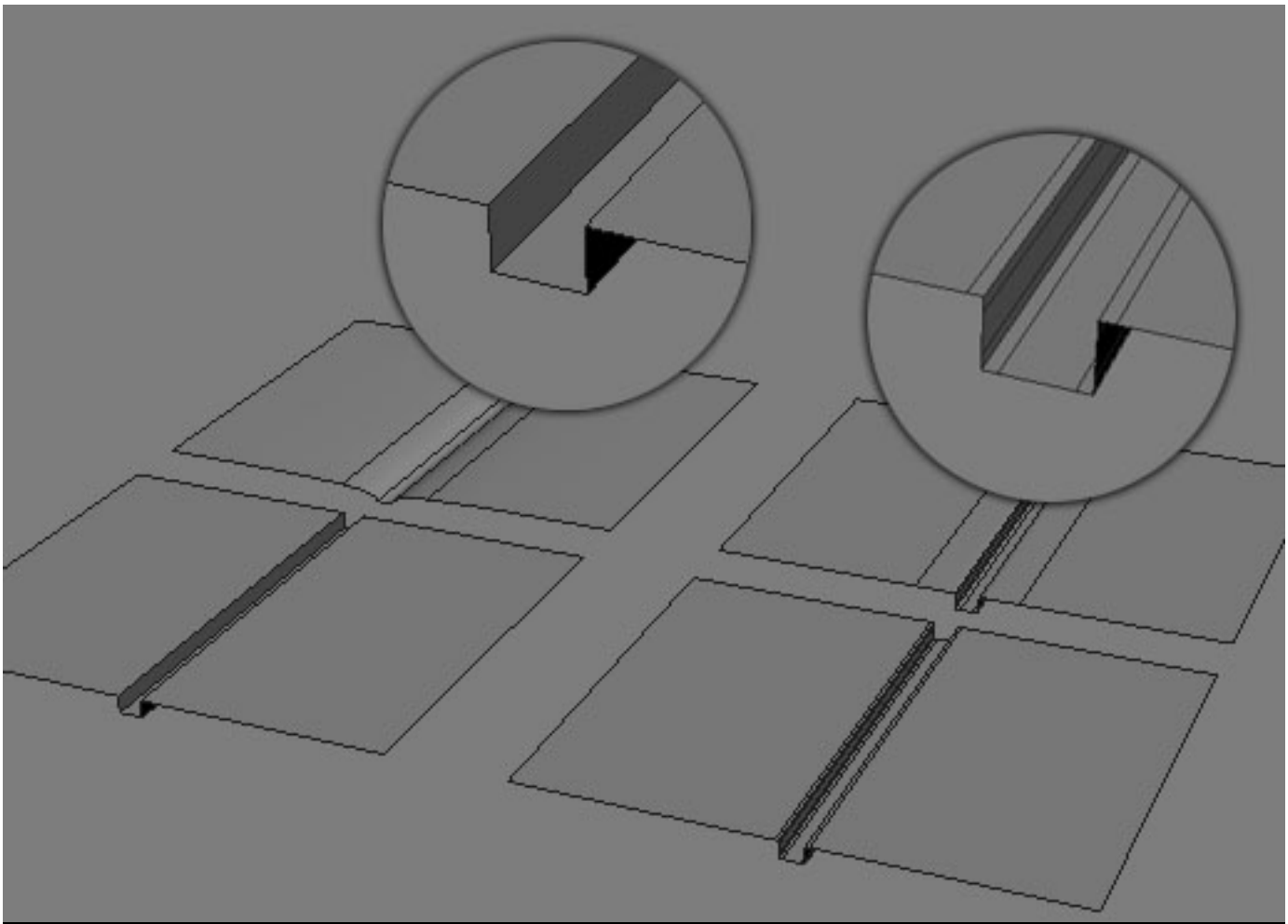
Edge constraints when moving vertices

Use edge constraint if you want to move vertices according to a certain edge.

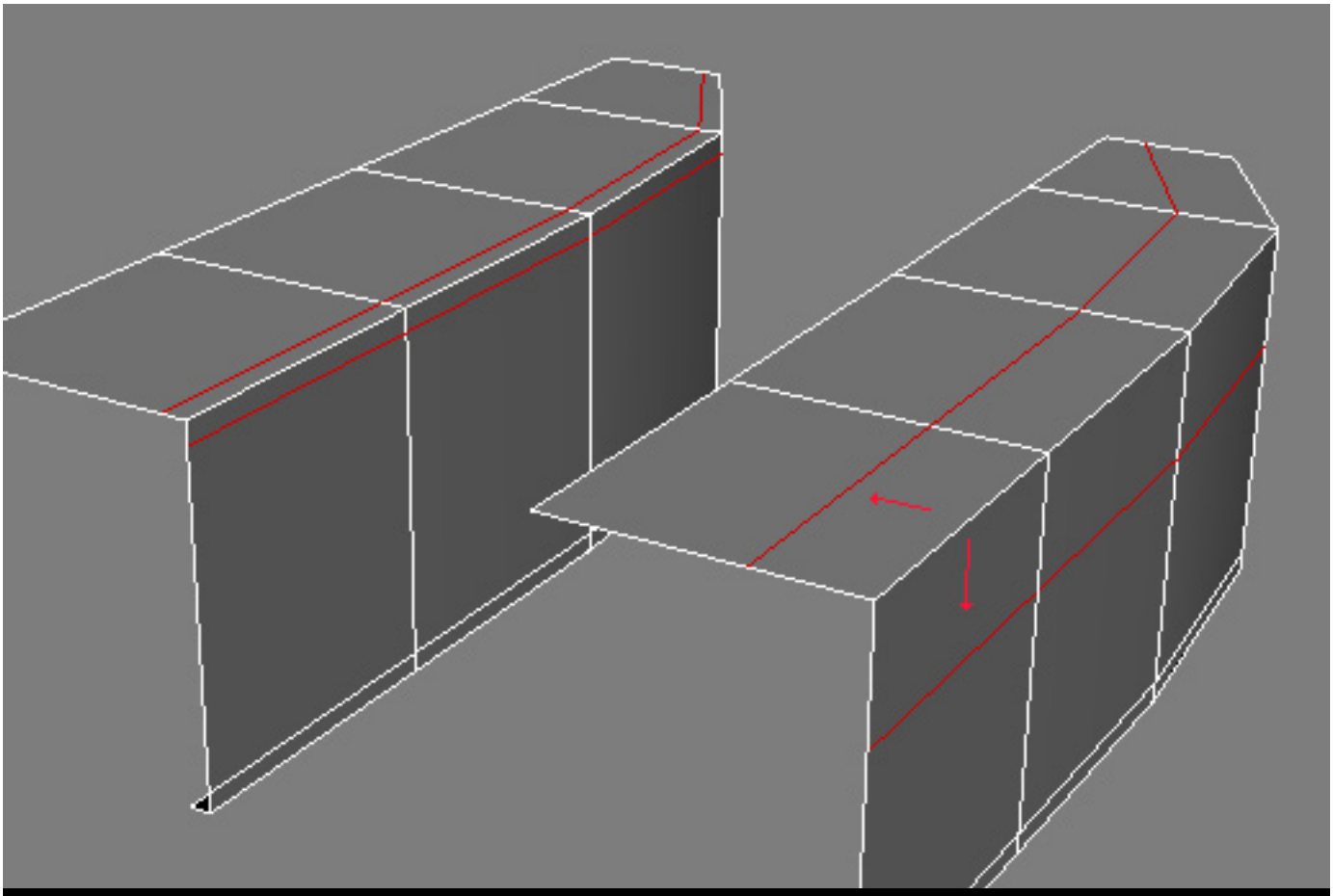
Why does an edge loop stop?

Perhaps you have already wondered why an edge loop suddenly stops, and what the reason for this would be.

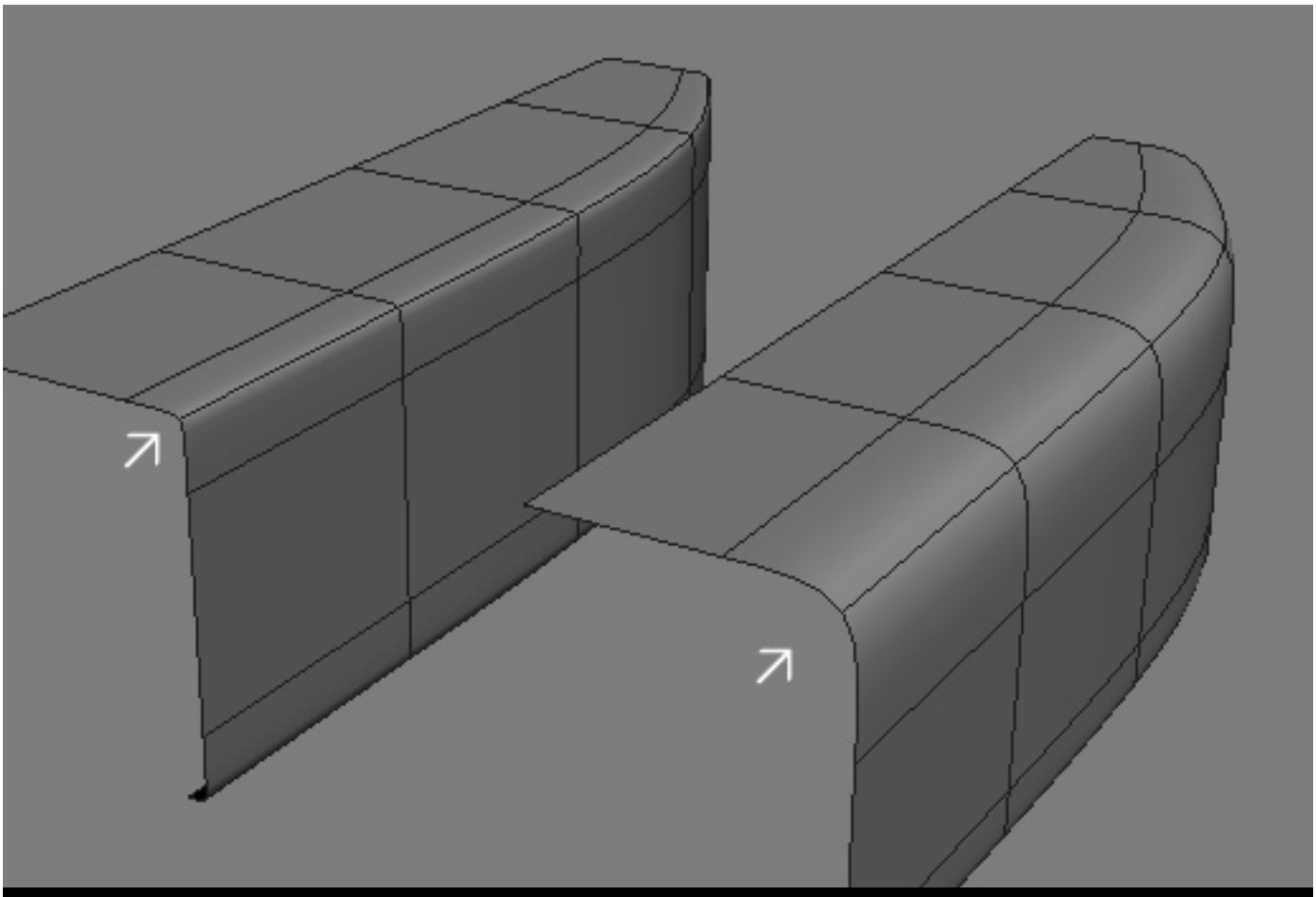
An edge loop stops when an edge needs to pass through a “pole”.



