

# Fitness for use

Introduction to  
professional  
standards in  
modeling



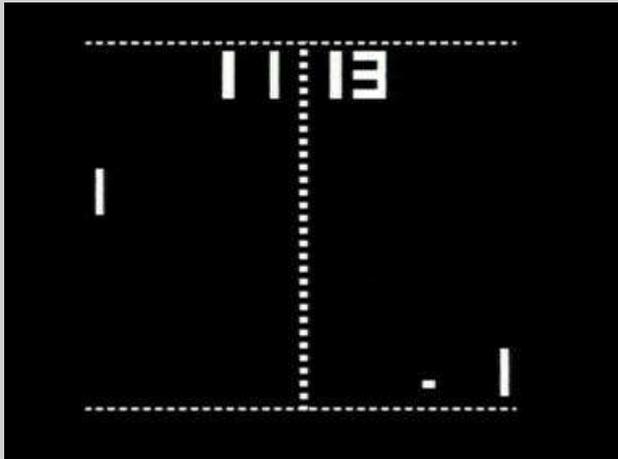
# Why this is important

- Every error type described here occurs, and at every level
  - Students
  - Third party content creators
  - Model libraries
  - Primary studios
- Because models are the first stage of the production process, fixing mistakes is costly
  - Who fixes it?
    - Other studios
    - Other modelers
    - Texture artists
    - Animators
    - Lighting artists
    - Compositors

# Target audience

- This lecture is designed for:
  - Students who want to learn what is expected in the CG industry
  - Early career professionals who want to fill in the gaps in their basic knowledge of CG
- This lecture will:
  - Identify the kind of problems most commonly associated with weak CG skills
  - Provide insight into how to fix errors associated with lack of professionalism

# Tool choice



Most professionals use more than one package

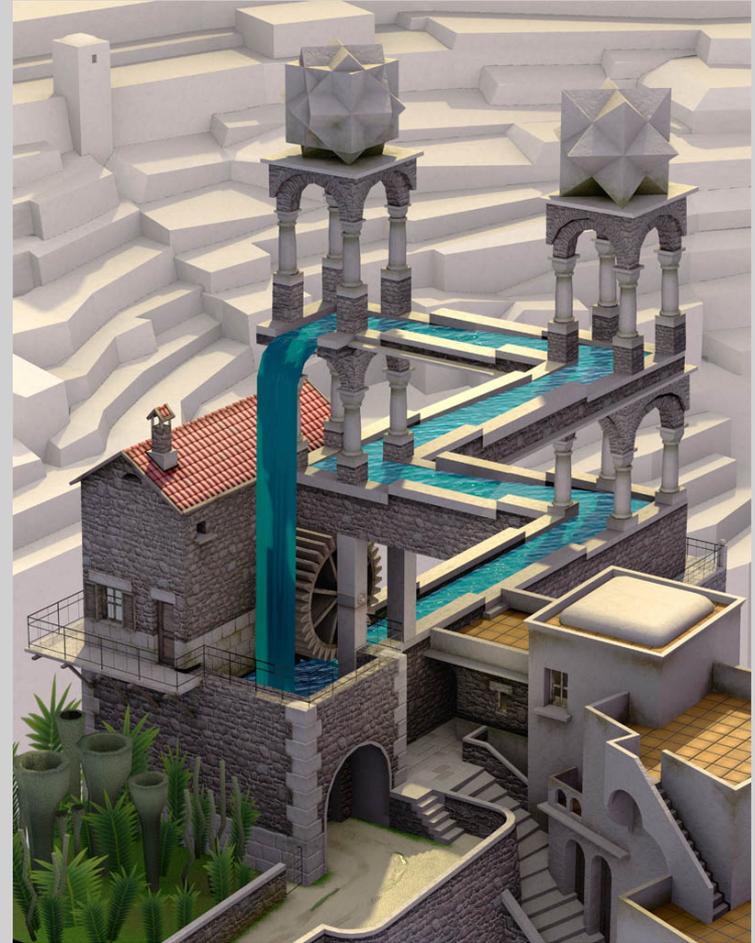
- All applications are based on the same principles
- An artist uses the tool, not the other way around
- Learning an interface is easy compared to becoming a skilled artist
- The application used is not the measure of skill or ability

# The same page

- Professional criteria in modeling are based on common understanding
- We start at the beginning

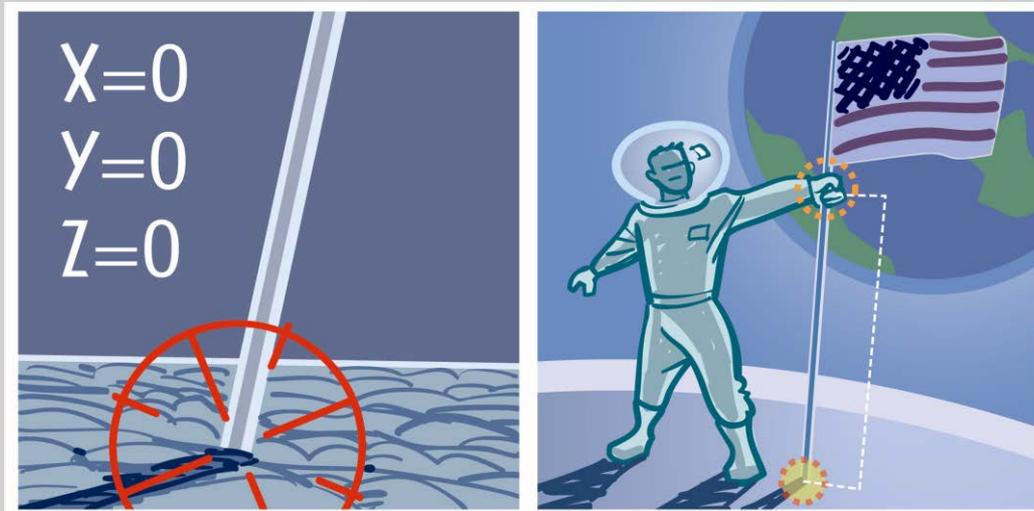


Jaguar E-type Roadster S1 - By Rachel van der Meer 2012

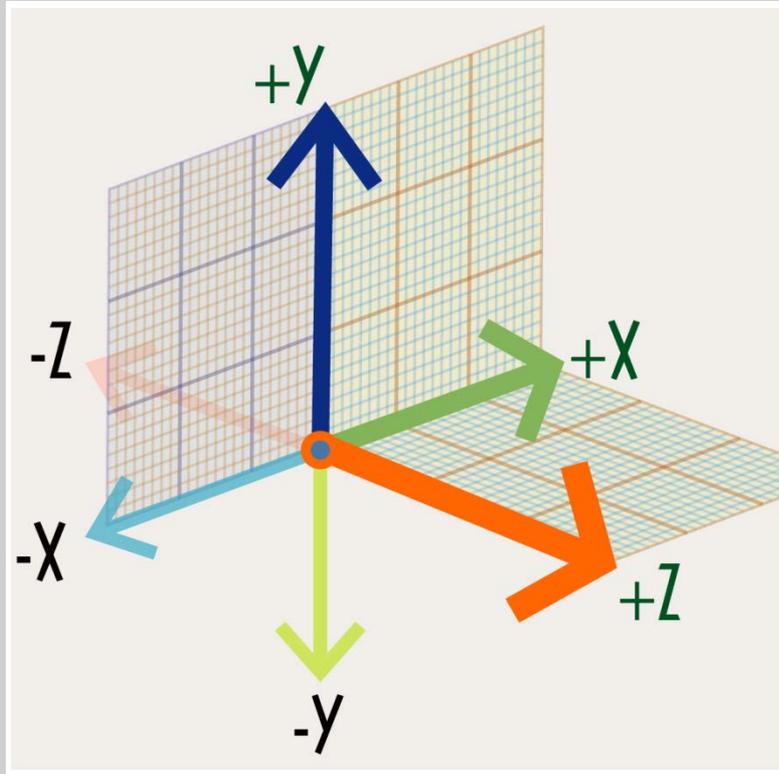


# The starting point

- Global origin
  - No scene without a global origin
  - Provides a zero point
  - Impact must be considered in all operations
  - Origin offset can cause dramatic technical errors

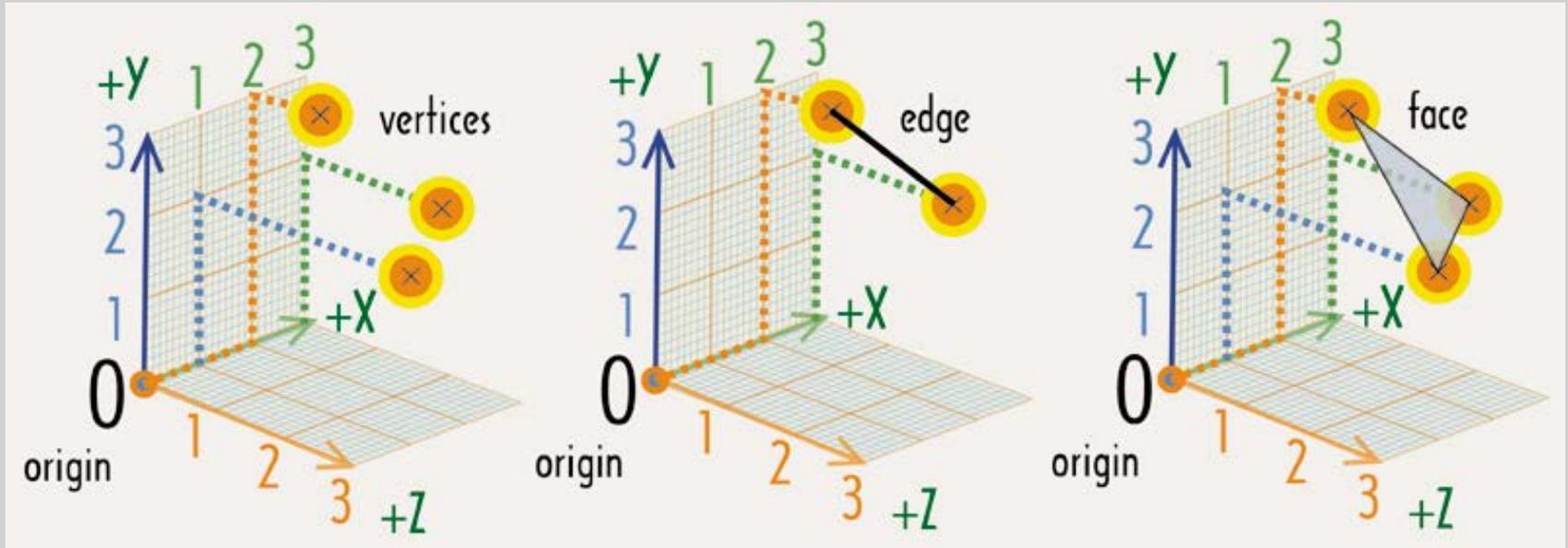


# World origin axes



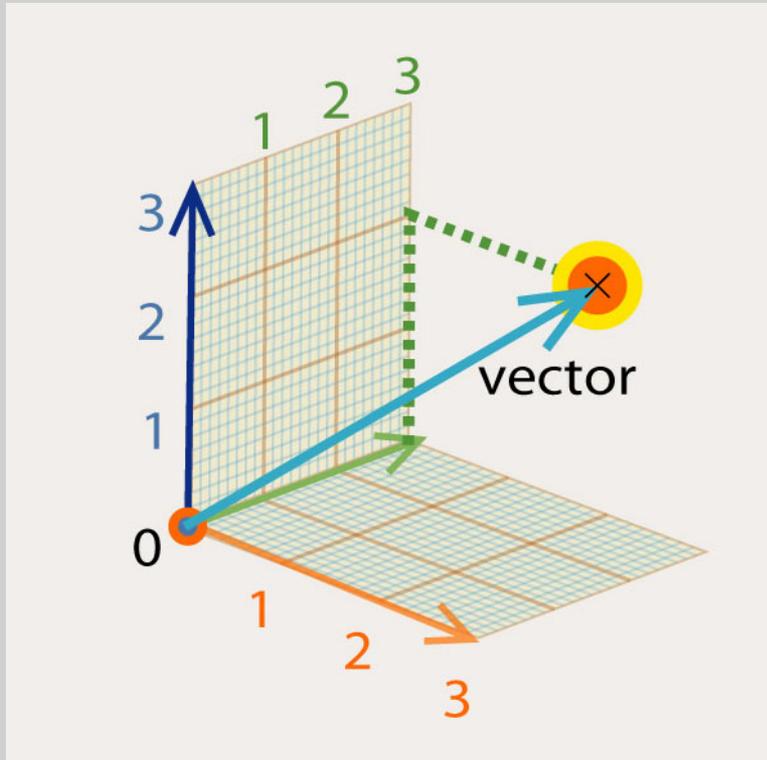
- All measurements relative to the six global axes
- All non-zero positions are measured as distance from zero
- Three axes allows triangulation of location
- Difference between modeling in world or local axes can cause likeness errors

# Geometric elements



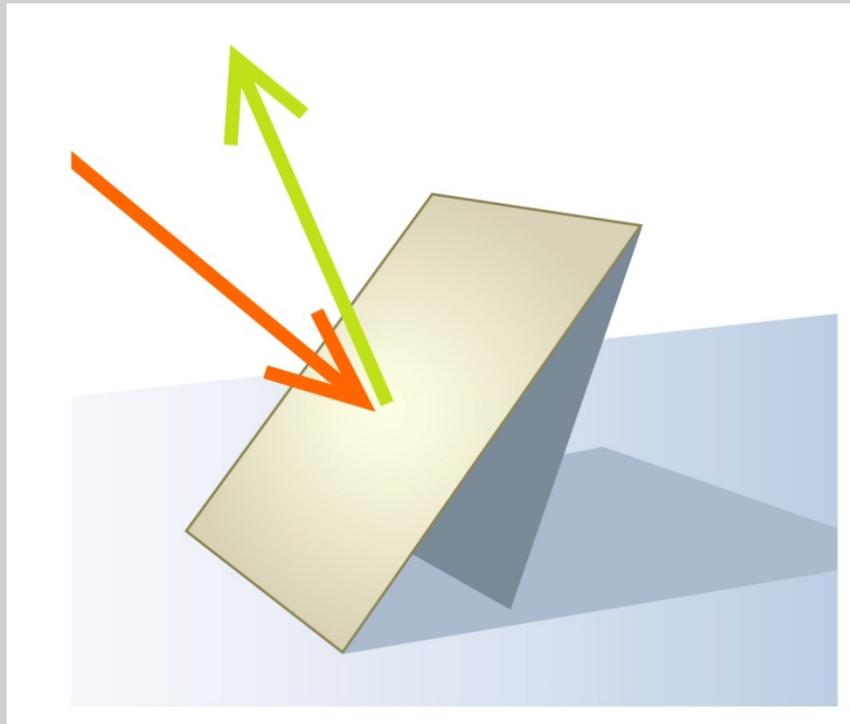
- Vertices, edges, and faces
- Understanding each of these is crucial to making professional models
- Most modeling errors involve one or more of these components

# The vector



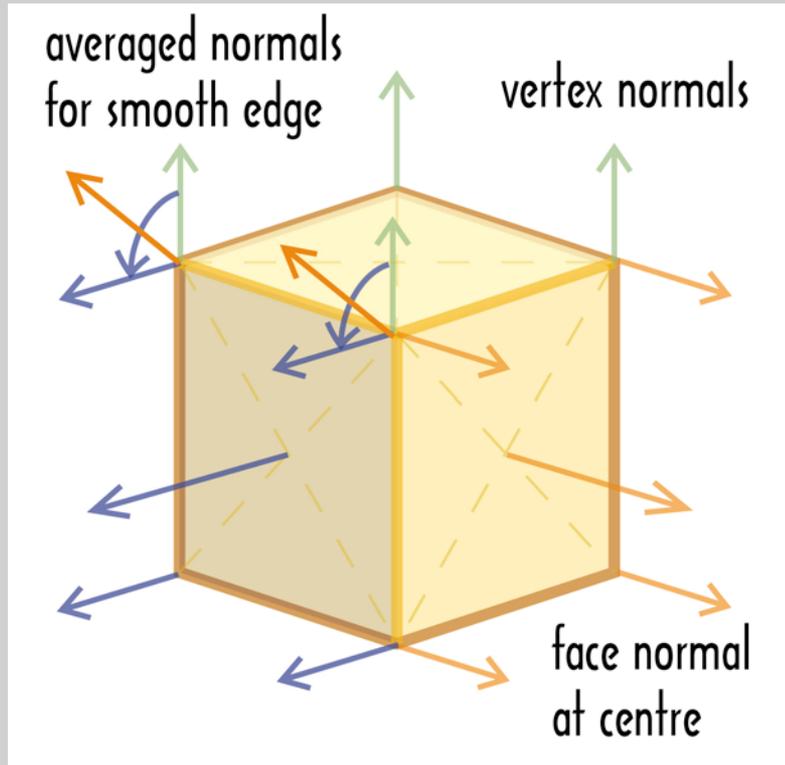
- A direction defined by 2 points:
  - The origin is always point 1
- The vector is drawn through a second point from the origin
- The vector is written as a single point

# Render requirements



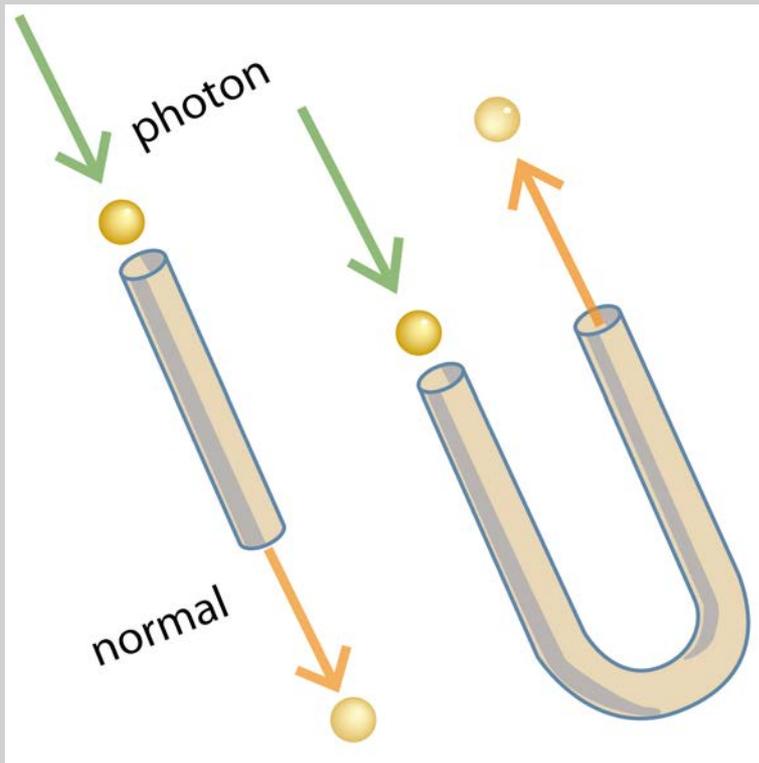
- A face
- Light
- Ray (in orange)
- And one other thing...

