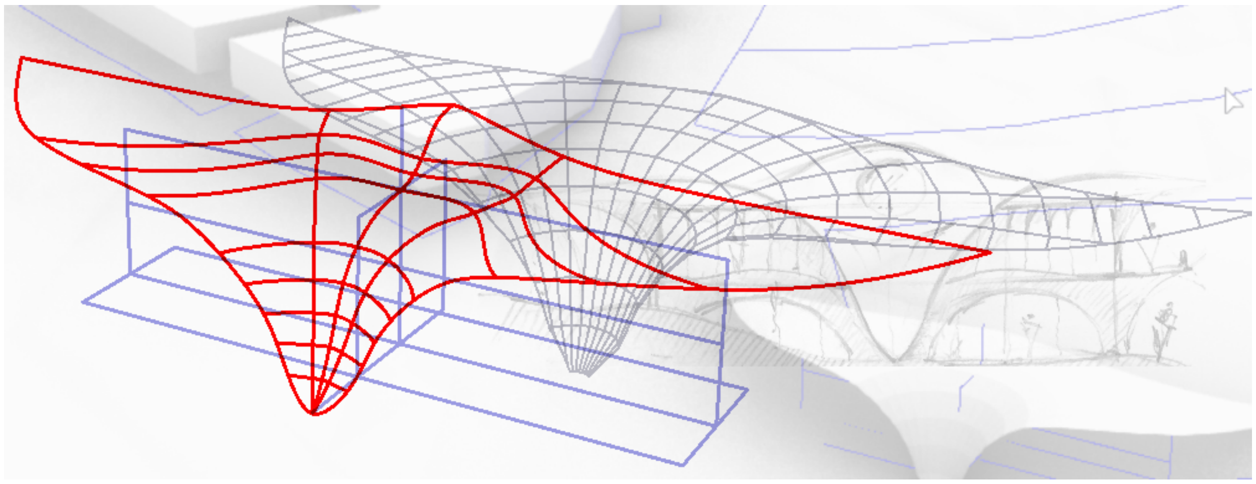
Rhinoceros

Modeling Workflows in Architecture



© Robert McNeel & Associates 2021 All Rights Reserved.
Printed in USA
Copyright © by Robert McNeel & Associate

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission. Request permission to republish from: Publications, Robert McNeel & Associates, 3670 Woodland Park Avenue North, Seattle, WA 98103; FAX (206) 545-7321; email permissions@mcneel.com.

Table of Contents

Preface

Digital modeling has taken leaps and bounds in the past few decades. From basic tools used to optimize drafting, to an intelligent medium that is changing how we design and build. Digital means are infiltrating every single aspect of design and construction today. This fast pace development of technology, and the inclusion of end users in the making of digital tools, is forcing a volatile and fast-changing platform with numerous workflows. In commercial software available today, we can distinguish four dominant approaches to digital modeling: direct, algorithmic, object-based and parametric. Rhinoceros and its plugins support all four modeling methods, which makes the task of presenting Rhino to new users very challenging. The versatility of Rhino and the ease in which they can be extended through plugins means that it is very hard to be specific about how Rhino is used, by who and when.

While contemplating best ways to present the Rhino modeling environment to architects, I thought it might be best to base the presentation on modeling methods and workflows, rather than a description of a finite set of tools. My hope is that this approach will promote critical understanding of the tools, and encourage users to be creative in how they use Rhino. This is perhaps the best way to understand Rhino's best potential, but also help challenge and push it in a positive direction.

The content is presented in three parts. The first part presents common modeling vocabulary, and how it manifests in Rhino. The second part, which is the bulk of this document, is about modeling workflows in architecture using Rhino core modeling tools. It covers four critical stages of modeling: setting up the modeling environment, concept modeling workflows, detailed modeling workflows, and prototyping. The third involves a brief discussion of digital modeling methods in architecture, and how Rhinoceros, Grasshopper and other McNeel plugins fit within these methods.

While this is by no means a comprehensive manual, I hope that it will serve as an introduction to help getting started with Rhino and Grasshopper.

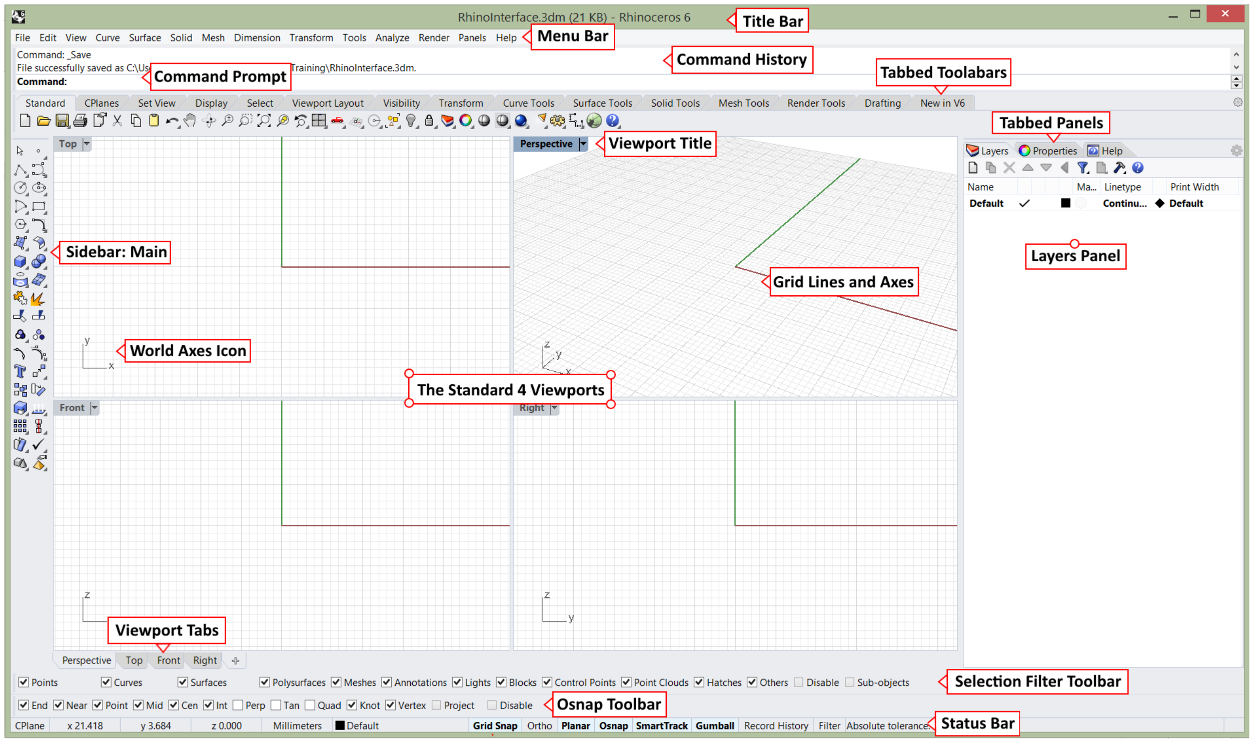
Rajaa Issa
Robert McNeel & Associates
January, 2021

Part I: Modeling vocabulary

3D modeling is the process of using computers to create, reflect on, and share three dimensional forms. Computers use geometry as the primary conduit to represent design ideas and use data to attach information to the geometry. Most 3D modeling applications share common vocabulary and the very first step to learn how to model within any environment is to understand its vocabulary. That includes workspace organization, access to different tools, interface and modeling controls, what geometry is supported and how it aggregates and how data is applied and managed. Visualization, analysis and presentation of 3D forms is an important part of any modeling application. Models are stored and exchanged using files and file formats are either specific to the application or more widely used across multiple ones. This chapter discusses the modeling vocabulary in Rhino.

Workspace interface

Once you open the Rhino application, you will see a screen with a graphic area in the center (viewports) surrounded by menus, toolbars, controls and panels. Most of the workspace elements can be rearranged, removed or expanded.



Command: _Save
File successfully saved as C:\Us... Training\RhinoInterface.3dm.
Command:

RhinoInterface.3dm (21 KB) - Rhinoceros 6
Title Bar
File Edit View Curve Surface Solid Mesh Dimension Transform Tools Analyze Render Panels Help
Menu Bar
Command History
Command Prompt
Tabbed Toolabars
Standard CPlanes Set View Display Select Viewport Layout Visibility Transform Curve Tools Surface Tools Solid Tools Mesh Tools Render Tools Drafting New in V6
Tabbed Panels
Layers Properties Help
Name Ma... Linetype Print Width
Default Continu... Default
Layers Panel
Top
Perspective
Viewport Title
Sidebar: Main
Grid Lines and Axes
World Axes Icon
The Standard 4 Viewports
Front
Right
Viewport Tabs
Perspective Top Front Right
Points Curves Surfaces Polysurfaces Meshes Annotations Lights Blocks Control Points Point Clouds Hatches Others Disable Sub-objects
Selection Filter Toolbar
End Near Point Mid Cen Int Perp Tan Quad Knot Vertex Project Disable
Osnap Toolbar
CPlane x 21.418 y 3.684 z 0.000 Millimeters Default Grid Snap Ortho Planar Osnap SmartTrack Gumball Record History Filter Absolute tolerance
Status Bar

The Rhinoceros interface in Windows

Title Bar	Displays the filename and the Rhino version used.

Menu Bar	Access commands, options, and help through text-based drop-down menus.
Command Prompt	Area to type and run commands, and see options and other information displayed by the command while running or after it exits.
Command History	Displays the most recently used commands and all their prompts.
Tabbed Toolbars	Groups are containers with one or more toolbar, with a tab at the top for each toolbar.
Sidebar	Toolbar to access common commands and options related to the selected tabbed toolbar.
Tabbed Panels	Rhino controls such as layers, properties, materials, lights, display mode, help and more are displayed in panels with tabs. Tabbed panels can be arranged side by side, or vertically.
Layer and Property Panels	One of the important panels that can be tabbed. The other important panel is the properties panel. Most users have at least these two panels visible.
Status Bar	Displays the coordinates of the pointer, the units and the current layer of the model, toggles and other options.
Osnap Toolbars	Used to set object snap settings.
Selection Filter Toolbar	Used to narrow down what geometry can be selected.
Viewport Tabs	Click to make active a viewport. Also, use the “+” to add more viewports and Layouts (paper space).
Standard Viewports	Displays different views of the model within the graphics area. Viewports can show a grid, grid axes, and world axes icon. Any number and combination of viewports can be arranged within the graphic area. The default viewport layout displays four viewports (Top, Front, Right, and Perspective).
Viewport Title	Display the viewport projection. Click the arrow to access viewport settings through the drop-down menu.
Gridlines and Axes	Used to aid modeling. Their visibility, colors and other settings can be customized.
World Axes Icon	Used to aid modeling. It changes when rotating the model.

Description of Rhinoceros interface elements

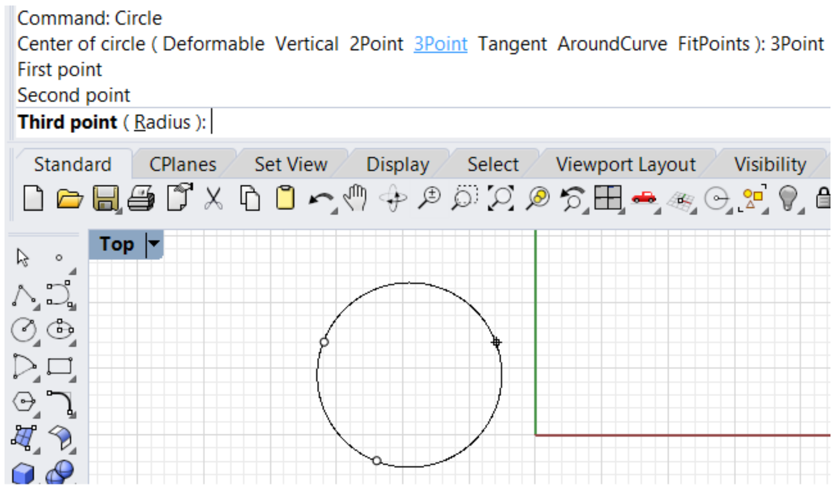
User Interface tutorial
User interface for Windows and user interface for Mac video tutorials

Commands

Commands are actions or functions that perform specific operations. For example, ”Circle” is a command that helps create and add a circle to the 3D model. Rhino allows invoking the same commands in many different ways: menus, toolbars, command line and interactively through widgets, mouse and keyboard events. For advanced users, Rhino also supports creating new custom user commands and toolbars through macros and scripting. The different ways to access Rhino commands allows users flexibility to access in the way they are most comfortable with and to help increase their productivity.