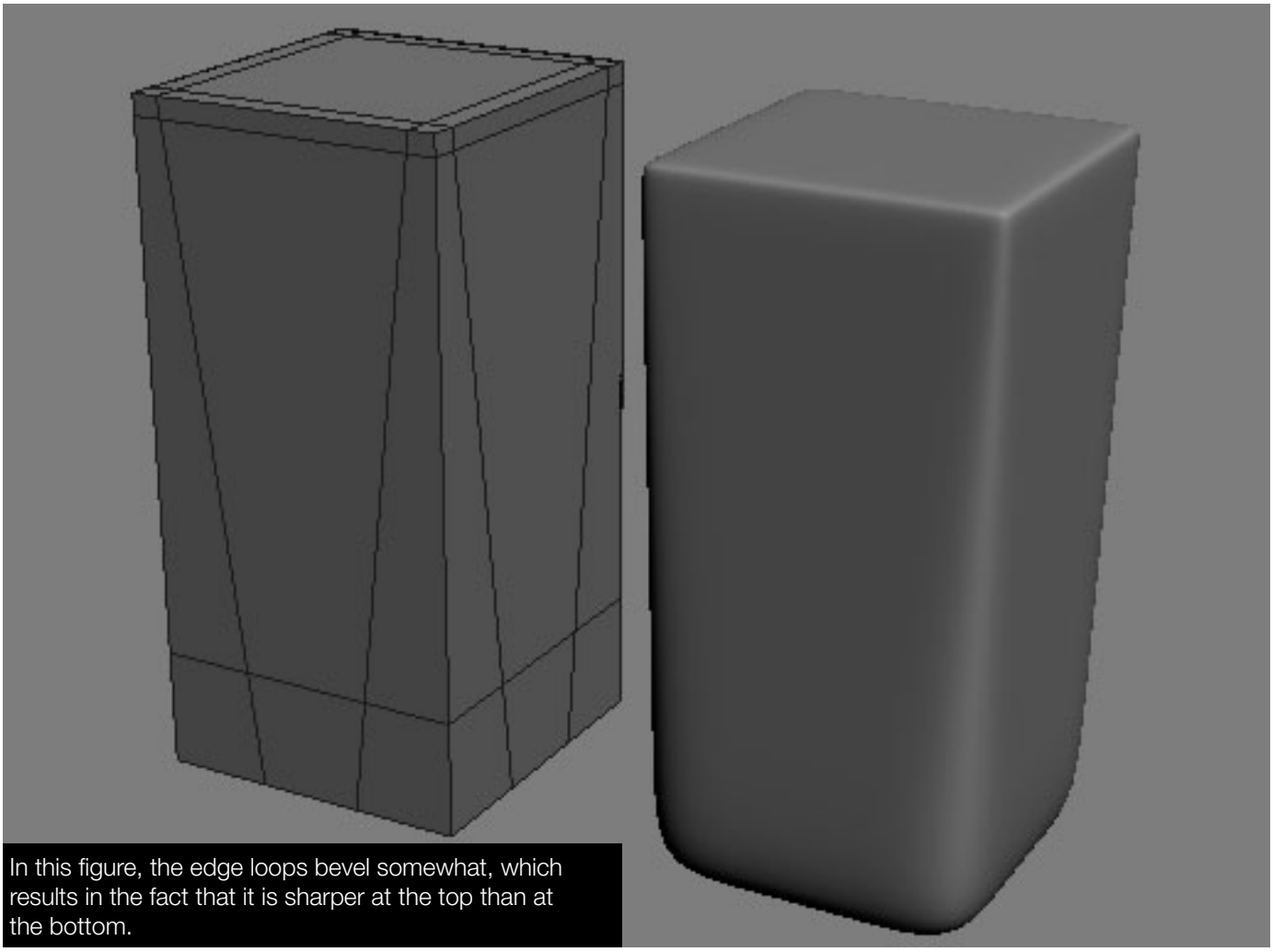
Make grooves throughout your mesh. By giving the angles additional edges, you make the rounding sharper.



Thanks to edge loops, you can quickly determine the size of your rounding. If you move edges further apart, the rounding will also be bigger. In this way, you control the rounding. Note that the original edge at the corner still exists!



The result of the figure above after rounding via the turbosmooth modifier (2 iterations, isoline display checked). The right mesh has a bigger rounding because the edges are further apart.



In this figure, the edge loops bevel somewhat, which results in the fact that it is sharper at the top than at the bottom.

POLES

A pole is the name for a vertex that looks somewhat different than the “regular” vertices.

There exist two types of poles, however, perhaps first a definition of what a “regular” vertex actually is.

Regular vertex

A vertex where 4 lines meet is regarded as a “regular” vertex.

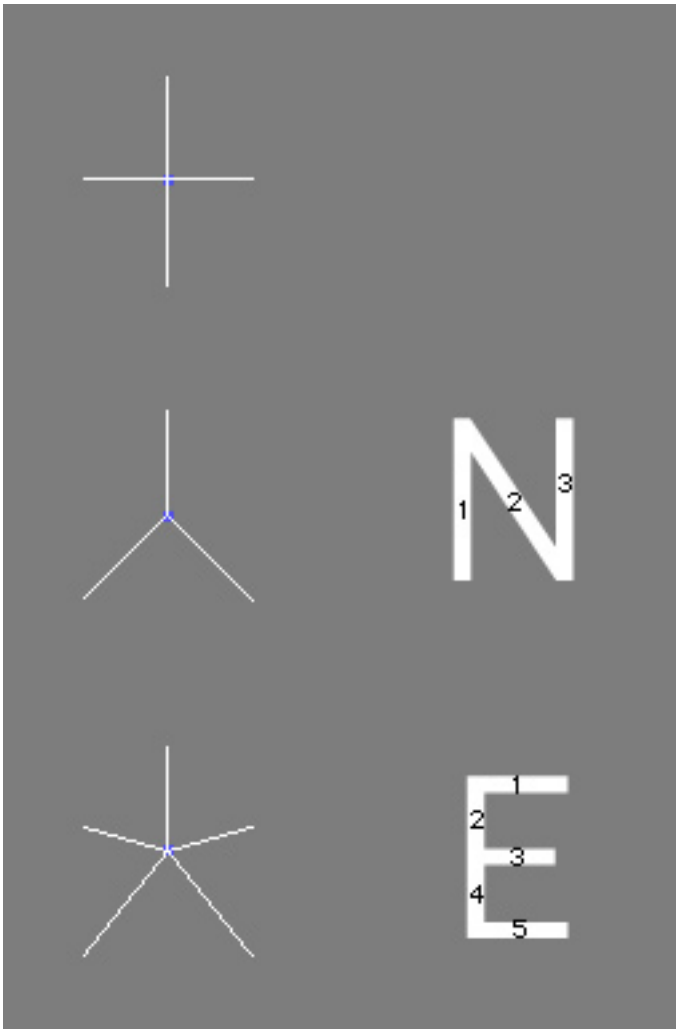
A pole is a vertex where 3 or 5 (or more) edges meet, so not 4.