# Normals



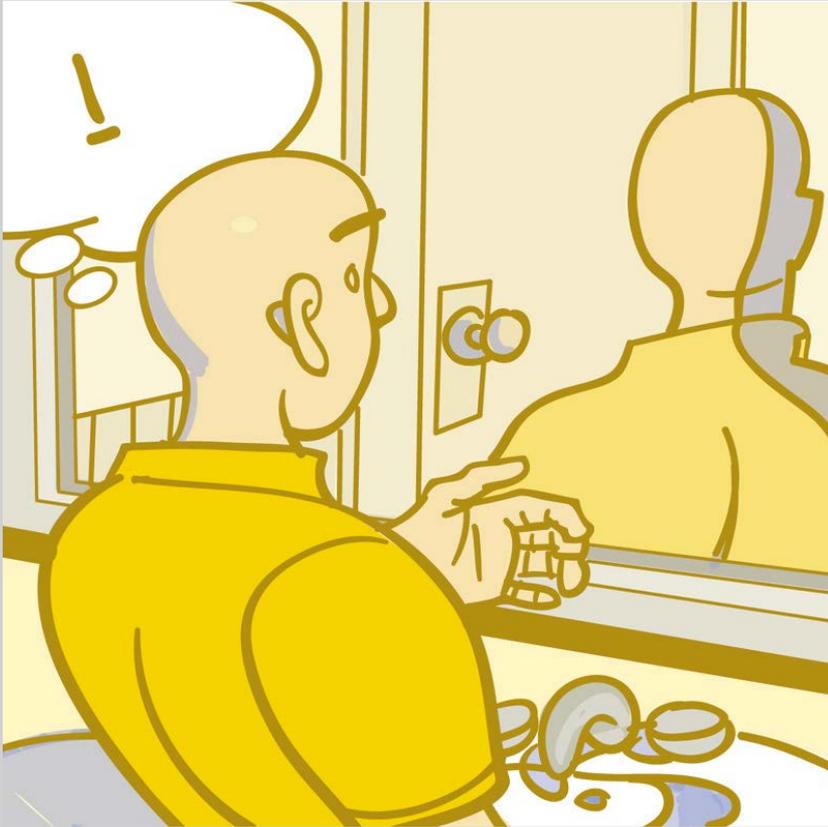
- Normals orientate geometry elements
- A normal is a vector
  - Attached to geometry element
  - Usually perpendicular to faces
  - Vertex normals borrowed from face normals
- Has strong effect on lighting

# Normal behaviour



- In CG, the intersection of a ray and a normal returns a value
- If the normal vector is wrong or backwards, it will cause render errors

# Backwards normals



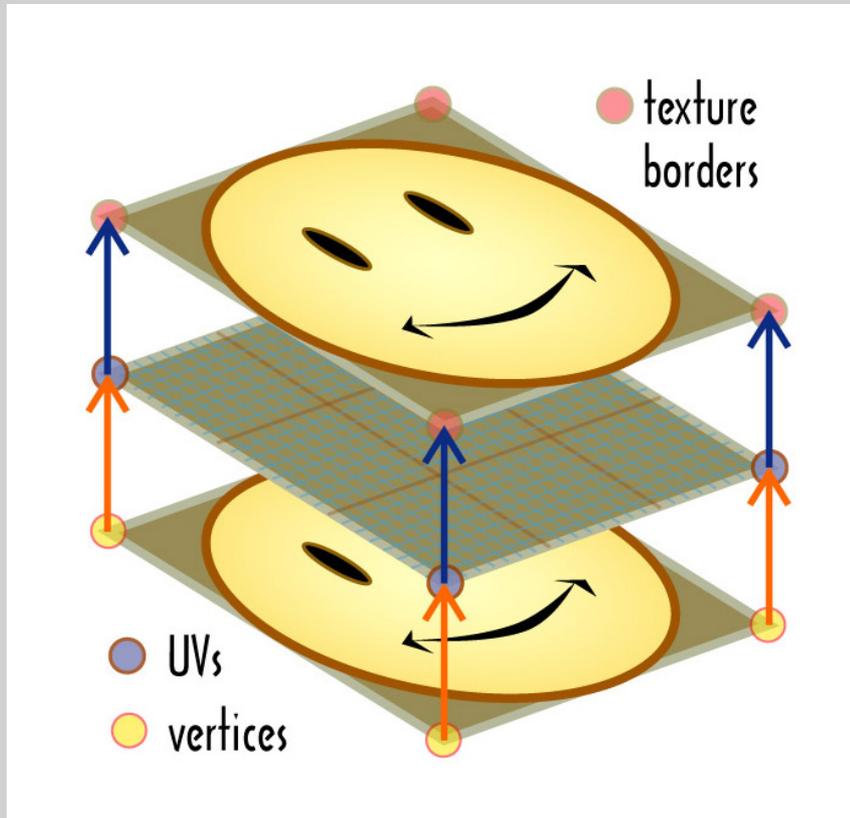
- One of the most common errors in CG
- Causes objects to render inside-out
- Work with backfacing display off to avoid

# Texture maps



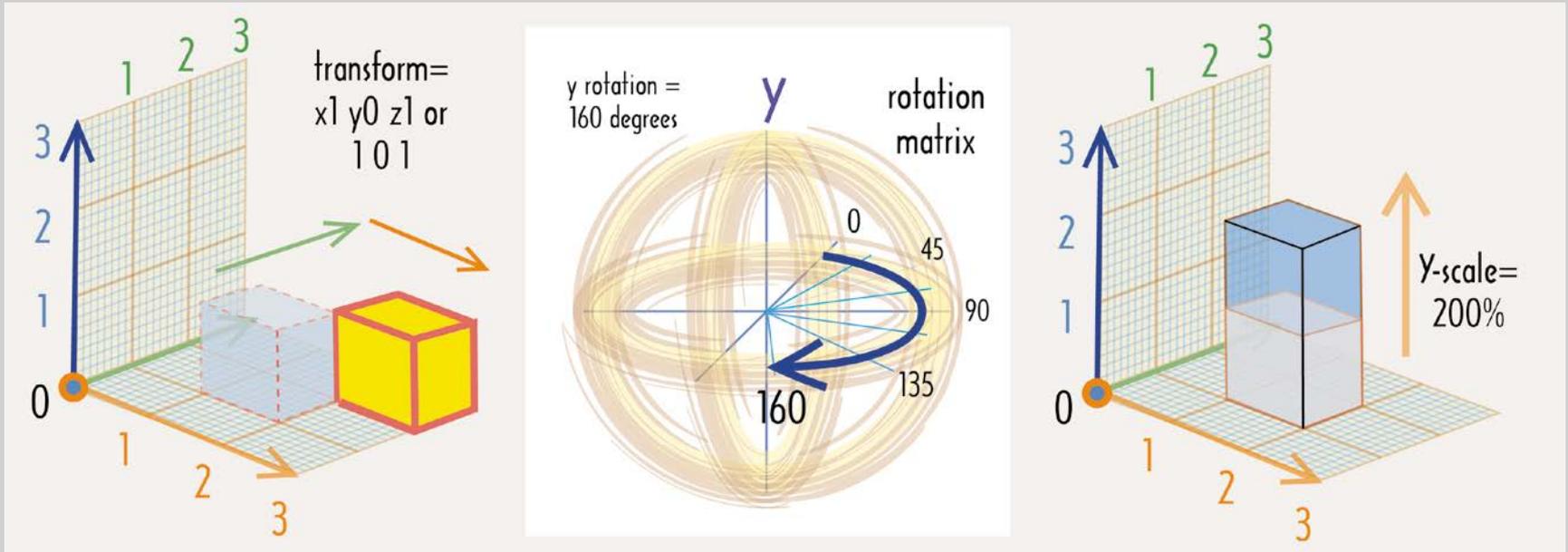
- Most objects will be textured
- To apply a texture map, you must tell the application where it belongs
- The map must contain the right information

# UVs



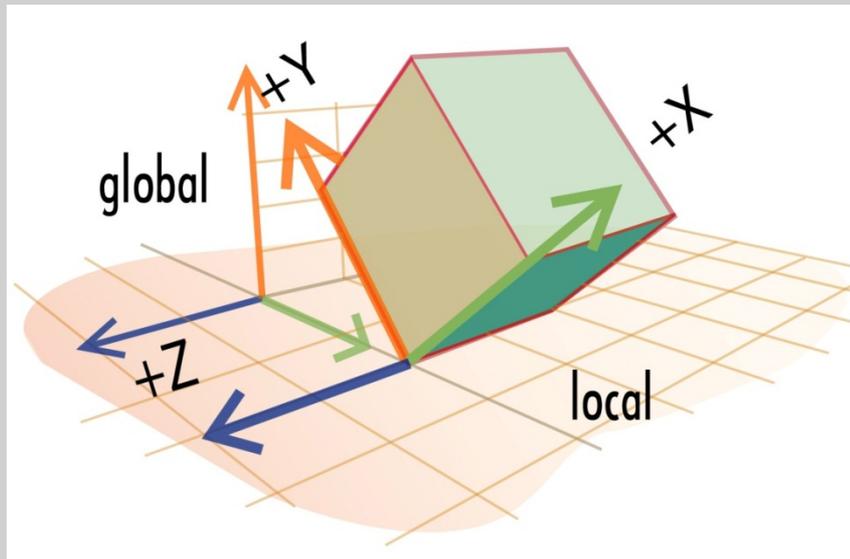
- A “UV” is a texture coordinate
- U and V correspond in object space to X and Y in an 2D grid
- Uvs are attached to vertices
- Texmap position based on Uvs
- Edited improperly, these can be a serious problem

# Move=transform



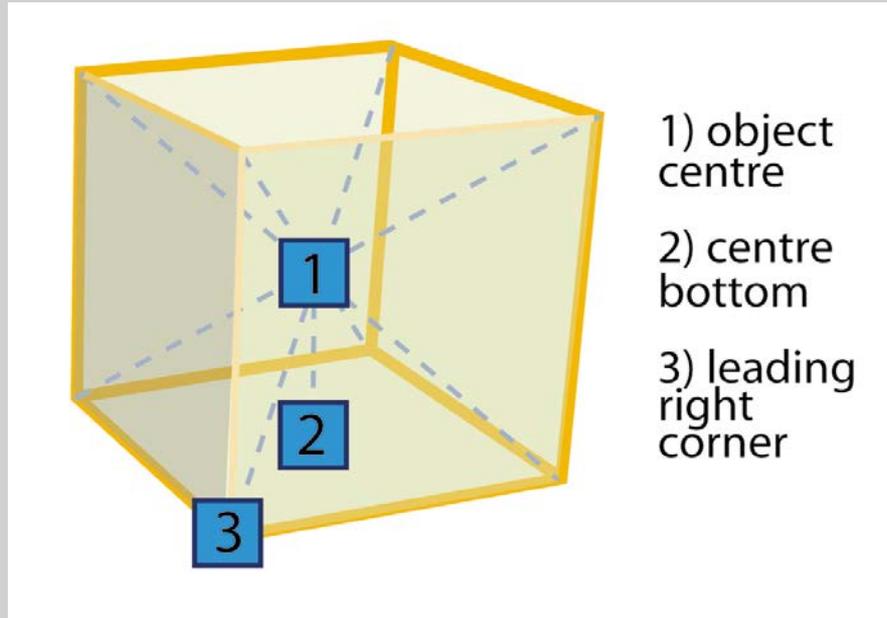
- 3 primary ways to transform an object:
  - Move: linear translation along one or more axes
  - Rotate: radial transform along one or more axes
  - Scale: projection along one or more axes
- Always zero transforms when done

# Pivots



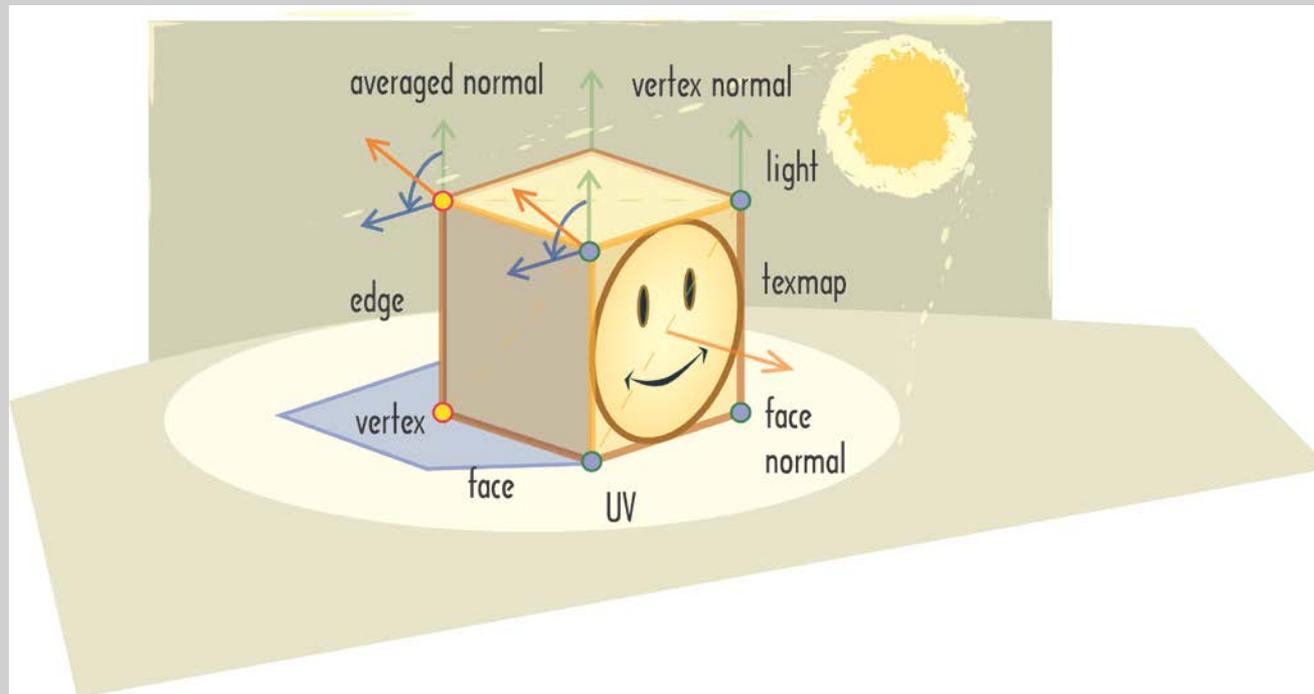
- Pivots are required for all translation operations
- Objects have local coordinates
- The pivot is part of the local coordinate system
- Pivots are frequently moved
- Pivots are essential to animation

# Pivot locations



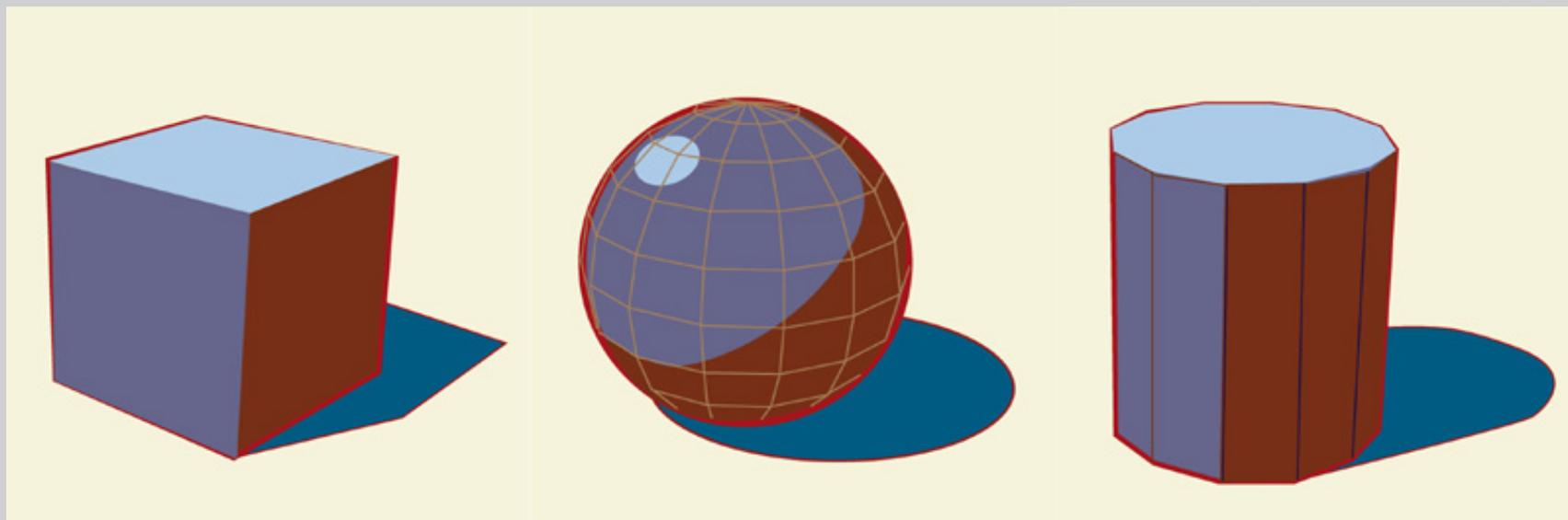
- Pivots can be anywhere
- The most common locations are illustrated here
- Always set properly before submitting file

# Basic elements of CG



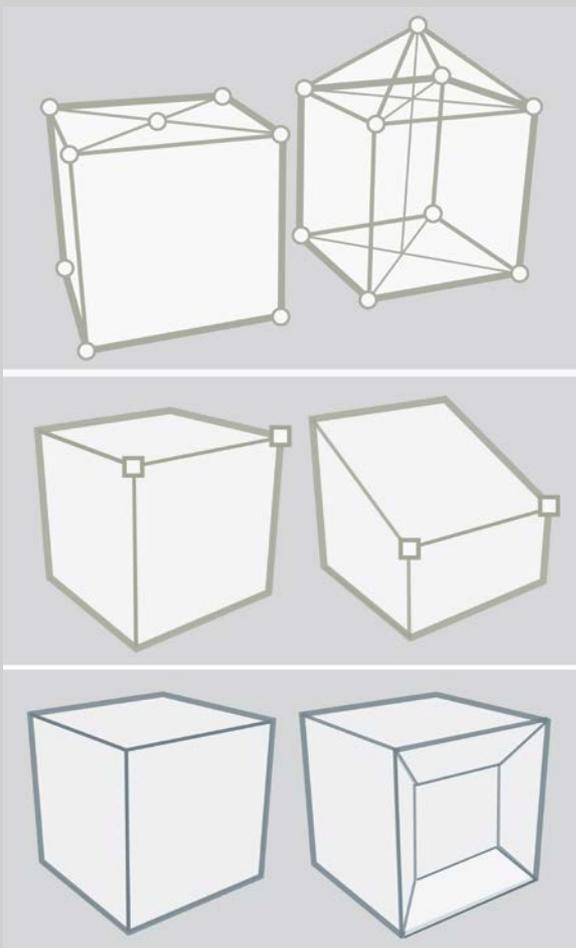
- All errors can be reduced to these 7 elements:
  - Normals, rays, UVs, faces, edges, vertices, texmaps

# Building blocks



- Pre-defined shapes are rarely exactly what you need
  - Even if you want a cube, cylinder, or sphere
- These are used to start a model

# Component editing



- Move, rotate, or scale
- Most modeling operations are component editing
- This is where most mistakes will be made
- Without care, the way components are edited will violate professional standards
- Reluctance to edit components causes errors

# Clean geometry

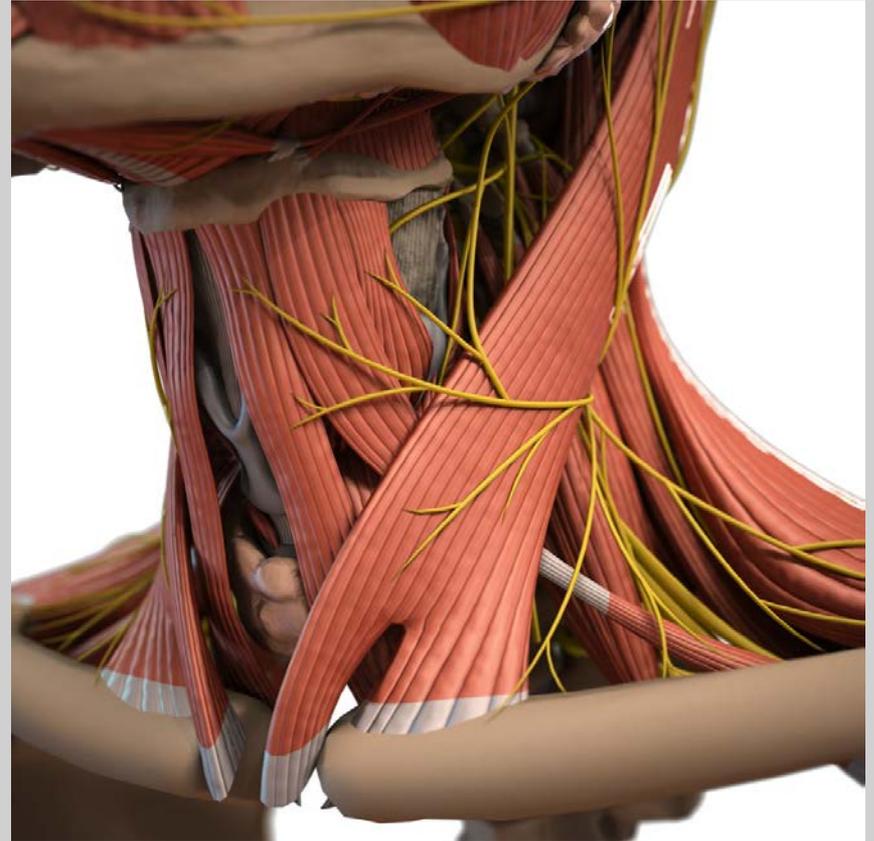


# Clean at every stage

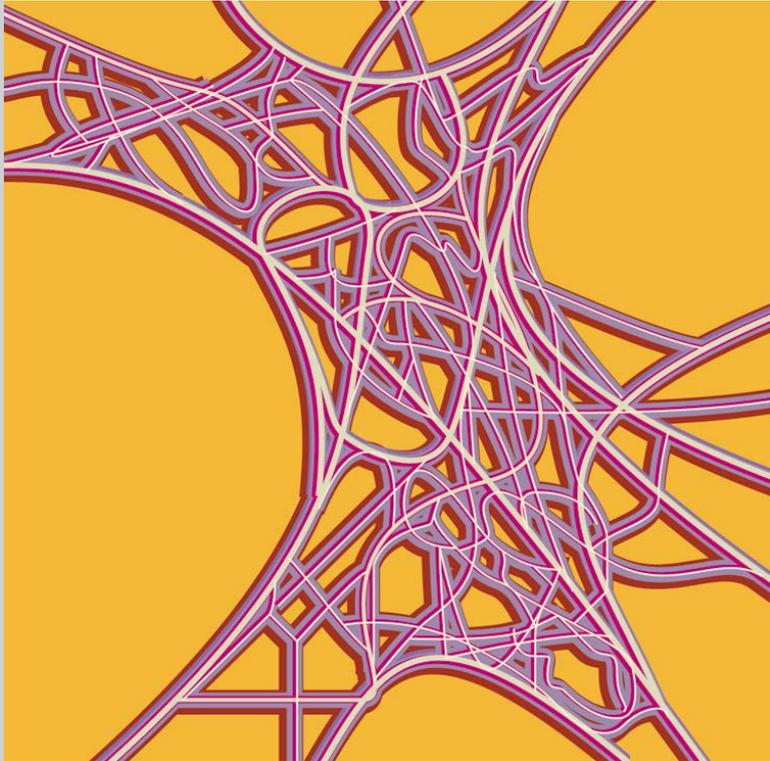


# Errors

- “Clean geometry” is error-free
- Clean geometry is expected of any professional artist
- If you cannot recognize and fix errors, your work may not be fit for use