Command Line

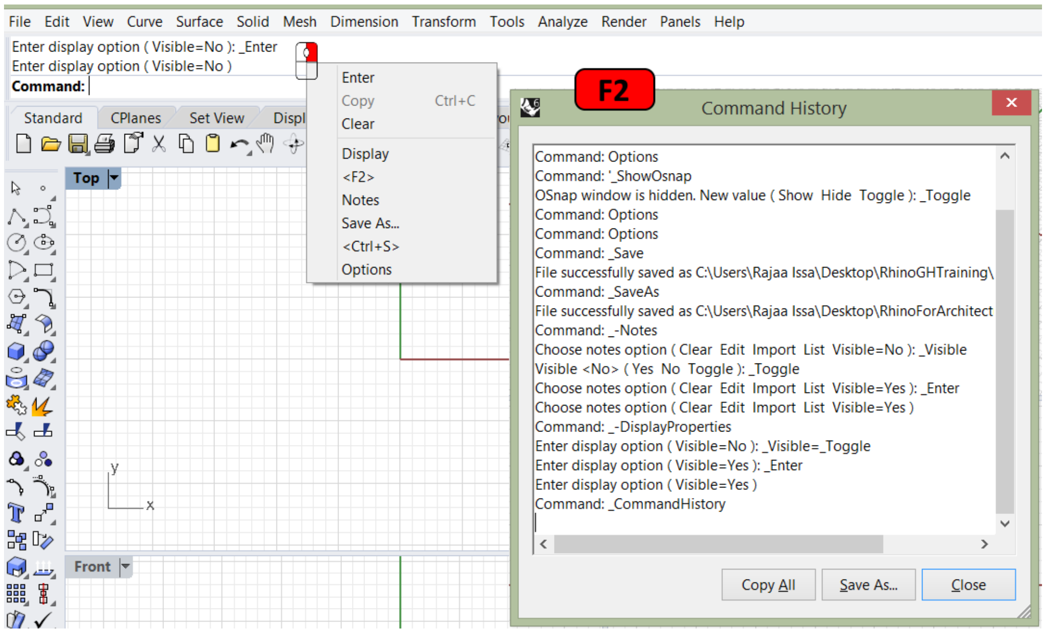
This is a widely used method to access commands in Rhino especially among users familiar with Rhino and the command set they commonly use. Commands are run by typing their names (or the first few letters of the command) in the Command Prompt. Typing in the command prompt shows a list of all commands starting with the same letters. Frequently used commands show on top and it's also possible to repeat recently run commands. You can press “Enter” to run the last command (or right-mouse-click), and press “Esc” to cancel a running command. When inside a command, the command prompt area shows the command instructions and options. Many Rhino commands do not support dialogs and hence command options can only be set through the command prompt area.



```
Command: Circle
Center of circle ( Deformable  Vertical  2Point  3Point  Tangent  AroundCurve  FitPoints ): 3Point
First point
Second point
Third point ( Radius ):
```

Figure (14): The command prompt shows options and record steps in the command history area

The command history area shows recent commands and options used. Right-click on the command line to view recently used commands. You can Press F2 to view the command history and options used.


File Edit View Curve Surface Solid Mesh Dimension Transform Tools Analyze Render Panels Help
Standard CPlanes Set View Displ
Top
Front
Enter
Copy Ctrl+C
Clear
Display
<F2>
Notes
Save As...
<Ctrl+S>
Options
F2
Command History
Copy All
Save As...
Close

```
Enter display option ( Visible=No ): _Enter
Enter display option ( Visible=No )
Command:
```

```
Command: Options
Command: '_ShowOsnap
OSnap window is hidden. New value ( Show  Hide  Toggle ): _Toggle
Command: Options
Command: Options
Command: _Save
File successfully saved as C:\Users\Rajaa Issa\Desktop\RhinoGHTraining\
Command: _SaveAs
File successfully saved as C:\Users\Rajaa Issa\Desktop\RhinoForArchitect
Command: _-Notes
Choose notes option ( Clear  Edit  Import  List  Visible=No ): _Visible
Visible <No> ( Yes  No  Toggle ): _Toggle
Choose notes option ( Clear  Edit  Import  List  Visible=Yes ): _Enter
Choose notes option ( Clear  Edit  Import  List  Visible=Yes )
Command: _-DisplayProperties
Enter display option ( Visible=No ): _Visible=_Toggle
Enter display option ( Visible=Yes ): _Enter
Enter display option ( Visible=Yes )
Command: _CommandHistory
```

Figure (15): Right-click on the command area to access recent commands. Click F2 to navigate the last 500.

Menus

You can find most of the Rhino commands in the menu bar. Menus group similar commands together. There are many submenus within the drop-down menu. Third party plug-ins sometimes add their own menus to the menu bar. Each menu groups similar commands together. For example the “Curve” menu includes curve creation and editing commands.

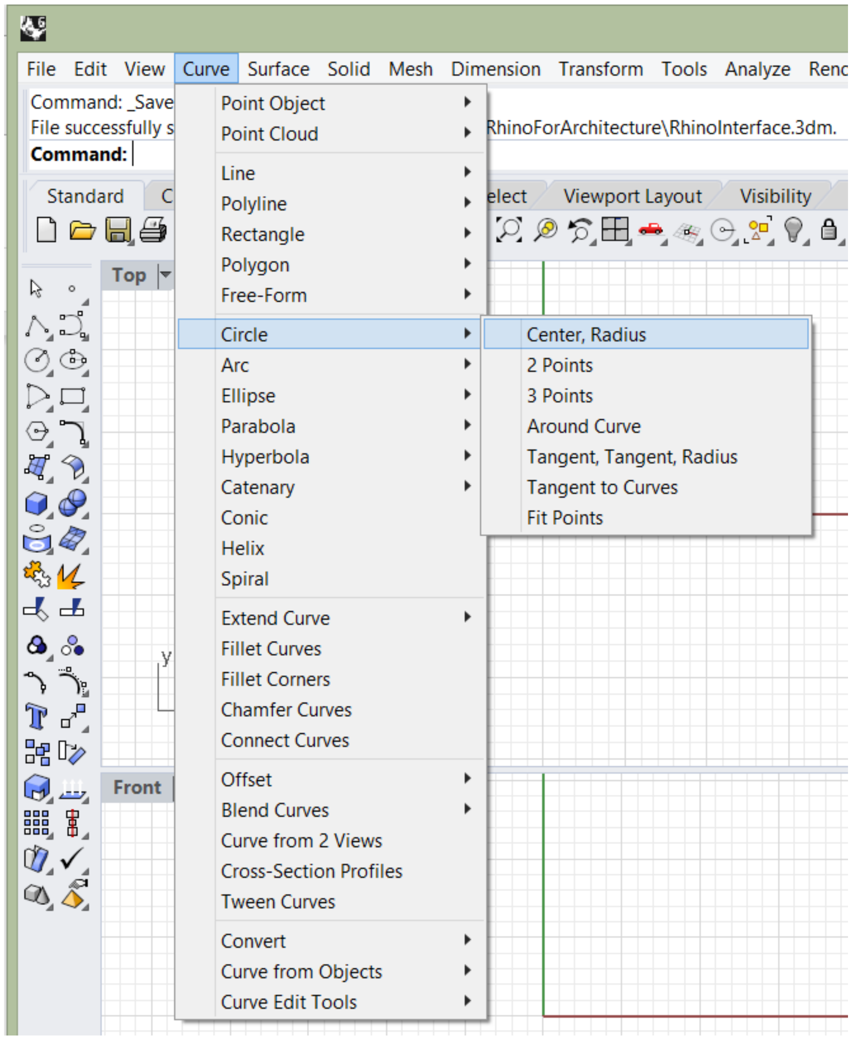

Command: _Save
File successfully s RhinoForArchitecture\RhinoInterface.3dm.
Command:

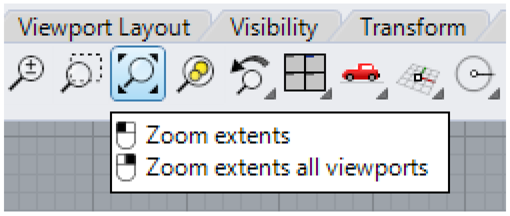
Figure (13): Curve menu and submenus include all Curve commands

Toolbars

Rhino toolbars contain buttons that provide shortcuts to commands and options. You can float a toolbar anywhere on the screen, or dock it at the edge of the graphics area. Rhino starts up with the *Standard* toolbar group docked above the graphics area and the Main toolbar as the sidebar on the left.

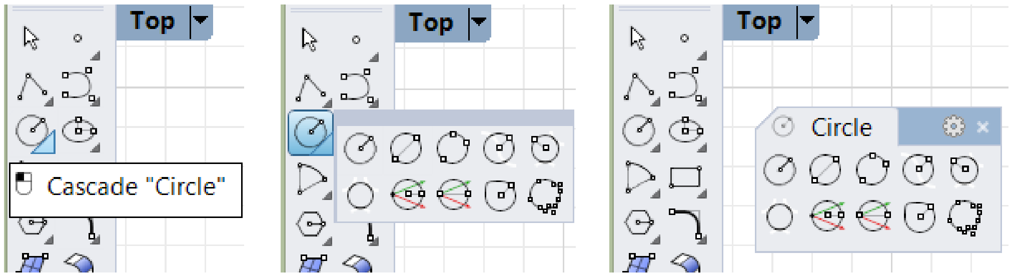
Tooltips

Tooltips tell what each button does. To access the tooltip, move your pointer over a button without clicking it and a small tag with the name of the command appears. In Rhino, buttons can be set to execute multiple commands and set specific options. The tooltips aslo indicate which buttons have dual functions as in the following example.



Tooltips give hints about what the button command does

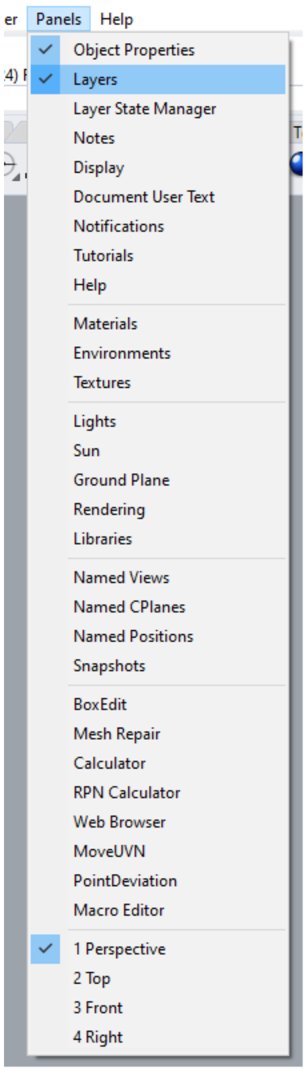
Cascading toolbars
A button on a toolbar may include other buttons in a cascading toolbar. Usually, the cascading toolbar contains variations of the base command. After you select a button on the cascading toolbar, the toolbar disappears. Buttons with cascading toolbars are marked with a small black triangle in the lower right corner. To open the cascading toolbar, hover over the black triangle and LMB click. You can select a button from the cascade toolbar, or peal out by clicking on the top margin of the toolbar.

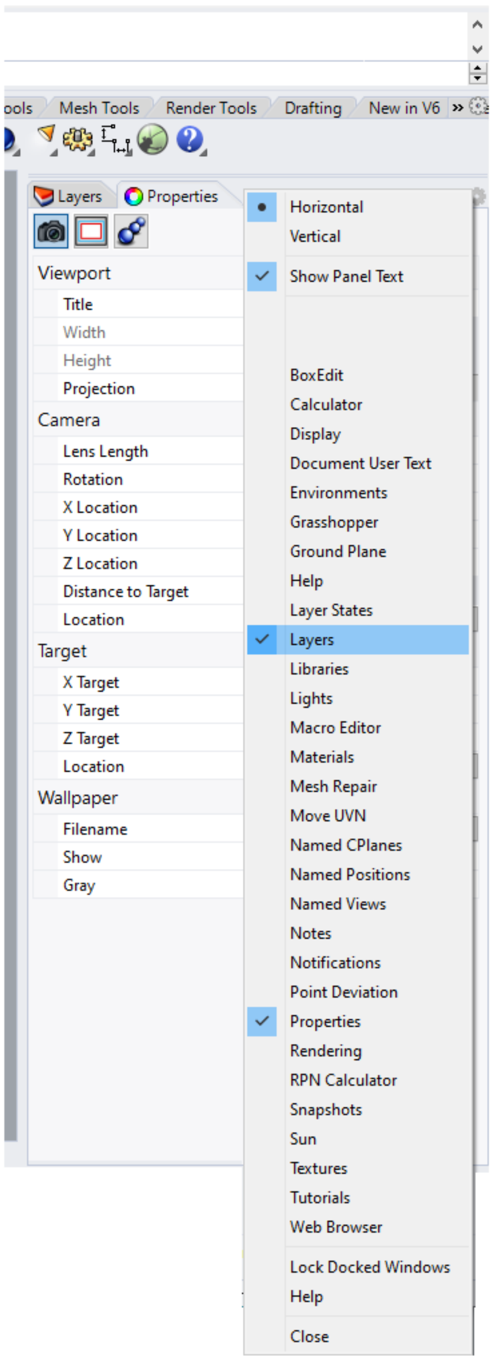


Copy of the cascade toolbar can be peeled off

Panels

Many Rhino controls are contained in tabbed panels. Open any panel from the Panels menu or right-mouse-click on the margin of another open panel and check the desired panel. Once a panel is opened, you can click on its title and drag to dock.