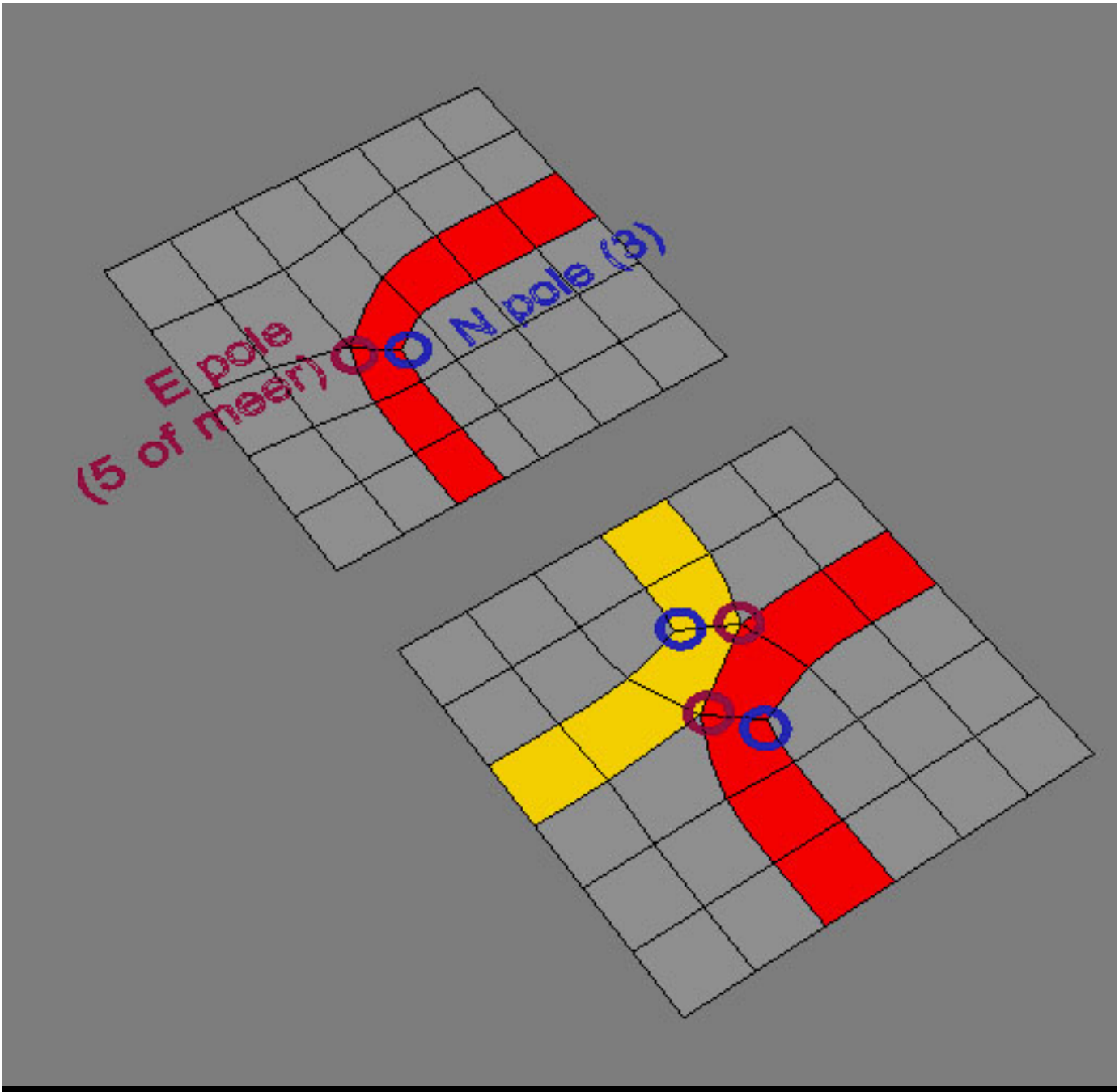
N-pole and E-pole

A real standardized name does not exist; however, unofficially we speak of an N-pole (3) and an E-pole (5 or more).

A reminder to know the difference between an N- and an E-pole: the letter N is built out of 3 lines, whereas the E is built out of 5.

The E-Pole is also sometimes called the “Extrude pole” because you obtain it when you extrude a polygon.

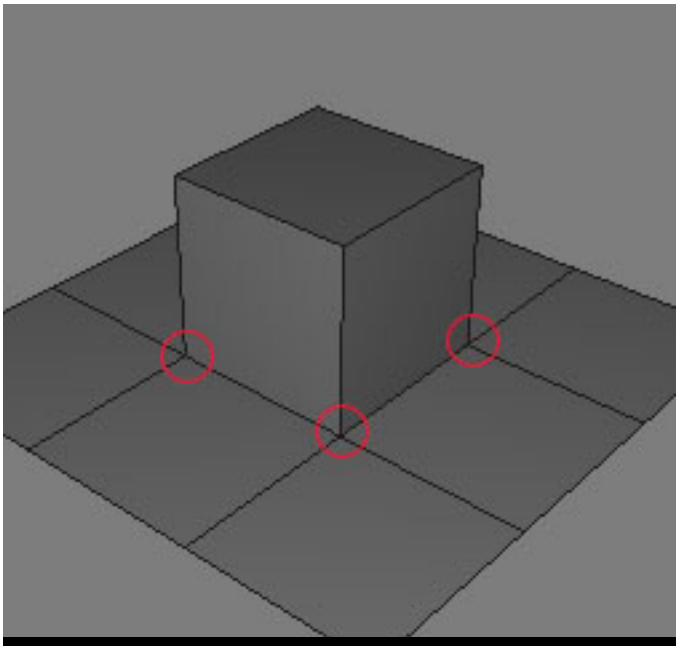




E-pole and N-pole create a twist in the mesh.

Poles create a twist/change in the mesh. Guideline: never put a pole on the edge of the mesh; however, place it somewhere in an open spot on your mesh. When you place it at the edge, the mesh can give a chamfer at that place and you will see the deformation.

Note: Poles are not bad! They will certainly be present when you SubD model. Just know it is best to put them in places where they do not disturb too much.



E-pole is sometimes also called Extrude Pole.

DARE TO CUT INTO YOUR MESH

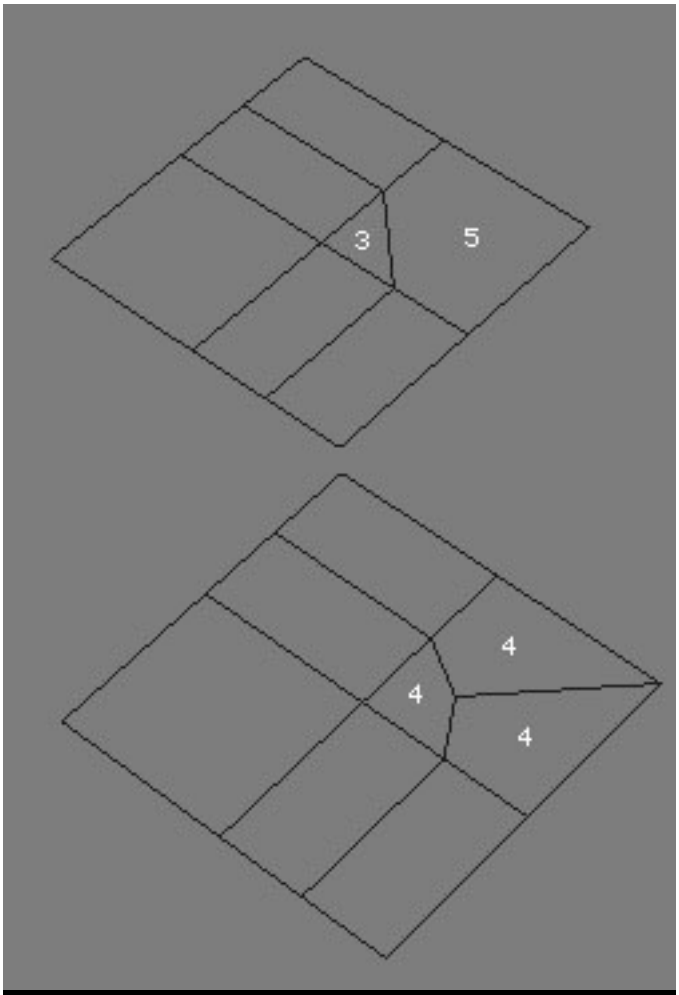
Cutting (holes) in the mesh

One of the aspects students have difficulties with at the beginning, is determining where certain cuts and chamfers should be. The modeler himself needs to be able to determine where a certain detail must be without depending on the mesh. You need to be able to cut where you want and you need to be able to solve the thus arising problems (non quads).

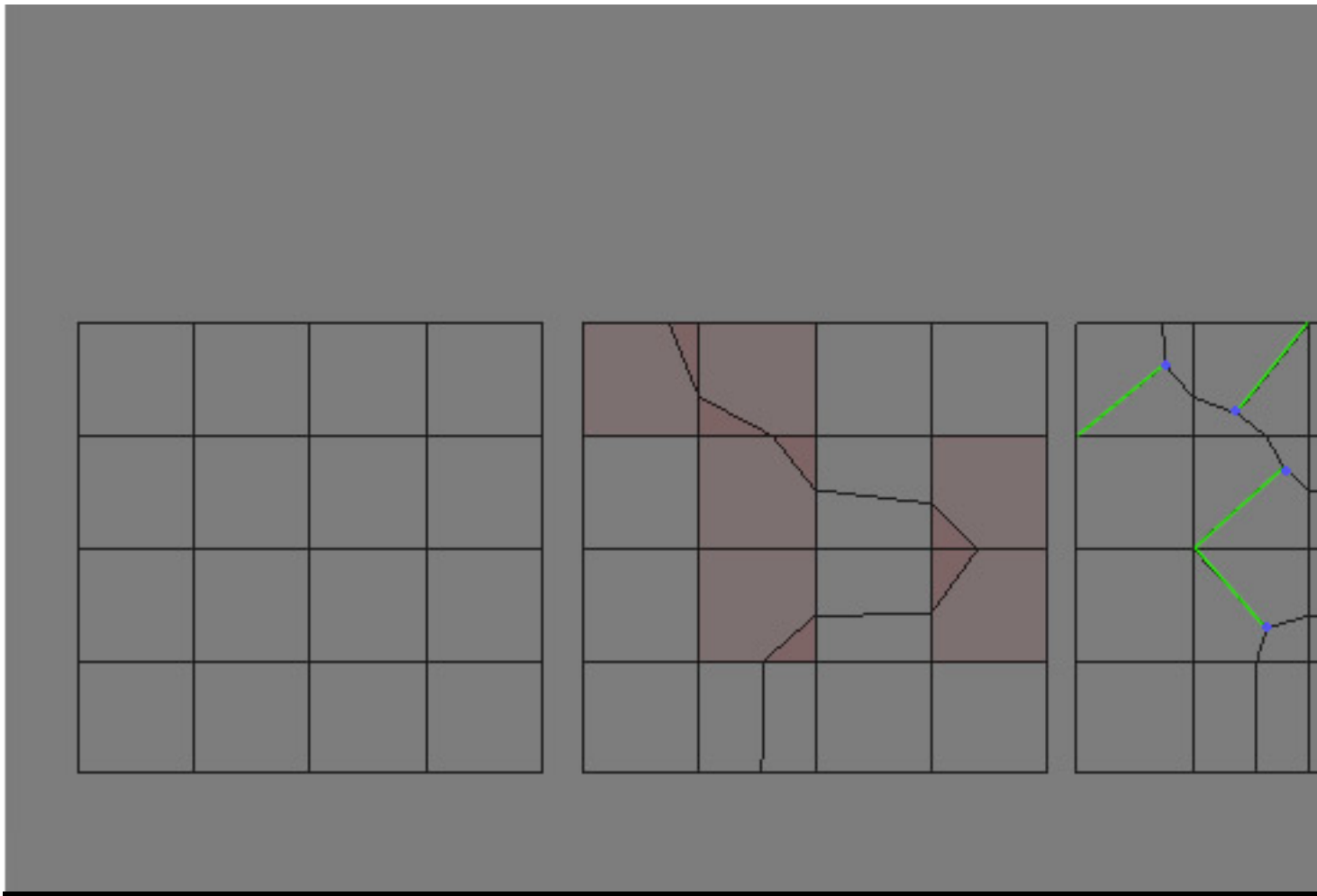
Tip: use splines as a reference! For it is extremely difficult to for instance cut a perfect circle without reference.

