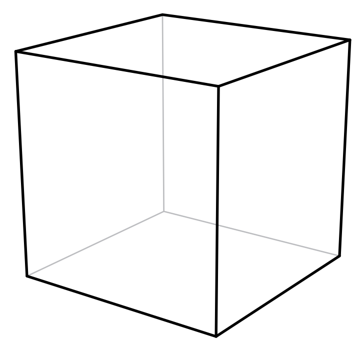
Foundations

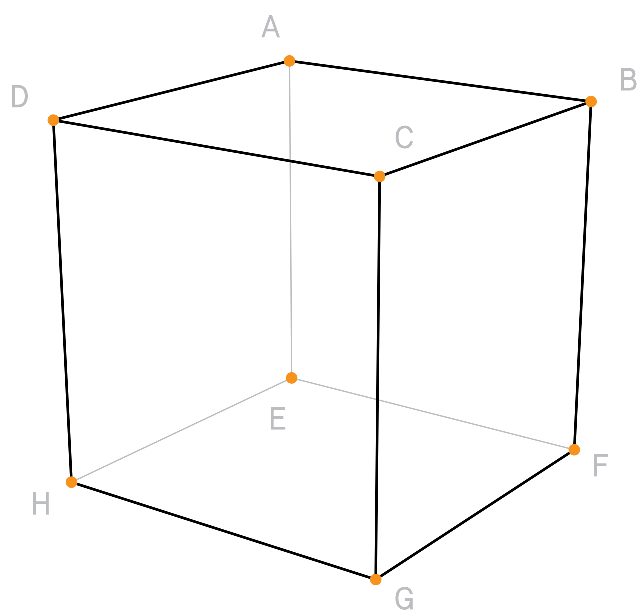
Members of the 3D modelling community fall broadly into two cohorts.

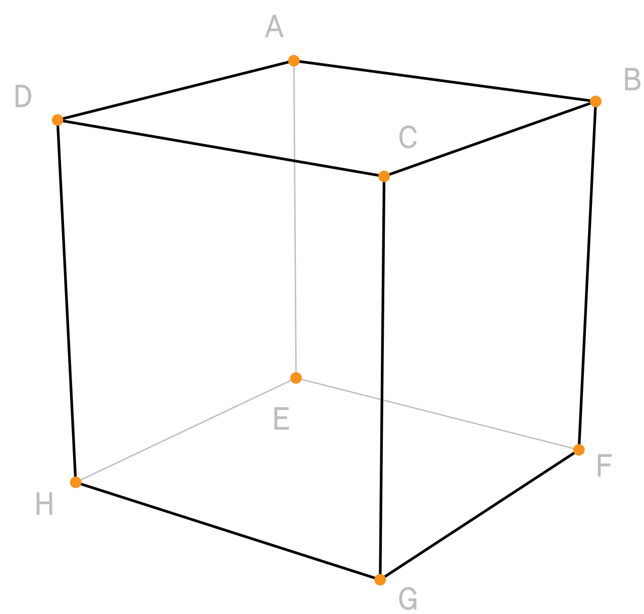
On the one hand, architects, designers, and engineers use 3D Computer-Aided Design and Manufacturing packages in order to visualize and understand the complex systems they work with. Modeling suites allow them to express complicated geometries to the exact technical specifications they require, and to translate those specifications into machine-readable NC formats for manufacturing.



On the other hand, animators and artists rely on the power of modeling tools to express complex formal propositions in a repeatable manner. Here, the power of such tools comes from the separation of the spaces they describe from the the real world. Instead of worrying about the constraints of construction time and cost, artists are free to impose their own constraints.

The ability to impose computational and semantic constraints on a modeled object is powerful becuase it provides a way to instruct an object about its purpose. Typical modeling software relies on point clound representations of geometry - a representation that uses what amounts to a list of vertices and a list of edges between vertices. The notion of the object is intimately connected with its location in space. We find this problematic, because, more often than not,what is interesting about an object is not where that object happens to be located in space, but rather, what that object does, how that object relates to itself and other objects.





A is located at (a,b,z).
B is located at (c,b,z).
C is located at (c,d,z).
D is located at (a,d,z).
E is located at (a,b,y).
F is located at (c,b,y).
G is located at (c,d,y).
H is located at (a,d,y).

et cetera.

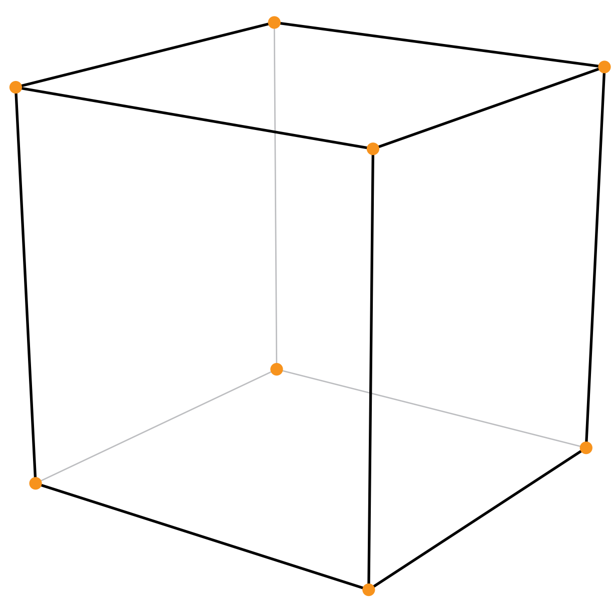
(A,B) is perpindicular to:
(B,C), (A,D),(E,H),
(F,G), (B,F),(C,G),
(A,E), (D,H).