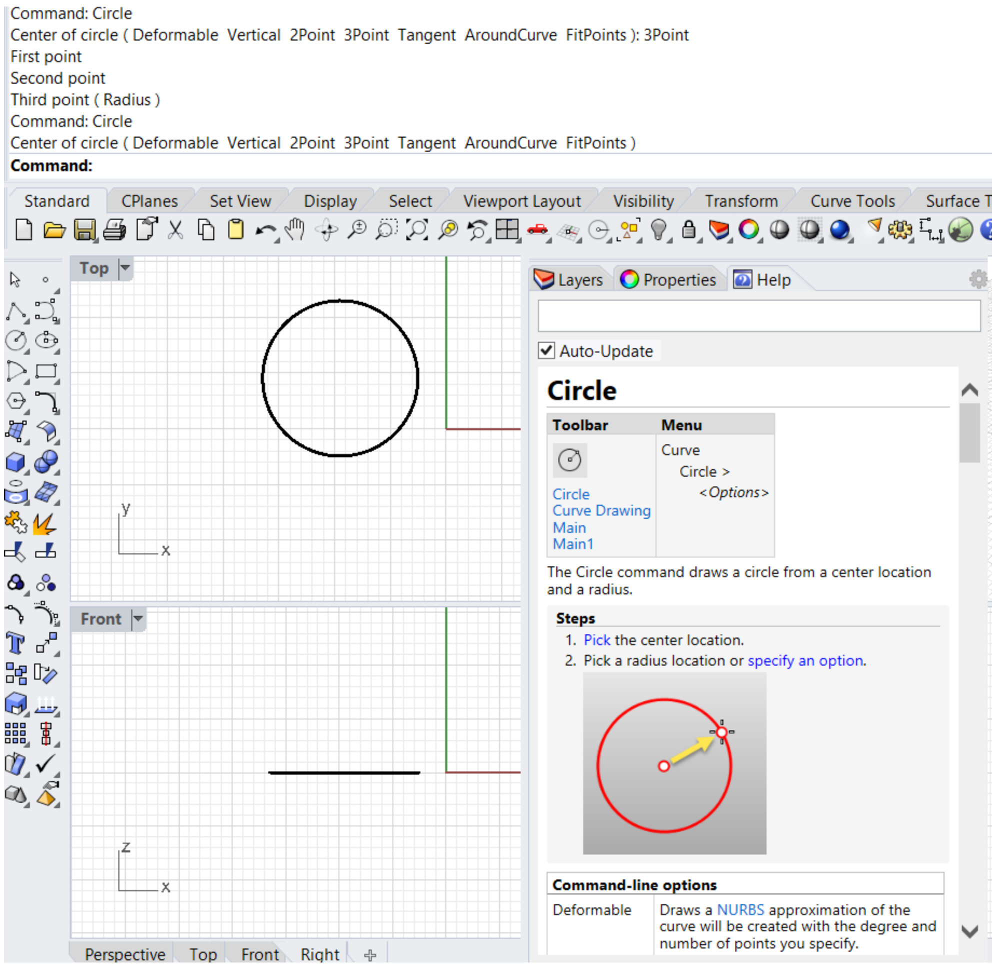




Open panels through panel menu or by clicking on the gear of the tabbed panels or right-mouse-click the margin

Help

The help documents the full functionality of any modeling application. Many times the help is accessible online. The CommandHelp in Rhino displays help topics in a dockable panel. The help offers detailed description of the commands and short videos that show the workflow. If you check Auto-update, the help for the current command displays.



```
Command: Circle
Center of circle ( Deformable  Vertical  2Point  3Point  Tangent  AroundCurve  FitPoints ): 3Point
First point
Second point
Third point ( Radius )
Command: Circle
Center of circle ( Deformable  Vertical  2Point  3Point  Tangent  AroundCurve  FitPoints )
Command:
```

☑ Auto-Update

Circle

Toolbar	Menu
Circle Curve Drawing Main Main1	Curve Circle > *<Options>*

The Circle command draws a circle from a center location and a radius.

Steps

1. Pick the center location.
2. Pick a radius location or specify an option.

Command-line options

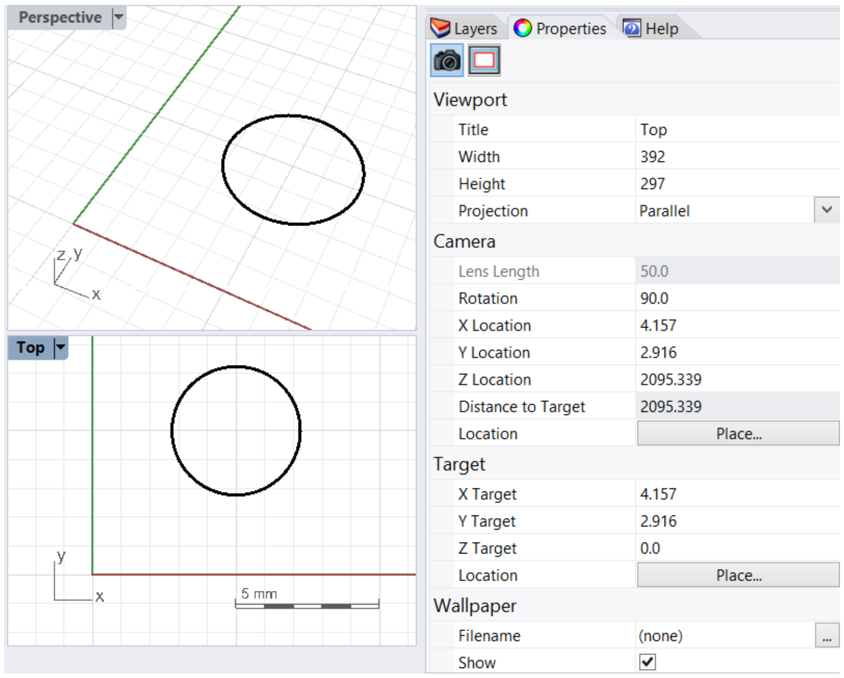
Deformable	Draws a NURBS approximation of the curve will be created with the degree and number of points you specify.

Help can be set to display help for the current command

To access the entire Rhino help document in a browser window, go to “Help > Help Topics” menu, or press F1.

Viewports

The 3D modeling space is contained inside special windows called viewports. You can think of viewports as cameras looking at the same model from different angles. Viewports can be set to parallel, isometric or perspective projection. To assist modeling, viewports include origin, coordinate axes and a grid drawn on a default plane called *construction plane*. Construction planes are the default plane where geometry is drawn unless coordinates are keyed in, or snap to some other geometry.

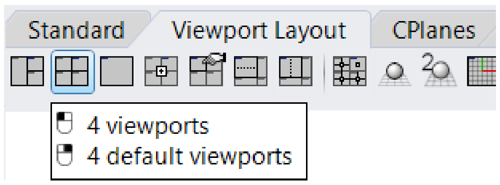


Viewport	
Title	Top
Width	392
Height	297
Projection	Parallel
Camera	
Lens Length	50.0
Rotation	90.0
X Location	4.157
Y Location	2.916
Z Location	2095.339
Distance to Target	2095.339
Location	Place...
Target	
X Target	4.157
Y Target	2.916
Z Target	0.0
Location	Place...
Wallpaper	
Filename	(none)
Show	✓

Viewports specify camera information and projection

You can customize the viewports and their position to suit your preferences. The position of viewports is adjustable. To move and resize viewports, drag the viewport title or borders. You can create new viewports, rename them, and use predefined viewport configurations. To toggle between a small viewport and one that fills the graphics area, double-click the viewport title. You can display the viewport titles in tabs if you prefer. The highlighted tab designates the active viewport. Tabs make it easy to switch between viewports when using maximized or floating viewports. One unique feature about Rhino is that each of the standard four viewports has a different construction plane (except perspective, which uses the Top CPlane by default).

With Rhino, you can open an unlimited number of viewports. Each viewport has its own projection, view, construction plane, and grid. If a command is active, a viewport becomes active when you move the mouse over it. If a command is not active, you must click in the viewport to activate it. You can divide your viewport to have multiple viewports with different projections from Viewport Layouts, then split it either horizontally or vertically. You can go back to the standard four views.

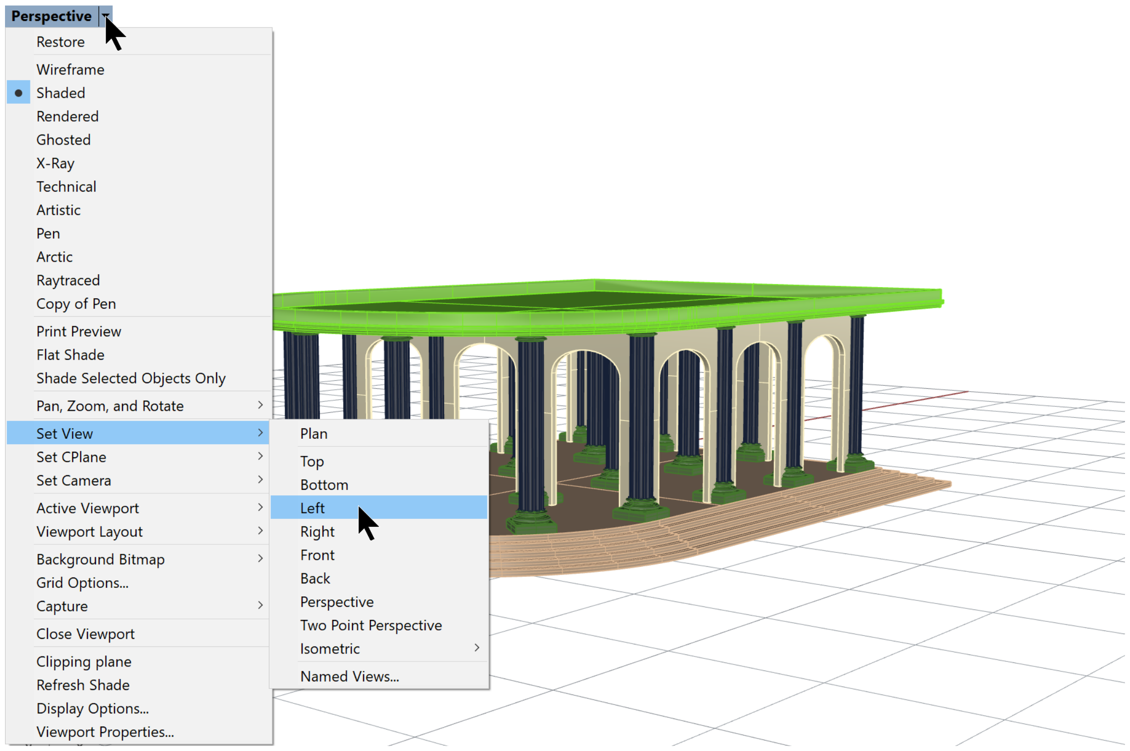


Reset viewport layout to 4 views

Each view in a viewport is seen through a camera lens. The invisible target of the camera is located in the middle of the viewport. You can assign different projection, zoom and camera angles for each of the viewports as needed. To change your view:

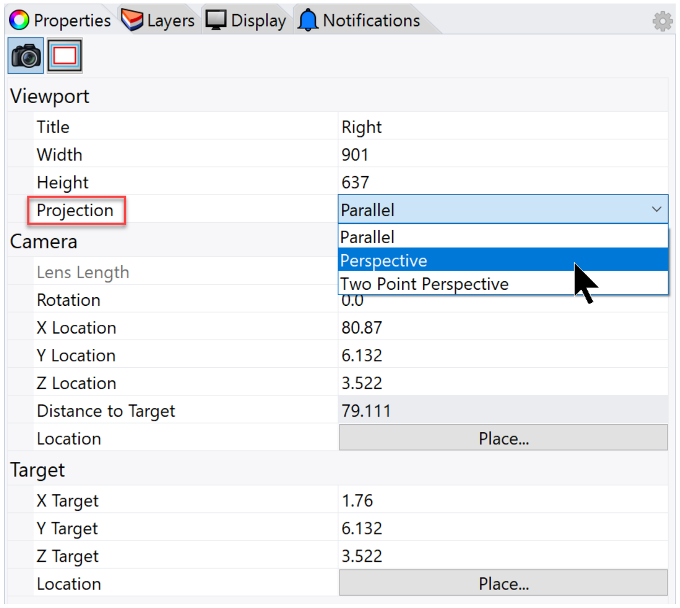
1. Click on the arrow next to the viewport title.
2. Select Set View submenu.
3. Choose your preferred view.

Access these steps from the command line using the "_ SetView" command.



Viewport options

To toggle a viewport between parallel and perspective view: Under the properties panel > Projection > change from Parallel to Perspective to two point perspective.

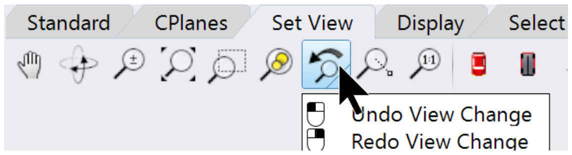


Viewport	
Title	Right
Width	901
Height	637
Projection	Parallel
Camera	
Lens Length	
Rotation	0.0
X Location	80.87
Y Location	6.132
Z Location	3.522
Distance to Target	79.111
Location	Place...
Target	
X Target	1.76
Y Target	6.132
Z Target	3.522
Location	Place...