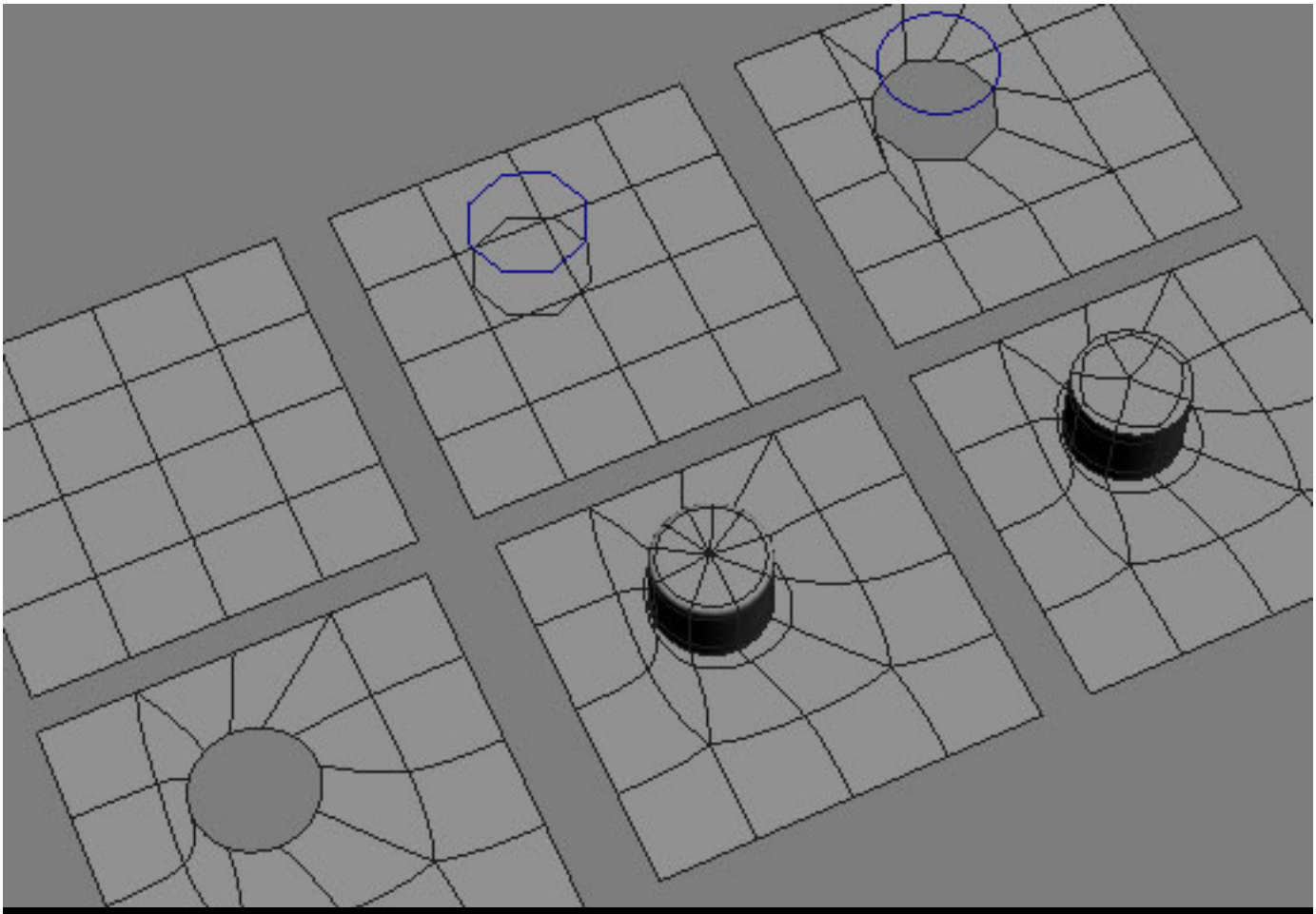
Draw a circle and put it on top of your mesh where you want to cut a circle. Because the circle now is perfectly round, and we let a smooth modifier rounding over the mesh, we should actually cut a low poly circle. Therefore put the interpolation of the circle on 1 (you can also work with an Ngon instead of a circle).
Go in Top view. Cut in the mesh where the angles of the (low poly) circle are. You now obtain a mesh in which possibly will be triangles and Ngons, which you now need to solve and make quad.

In the example below, you can do the following: when you cut a pentagon through from an angle, then you obtain 2 quads. The adjacent triangle then also turns into a quad because the latter gets an additional vertex.

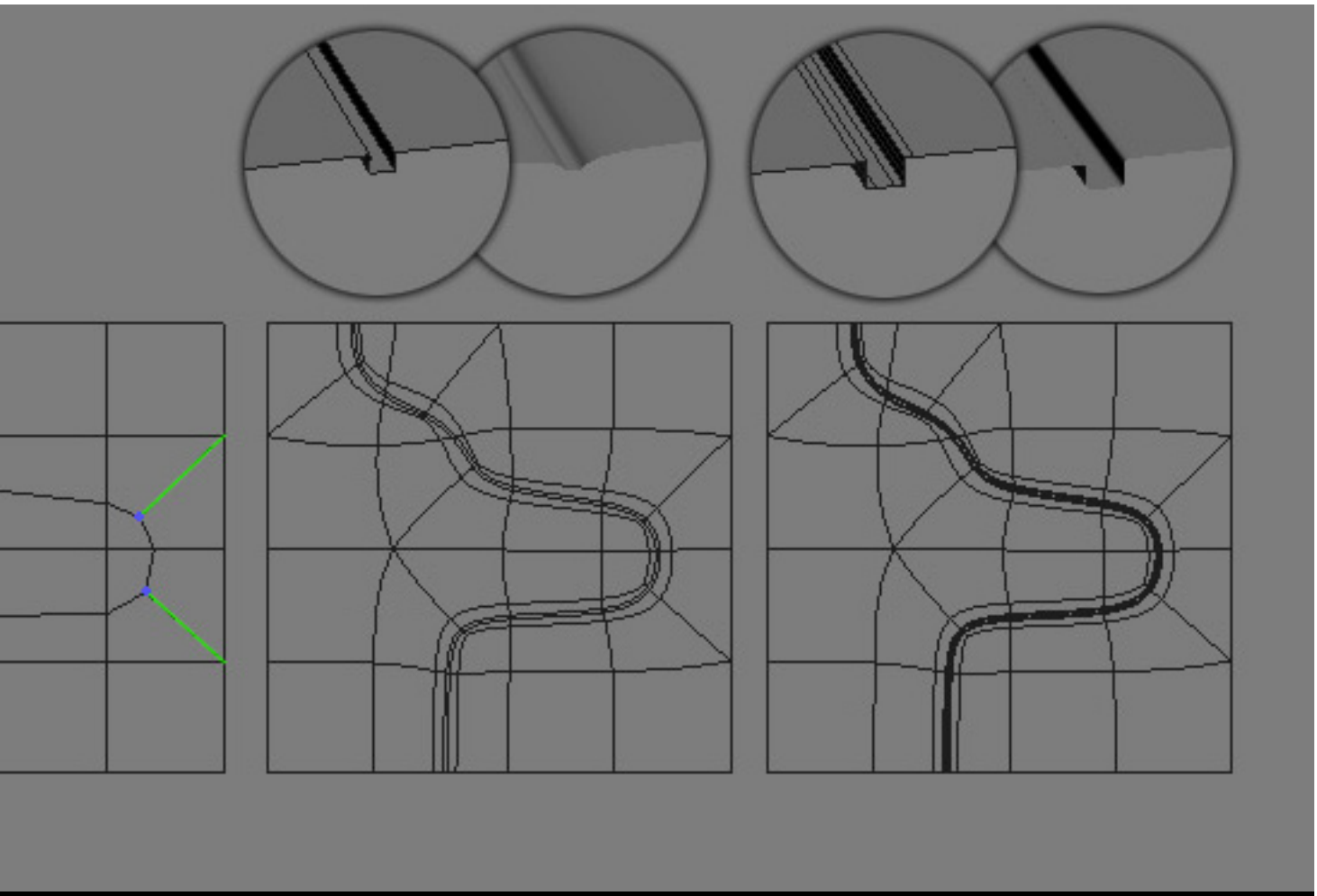


The triangle + Ngon solutions: cut the Ngon in half.