(A,B) is parallel to:
(D,C), (H,G), (E,F).

(A,B) is coincident with:
(D,A), (B,C), (A,E),
(B,F).

et cetera.

Our ideal user is a user who works in 3 dimensional spaces, and who depends on a very high level of accuracy in their work - a level of accuracy that is better suited to being maintained by a computer, than a brain. We envision a tool that helps designers, architects, animators, and makers in general to express and understand dependencies between points, lines, and planes in space. Rather than taking vertices or lines or shapes as atomsof our system, dependency relationships are primitive. By lifting away the specificity of spatial location and orientation, we allow constraints to be applied to arbitrary geometry. the role of the system is to solve the user’s constraints. This allows for a declarative, rather than procedural approach to modeling, and further, it allows for powerful, geometric invariants to be expressed inside of the system - invariants that are critical when preparing object ideas for manufacturing and production.



User Stories: Designer

The following questions ask generally about the way people use their modeling tools, and in what capacity 3D modeling is part of their work and workflows.

How does your work require you to use 3D modeling tools?

There's probably a couple different categories of use, there's like sketching to figure out formal or spacial ideas, which is quick and dirty, which is modeling which doesn't necessarily validate for fabrication or produce shop drawings or specifications, but just helps get ideas across, and then there's drawing or modeling for making drawings and other types of representations like rendering or construction documents or schematic drawings, and finally there's modeling for fabrication and production, which usually is some sort of modeling and then exporting to certain formats like STL or OBJ or some kind of mesh format. And that has a sort of one way process that where you model something and then export it, so when you make another model or change the model, you don't get a new output. You're kind of locked in to this single vertical chain of production.

What part of your specific 3D modeling program do you find the most useful? What features would are absolutely essential to your work?

I use Rhino almost all the time. I might use other things if I need to export a file or change a format, but I prefer Rhino pretty much all the time. The most basic reason why I like Rhino is that I was trained in Autocad first, and there's a pretty distinct similarity in the command structure and the commands. There's a familiarity - it's comprehensive in its operations, a lot of people use it, it is capable of extremely complex form and extremely precise and accurate operations - it's basically just not missing anything I need.

Which parts of your platform do you find yourself using the least? Which aspects are typically forgotton or unutilized?

There's no real parametric workflow build in, but Grasshopper, the plugin for Rhino, allows parametric operations to occur. You have to write every single custom parametric operation every time you want to do it, but its extremely powerful, and you can map it to fabrication, you can map it to documentation, whatever. And it can really change the way you think about producing something.

The vertical chain of fabrication I was talking about is a downfall in Rhino - you have a model and you're not always getting a producable output. Something that would be a similar thing, but for rendering, would be this program Keyshot, where you have a model and you can change it, adjust the view, and it's always producing a rendering. You just say "save view" and it gives you a high-quality rendering, whereas in a program like VRay in Rhino or 3dStudio Max or in Maya, when you're producing a rendering, you're setting up the shot, setting the materials, and so on, and so on, and then rendering. So Keyshot is a real-time renderer, and if there was some kind of real time fabrication validation, or real time fabrication file production... If I'm looking at a model and always having a toolpath, or looking at a model and always having a cut sheet for the laser, that would be extremely valuable. Even if I had to work to make it happen, even if I had to set up all these parameters and define exactly how it should export - I would still want to use that.

Which problem or problems do you find yourself solving over and over again in your current platform, either in the same or different ways?

Probably some sort of validation. Often, with time crunches on projects, you don't go back and double, triple check everything, make sure things are properly aligned - and you have to decide what “properly aligned” means, what “properly dimensioned” means, what “properly sized” means, but I don't have a way to set that up and make sure its constantly validating, make sure that things are square and perfectly measured.

Thickness for 3D Printing, too, thickness of the shells, the walls, the wires - I have to keep remaking the model and keep almost fully rebuilding each time to make a change like that, instead of determining a thickness somewhere.

Versioning is not great, but I guess that's never really solved properly, that’s more of a user thing, I think.

I mean, another thing that has to happen over and over and over, is if you have to change the profile of a cut, and you're tabbing that profile, at least in our current workflow we're exporting it into a second software and

User Stories: Architect

The following questions ask generally about the way people use their modeling tools, and in what capacity 3D modeling is part of their work and workflows.

How does your work require you to use 3D modeling tools?

3D software isn't necessarily a requirement but it speeds up the design process by allowing for models that can be visualized and adjusted easily (good replacement for cardboard models). 3D models can also be transformed into 2D construction documents for contractors and images for clients.

What part of your specific 3D modeling program do you find the most useful? What features would are absolutely essential to your work?