Parallel
Perspective
Two Point Perspective

Change view projection

There is a separate undo command to undo and redo view changes. Click in a viewport, press your Home or End key on your keyboard to undo and redo view changes, or click on Undo/Redo view changes under the Set View Tab.



View undo and redo

Navigation

Navigating modeling space refers to the ability to reach certain parts of your model and be able to view them up close or from far at any angle and projection. Terms such as “zoom”, “pan”, “parallel projection” are standard in all modeling software. The computer mouse is usually used to navigate models along with specialized commands or tools to help quickly get around. Screen gestures and virtual reality tools allow navigating touch screens and VR.

View navigation

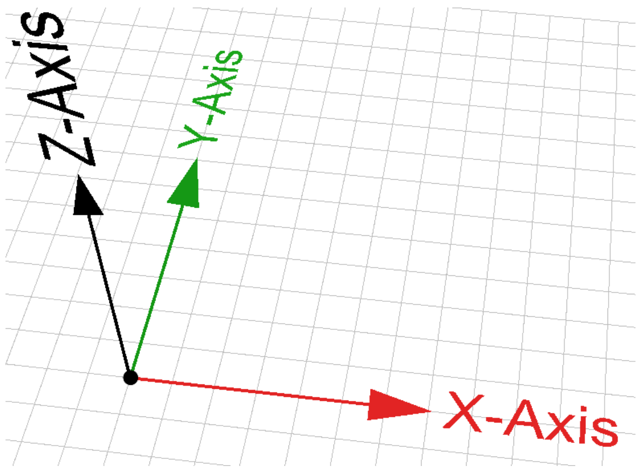
View navigation includes panning, zooming and orbiting. Panning means shifting the view without changing the camera angle. Rhino Pan command supports panning at any projection by holding down the left mouse and moving the mouse to Shift. Otherwise, the simplest way to pan is to hold down the Shift key and drag the mouse with the right mouse button held down (no need for holding the Shift if you are in a parallel projection view). To zoom in and out, use the mouse wheel, or hold down the Ctrl key and drag your mouse up and down with the right mouse button down. Orbiting is only active in perspective views. With the right mouse button down, moving the mouse rotates the view around the center of the view. You can pan, zoom or orbit your view in the middle of a command to see precisely where you want to select or snap to a point.

Function	**Mouse action**
Repeat last command or End command (Enter)	Right mouse button
Selects objects	Left mouse button
Customizable Pop-up menu	Press wheel
Orbit the perspective viewport and Pan Parallel Viewports	Drag with right mouse button down
Pan perspective viewport	Shift + Drag Right mouse button
Zoom in and out	Roll the wheel
Zoom in and out	Ctrl + Right mouse button

Quick reference to Rhino mouse functions

Coordinate system

All modeling software uses a coordinate system to describe the location of points in space. The most common one for architectural applications is the Cartesian coordinates. Rhino uses left-hand Cartesian coordinates with three World planes (XY, XZ, and YZ) that meet the origin. The location of a point is described with three ordered numbers (tuple). The World origin is located at (0,0,0). Points in Rhino can be defined in either absolute or relative coordinates. You can also use polar notation to describe a point.



Rhino uses left-hand Cartesian 3D coordinate system

The following table includes examples of specifying points in Rhino.

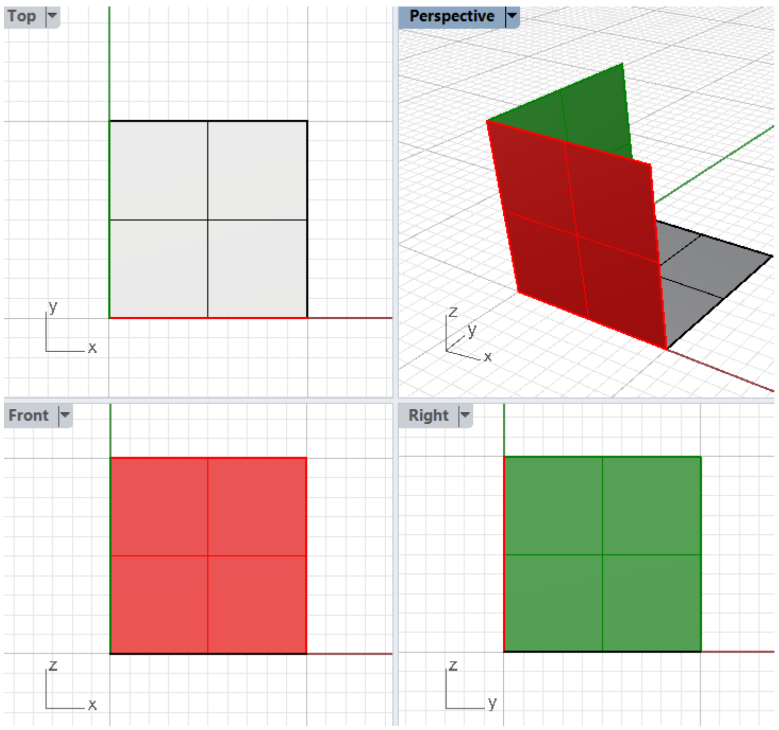
Construction plane coordinates	Enter x,y,z values to place points relative to the current construction plane. You may omit z and y values (they are set to 0 in this case). Examples: 0 = (0, 0, 0) in CPlane coordinates 1,3.5 = (1, 3.5, 0) in CPlane coordinates 1,3.5,6 = (1,3.5,6) in CPlane coordinates
World coordinates	When the construction plane is different from the World planes, you need to specify that your point is relative to World coordinates by preceding the x,y,z by “w”. Examples: w0 = (0, 0, 0) in World coordinates w1,3.5 = (1, 3.5, 0) in World coordinates w1,3.5,6 = (1,3.5,6) in World coordinates
Polar coordinates	Use distance, angle and z value (from CPlane origin). Examples: 17<45 (radius<rotation angle) 17<45,8 (radius<rotation angle,z)
Relative coordinates	This helps locate a point relative to the last point used. Suppose last point used was (1,1,0): **@3,4** = 4,5,0 (move 3 units in the x direction and 4 units in the y direction from the previous point) Alternatively, use “R” or “r” instead of “@” to express relative coordinates: **R3,4** (or **r3,4**) You can also use relative polar coordinates: **r4<45** or **@4<45**

Coordinate notation in Rhino

Construction planes

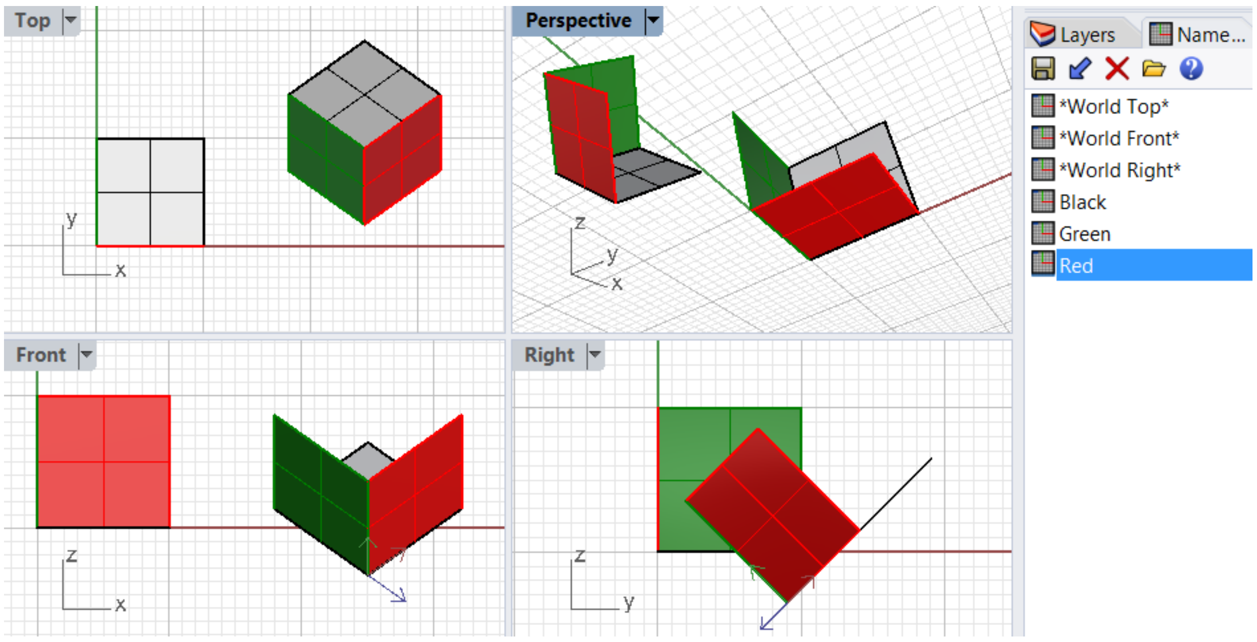
Construction planes (CPlane) are used in modeling software to orient and guide modeling. They align to the World origin and direction by default but can change to be relevant to the model. Creating moving and viewing geometry is highly influenced by the orientation of the construction planes. For example, Points you pick are always on the construction plane unless you use coordinate input, elevator mode, or object snaps.

Rhino default parallel views each have their own construction plane that is parallel to the view itself but share the World origin. Perspective view uses World Top construction plane.



Four Rhino view construction planes

Rhino supplies a rich set of commands to realign the construction plane, and synchronize other viewports to follow if needed. If your model has specific directions that you need to use often, then you should save in the Named construction plane table. This way you can reference quickly and not have to recreate every time you need them.

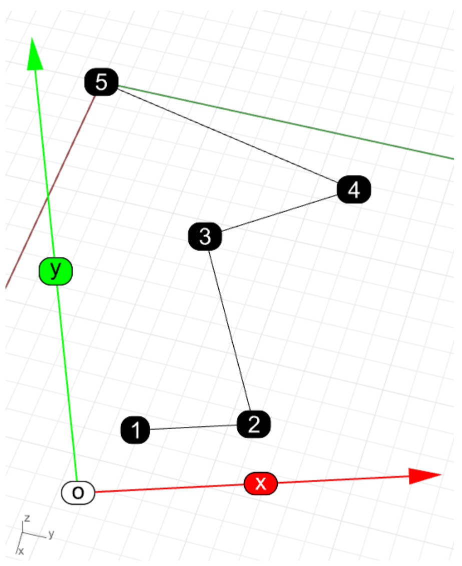


Save construction planes in Perspective view only

Exercise: Accurate modeling

Given a custom orientation is space, create a polyline connecting points 1 to 5 given the following information:

pt **1**: 3 units from custom origin at 45 degrees
pt **2**: 4 units from point 1 in the custom x direction
pt **3**: 10 units from origin at 60 degrees
pt **4**: 10x10 units in the custom x and y directions
pt **5**: world origin



Resource 1: Accurate modeling

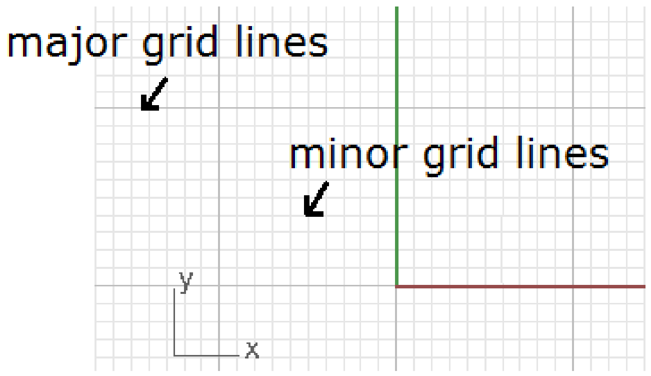
For details about accurate modeling in Rhino, please check the Rhino help on this topic.

Modeling aid and constraints

Interactive digital modeling can be challenging without guides and constraints. They serve as modeling aids to control preferences and pick appropriate locations in 3D space. Modeling guides include grids, coordinate axes and smart tracking mechanisms. Constraints include ortho, geometry constraints, filters, selection constraints, gumball, and construction planes. We will cover some of these, and you can reference the details in the Rhino help file.

Grid and Grid Snap

Grids are a handy visual reference of the orientation and the scale of the modeling space. You show, hide, change the number of grid lines, spacing and intervals in which you can snap to.

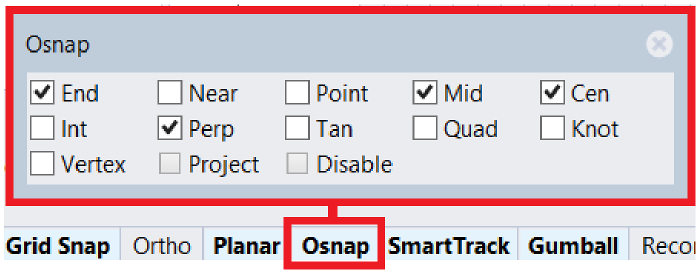


Grids align with construction plane and has major and minor lines

Grid Snap helps snap on grid intersections. You can also toggle Grid Snap on and off by pressing F9 or typing the letter S and pressing Enter. Pressing F7 hides or shows a reference grid in the current viewport of the graphics screen at the construction plane.