Randomly cutting through a mesh results in triangles and pentagons. You can solve these by cutting the pentagon in half (green



Use a circle (interpolation 1) as a reference to know where to cut. Next, you need to make everything quad and this sometimes requires some thinking.



n line). Then, you obtain everything quad again! Subsequently, you can finish and make a chamfer.

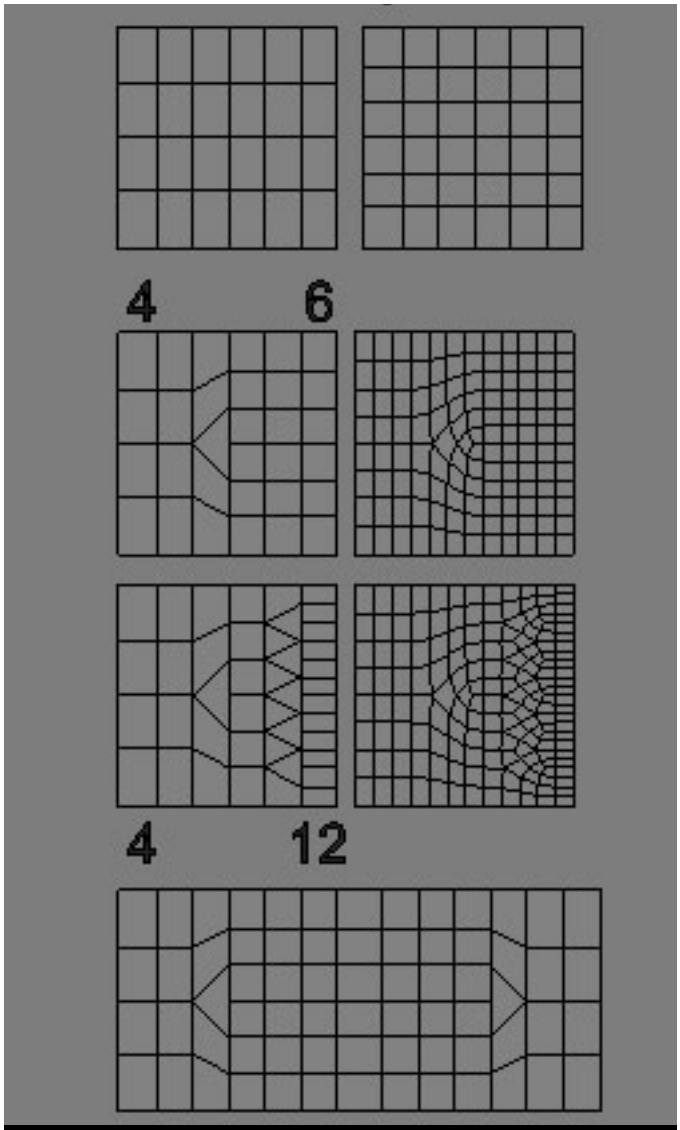
FROM FEW TO MANY DETAILS

Putting detail in a specific place

One of the aspects you will certainly encounter is the following problem: how to put more detail at a specific place?

The obvious possibility is to cut everything in it and to extend the edges throughout the entire mesh, with as a result more detail in the entire mesh. However, to avoid this and to keep the mesh clear, it is better to put the details locally. A useful technique is shown in the adjoining illustration.

The adjoining figure shows the following: the less good version at the top, by which additional edges are cut over the entire mesh. This creates a lot of overhead.
Underneath, you can see a transition from 4 to 6 polygons by building in “fake” triangles; for it exists of 4 vertices and therefore is a quad. Thus you can build out this system further and give it more detail.



Making the mesh more detailed is possible as described above.

Creating additional edges over the entire mesh. From less detail to more detail and back to less detail.

POLYGON LOOPS (of EDGE RINGS)

Polygon Loops are easy to select “flows” in your mesh. They strongly determine the structure of your mesh.

First of all, there is the name confusion. Edge Loops and Polygon Loops are sometimes used one for the other. However, there actually is a difference. The difference is in selecting.

You can select an Edge Loop by selecting one edge of the loop and then pressing the button “loop”.

You need to select a Polygon Loop by selecting a transverse edge and by pressing the “ring” button. This is why it would perhaps be better to speak of “Edge Rings”, though this is not the case.

Underneath you can find a number of examples:

O loop

The most used loop is probably the O loop. An O loop is nothing more than an inset of a number of polygons. The newly obtained structure is somewhat rearranged by moving the vertices and thereby obtaining a circular (or oval, or the like).

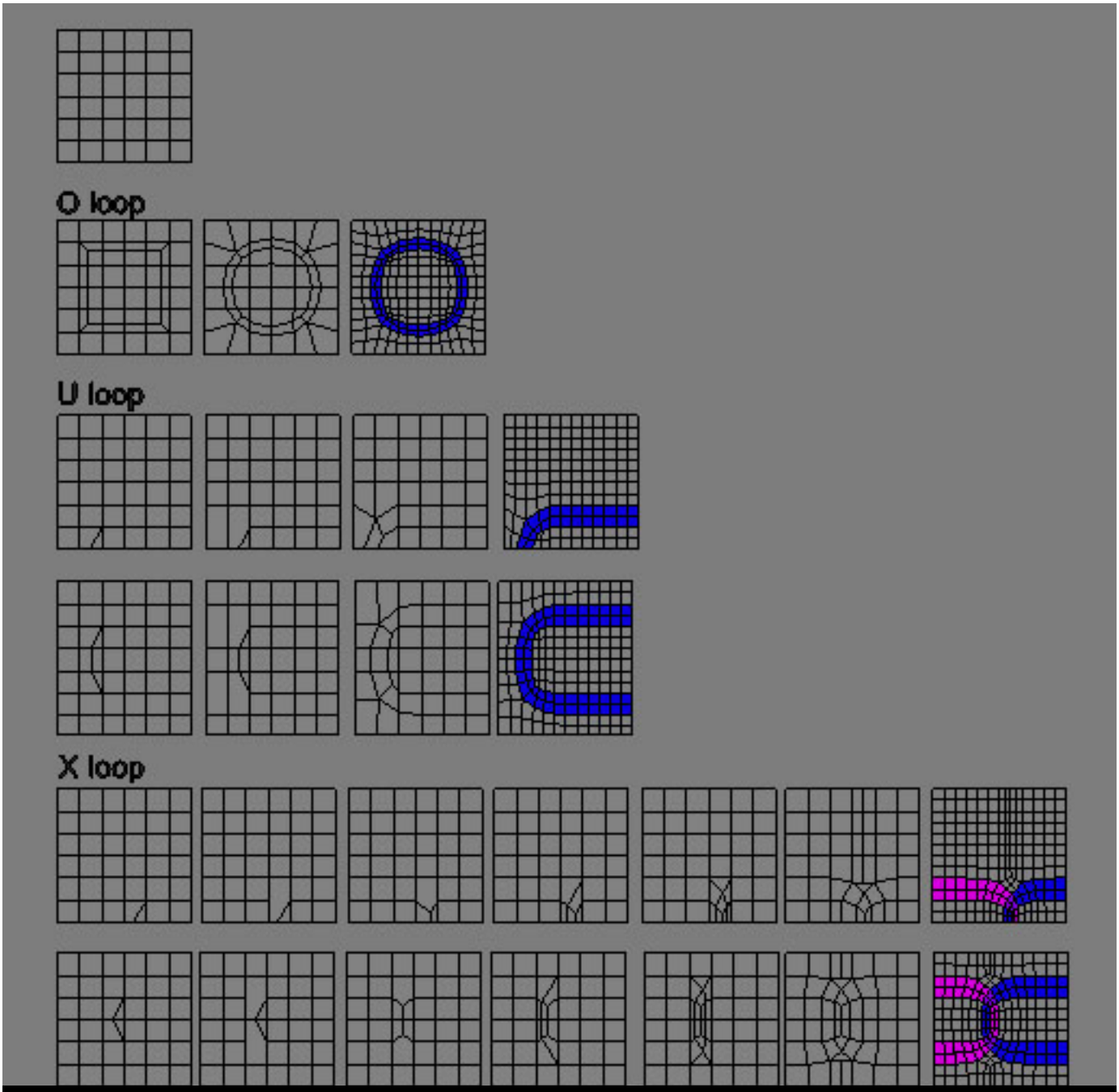
U loop

The U loop is useful if you want to obtain a certain wave in your mesh.