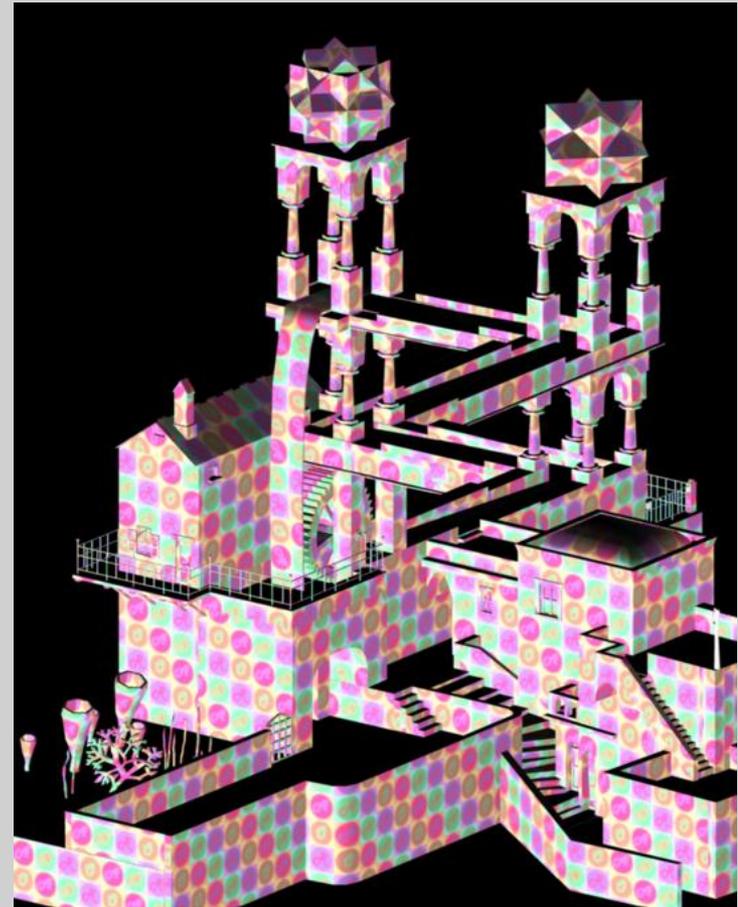
# “Clean” isn’t “dirt-free”



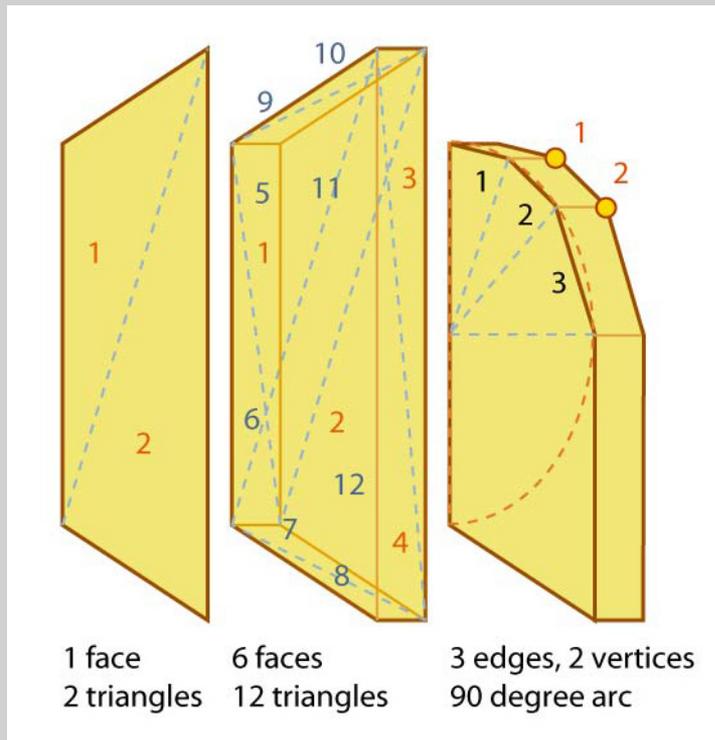
- If you are building dirt, it should be built cleanly
- Even a cobweb can be clean
- “Clean” is error-free, efficient, and well-organized

# Classifications

- Likeness errors
  - Measurement and estimation
- Technical errors
  - Corrupted geometry
- Optimization errors
  - Excessive or inconsistent resolution
- Construction errors
  - Left-over geometry
- Organizational errors
  - Poor parenting, naming, and grouping

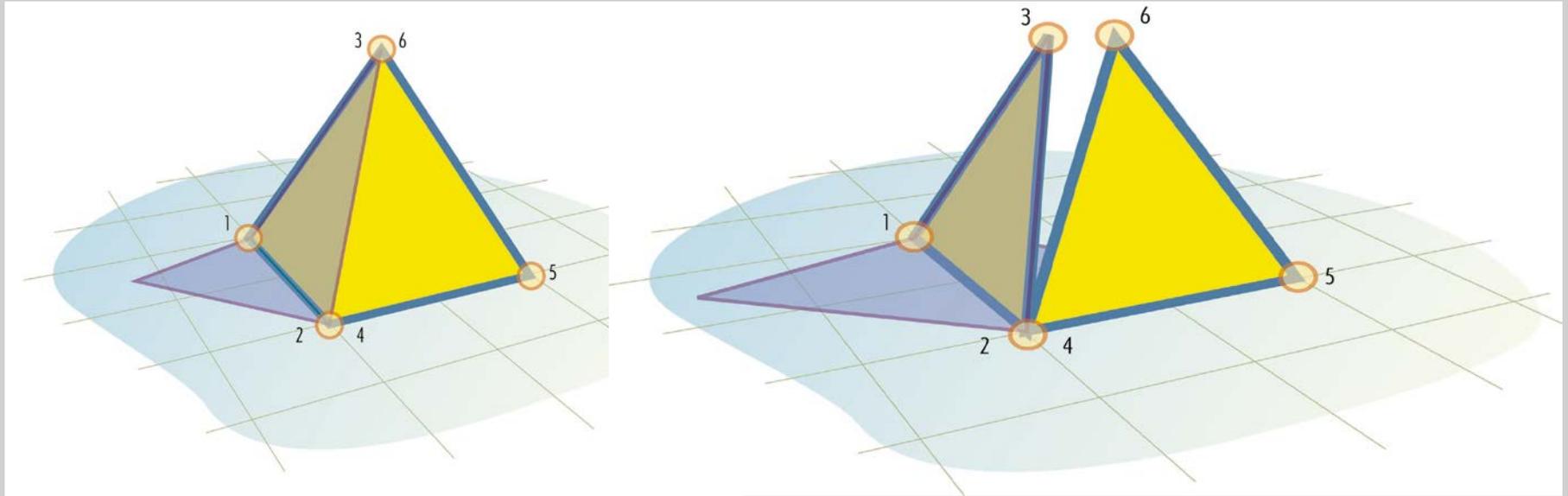


# Efficiency



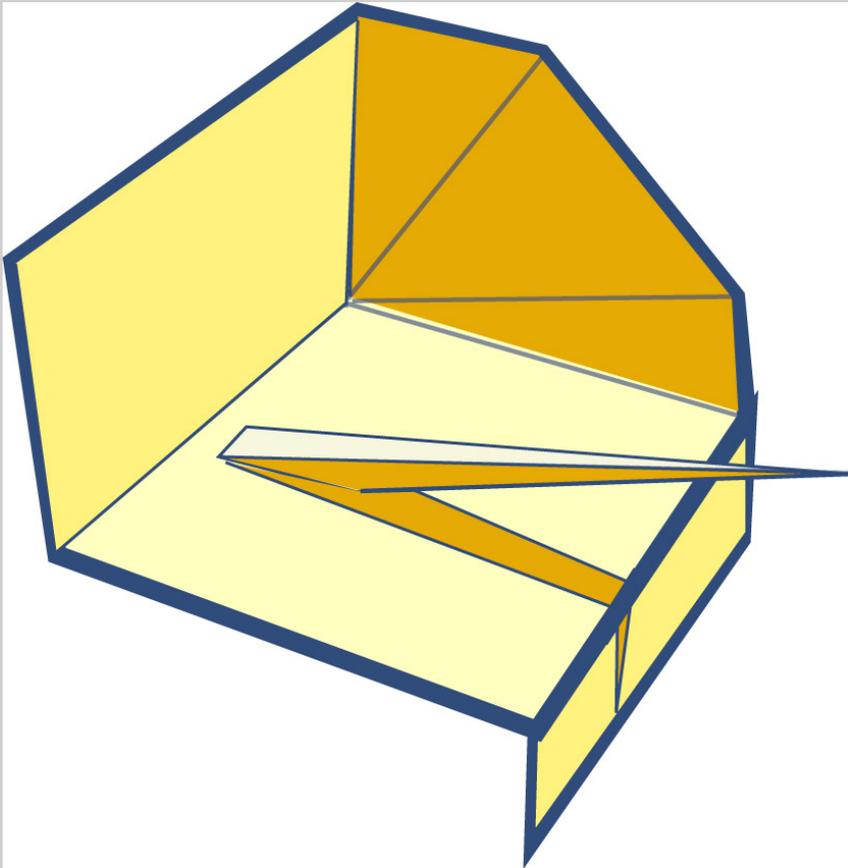
- Minimum number of triangles required for:
  - Shape, texture, lighting, and animation
- Consistent number of segments relative to arc-length
- Vertex spacing along curves consistent for distance and curvature

# Coincident vertices



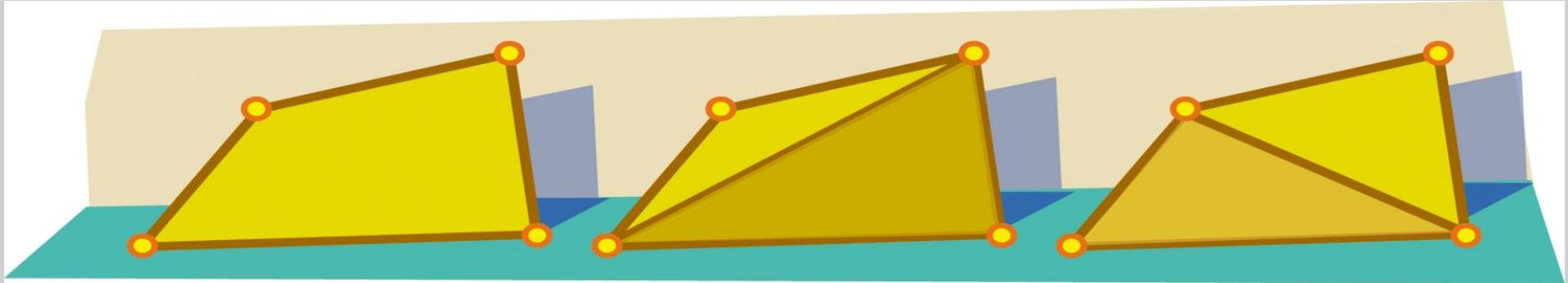
- Allowed between objects
- Not allowed within an object\*
  - \*With exception of special animation requirements

# Spikes



- Construction error
- Caused by careless selection and transformation
- Will not be found by most automatic repair tools
- Solution is to manually delete and rebuild

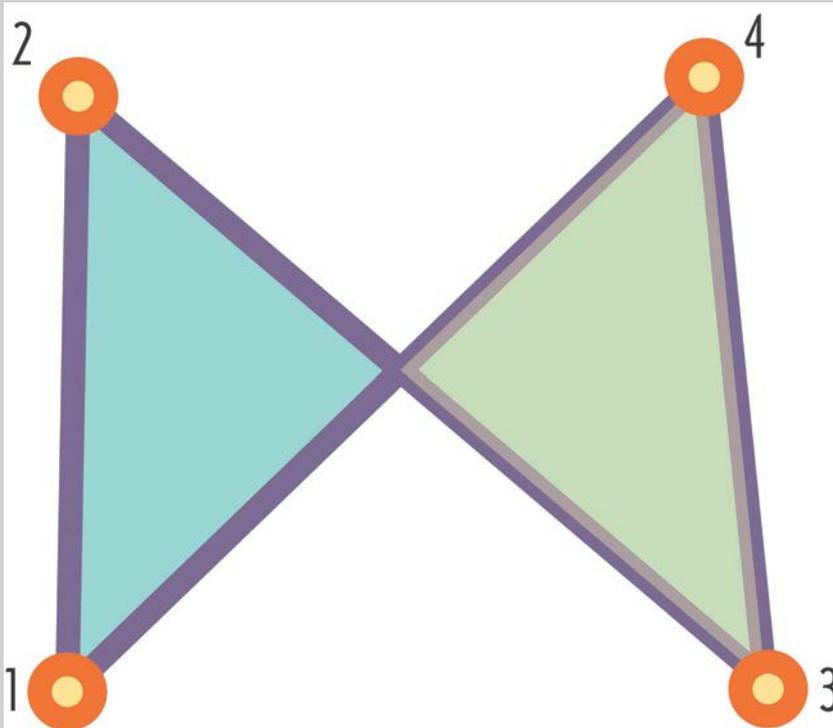
# Non-planar faces



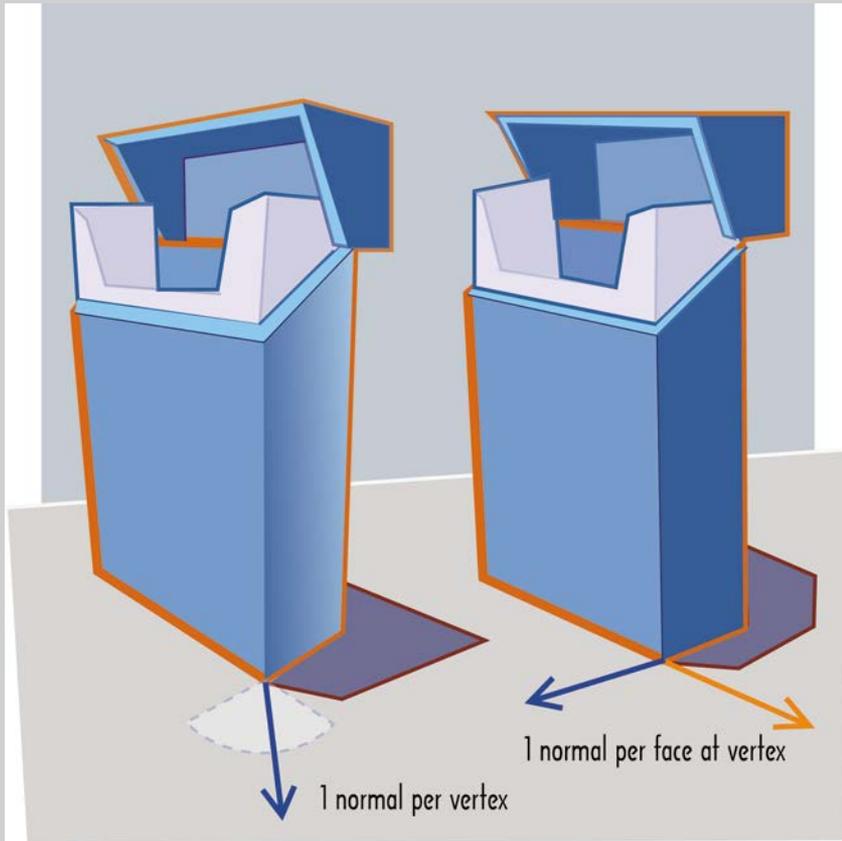
- Non-planar faces can be triangulated in more than one way
- Depending on how severe the difference is, this may be unacceptable
- If part of an animated soft body, smoothing across the edge may not be enough to hide this error

# Bow-tie faces

- Non-planar face
- Caused when any vertex in a 4-sided polygon crosses from one side of a vertex pair to the other
- Can be hidden in small crevices
- To find, highlight non-planar

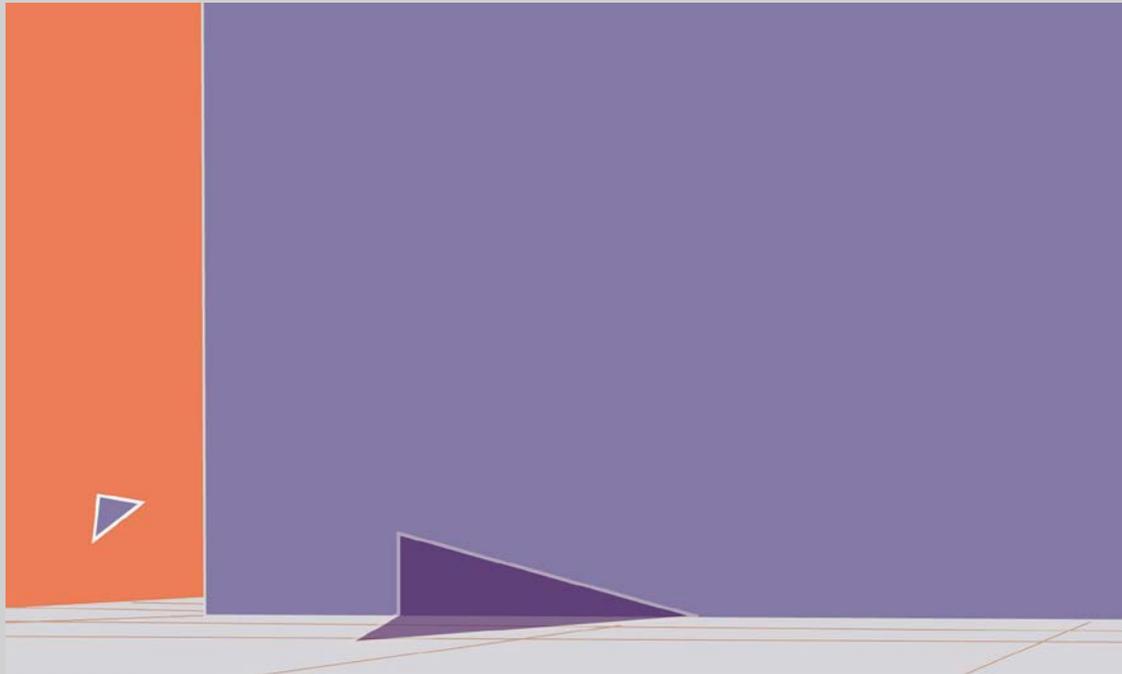


# Smoothing normals



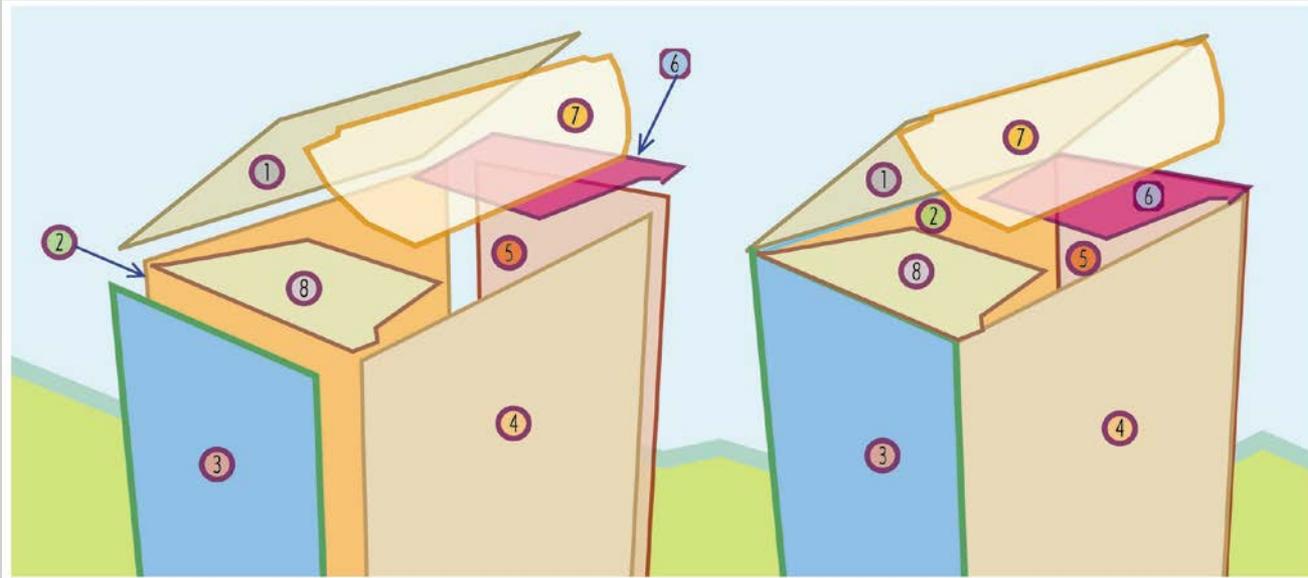
- Object normals affect lighting
- If incorrect, smoothing will be incorrect also
- Normal maps will cause renderer to ignore
- However, it is better to make the normals correctly prior to mapping