I have used SketchUp and Revit which is common among architects (owned by Autodesk who made AutoCAD). Many architecture students are now using Rhino.

It's useful that objects have structural properties integrated into their forms. For example, if I create a door it will know its own dimensions, weight, location, etc. It also then gets added to a list so I can use it again. The most useful part is the ability to understand and visualize forms. Being able to spatialize designs and to view rotations and angles in 3D space is very helpful.

Accuracy is essential in designs. Keeping parallel planes parallel, keeping curved surfaces curved, aligning points and lines, etc. The ability of lines, points, and objects to learn information about themselves and where they belong helps this.

Which parts of your platform do you find yourself using the least? Which aspects are typically forgotton or unutilized?

It's hard to know what I don't use but it is frustrating at least having to know exactly where points lie in space. If I draw a bridge on paper you'll know what it is regardless of where it's at in space. The important part is how everything relates to everything else.

If you could change anything about your current platform, what might that be?

This probably isn't really what you're looking for but being able to draw and form things with my bare hands would be so cool (Leap motion style).

Which problem or problems do you find yourself solving over and over again in your current platform, either in the same or different ways?

When I used AutoCAD (not 3D) my favorite commands were copy and mirror. Some sort of scripting or the ability to run a series of commands repeatedly would be useful.

What would you say the greatest source of error in your work is, currently. How does your program work to contribute to or alleviate this?

We typically don't spend time enough time cutting cross-sections of buildings to understand in detail what is going on. This causes us to end up with things we think will work or go together but don't actually. If a computer algorithm were to solve some of these problems however it should ultimately be up to the designer whether or not changes should actually be made.

The next set of questions asks users to reason through a hypothetical problem, in the language of their current modeling platform.

Imagine for a moment that you're working on a project where it's important fot two different geometries to maintain some size and shape relative to one another, irrespective of resizing or relocation. How would you go about expressing this in your model? Be as general or specific as you like.

I would make them both part of the same object in order to move/scale them together. It's not a solution but it's close to what you're asking.

Now imagine that you've drawn a shape in your program, and you want to ensure that it tessellates. How would you go about checking this quality?

I would use array commands to duplicate an object repeatedly and set what edge or angle to copy over.

Finally, let's say that you've just finished modeling a huge project with hundreds of thousands of points and polygons, when you realize that a large set of edges in the model, which are supposed to be coplanar, are not. What do you do next?

First reaction is to undo it and do it right. Most definitely don't continue. Don't really know how else to go about changing it.

adding tabs, and then if we have to change that profile by one millimeter, we have to reexport and re-tab, which is crazy. I'd say I'm redoing more things on the tool path side than on the modeling side.

What would you say the greatest source of error in your work is, currently. How does your program work to contribute to or alleviate this?

Not modeling in 3D is the biggest source of errors. Drawing tool paths without making a 3d model and checking the validity of that part, is probably the biggest src of erros. but you know the fact that there is this waterfall workflow, is probably a source too, you know when you want change a tool path to redo all the models and properly version then is sometimes just not economical, especially when you're on a time crunch. To rebuild all that stuff just because you want to change one dimension is crazy.

The next set of questions asks users to reason through a hypothetical problem, in the language of their current modeling platform.

Imagine for a moment that you're working on a project where it's important fot two different geometries to maintain some size and shape relative to one another, irrespective of resizing or relocation. How would you go about expressing this in your model? Be as general or specific as you like.

I would decide that in my head, decide that lets say a leg has to fit into a whole and then i would model it with making each form separately. I'd make the solid and make the peg and then use a command like booleandifference to subtract the peg from the solid, and then I would just have decided that those two pieces are linked, but I wouldn't do something empirical to ensure that, and if I was just working with someone else i would just tell them that. so if i were handing off a file, I would have to write notes, write specifications that would prove that that thing stays where it is and the dimensions are linked.

If i really wanted to make sure it would happen, i would make a grasshopper definition, that would take the two forms as inputs and and intersect them to create the proper opening, and keep them linked, but then you have to hand off the grasshopper definition, so that's a two part process, not baked into rhino.

Now imagine that you've drawn a shape in your program, and you want to ensure that it tessel-lates. How would you go about checking this quality?

I'd write a grasshopper definition that takes the input surface and however it is to be divided, I would write a definition to handle that, if its triangulation, I'll divide the surface into U and V definitions, and then draw polygons that connect the grid that those coordinates form. once you have that, its fairly easy to fold in sliders or parameters that allow those shapes to be trans-formed or scaled consistently.

Finally, let's say that you've just finished modeling a huge project with hundreds of thousands of points and polygons, when you realize that a large set of edges in the model, which are supposed to be coplanar, are not. What do you do next?

Hmm. Sounds like I would rebuild it. I don't know, I mean it depends whether or not I feel like I can select the right parts - if I feel like I can get the right view of the object and select the bad parts with the right window, I would do that, but if I can't i'd just rebuild them. Hopefully, if its meshes I have some kind of NURBS surface that it came from, so I can adjust that and then remesh it, but if not, I guess I'd just have to rebuid. I don't know what else I could do.