Osnap

Object snaps constrain the marker to an exact location on an object such as the end of a line or the center of a circle. These are very important tools to help model accurately and quickly. You can customize Osnap on one or more constraints. To quickly use one constraint, right mouse click on it to disable the others. Another right mouse click on it will restore the previous Osnap state. One special Osnap is "Project". If checked, your geometry will be projected to the view construction plane even if you snap outside it.

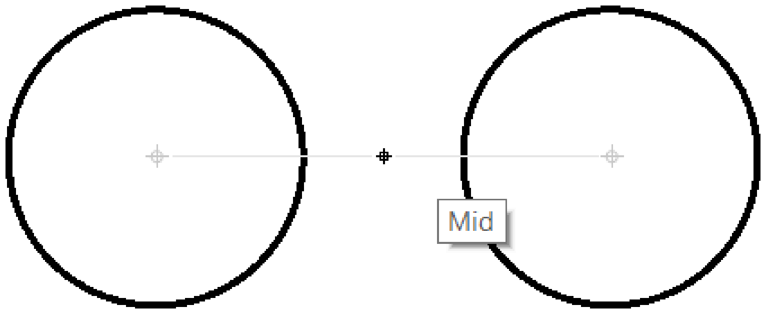

Osnap

✓ End	Near	Point	✓ Mid	✓ Cen
Int	✓ Perp	Tan	Quad	Knot
Vertex	Project	Disable		

Object snaps

Smart Track

Smart Tracking uses temporary reference lines and points that are drawn in the Rhino viewport using implicit relationships among various 3D points, other geometry in space, and the coordinate axes' directions. Smart Track uses OSnap settings to create the reference lines.

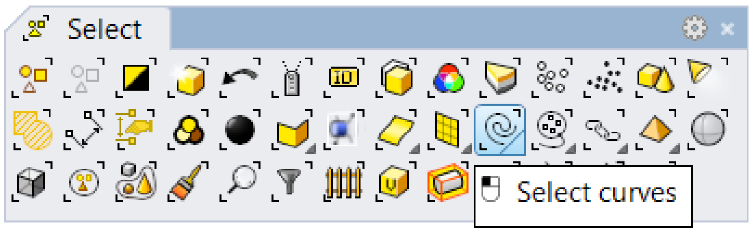


Smart Track to create temporary guides and snap on these guides

It is also possible to set up persistent modeling guides that appear when selecting points. Use AddGuide and RemoveGuide to add and remove these guides.

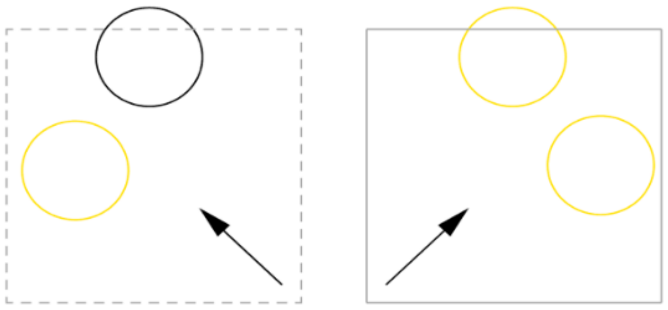
Selection

Rhino supports different ways to select geometry. Selection commands allow selecting all objects in the file that are of a certain type or share certain attributes.



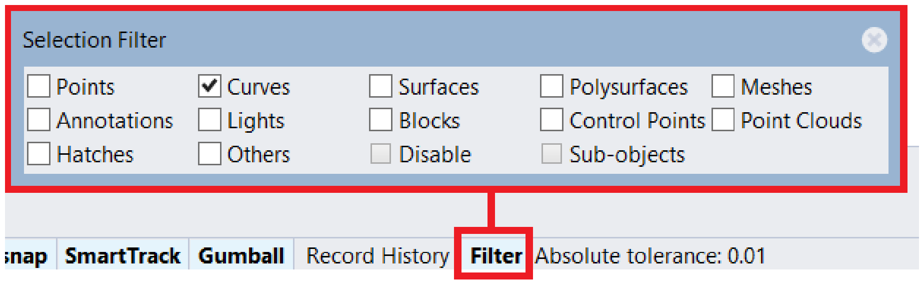
Selection tools

A cross or box window selection is used to select geometry by dragging the mouse (with left button down) around a specific area inside a viewport.



Left: cross window selection from right to left selects objects completely within the window. Right: box window selection from left to right selects all objects wholly or partially within the window

The selection filter allows isolating certain object types. The window selection would then select only the types permitted by the selection filter. For example, if only “Curves” is checked, then no other object type is selected even if they are within the window.

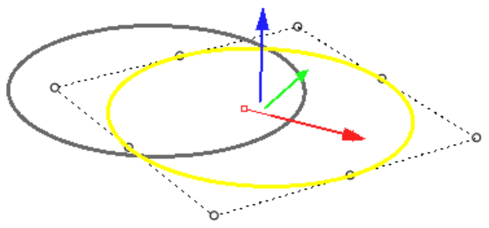
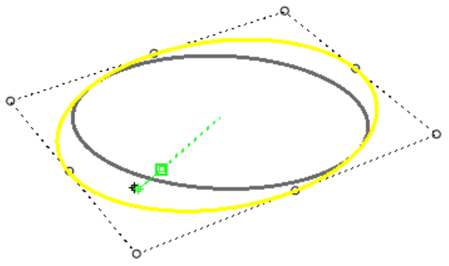
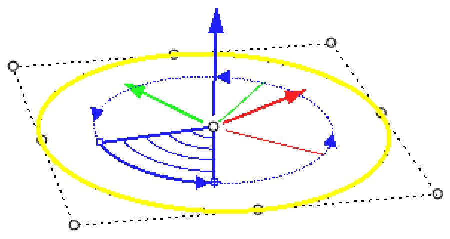

Selection Filter

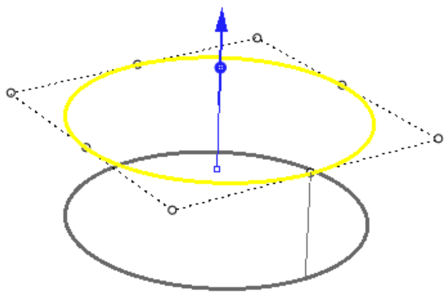
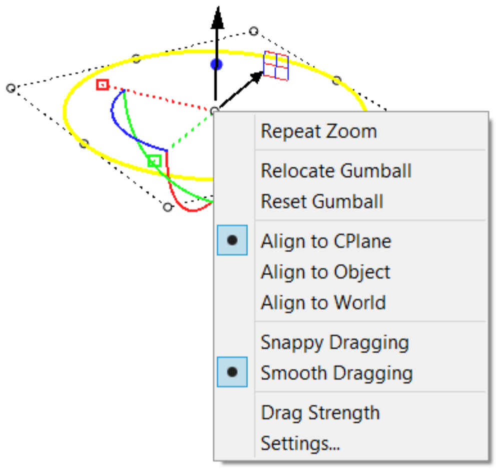
☐ Points	✓ Curves	☐ Surfaces	☐ Polysurfaces	☐ Meshes
☐ Annotations	☐ Lights	☐ Blocks	☐ Control Points	☐ Point Clouds
☐ Hatches	☐ Others	☐ Disable	☐ Sub-objects	

Selection filter to isolate types for selection

Gumball

Gumball displays a widget on a selected object which is used to facilitate direct editing. The Gumball provides move, scale, and rotate transformations around the Gumball origin.

Dragging Gumball arrows move the object in the arrow direction. Note that pressing the Alt key makes a copy of the object.	
Moving the Gumball handles scales the object in the handle direction. If dragged with the Shift key down, the scale becomes uniform in both directions	
Dragging Gumball arcs rotate the object.	

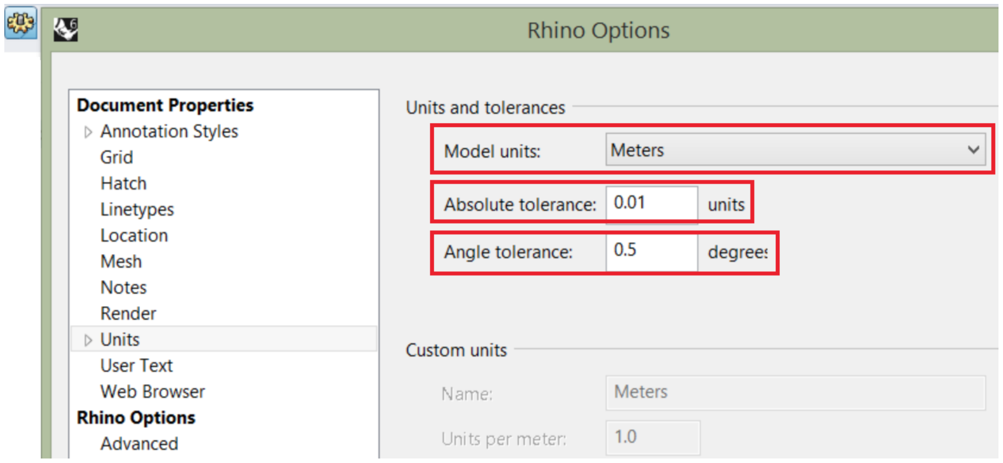
Dragging the arrow handle extrudes the object. With Shift, it extrudes in both directions.	
Other Gumball functions can be accessed with a right mouse click at the center handle.	 Repeat Zoom Relocate Gumball Reset Gumball ● Align to CPlane Align to Object Align to World Snappy Dragging ● Smooth Dragging Drag Strength Settings...

Gumball functions

Gumball
Gumball tutorial: https://vimeo.com/260472052

Units and tolerances

Rhino is always modeled in real full scale. So if your building is measured in meters, you should use “Meters” as your units in Rhino. You can set your model units under Document Properties>Units. It is important that you check your units before starting any modeling. Units affect the scale of your model.



Units and tolerances

Model units: Meters

Absolute tolerance: 0.01 units

Angle tolerance: 0.5 degrees

Custom units

Name: Meters

Units per meter: 1.0

Units and tolerances settings

Rhino modeling involves two types of operations. One is exact, and the other is approximate. For example, when you create a circle, you specify the exact location in space for the center, and exact radius. Now if you project this circle on a NURBS surface, this cannot be an exact operation in NURBS modeling. The amount of approximation is made within some tolerance. You can set that tolerance in the "Absolute tolerance" field. The smaller the tolerance, the tighter the model, but it may involve more complex structure, or take longer to calculate. You should consider what the model will be used for downstream when deciding what tolerance value to use. If the model is only used for visual representation, or renderings, then tolerances can be relaxed. If you will pass the model to be printed or analyzed in some engineering application, you need to check acceptable tolerances in the target application. Angle tolerance is used to evaluate if two curves or two surfaces are tangent within that tolerance.

Set up tutorial
Tolerances and precision in Rhino: https://vimeo.com/85108857

Geometry

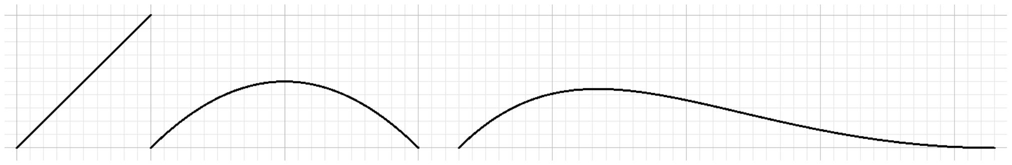
Modelers support various geometry types. The most common types are NURBS and polygon meshes. NURBS is the primary geometry type in Rhino, and it enables high precision free-form modeling in very small tolerances.

NURBS geometry

Rhino uses NURBS curves and surfaces as its primary method to represent geometry. NURBS is an accurate mathematical representation of curves that is highly intuitive to edit.

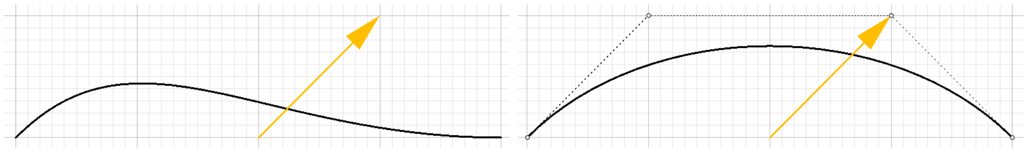
NURBS Curves

It is effective to represent free-form curves using NURBS, and the control structure makes it easy and predictable to edit. The control structure of a curve consists of a list of points that are used to construct the curve and also to edit it. Rhino *Curve* command draws a NURBS curve by default. For example, here are the steps to create a curve.