# Floating faces



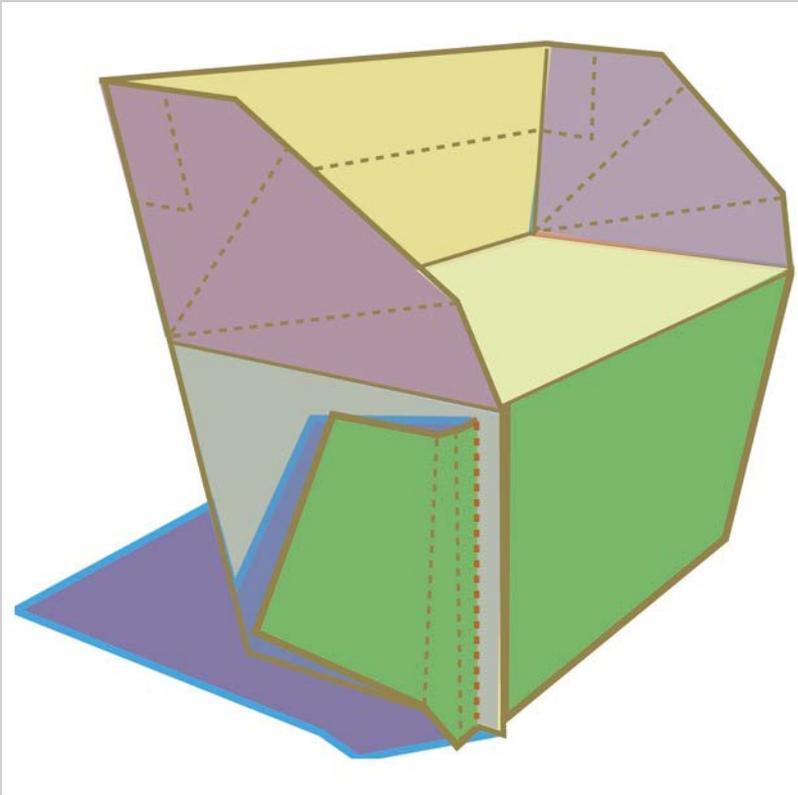
- Construction error
- Left-over result of boolean or other editing
- Usually very small, not visibly attached to polyset

# Separated faces



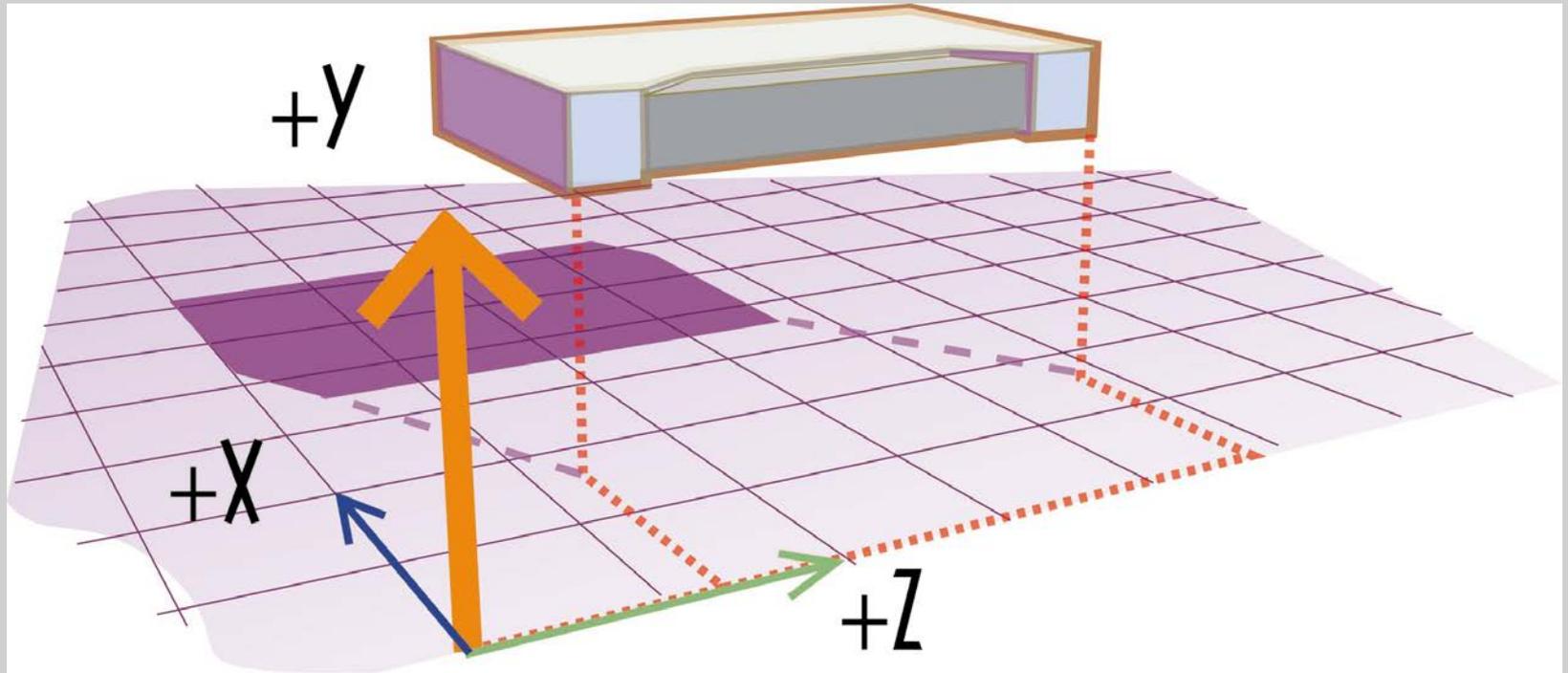
- Construction error
  - Occurs when parts are built separately or:
  - Faces are accidentally separated during construction

# Misaligned faces



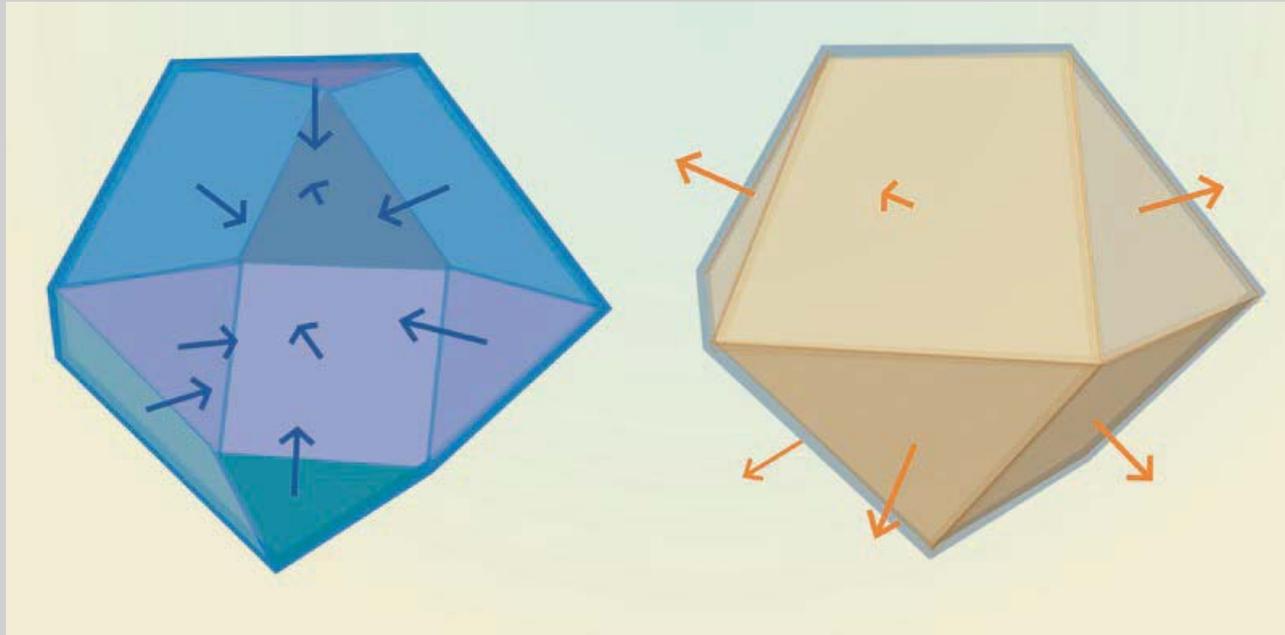
- Construction error
- Fix by translation of vertices or faces affected

# Origin offset



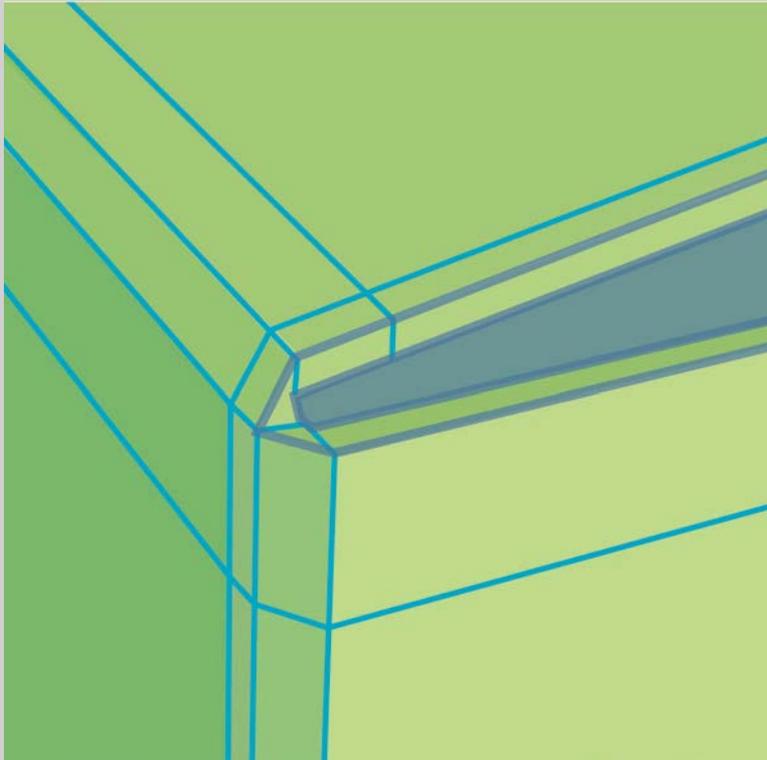
- The default position, orientation, and pivot location for objects must be correct for all objects in a scene

# Backwards normals



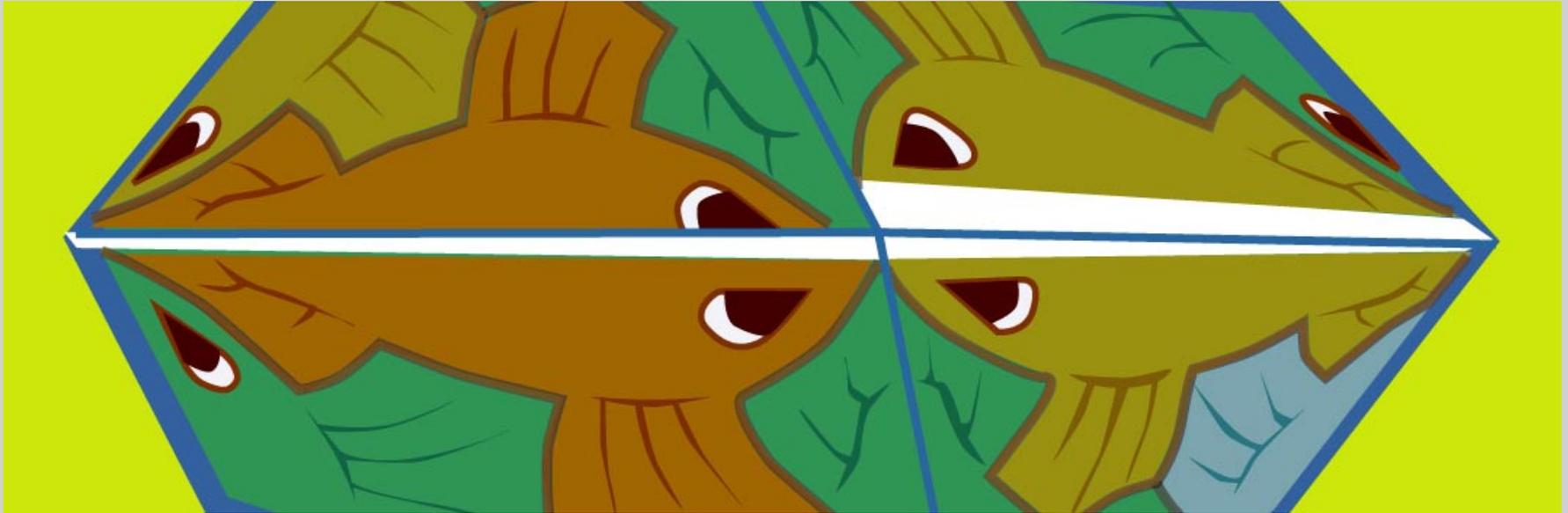
- When normals are backwards, renders will be unacceptable
- 2-sided faces will hide this, but significantly increase render time

# High poly count



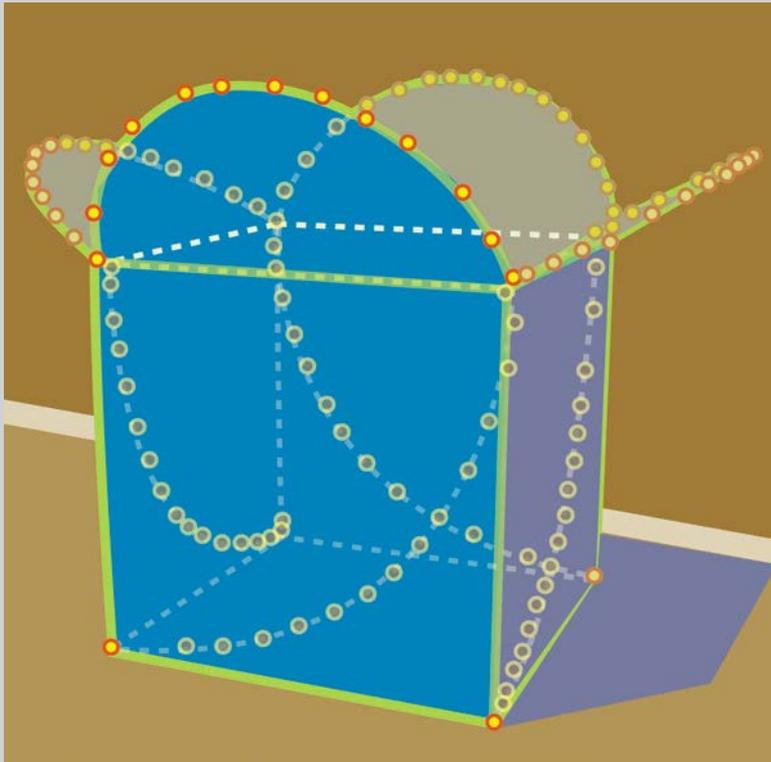
- Some tools radically increase the vertex count:
  - Edge loops\*
    - Box modeling
  - Booleans
  - Automatic operations performed on the full model
- These must be used carefully

# Misaligned texture



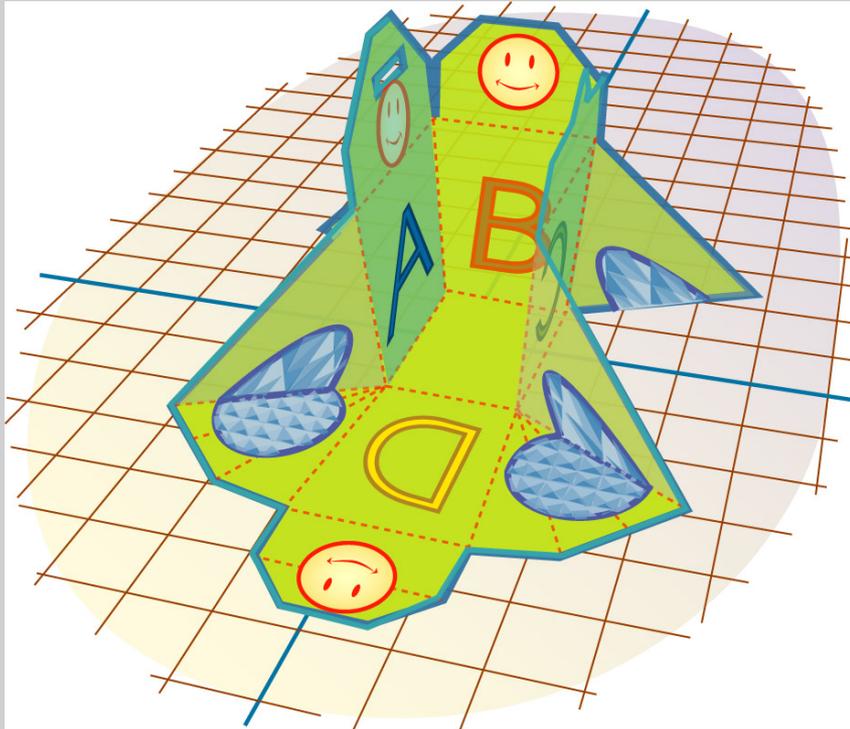
- This mistake is caused by an incorrect crop of the texture map combined with improper UVs
- It can be fixed by modifying the map, the UVs, or both

# Isolated vertices



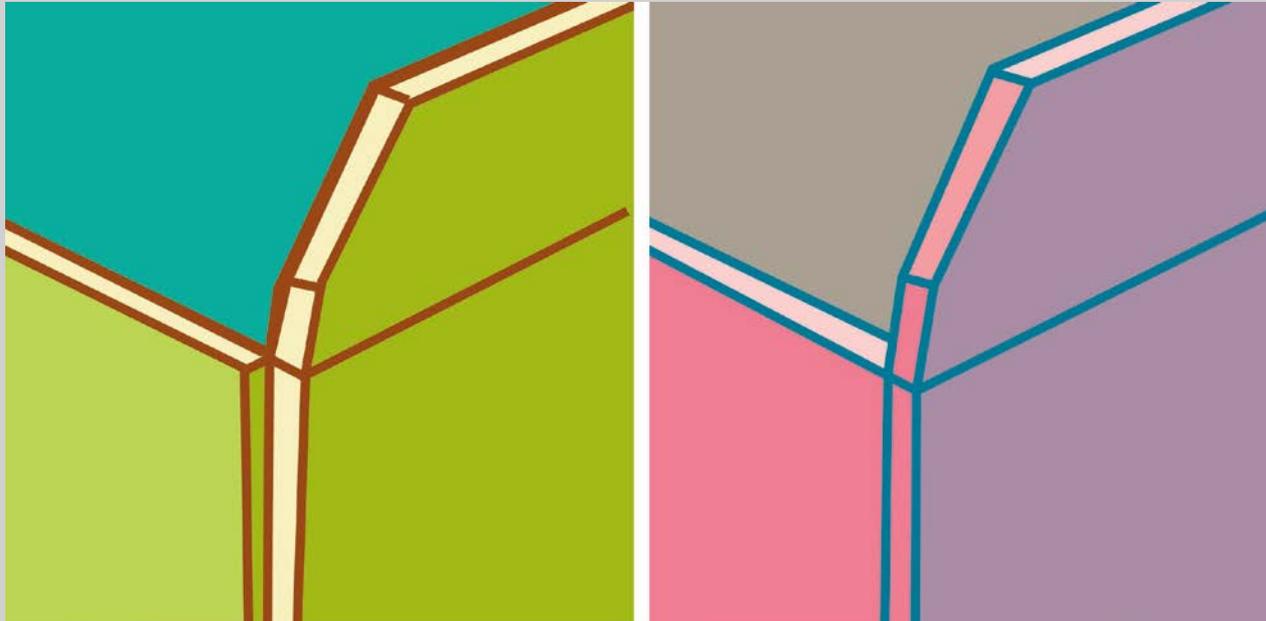
- An “isolated” vertex has only 2 edge connections
- This is not always illegal
  - All vertices in a four-sided polygon are isolated
- If it results in a polygon with more than 4 sides, it is a problem
- The face must be divided