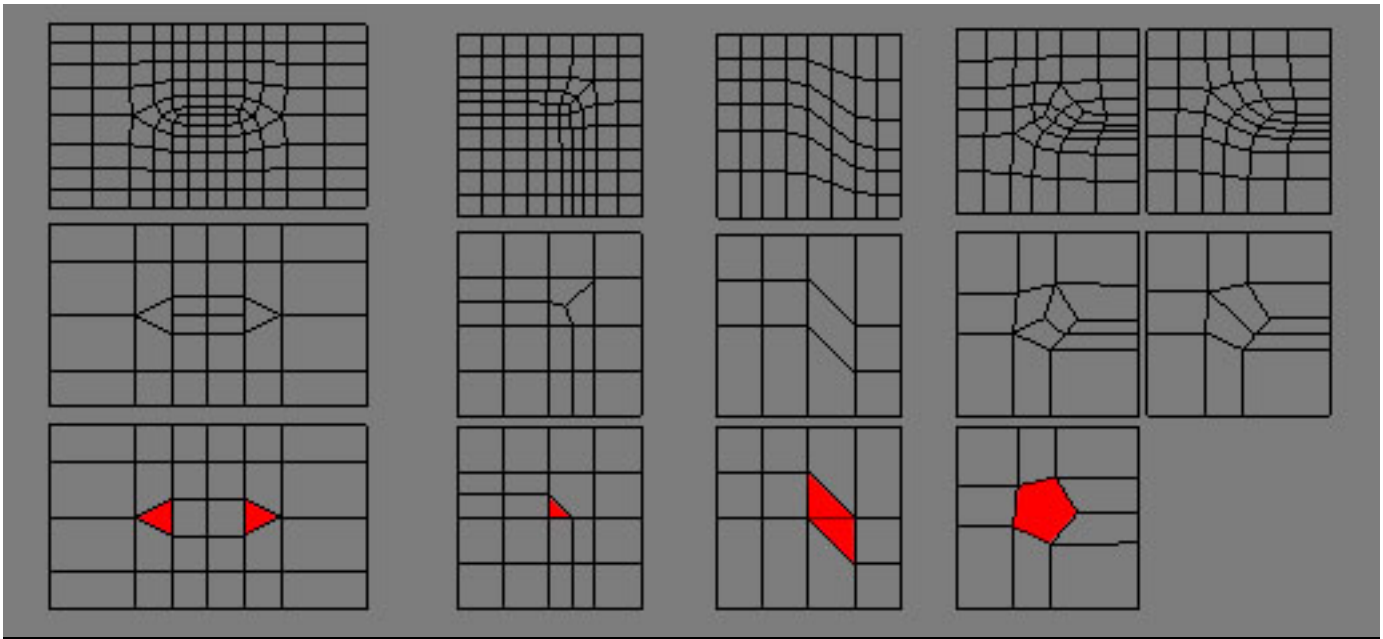
X loop

This is already for specific cases by which two flows are crossing one another.

The loops we worked out here, have possibly been approached very theoretically, however, you will encounter them in some of your models. It is possible that you will not notice them, nonetheless we ask you to study them well, so you understand what is actually happening.



In this figure, different loops are worked out. Study them well to know how they are built. Everything is quad!



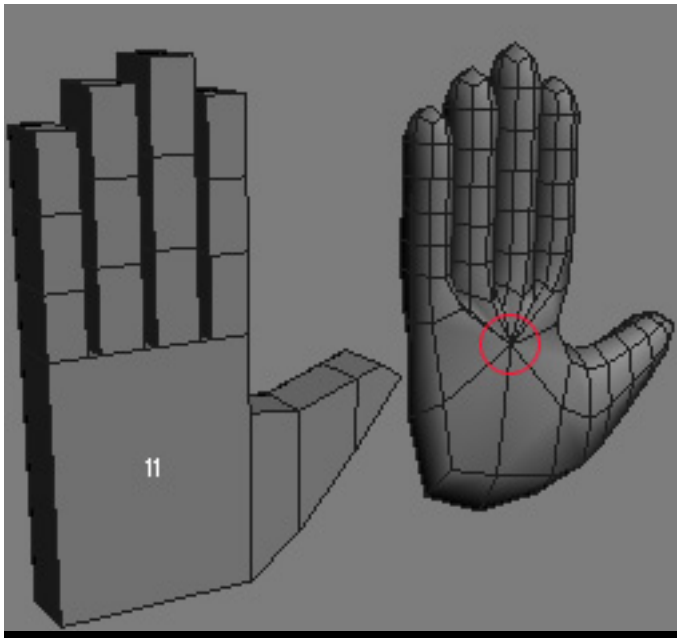
A few examples to solve non-quads. It is possible by removing edges or by additional cutting.

BLOB MESH = BAD MESH

You obtain a “blob” if you do not follow the rules described above and if you have too high hopes for the turbosmooth.

Of course, it is not easy to apply all the rules well. However, as always: practice makes perfect!

If you have polygons that are not quad, you already increase the chance that your mesh will have a blob mesh. At the right, you see a mesh of which one polygon has 11 vertices (=N-gon). The subdivision algorithm will try to solve this, however, this will result in a pinching effect and this will not be beautiful in a render.



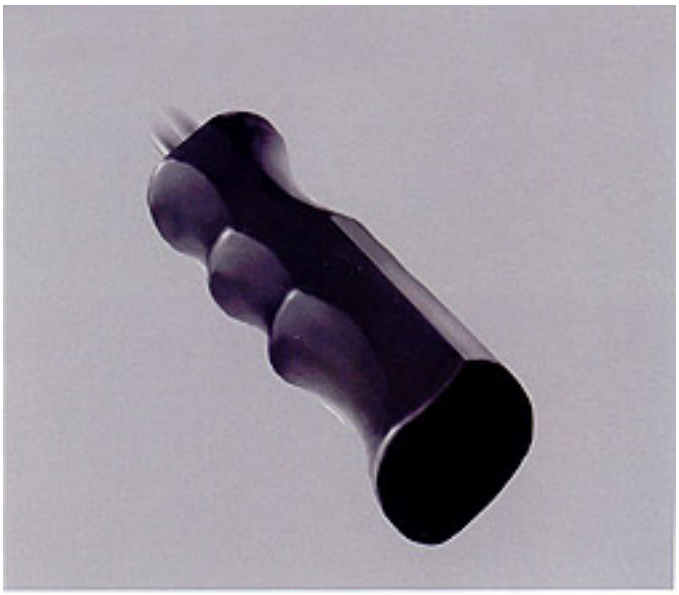
Say stop to the blob mesh! (Pinching by N-gon)

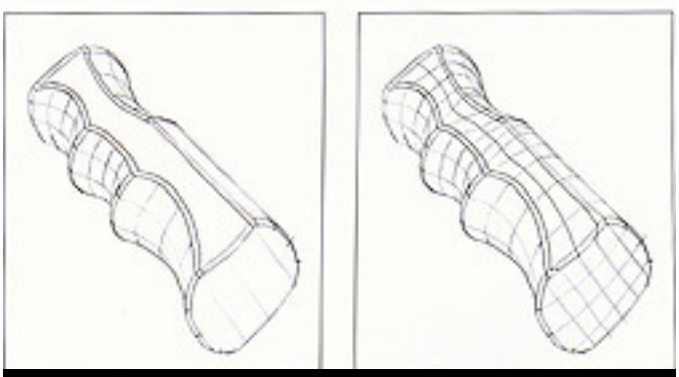
FLOW OF THE MESH

By flow, we mean how the polygons blend.

Make sure that the flow of your mesh is logical. This may seem very easy, however, it is not.

Tip: first sketch your mesh on paper. Start as follows: at the borders, put the edges close together, try to keep everything quad. Subsequently, also do this with the aspects that are very characteristic for the model (e.g. where buttons, extrusions or holes are). Then try to connect these aspects as proportionally as possible. Try to find a balance in the number of edges you use to connect everything. You do need to make sure that it remains clear, because you need to move all these vertices.




Make sure that the flow of your mesh is good.

A FEW GUIDELINES

High Poly, nonetheless …

Sub Division modeling is also sometimes called High Poly Modeling; however, this does not mean you have to be lavish with polygons. It is the smooth modifier that ensures that your base mesh gets more polygons. Keep your base mesh clean, legible and yet as low poly as possible.

Start with a simple base mesh and build further on it. If you start with a too high number of vertices, you will constantly need to adjust and correct these (excessive) vertices. In the long run, you will lose oversight and it will become a mess. Most of the time, you start with a box that has a small number of segments.

Tris on the mirror line

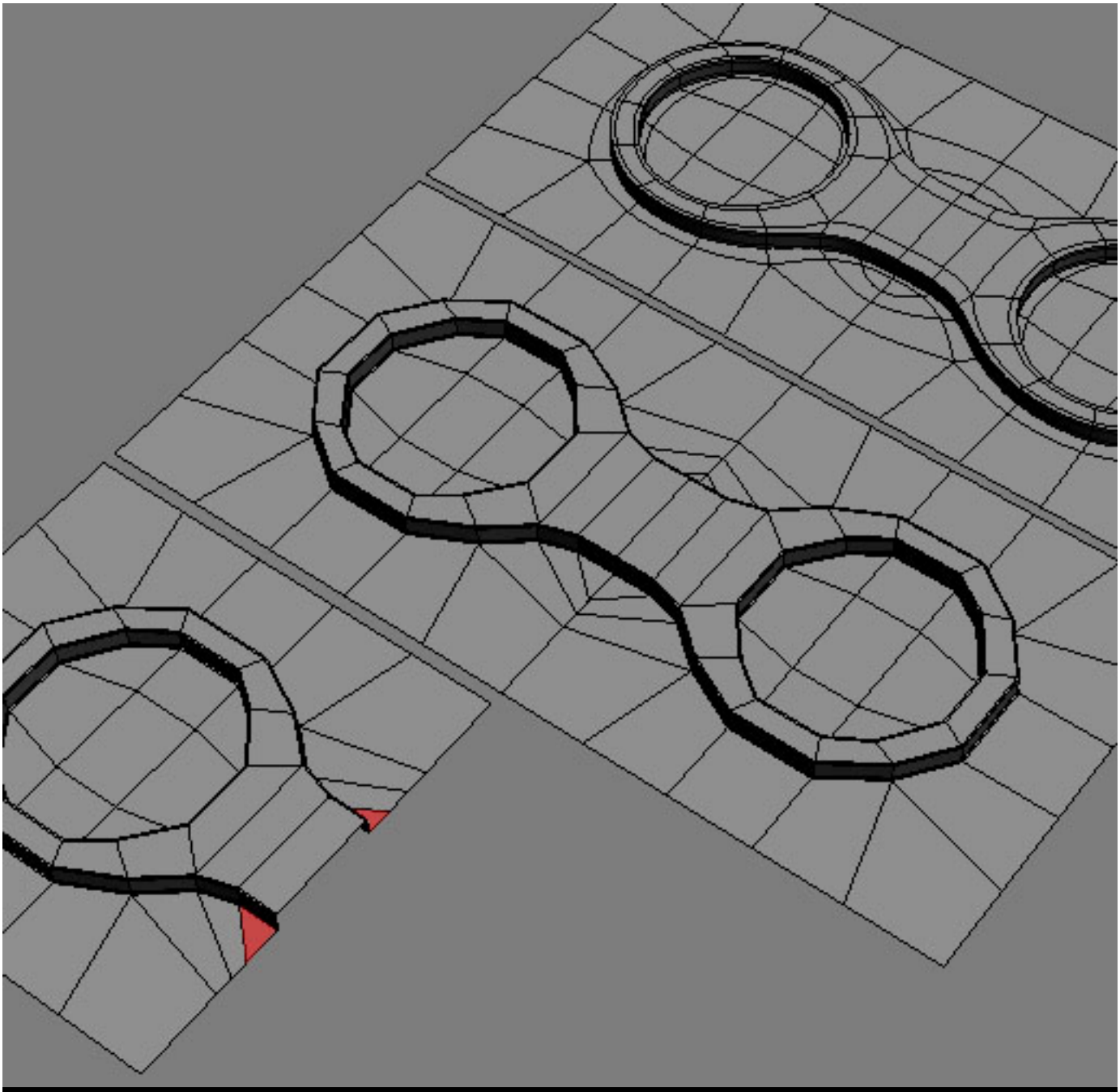
In theory, a triangle on the mirror line can do no harm, because it is mirrored, and two adjacent triangles form a quad. (The sides need to touch each other)