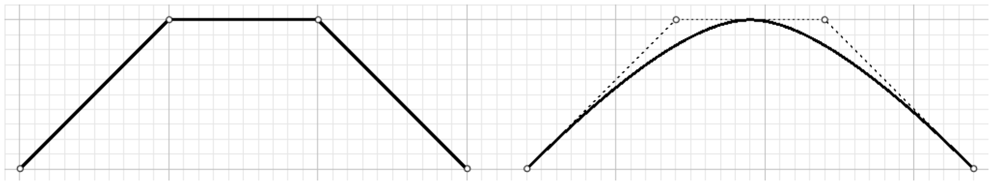
NURBS curves

At any point after you create the curve, control points can be turned on (use *PointsOn* command) and dragged to adjust the curve interactively (use *PointsOff* command, or *Esc* to turn off control points when done), as in the following.



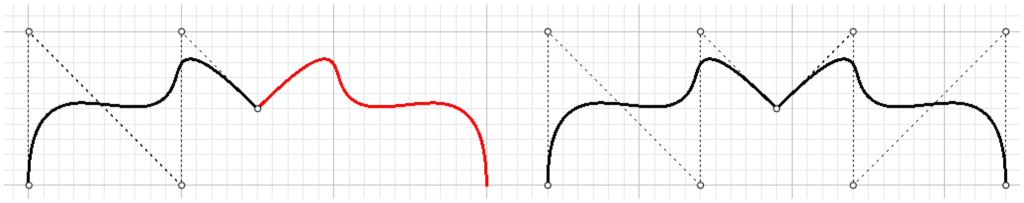
Curve editing by moving control points

Other than control points, you need to pay attention to the curve degree. The degree determines how smooth a curve is. In simple terms, degree 1 creates polylines, degree 2 creates arcs, and degree 3 creates smooth curves. Higher degrees simply increase the smoothness and are not very commonly used. The degree of the curve in the example above is set to 3 (the default when you run *Curve* command), but there is a *degree* option that you can change. Here is how our curve looks when set to degrees 1 and 2 using the same control structure.



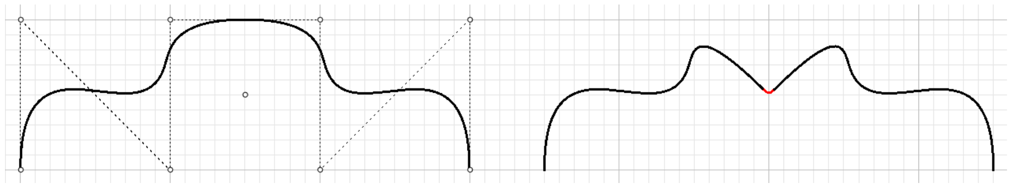
Curves degree affects how smooth they are

It is possible to join two curves together if their end points coincide. The new curve is called polycurve (or polyline if consists of lines).



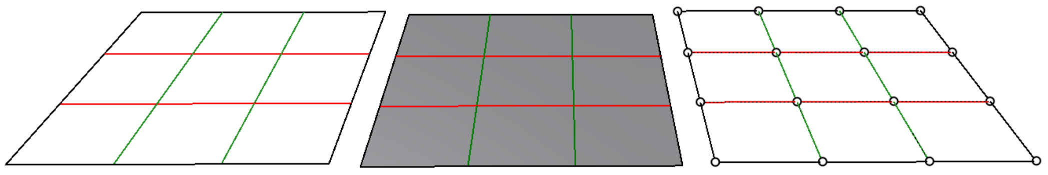
Joining curves create kinks

Notice that joining smooth NURBS curves together creates a “kink”. Deleting the control point directly on the kink fixes the curve continuity, but you can also join curves smoothly using commands such as BlendCrv.

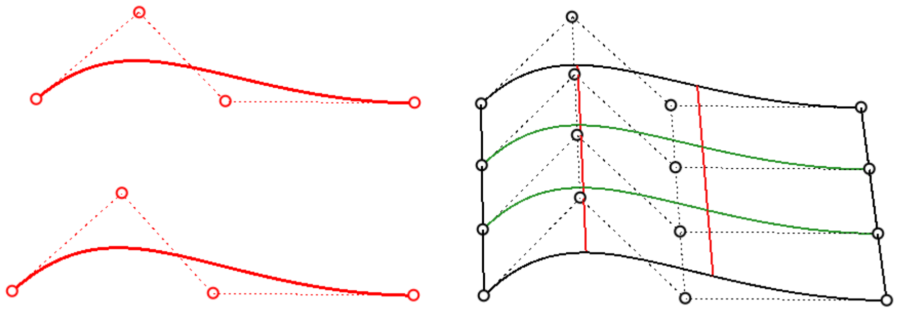

Delete control points or join curves with a blend helps create smooth curves from two curves input

NURBS surfaces

Surfaces in Rhino are created using NURBS. You can think of them as a network of NURBS curves in two directions. They are infinitely thin, infinitely flexible, mathematically defined digital membranes. Surfaces are represented on screen by either some outline curves plus some interior curves, called isocurves, or by a shaded picture which makes a surface appear to have some substance and to show light and shading. How surfaces are painted on the screen is dependent on the display mode in the viewport, and does not affect the surface in any way. The important thing to remember about surfaces is that they are defined with great precision at every point. They are not approximations.

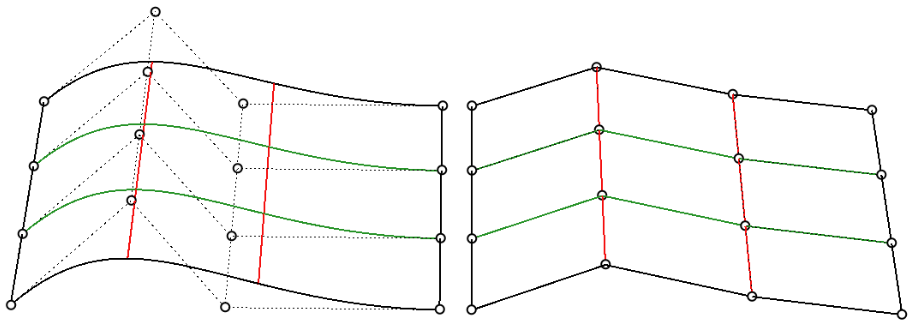

NURBS structure as a grid of control points in u and v directions

Two NURBS curves can be turned into a surface using Loft command. Notice that the control structure (the collection of control points) is very similar to the curves used to make the lofted surface. In the same manner, a NURBS curve can be edited by dragging control points, NURBS surfaces are modified when dragging control points.

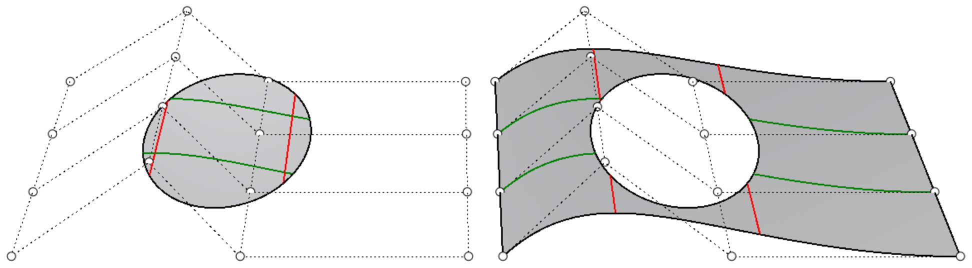

Loft two curves to create a surface preserves input NURBS curve structure when possible

There are a few concepts associated with NURBS surfaces are useful to remember. NURBS surfaces are rectangular with two main directions referred to as the “u” and “v” directions. The two directions do not have to be linear, so surfaces may bend in space. It is always useful to change the color of isocurves going in u-direction (red) from those going in v-direction (green). It keeps you aware of the general structure of your surface.

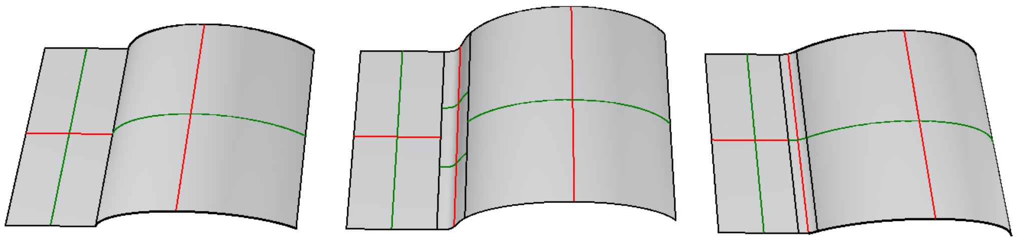
There is the idea of a “degree” in NURBS surfaces, and it defines the level of smoothness of the surface. With degree 1 surfaces, you end up with a faceted surface that has creases surrounding each unit area surrounded by adjacent control points. The higher the degree, the smoother the surface, but you typically need more control points to achieve it. Commonly used degrees are 1, 2, 3 and 5.


Surface degree affects smoothness

NURBS surfaces typically have a rectangular boundary. If the surface is non-rectangular, then it is likely to be “trimmed”. Trimming involves defining a boundary (using curves). The underlying surface remains rectangular, and you can always untrim the boundary to go back to the original surface. Adding holes to a surface is the same as adding an irregular boundary. It involves trimming an inner boundary. Again, the underlying structure of the NURBS surface remains intact in all cases.

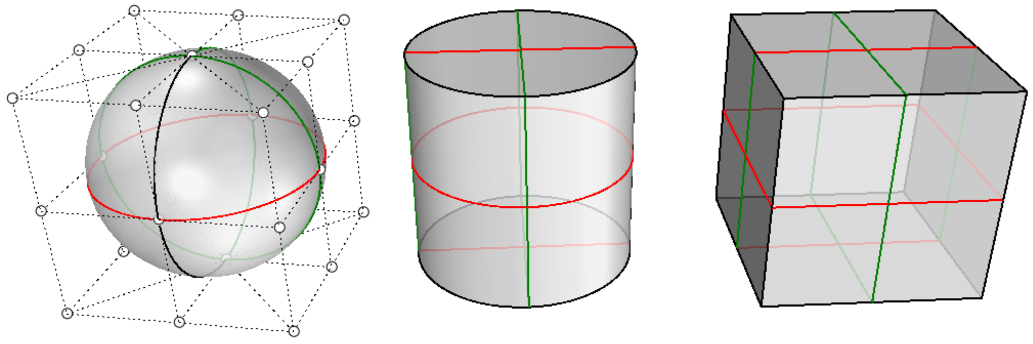

Trimming surfaces does not change the underlying NURBS structure

It is possible to join surfaces together to make a bigger object. That happens if the two surfaces can be joined along at least one edge with the model tolerance. If you don’t plan carefully, the two surfaces might not connect smoothly. There are tools in Rhino to blend surfaces with the desired smoothness (or what is technically called continuity). One example is to use BlendSrf command. It is also possible to blend or round the edges connecting two surfaces stitched together using a command such as FilletEdge and BlendEdge.

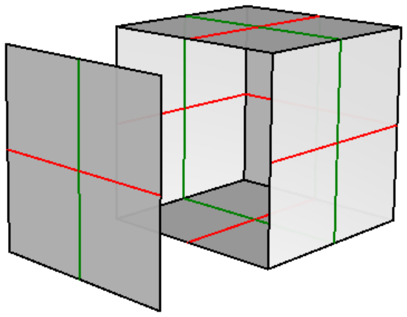

Joining two surfaces with or without blending

Solid geometry

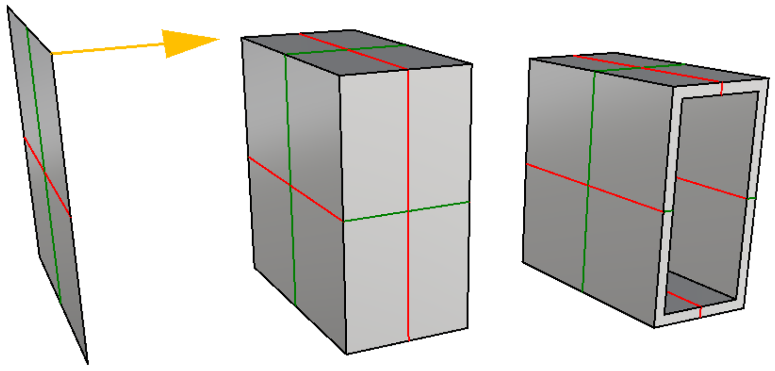
Rhino NURBS surfaces are infinitely thin; they have zero thickness. Enclosing a volume can be achieved with one or more surfaces stitched together as one object. Examples of single surface breps are spheres and ellipsoids. One problem with such surfaces is that they introduce "singularity", when one or both ends of the NURBS surface collapse in one point. Singularities are not desirable in modeling because they might not intersect well, for example. The Solid menu includes all primitive forms such as box, sphere, cylinder, etc.


Solid primitives

The more common way to create a closed polysurface or solid in Rhino is to join enough single surfaces together to enclose a space producing "boundary representation", or brep for short. A box is an example of this type of object. We call these objects solids, but it is important to remember that there is nothing inside them. They are volumes in space enclosed by the infinitely thin surfaces. If you remove one side of a box and look inside you will see the backsides of the five surfaces.