# Distorted geometry



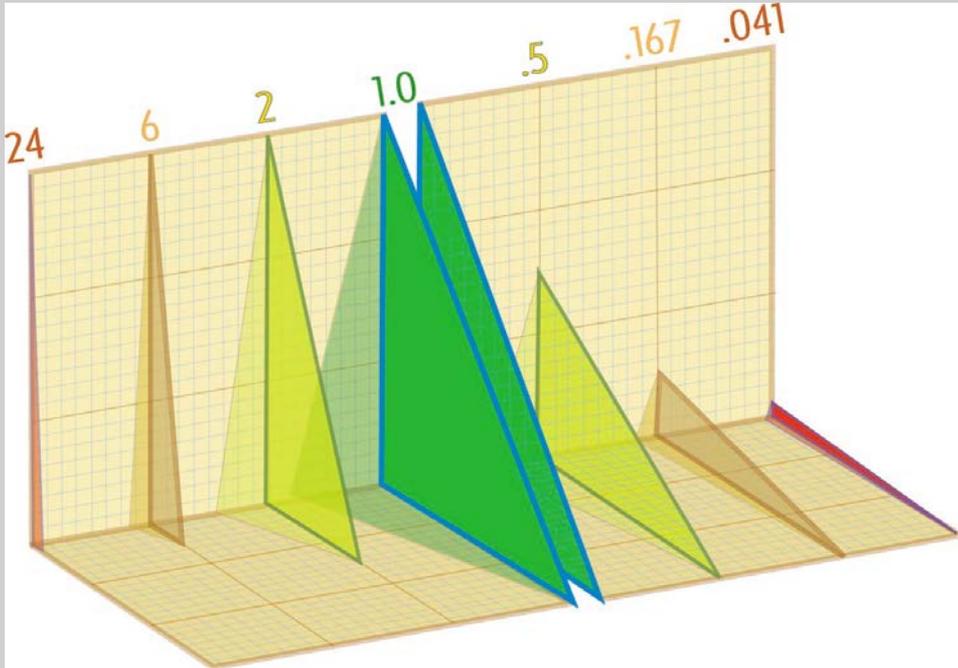
- If vertices are shared among more than one adjacent face, transformations cause shear
- To fix, cut along seams prior to transforming the section or use an animation rig

# Extrude error



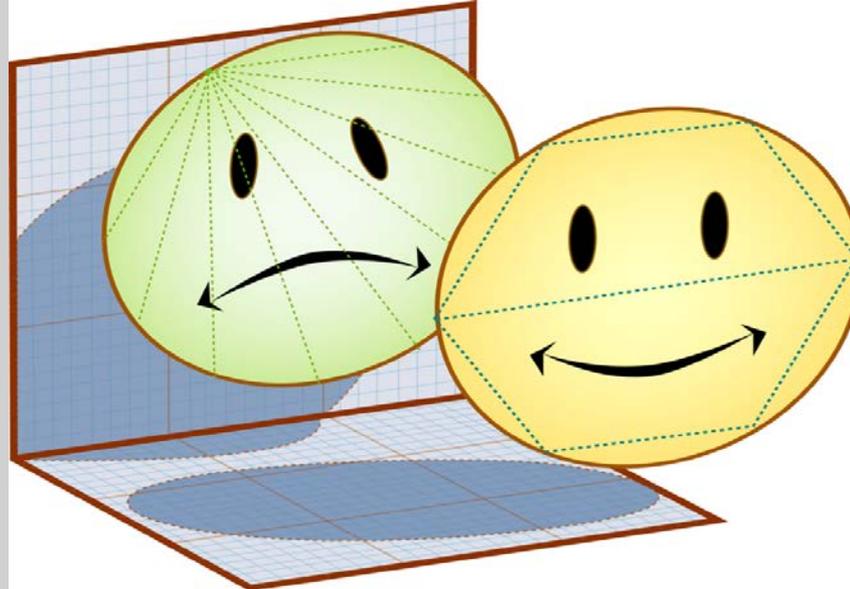
- Extruding faces can result in unwanted gaps, as shown here
- Check your tool's settings or hand edit to ensure this does not happen

# Aspect ratio



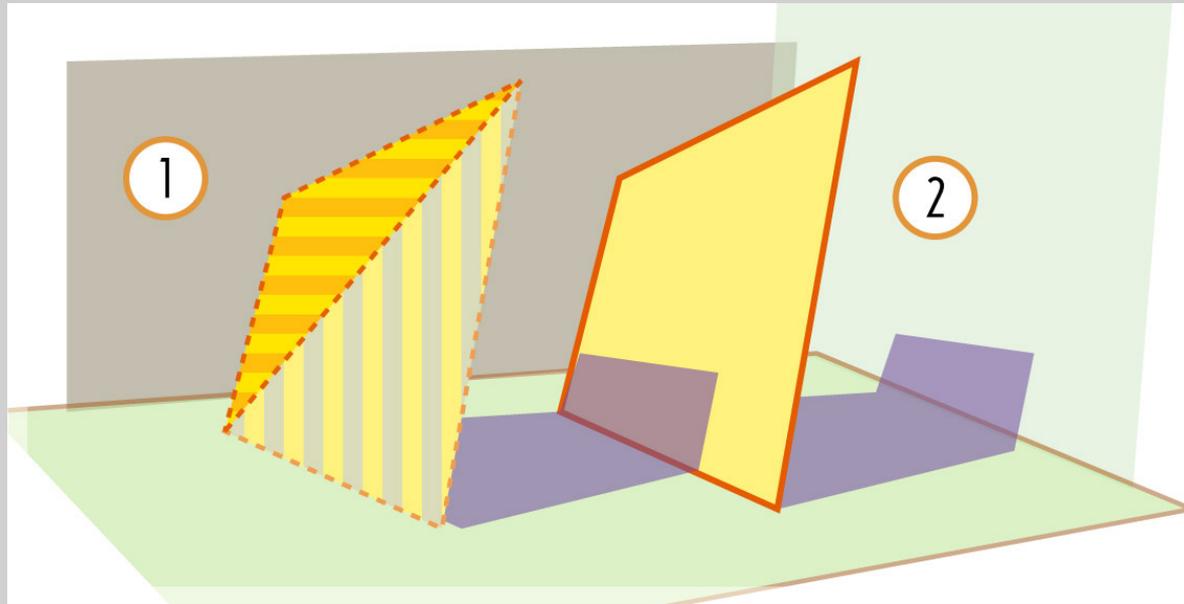
- 1.0 aspect ratio (AR) is ideal for rendering
- The greater the distortion, the more likely a rendering or animation problem will occur
- The optimal range is between 6.0 and its inverse .167

# Triangulation

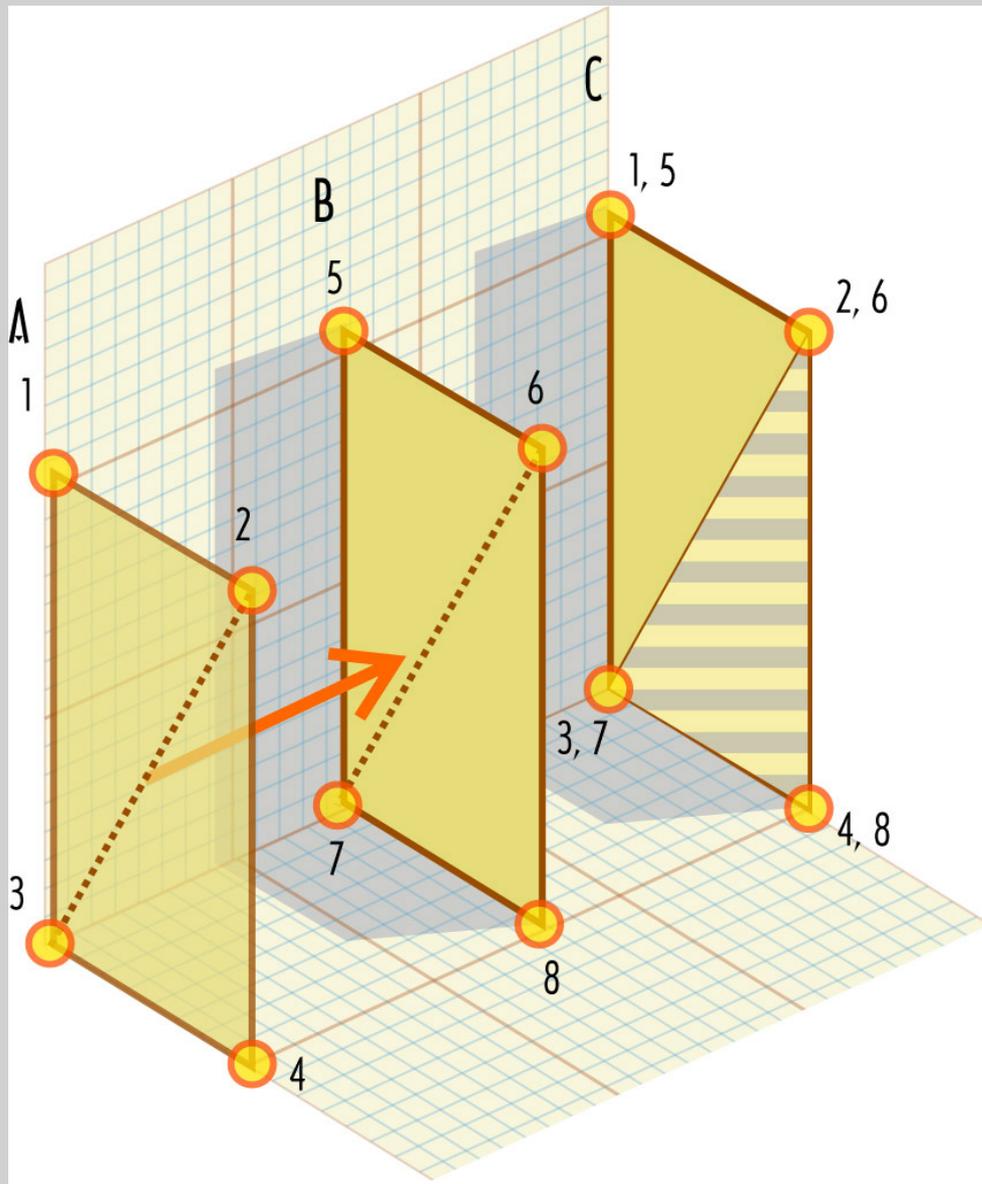


- Triangulation pattern affects rendering
- Symmetrical, shortest edge length patterns produce even lighting in vertex-based rendering

# Lamina faces



- Share the same vertices.
- These are illegal and can crash a render
- They can be found by tumbling the camera and looking for flickering faces or counting vertices

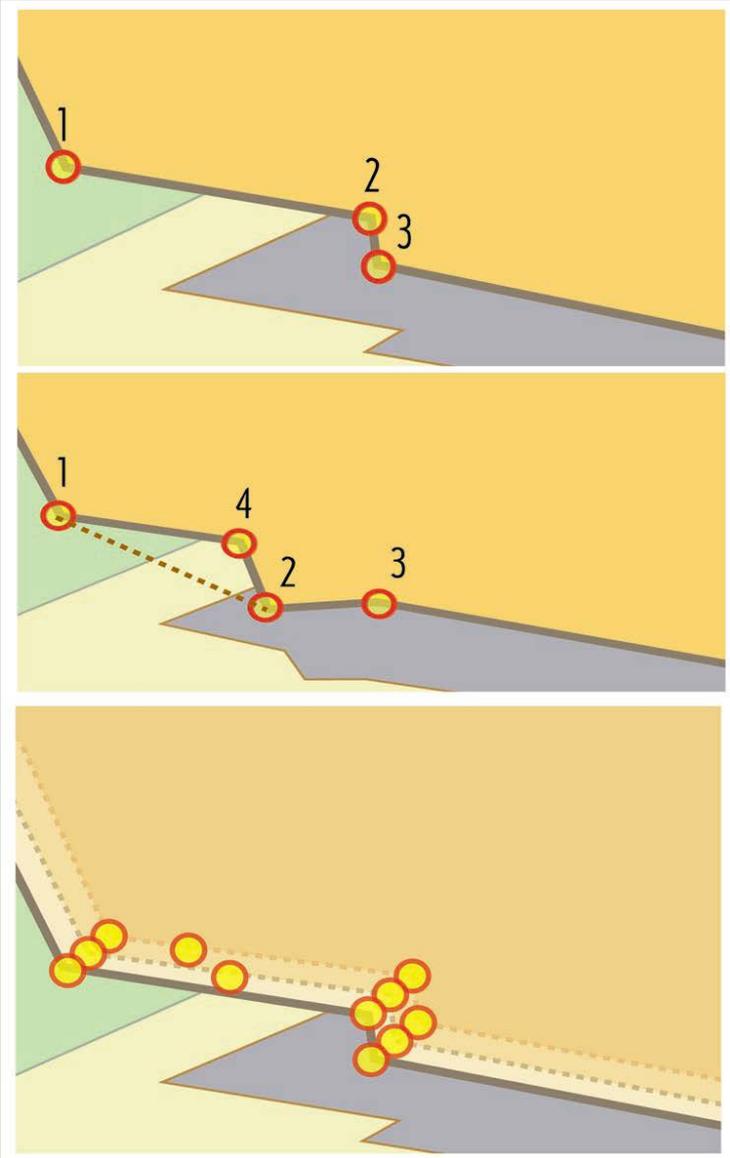


# Coincident faces

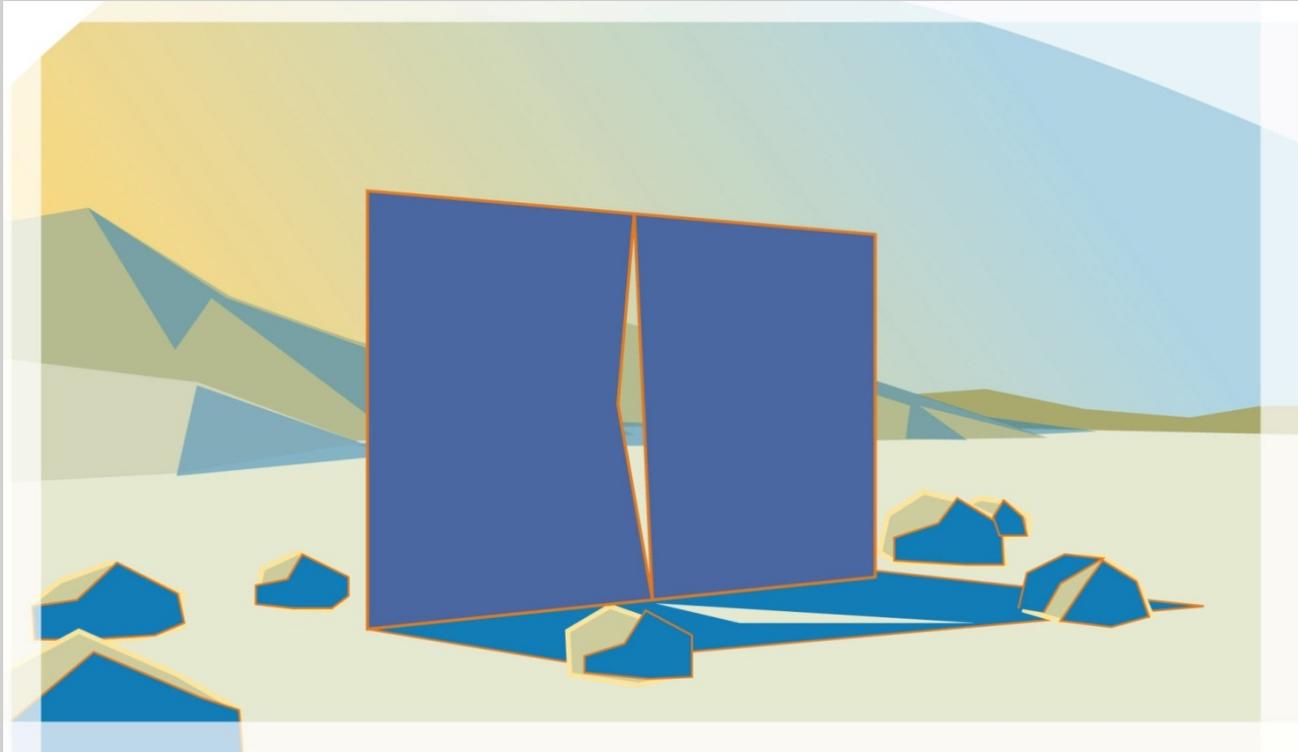
- Unlike lamina faces, coincident faces do not share vertices
- Coincident faces are not illegal and will not crash a render
- However, they will render incorrectly

# Duplicate edges

- Hide non-planar faces and isolated vertices
- Can be found by transforming individual vertices
- To fix, either delete or merge excess vertices

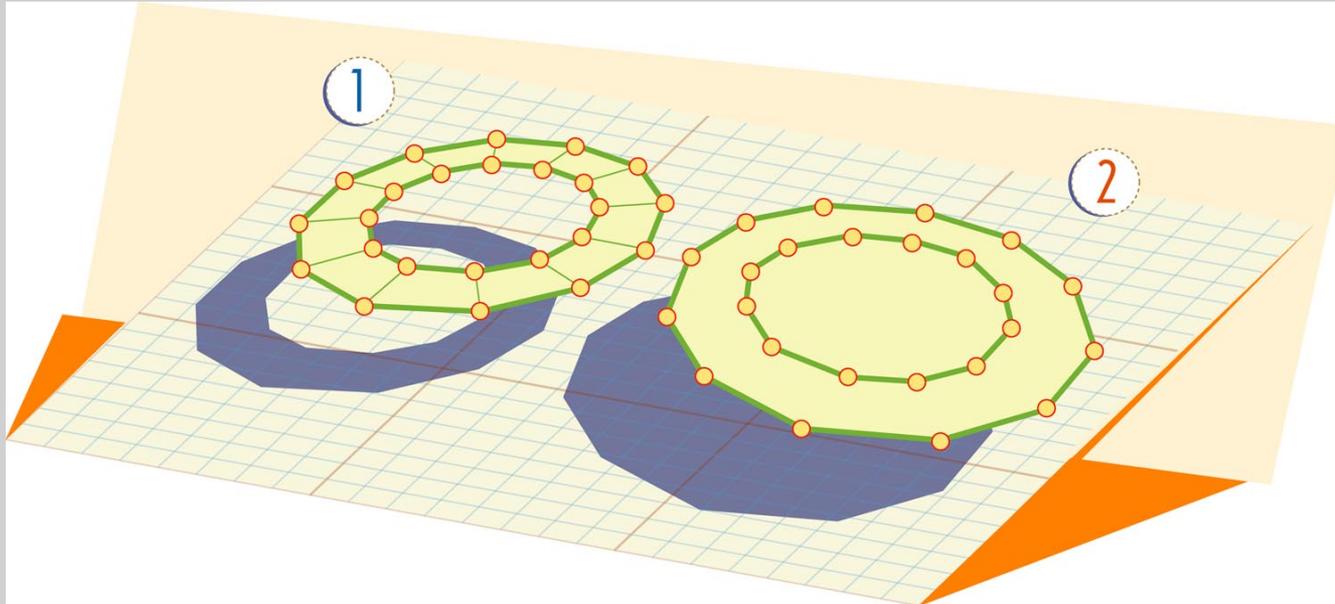


# Gaps



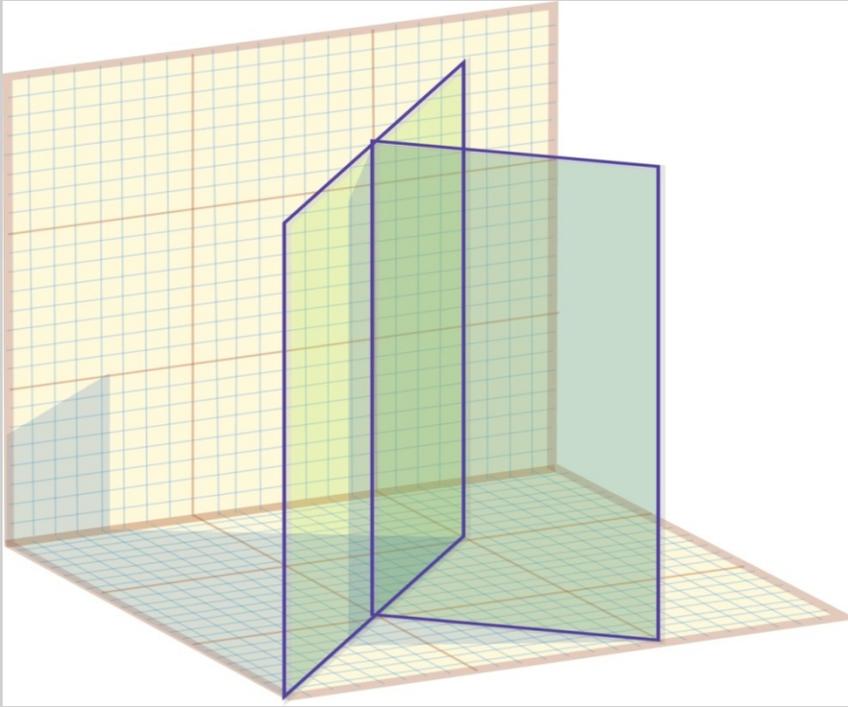
- Construction errors
- Gap should be filled in or surrounding faces adjusted