Equally divided mesh

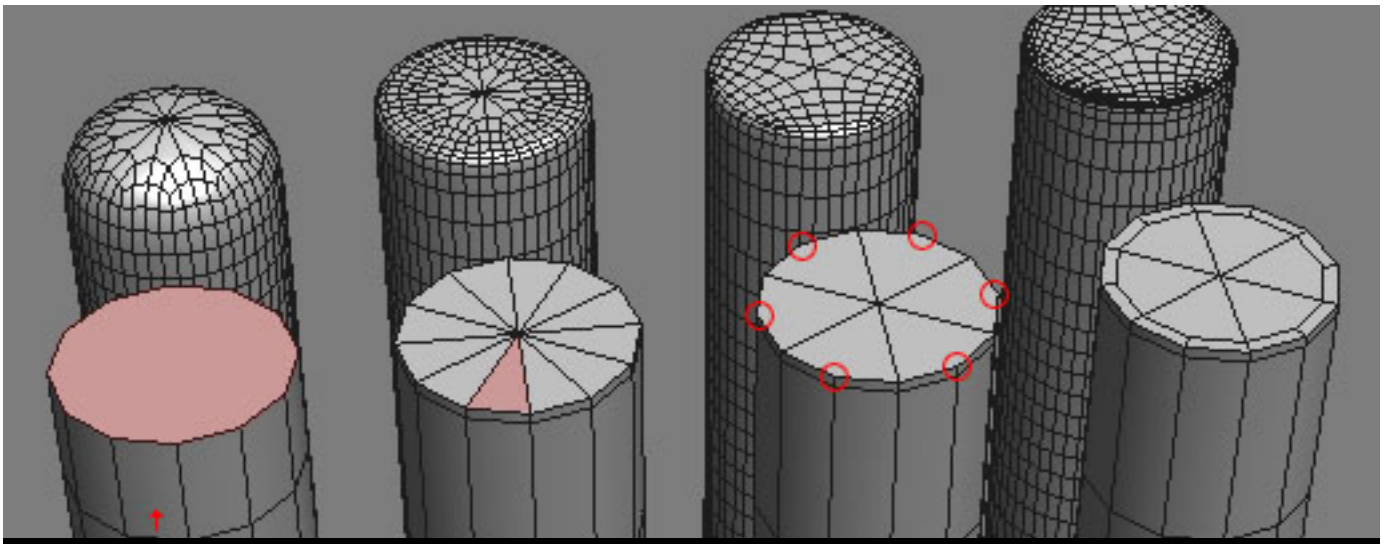
It is for the best that your mesh is equally divided and that you have no large surfaces with regard to the rest of your mesh. If not, you get the chance that there will be stretching in certain places.

Spin Edge Technique

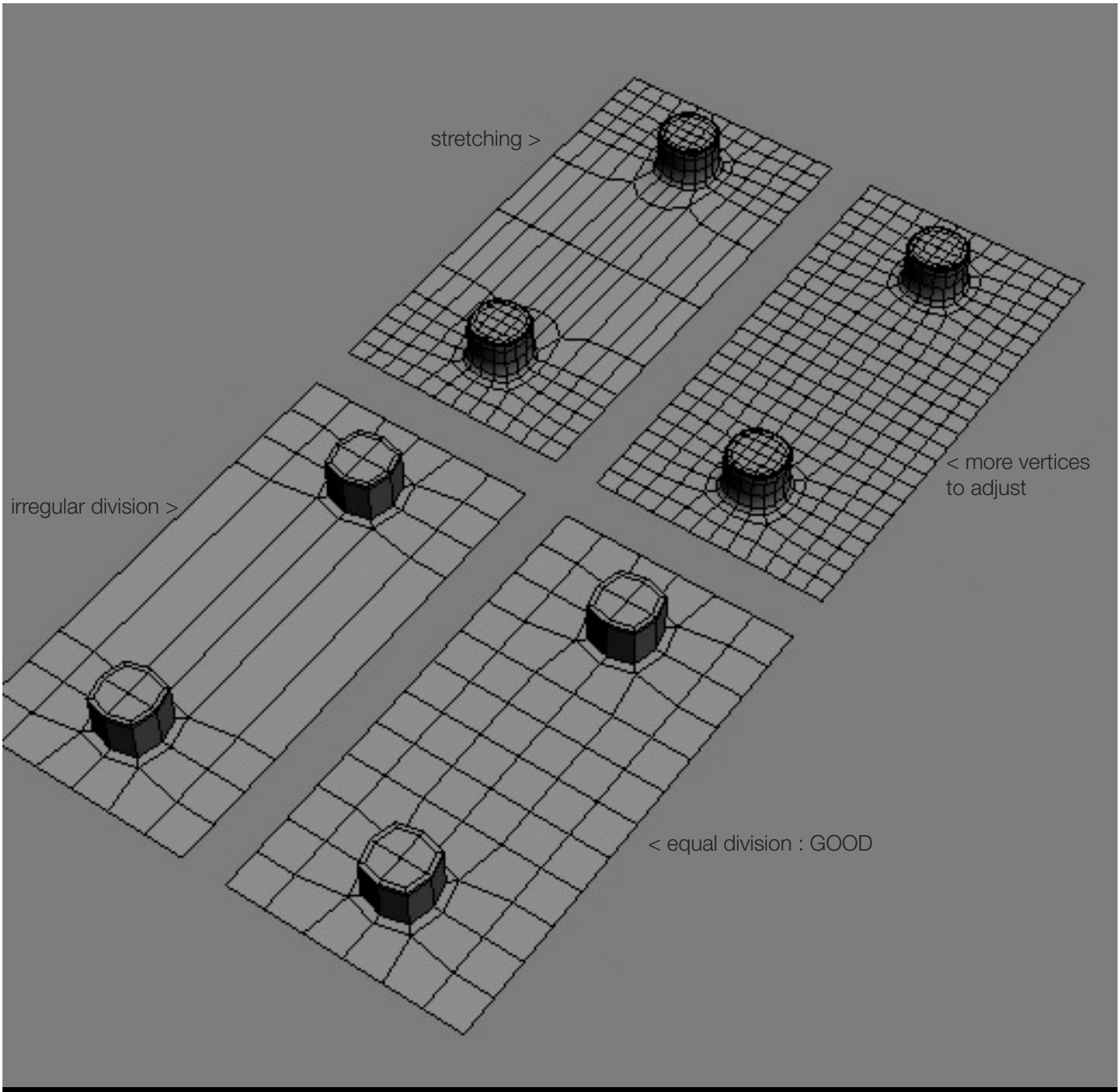
The spin edge technique rotates the common edge between two polys. For this technique, there is no button in 3dsMax, however, you can easily do it yourself. Via backspace, remove the common edge. Thus you obtain one large surface (non quad). Next, connect two vertices that are diagonally opposite to each other. Then you move the vertices somewhat to express the flow some more. Dependent on what diagonal you choose, you get another flow.



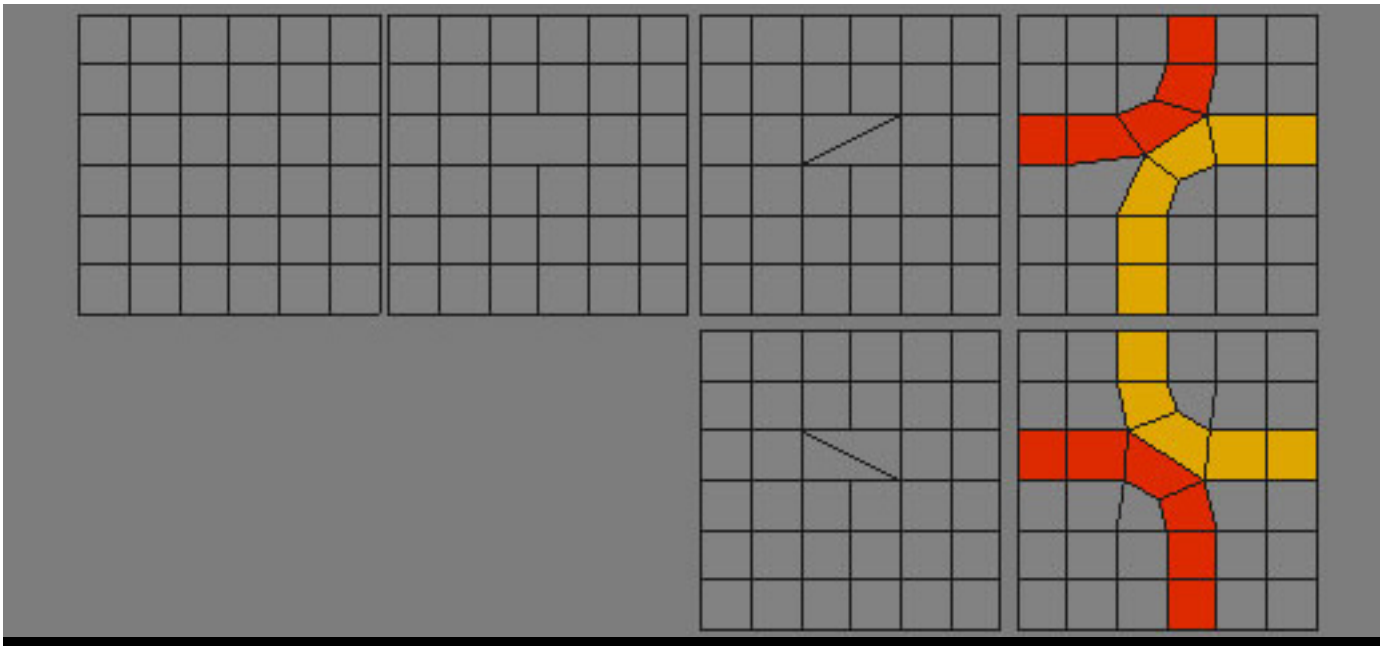
Triangles on a mirror axis are less “bad”, because they are mirrored and thus form a quad.