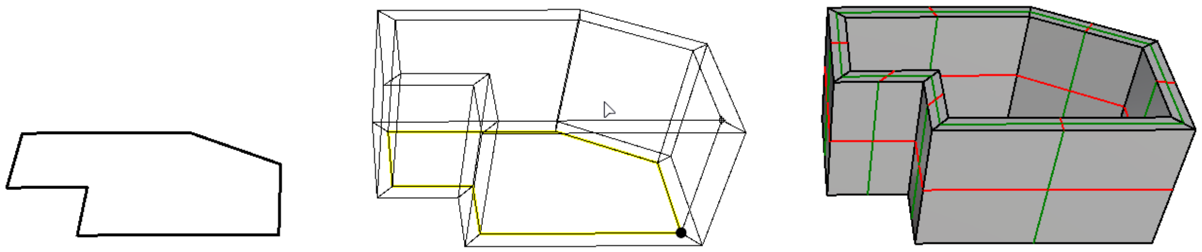

Solids are made out of surfaces joined together to enclose a closed volume

You can also create breps by extruding or offsetting surfaces. Shell command is another example of turning an open surface or polysurface into a solid with thickness.



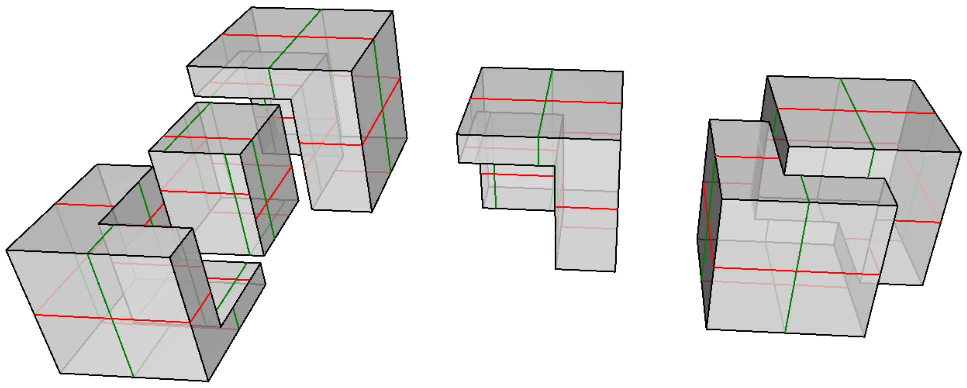
Solids from surfaces and polysurfaces

You can create solids from curves as input using commands such as Slab.



Slab command starts with a curve, offset, then extrude normal to the curve plane

There are modeling operations that work only with solids; most importantly the Boolean Operations. Commands such as BooleanUnion, BooleanDifference, and BooleanSplit ensure that the results remain solid.



Solid Boolean operations. BooleanSplit (left), BooleanDifference (center), BooleanUnion (right)

Extrusion geometry

Another object type that is related to a polysurface and a solid is the lightweight extrusion object. Lightweight extrusion objects use less memory, mesh faster, and save smaller than the traditional polysurfaces.

In models containing large numbers of extrusions represented by traditional polysurfaces, performance can be sluggish due to the relatively high demand on resources. If the same objects use a lightweight extrusion type, the model is more responsive and leaves plenty of memory available.

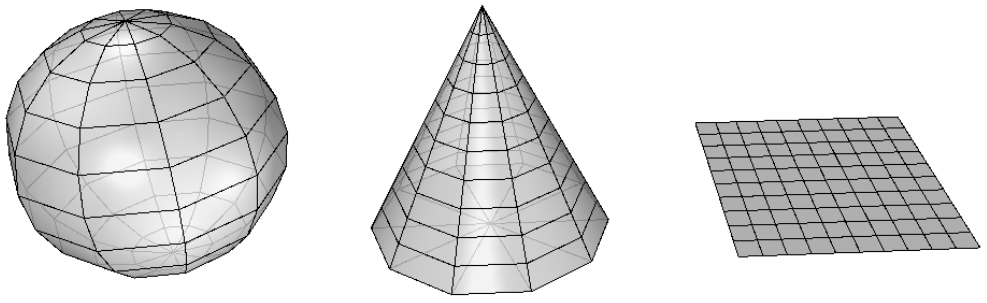
In Rhino 5 commands like Box, Cylinder, Pipe, and ExtrudeCrv create lightweight extrusion objects by default. Note that editing these objects might result in converting them to Breps (typically when the operation result has no reasonable extrusion replacement that can be found).

Mesh geometry

Rhino creates, edits, and otherwise uses polygon meshes. Polygon meshes are sometimes used to depict the same type of objects as surfaces, but there are important differences. Polygon meshes consist of a number, sometimes a very large number, of points in space connected by straight lines. These straight lines form closed loops of three or four sides, that is, polygons.

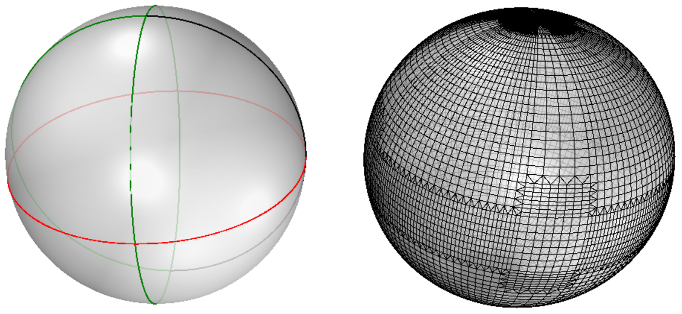
One important thing to know about polygon meshes is that the 3D data only exists for these points, or mesh vertices; the space between these points is not considered. Dense meshes are more accurate than very loose meshes, but not as accurate as surfaces.

Meshes in Rhino come in two forms, either as the primary object, or as part of the NURBS objects. The Rhino mesh menu allows creating both closed and open meshes.



Mesh primitives

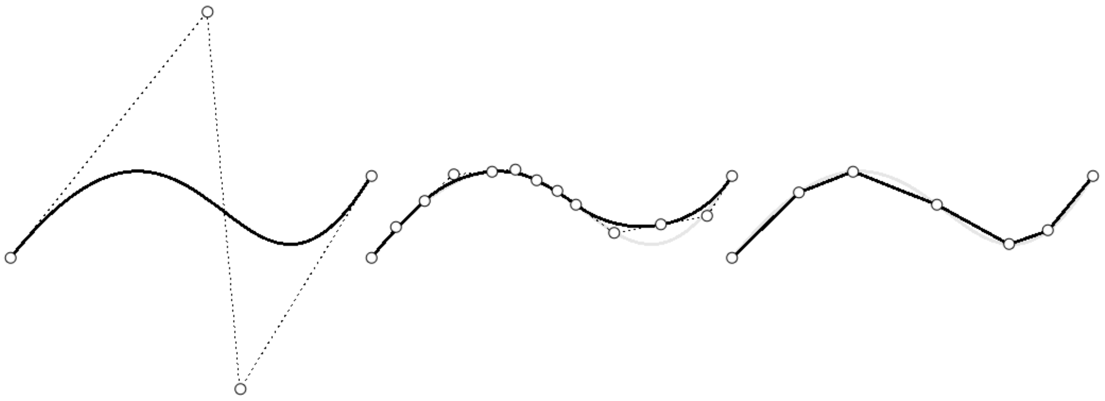
If you look at a surface in a shaded viewport, you actually see a polygon mesh derived from the surface to make a nice image on the screen (you can only shade meshes). All breps, which are NURBS surfaces joined together in one object, have a render mesh attached to them to help apply shading and rendering. The render mesh can be made more or less accurate based on speed and resolution requirements. Note that it is relatively easy to go from smooth breps to any resolution mesh, but it is hard to turn a faceted mesh into a smooth brep.


Render Meshes are generated for NURBS to be able to shade Brep objects on the screen

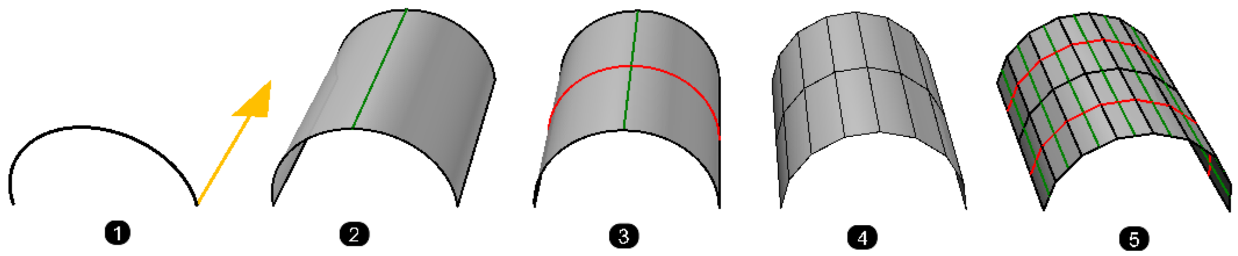
Typically, you will export mesh data for rapid-prototyped parts. Deriving accurate meshes from surface models is important. Rhino has several tools to help accomplish this. Rhino supports mesh operations to edit meshes, boolean and repair.

Transition between geometry types

Rhino supports converting between geometry types. For curves, a smooth NURBS curve can be turned into arcs or lines using Convert command.


Convert from NURBS curves to Arcs and Lines

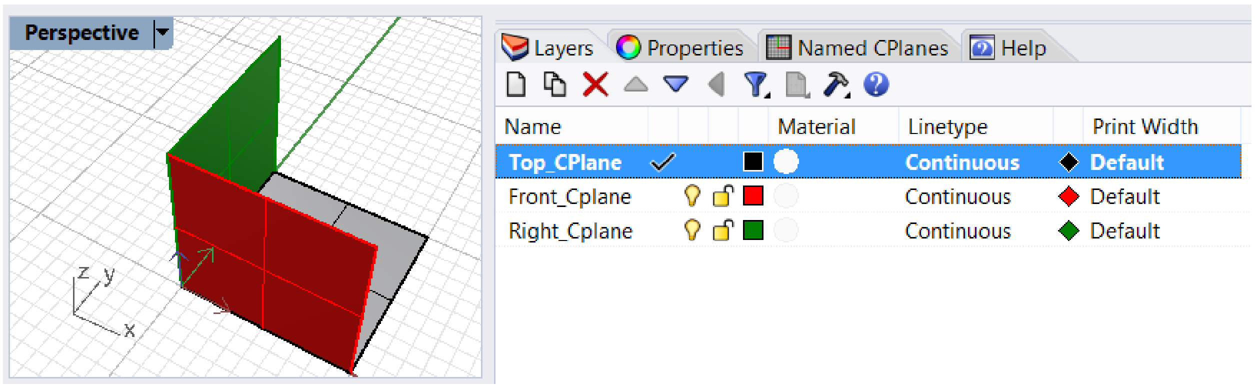
For surfaces, there are three main geometry types: extrusions, NURBS and meshes. There are a number of commands in Rhino to convert between surface types.



Convert geometry: (1) Curve (2) Extrusion surface (3) Convert to Brep (4) Mesh (5) Mesh to NURBS

Geometry organization and navigation

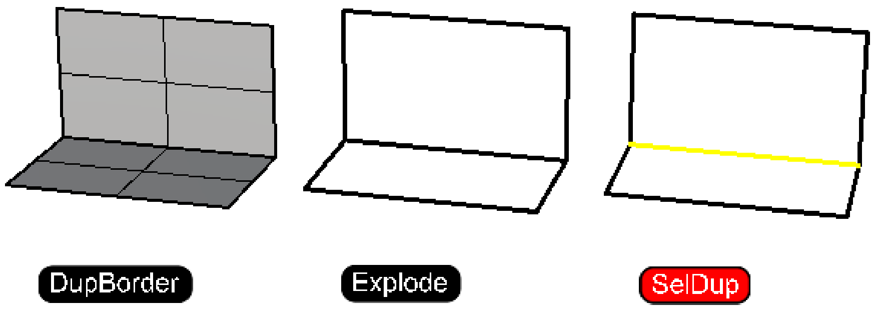
Designers typically organize their model geometry and data in layers. This helps isolate and group data in one place. For example, you might put all your lights in one layer, site geometry in another and so on. Other than grouping data, layers allow you to hide, select all objects, or assign attributes. For example, you can assign color, lineweight or material to the layer instead of objects. There is extensive information about layers in the Rhino help file.



Name					Material	Linetype	Print Width
Top_CPlane	✓			■	●	Continuous	◆ Default
Front_Cplane		💡	🔓	🟥	○	Continuous	◆ Default
Right_Cplane		💡	🔓	🟩	○	Continuous	◆ Default

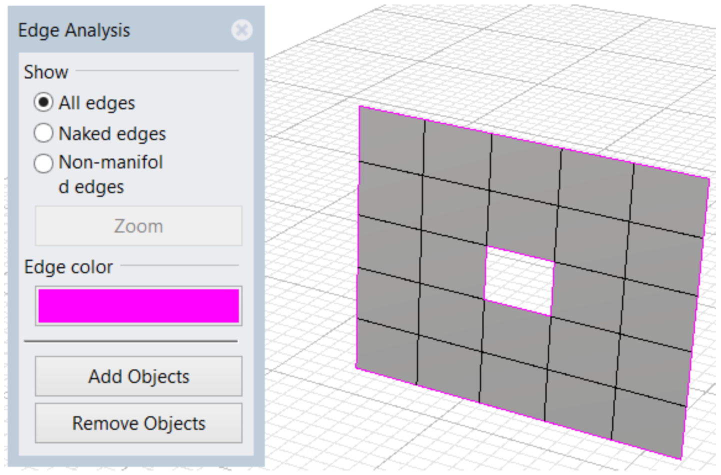
Layers to group data and assign general attributes

Selection tools by object type or attributes are other important tools to help navigate your model. For example, you might need to hide all points in the model. You can run SelPt, then Hide commands. You might also want to delete all duplicate geometry in a model. The command SelDup selects geometry that perfectly overlap (within your document tolerance), then runs the Delete command. It is also possible to select objects with a certain name or color. Knowing how to use selection commands can improve your productivity.