# Hole geometry



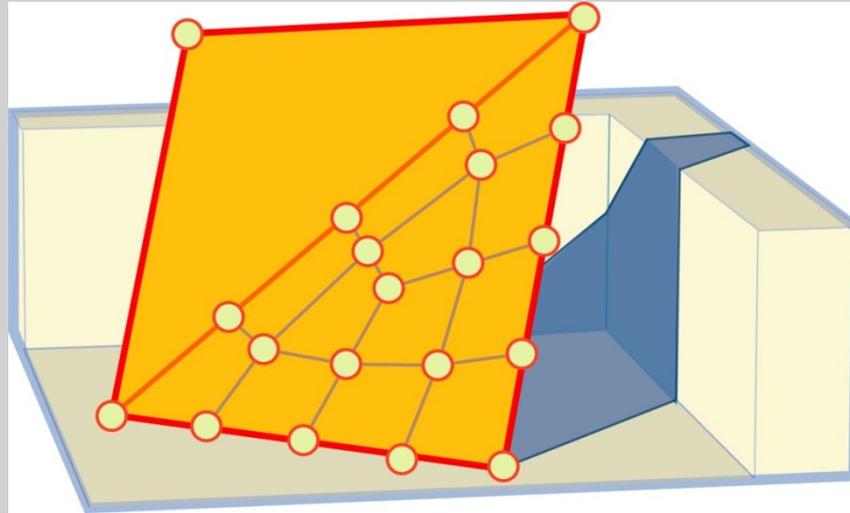
- A closed row of vertices within another closed row of vertices, without edge connections between them.
- This is illegal and must be fixed when it occurs

# Non-manifold geometry



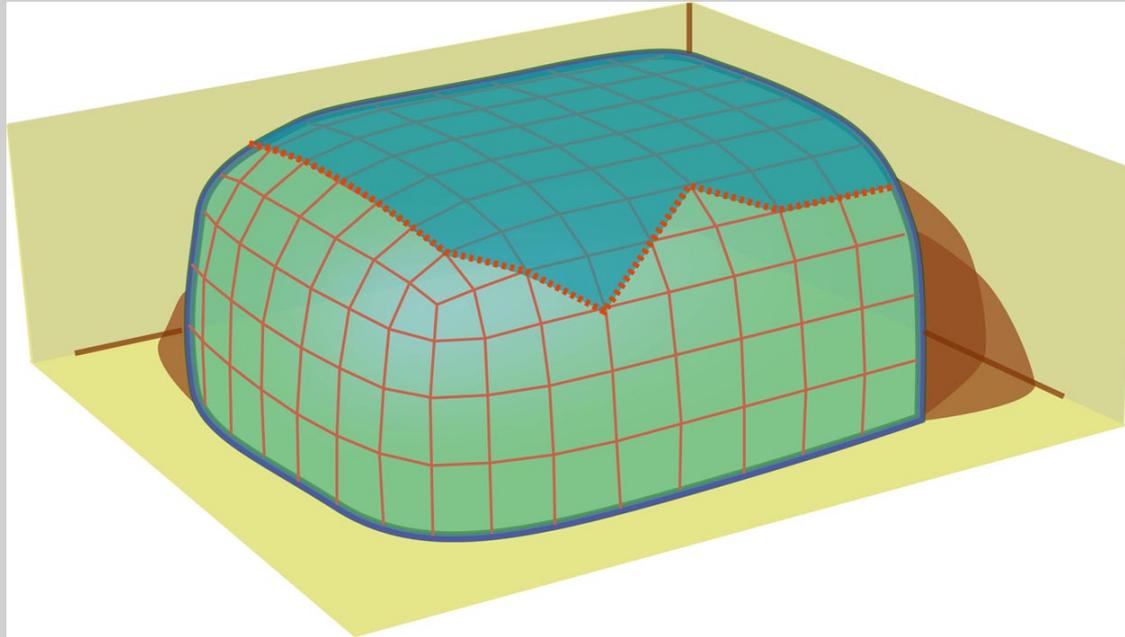
- Cannot be unfolded flat without seams or overlap
- Legal, but can cause problems during modeling and texturing

# Dense mesh



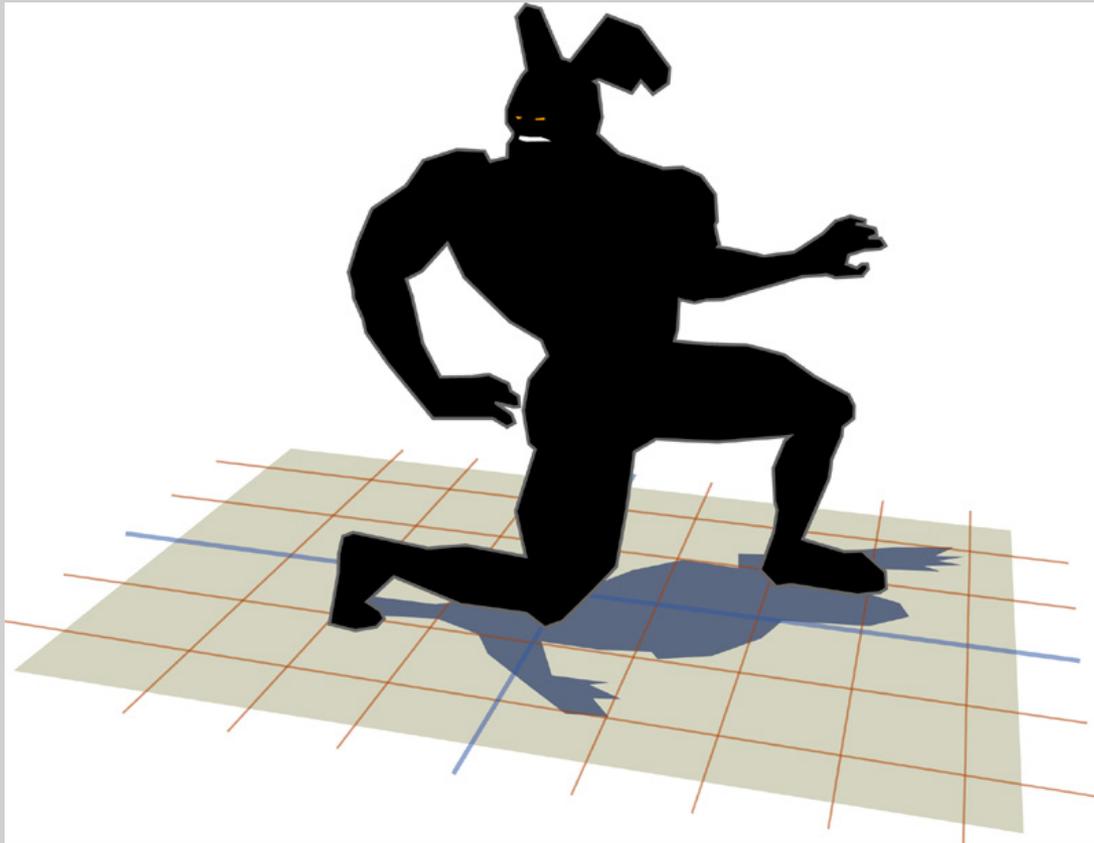
- Every vertex in your model should have a purpose
- Legitimate purposes are:
  - Define surface structure
  - Texture boundary
  - Even vertex distribution for lighting

# Ragged texture boundaries



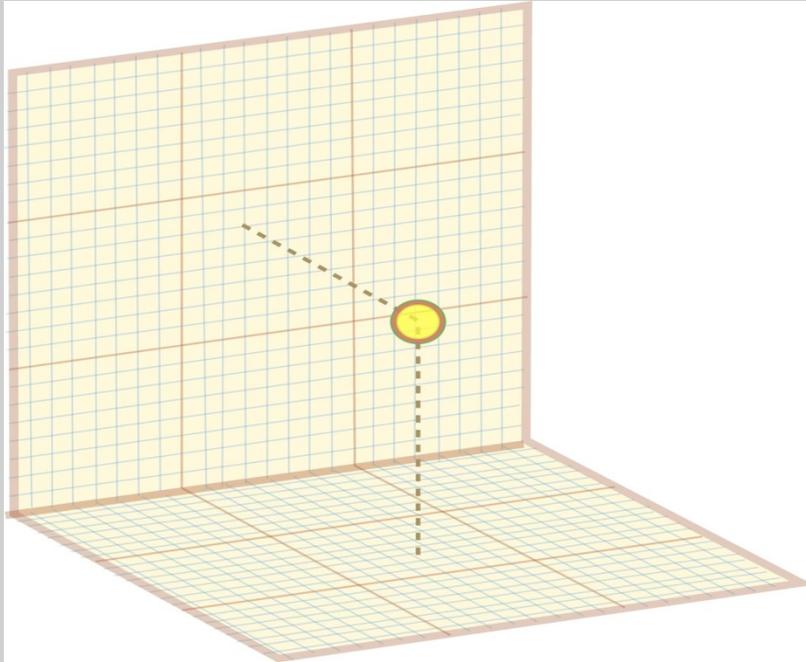
- During de-rez operation, it is easy to lose texture boundary definitions
- Boundaries should follow contours of your model

# Locked normals



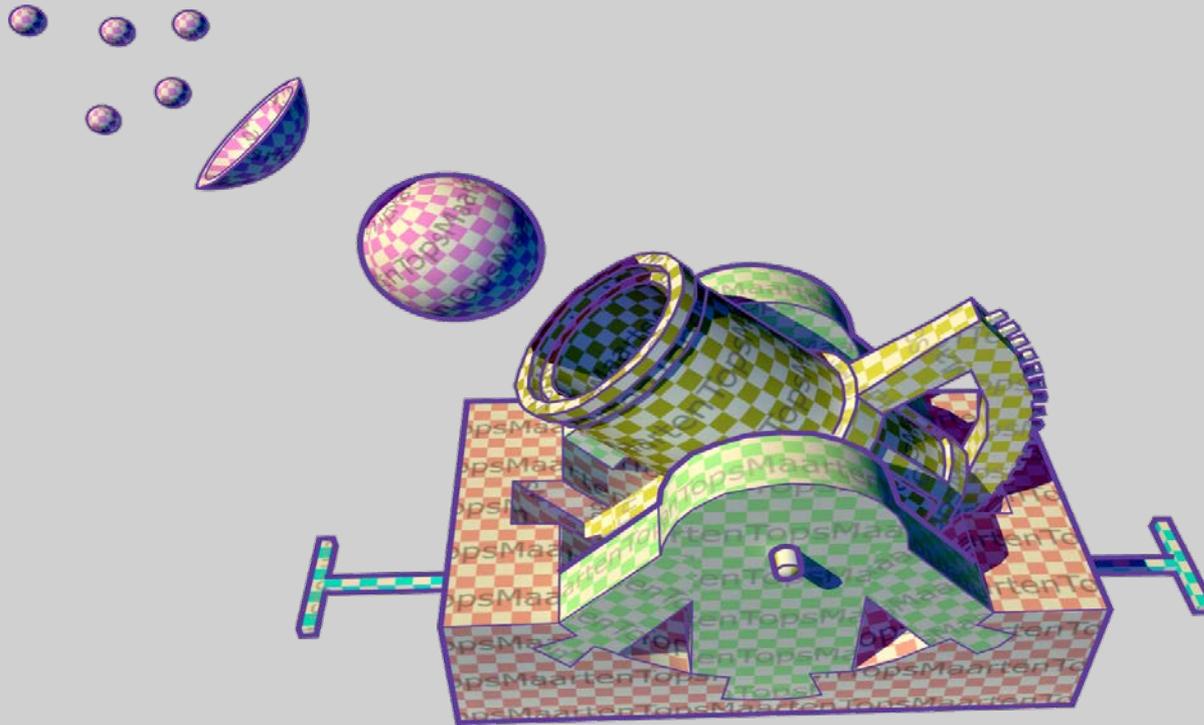
- If lights only barely affect your model, you may have locked normals
- Happens most often during import
- Recalculate normals to fix
- Test model before delivery

# Zero edge length faces



- A face with no dimension
- May cause crash at render
- Deletion may require selection of all neighboring faces

# Texture coordinates

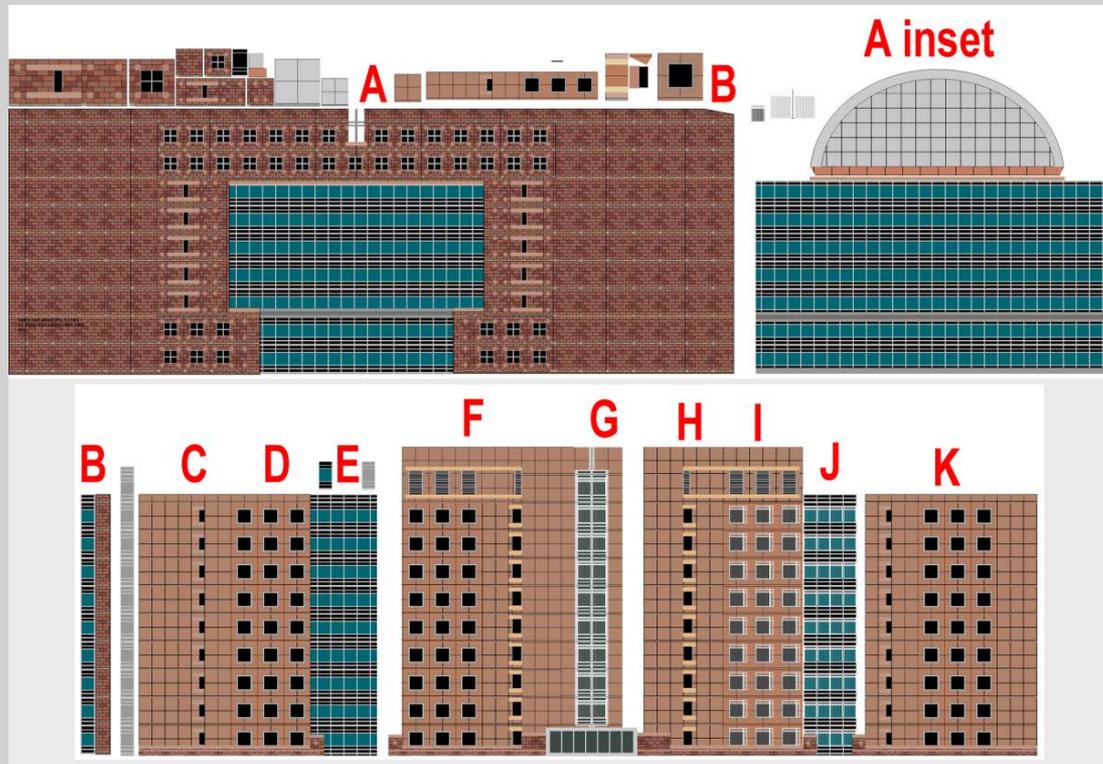


- Three goals: seamless, undistorted, efficient

# UV editing requires balance

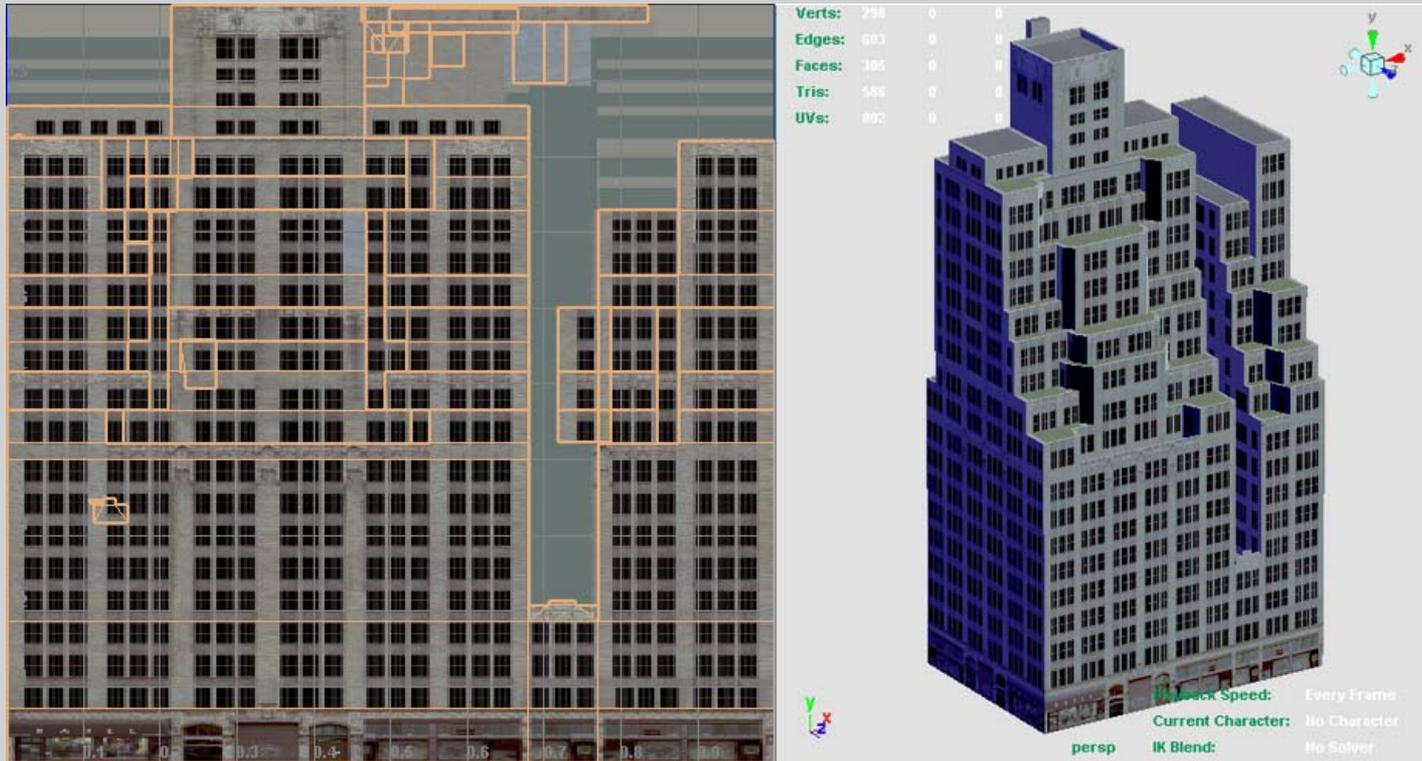
- Weak UVs can be the most expensive mistake made by a modeler
- Because poor UVs increase texturing time dramatically
- Good quality UVs can increase the amount of vertices required for texture boundaries
- 3 goals of UV editing are mutually exclusive
  - No seams
  - No distortion
  - Efficient

# 3D model projected into 2D space



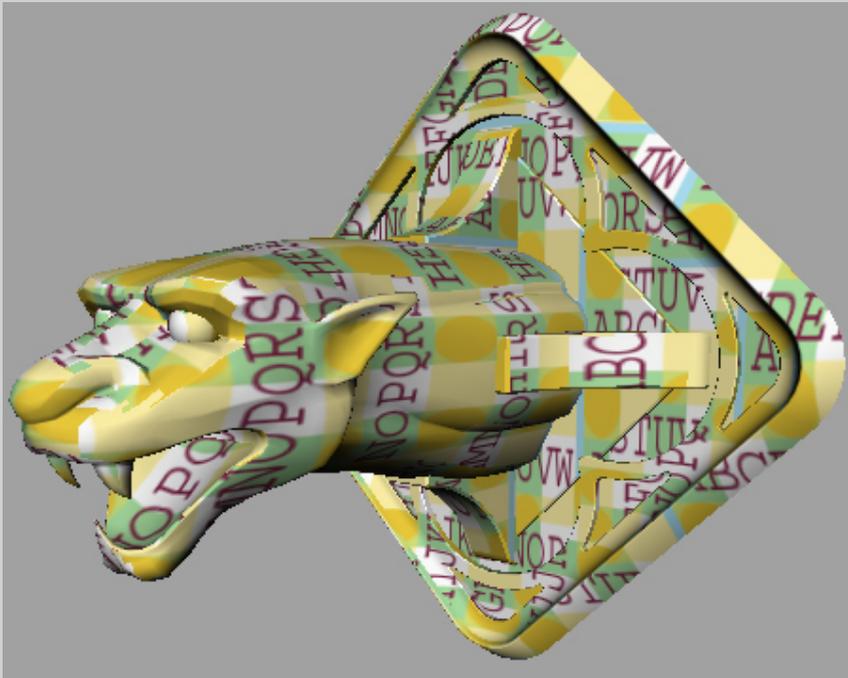
- The UV layout should be considered during the modeling process because it will affect the result