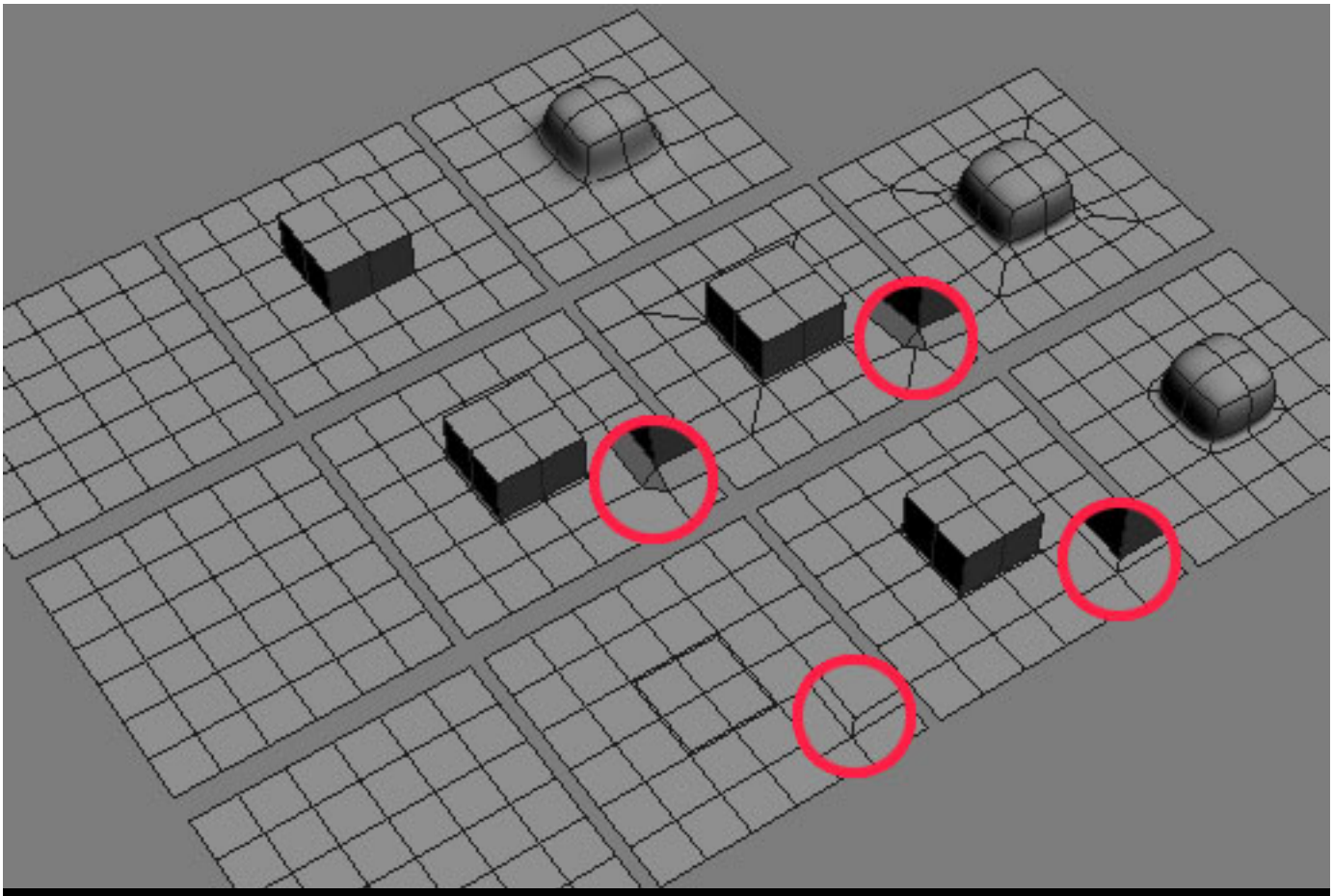
Step 1: the regular cylinder, above an N-gon. Step 2: and edge was moved towards the top of the cylinder and above an inset was made and all vertices at the center were welded together. You do get tris. Step 3: edges were removed and you obtain quads, however, you now have poles on the edges. Step 4: an inset solves the latter; the poles are removed from the edge. Consequently, you get a perfect SubD cylinder.



Irregular division can cause stretching.



Spin Edge technique allows to quickly obtain flows in your mesh.



Extrude a number of surfaces upwards and obtain a sharp edge on the rim: different ways are possible.
Top row: regular extrusion, gives a rather rounded result. In the middle row, the edge was chamfered and the triangles were worked out. Bottom line: first an inset and then the extrusion, is preferably.

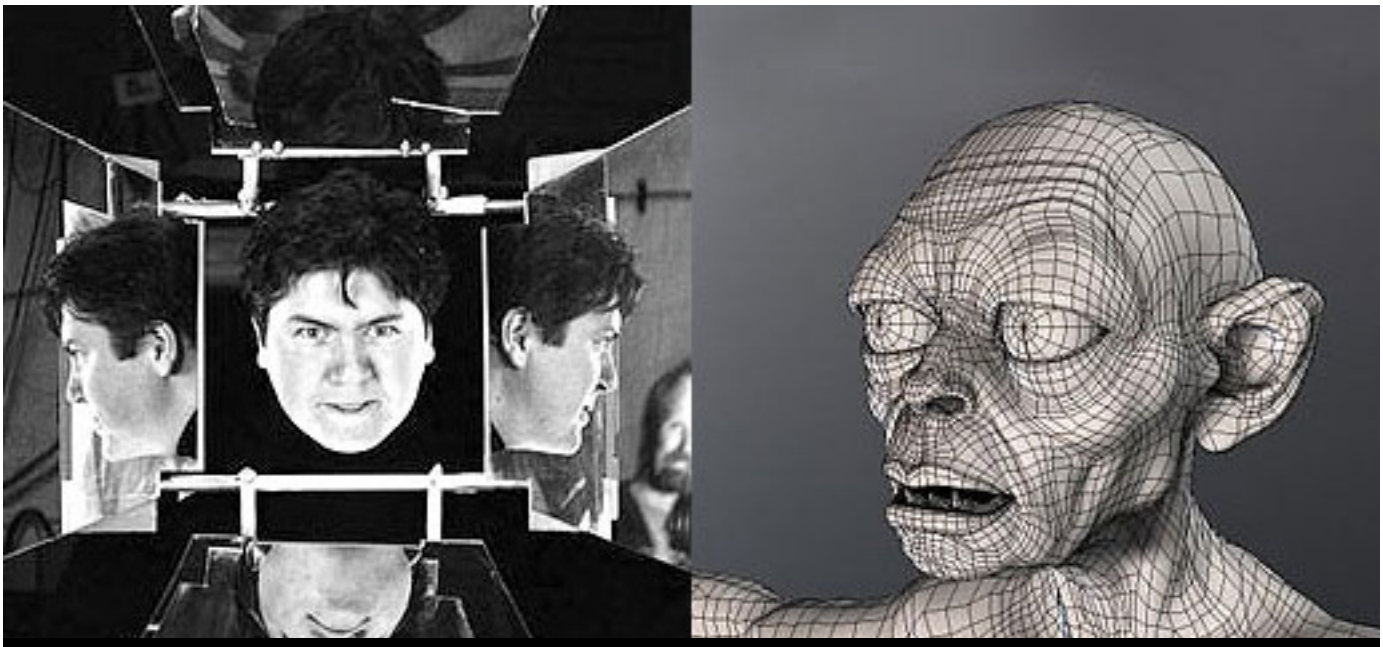
Multiple solutions

There are multiple solutions to obtain one and the same result. It is a matter of first thinking it through or to test what the best method is. Practice makes perfect. In fact, each modeler has his own way of modeling. We want to stipulate again that online, many tutorials on SubD can be found, however, most of the time, they do not explain why they do something and you just start copying. Be critical!

Conclusion

In this document, you find all the “tricks” to be able to SubD model well. Now, all it takes is some courage to start cutting into that mesh and to solve problems.

Now too, it is not possible to make something with just one click. It takes time and practice to master the tricks of SubD modeling. So practice, practice, practice!



Bay Raitt is the SubD modeler behind the famous LOTR character Gollem.

It may be useful to sketch the wireframe in advance before you actually start in 3D, so you have something to go by. Keep everything quad in your sketch; in this way, you force yourself to think it through instead of just trying in 3D. It is certainly possible that you will have to deviate from your sketch; however, you already have a beginning.

We too always need pencil and paper to sketch aspects to see if it would work.