# Non-manifold objects



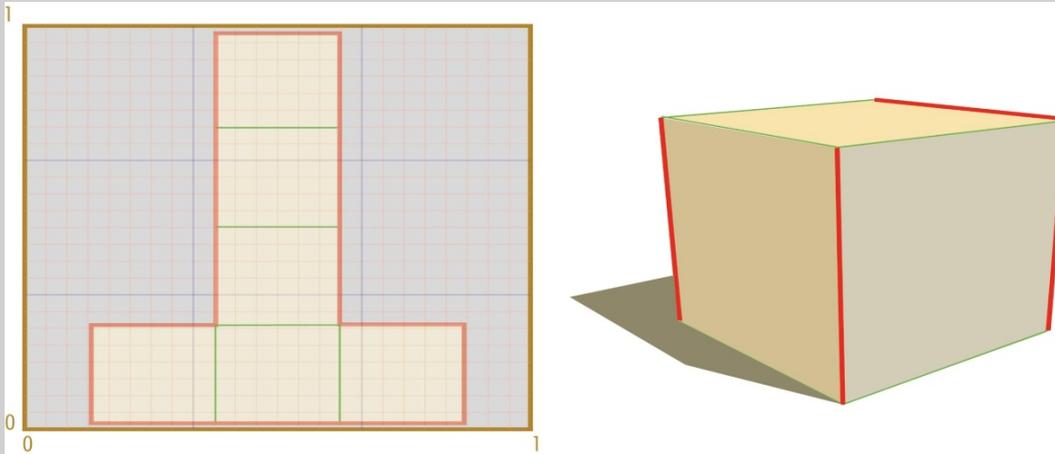
- Cannot be unfolded flat without seams or overlap
- Multiple projections required

# Hand editing

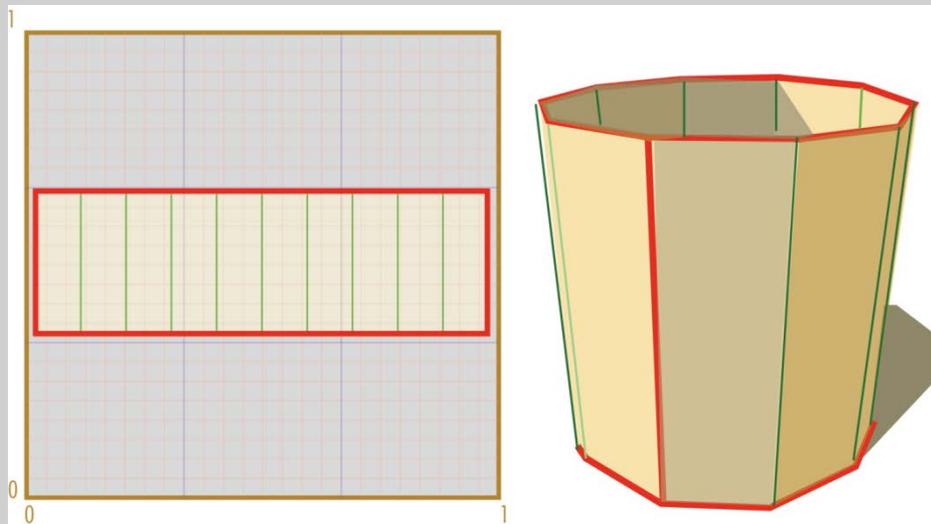


- Most objects cannot be mapped well in a single projection
- Assume that you will edit them by hand
- Be pleasantly surprised if it is unnecessary

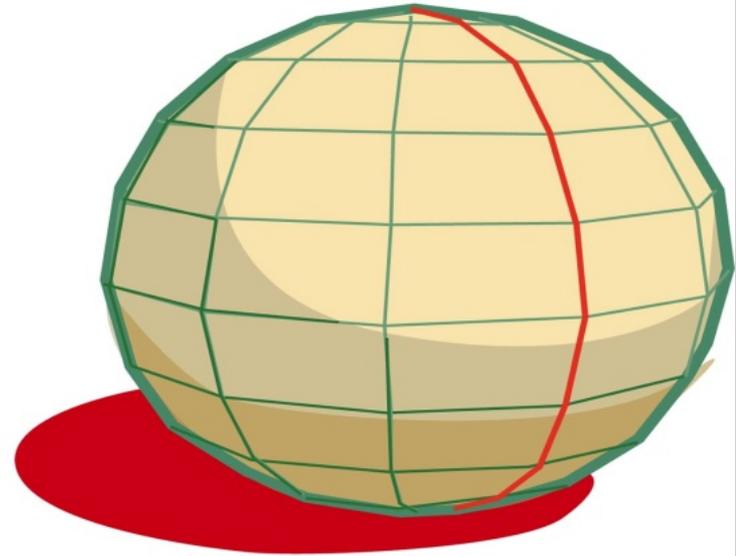
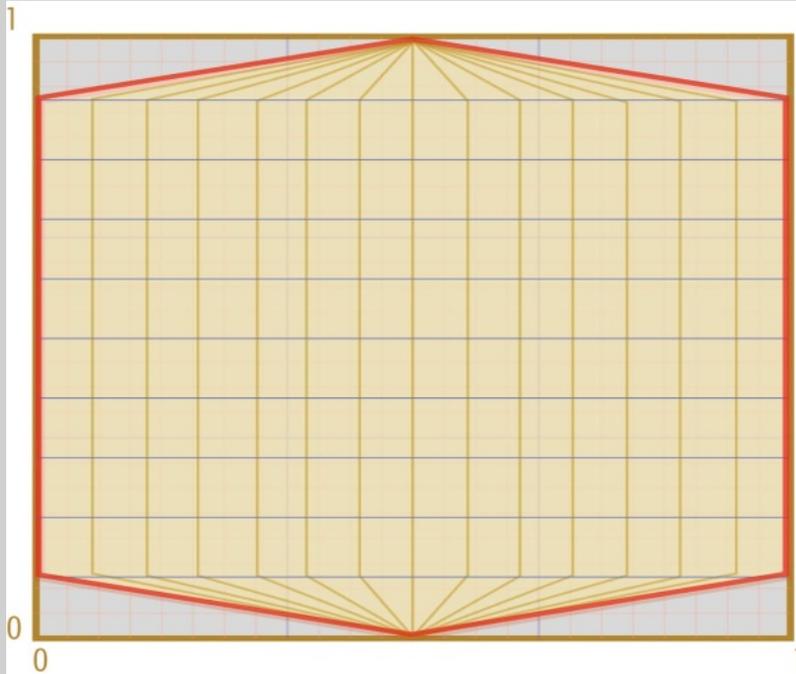
# Seams



- Decide where seams should be
- Build these into your model
- Notice how seam position affects texturing

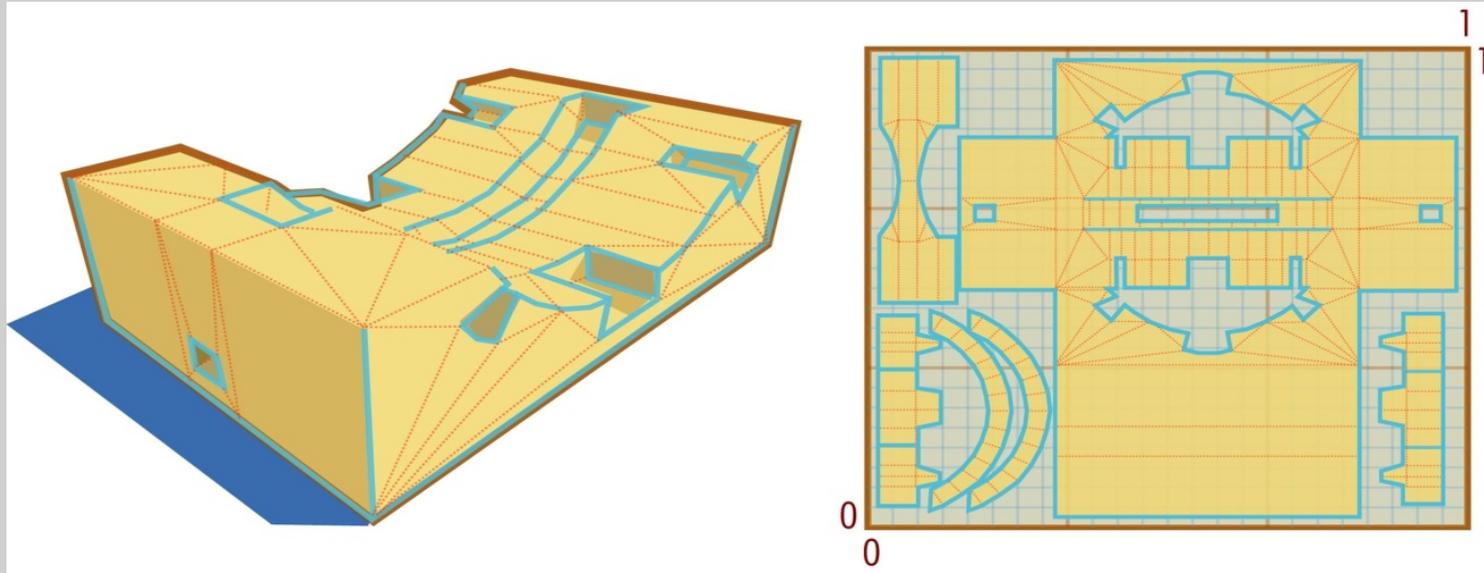


# Distortion



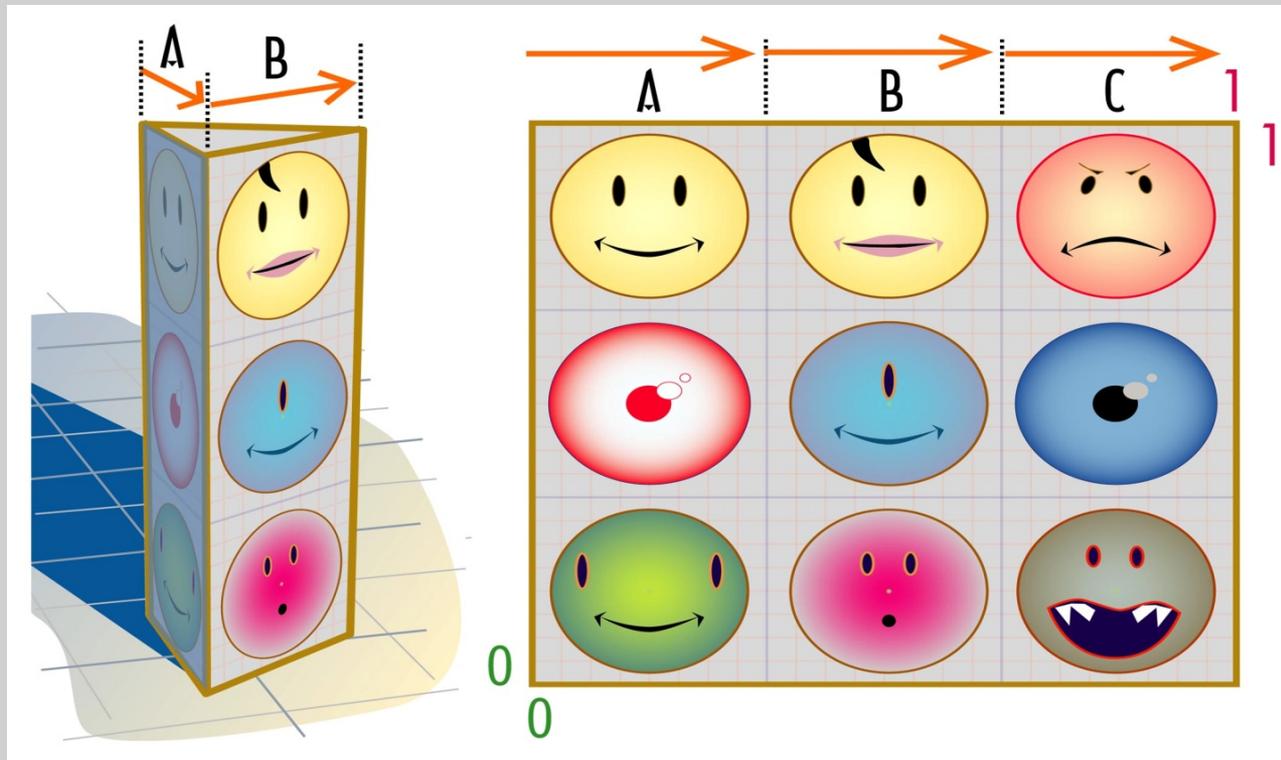
- All default sphere mapping methods have heavy distortion or numerous seams

# Distortion vs. seams



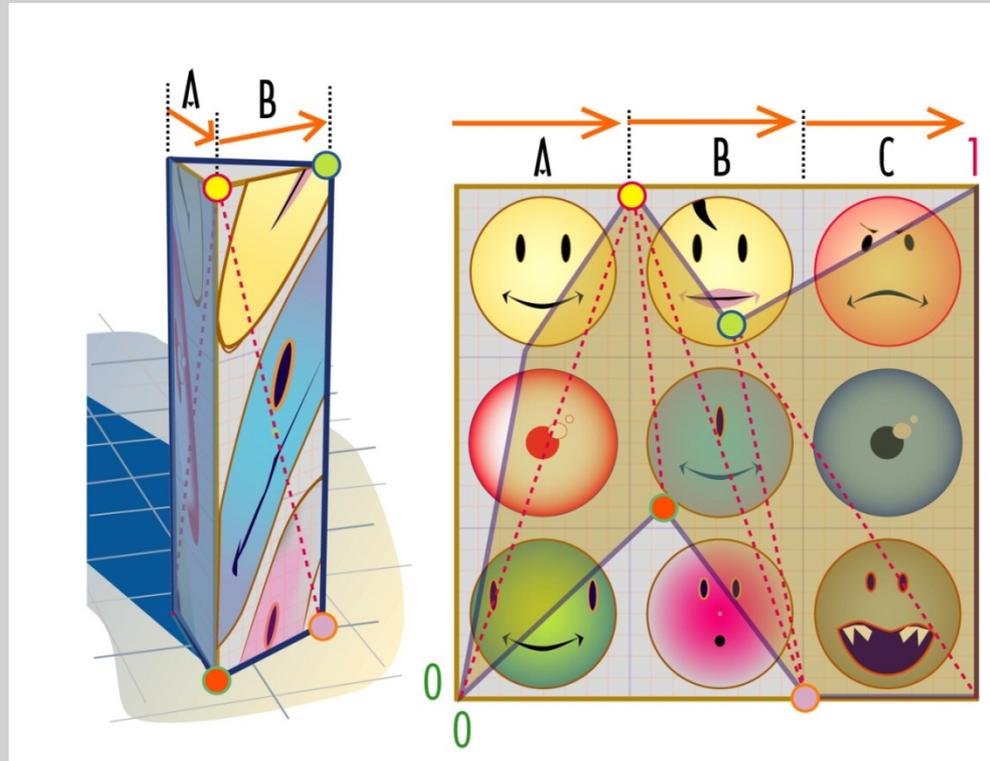
- The only objects that can be mapped without distortion have no insets or projections
  - Like a billboard
  - Or a flag

# Distortion-free rule



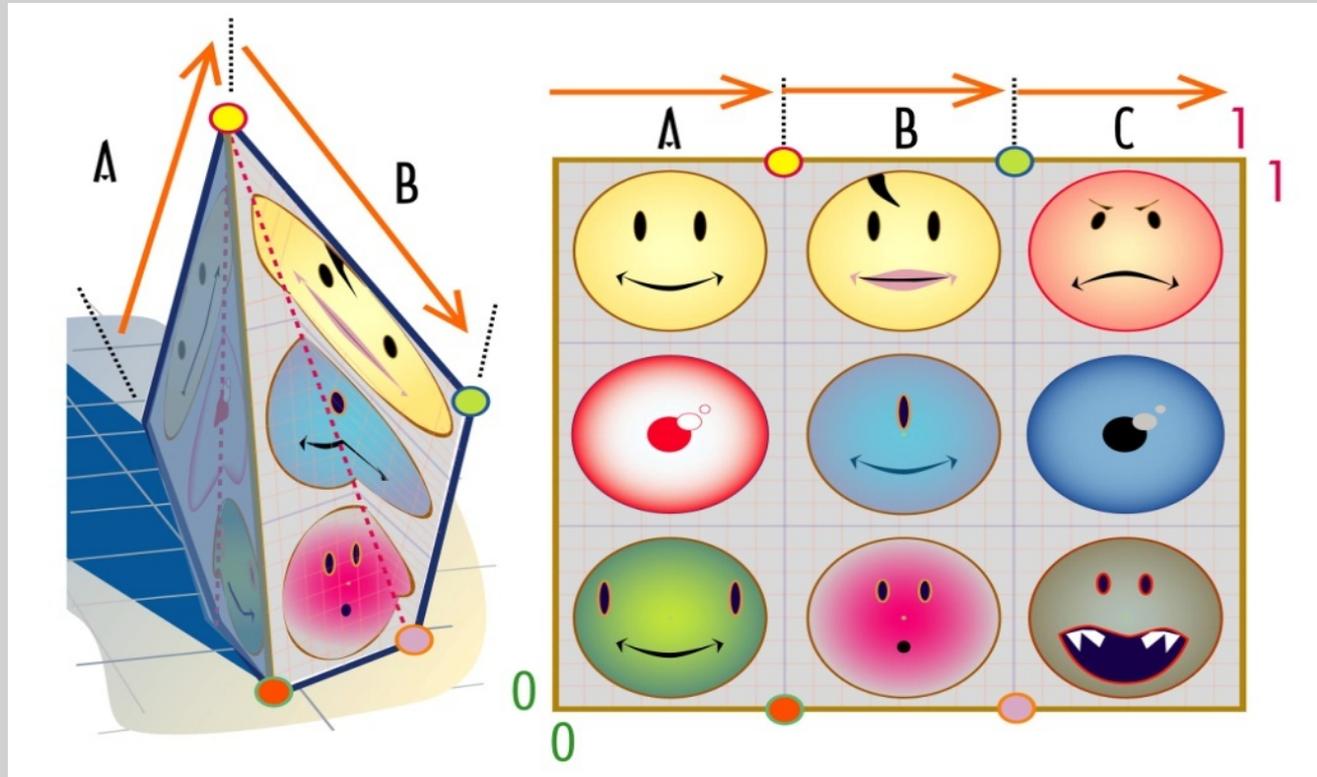
- If UVs are exactly the same shape as the polygons they are attached to, there will be no distortion

# Alternatively...



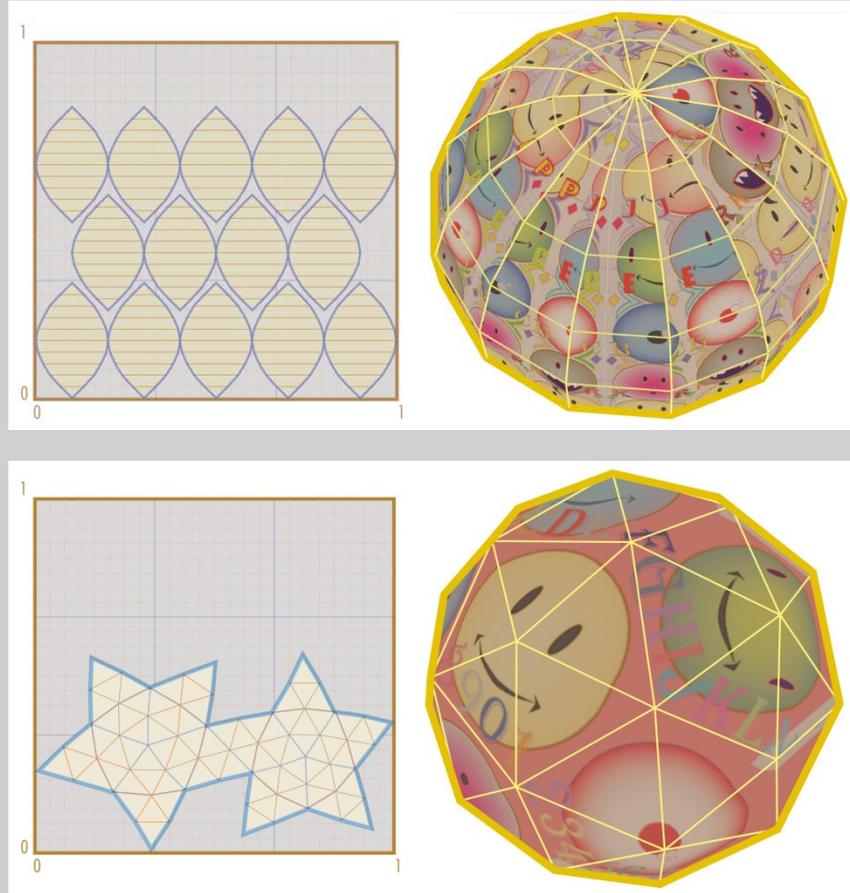
- If the shape of UVs does not match polygons they are mapped to, there will be distortion

# Another way to err...



- If vertices are moved relative to each other after UVs are assigned, distortion will occur

# Distortion-free spheres

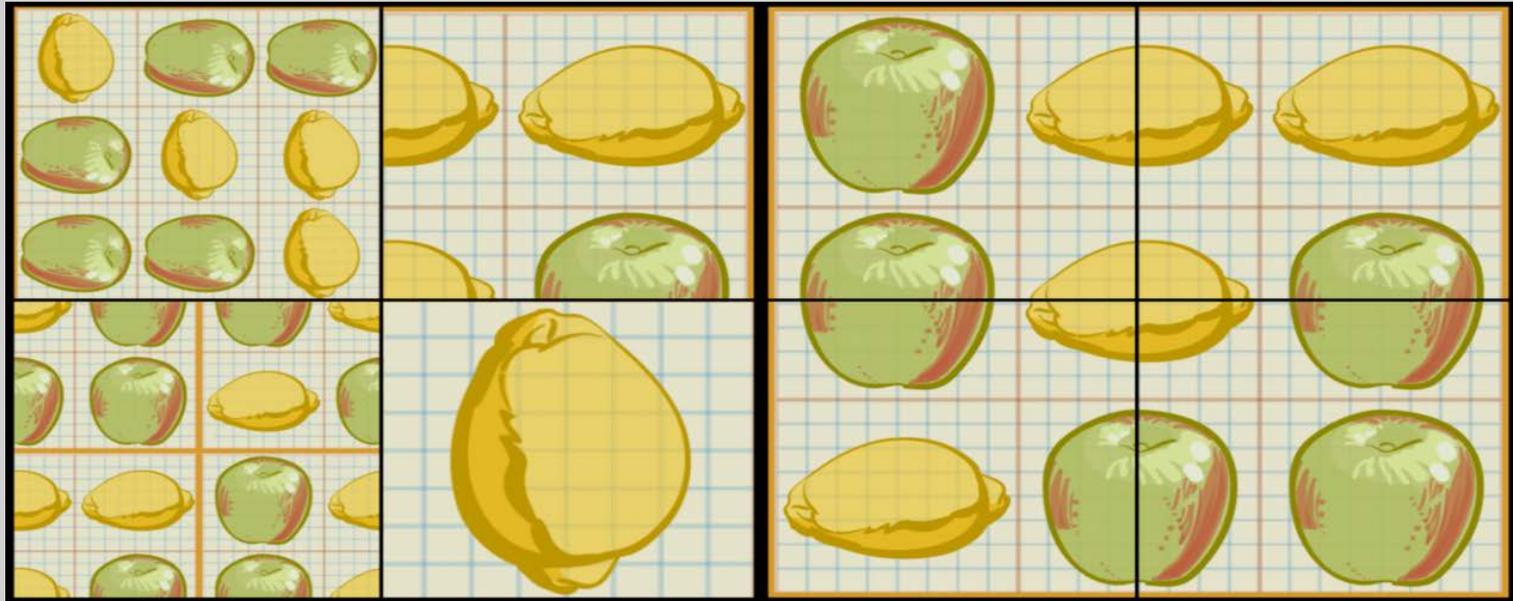


# Options

- Primitives can be mapped in many different ways
- Non-primitives have even more options
- Keep in mind that someone else must understand what you did

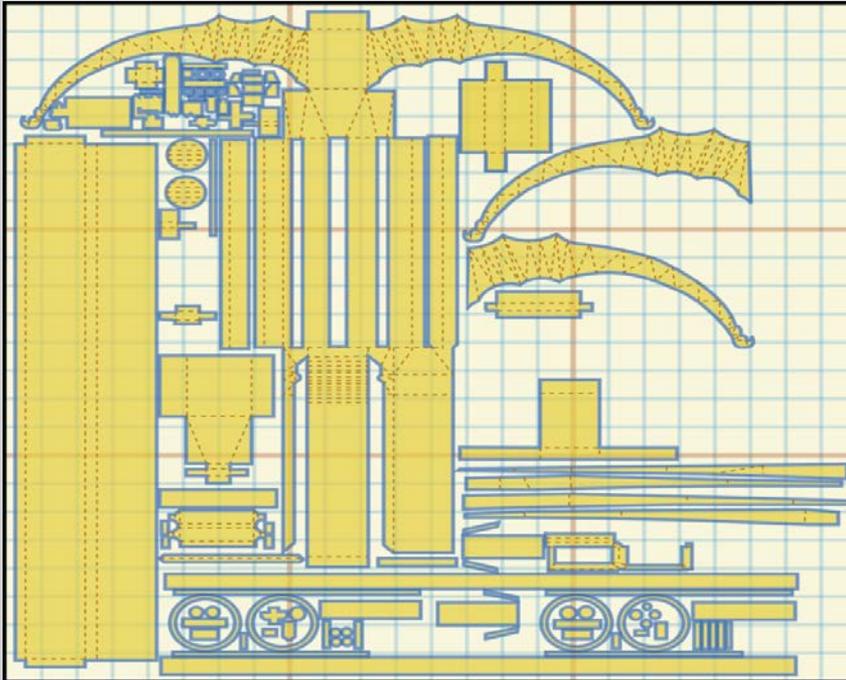


# Scale



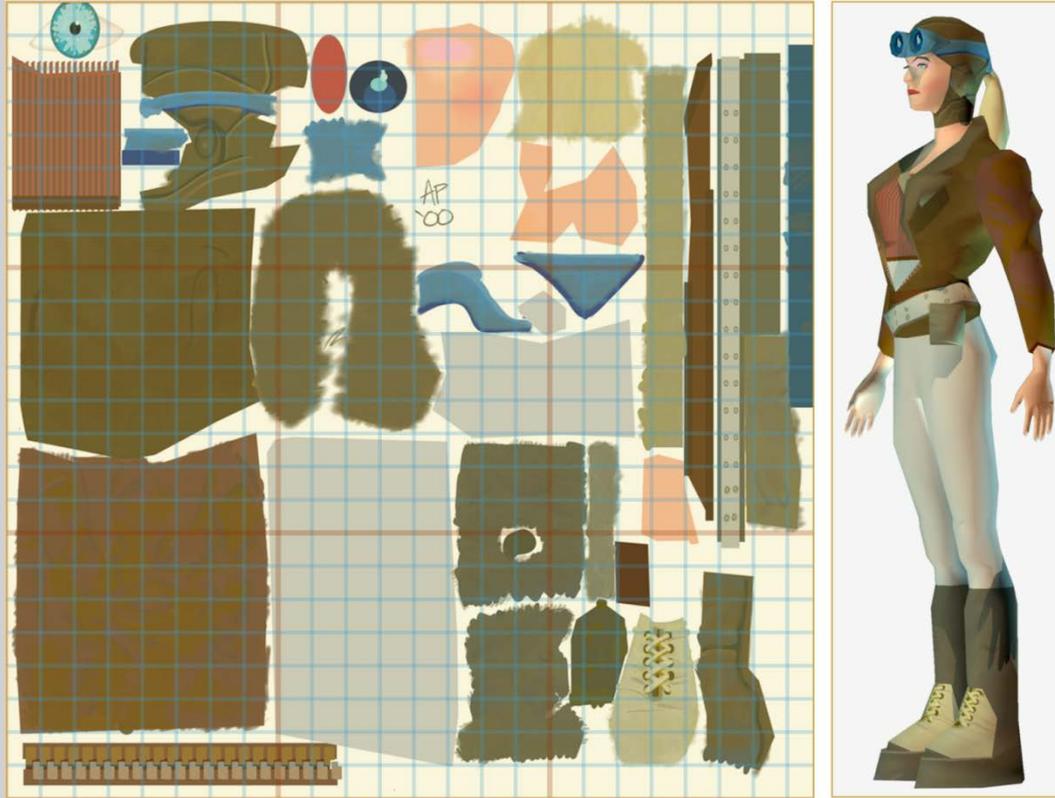
- UVs should utilize consistent pixel:polygon scale
- All objects of the same class should have the same number of pixels per unit of measure

# UV packing



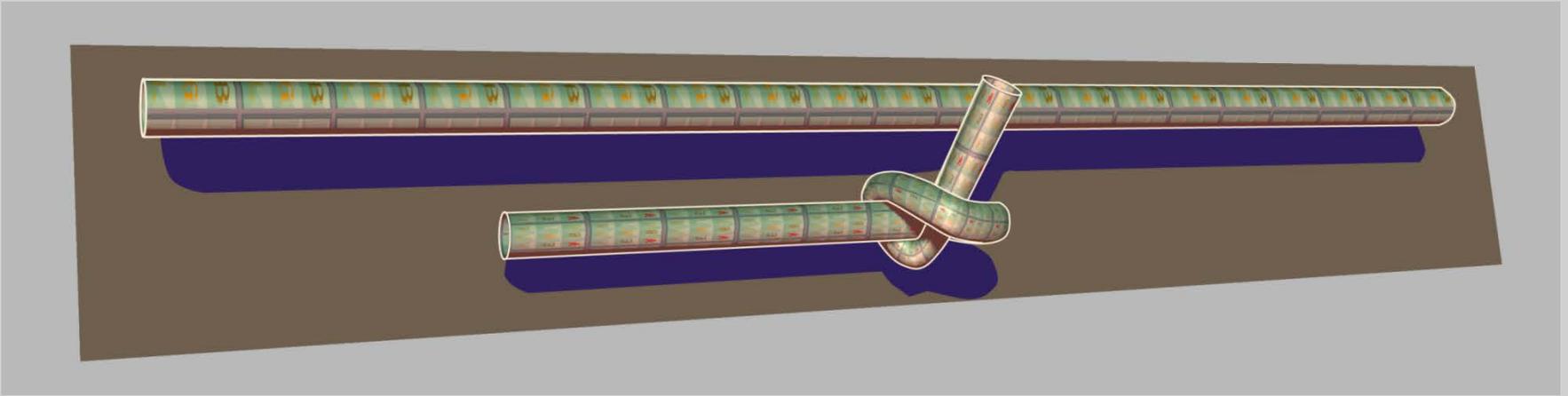
- This layout does not contain errors but is inefficient use of space
- Smaller pieces could go on different map
- Remaining objects can be rotated and scaled to fill the space better

# Mapping resolution



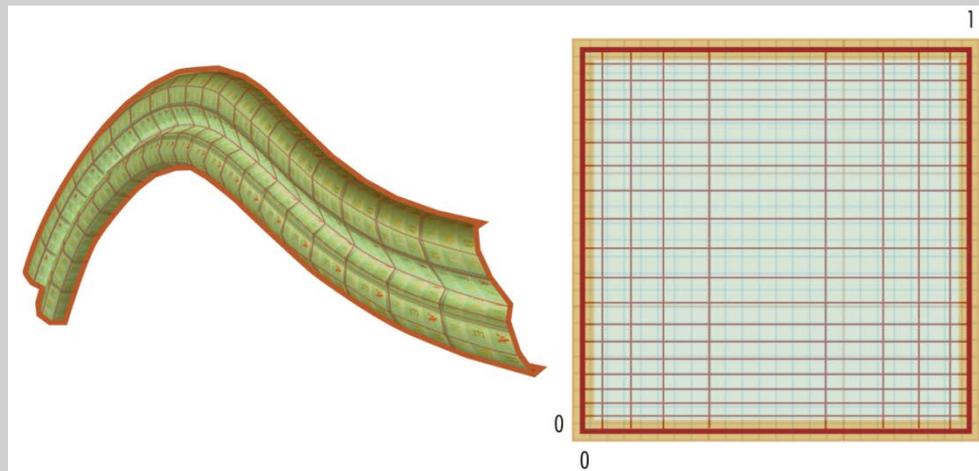
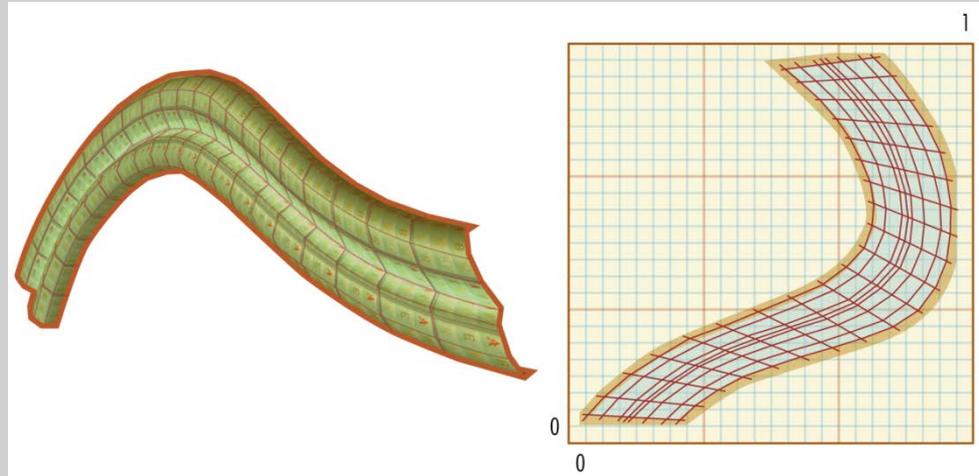
- Not the same as map resolution
- Affects amount of detail possible on object

# Knot mapping

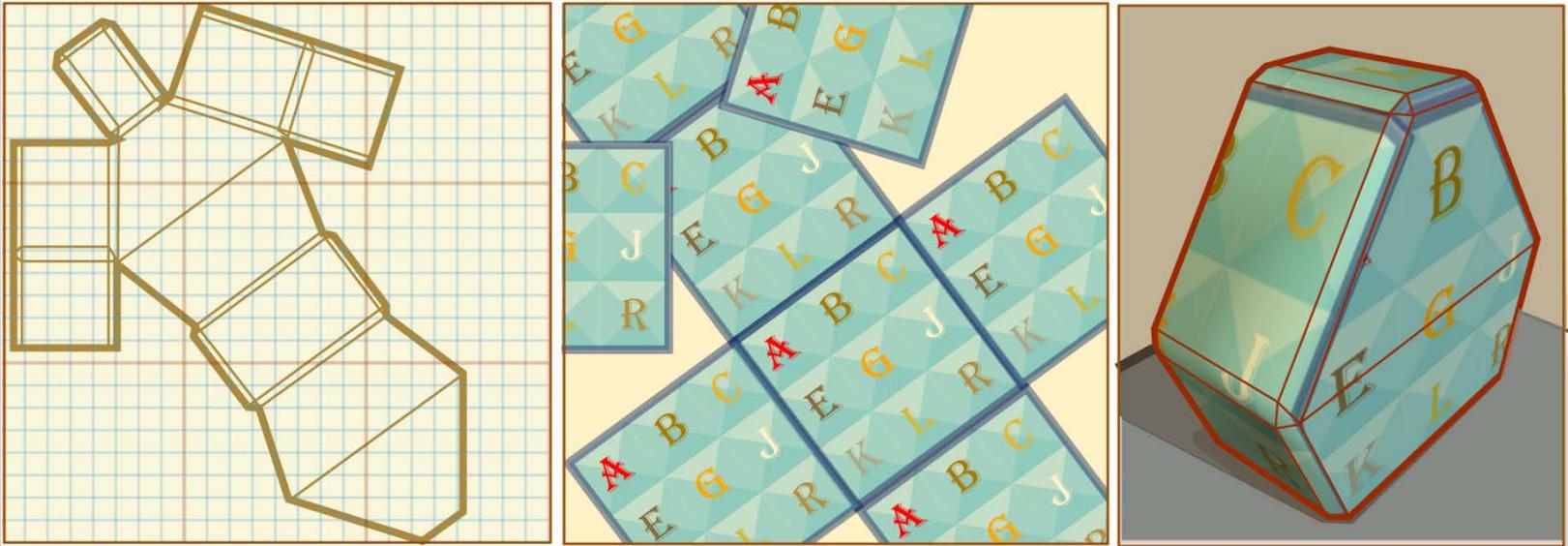


- If model is built in knot shape prior to applying UVs, it can be difficult or impossible to map well
- Better to apply UVs first, then manipulate
- A NURBS object is a good choice for this kind of object, then convert to polygons

# Projection VS. NURBS



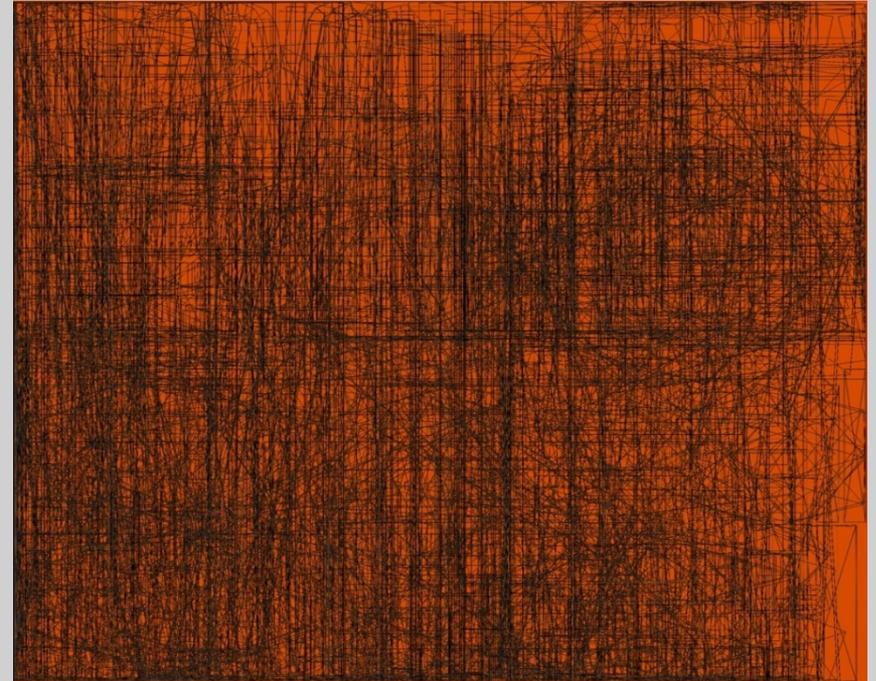
# Alignment



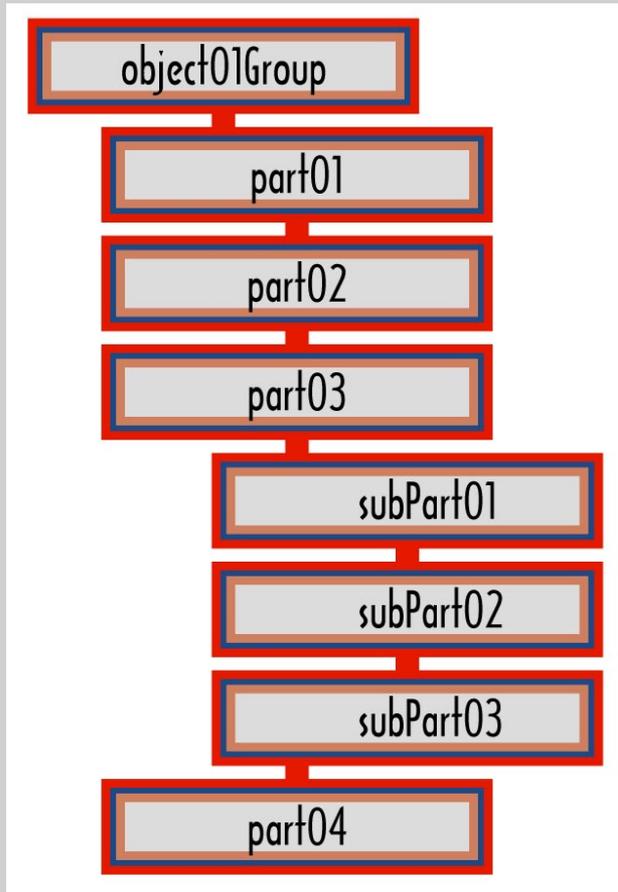
- Sometimes a well-packed layout requires a map that is rotated
- Be careful about doing this unnecessarily
- It can make texturing more difficult

# Overlap and grouping

- Can be preferred or undesirable, depending on the model
- Common in architectural models
- Uncommon in characters
- Grouping affects overlap



# Organization



- Pay attention to animation dependencies
- Naming conventions must be rigorously followed
- Avoid groups with large number of child objects
- Always have a root node

# A likeness is expected



# Professionalism

- Professional modelers can do more than match the target object
- Professionals make error-free models
- Professionals make well-organized models
- Professional models are efficient
- Professionals anticipate problems that might be encountered by team members
- The work of professionals can be easily integrated into a team effort



**Simon van den Broek**



End