Selecting duplicate geometry using SelDup

There are also specialized tools to identify specific information about your model. For example, you might need to zoom to a naked edge in a dense mesh (ShowEdges).

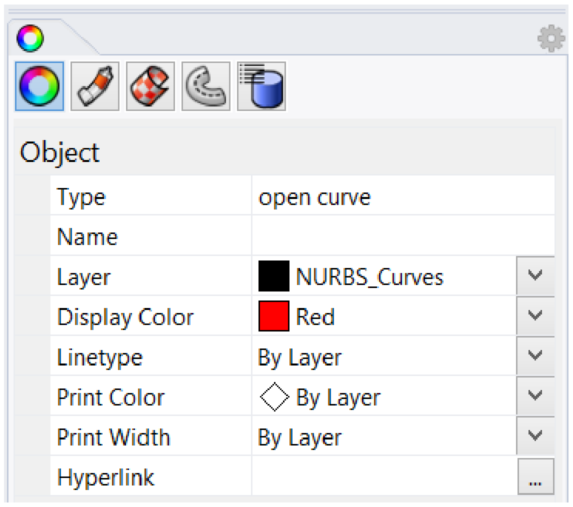


ShowEdge is a specialized geometry navigation tool

Data

Modeling objects consist of two parts: geometry and data. You can think of data as information that describes the geometry attributes such as name, color, material, linewidth, group affiliation, etc. Data also includes custom information specific to the modeler or the workflow.

Rhino objects have attributes that can be accessed and set directly using the properties panel. Each geometry type has a different set of data and attributes associated with it, and they all appear as icons under the properties main tab.

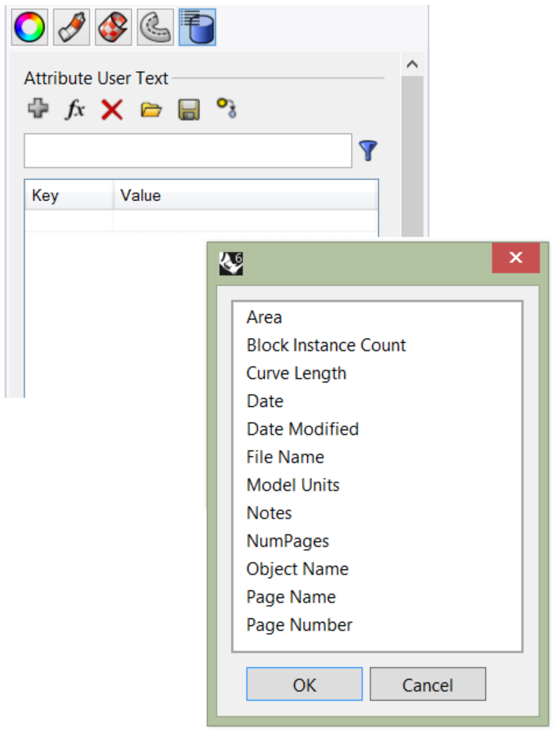


Object

Type	open curve
Name	
Layer	■ NURBS_Curves
Display Color	🟥 Red
Linetype	By Layer
Print Color	◇ By Layer
Print Width	By Layer
Hyperlink	...

Curve properties

Users can attach custom data to objects using a user string. You can add functional attributes such as length or area, but you can also create your own key and value that is specific to your modeling workflows. For example, you may tag all your roof panels with the "Area" Key and the area of each panel is calculated and attached. It also changes dynamically when the panels are scaled. Then you can select all curves that match a certain area.



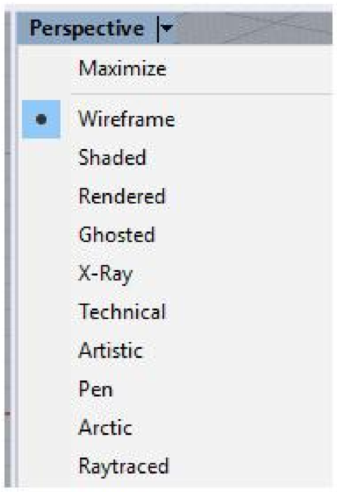
User text

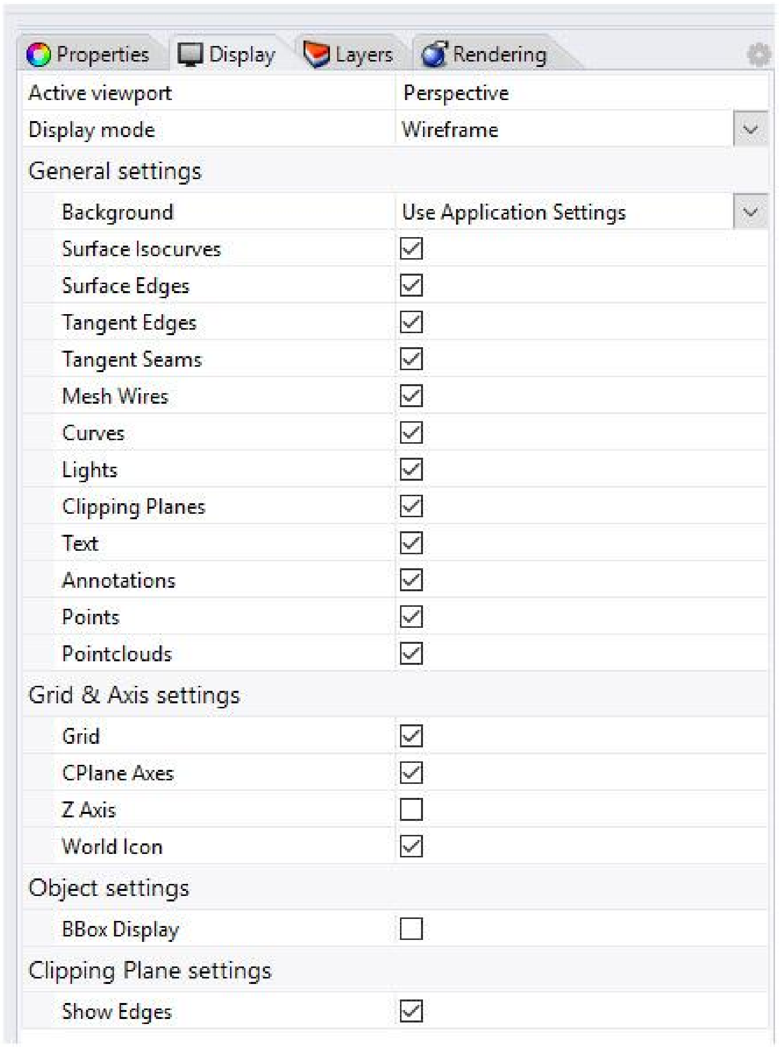
Rhino plugins can access objects' user data and attach any information that can inform how certain objects display or behave.

Visualization

Showing your design work to others is an integral part of any creative workflow. Bringing a team together on the direction of a project and presenting design options to a client are required throughout the design process in many industries. Rhino includes a variety of options for visualizing and communicating your work to others. Display modes, interactive rendering and a myriad of available third-party plugins for rendering ensure you can capture and communicate your design intent as you create.

Display modes in the Rhino viewport can be selected via the viewport drop-down menus or by way of the Display panel at the right of the Rhino UI.

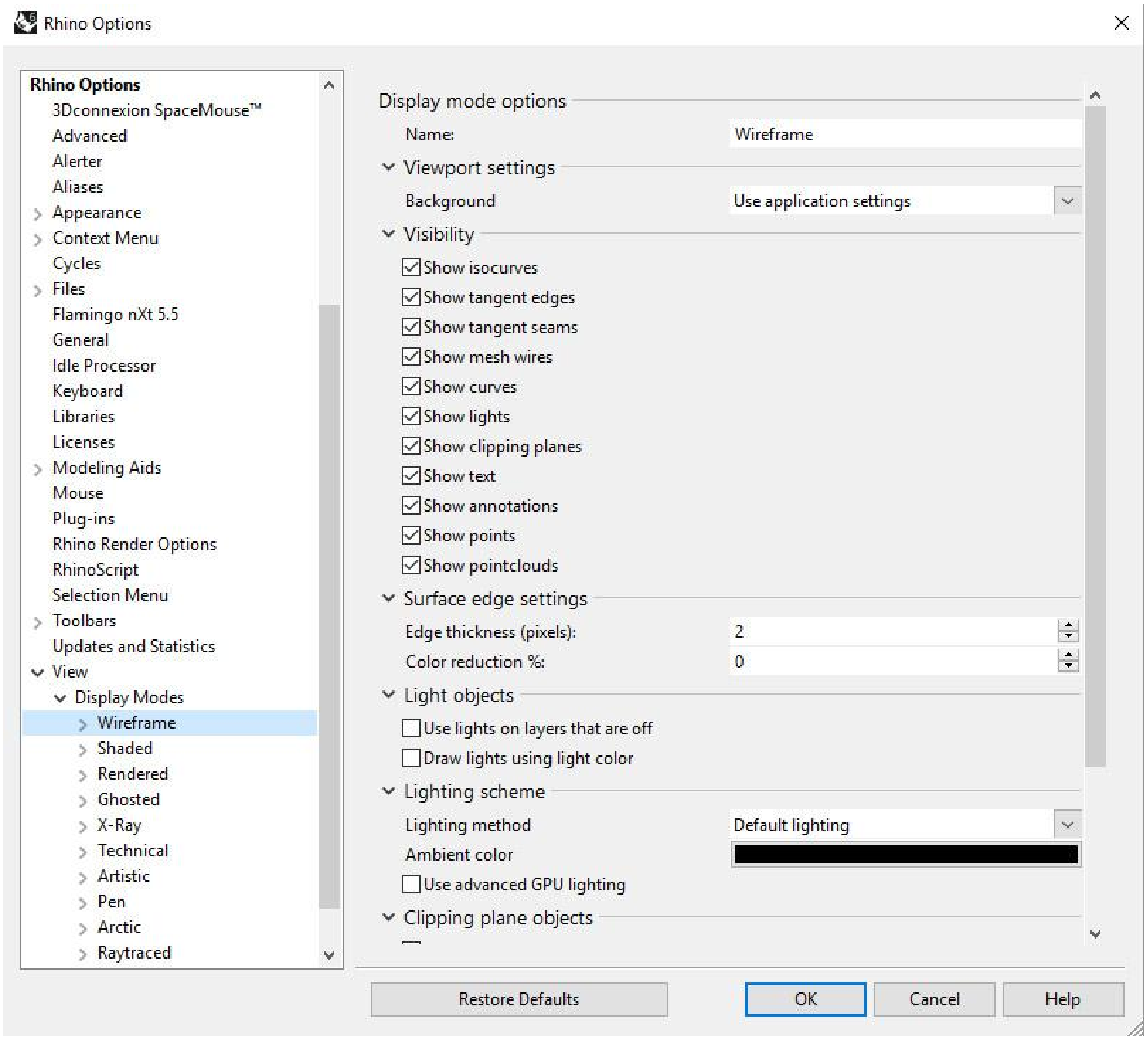




Active viewport	Perspective
Display mode	Wireframe
General settings	
Background	Use Application Settings
Surface Isocurves	✓
Surface Edges	✓
Tangent Edges	✓
Tangent Seams	✓
Mesh Wires	✓
Curves	✓
Lights	✓
Clipping Planes	✓
Text	✓
Annotations	✓
Points	✓
Pointclouds	✓
Grid & Axis settings	
Grid	✓
CPlane Axes	✓
Z Axis	☐
World Icon	✓
Object settings	
BBox Display	☐
Clipping Plane settings	
Show Edges	✓

Display models in Rhino

Display modes can also be created, customized and saved through Rhino Options > View > Display Modes.



Display mode options

Name: Wireframe

Viewport settings

Background: Use application settings

Visibility

- [✓] Show isocurves
- [✓] Show tangent edges
- [✓] Show tangent seams
- [✓] Show mesh wires
- [✓] Show curves
- [✓] Show lights
- [✓] Show clipping planes
- [✓] Show text
- [✓] Show annotations
- [✓] Show points
- [✓] Show pointclouds

Surface edge settings

Edge thickness (pixels): 2

Color reduction %: 0

Light objects

- [ ] Use lights on layers that are off
- [ ] Draw lights using light color

Lighting scheme

Lighting method: Default lighting

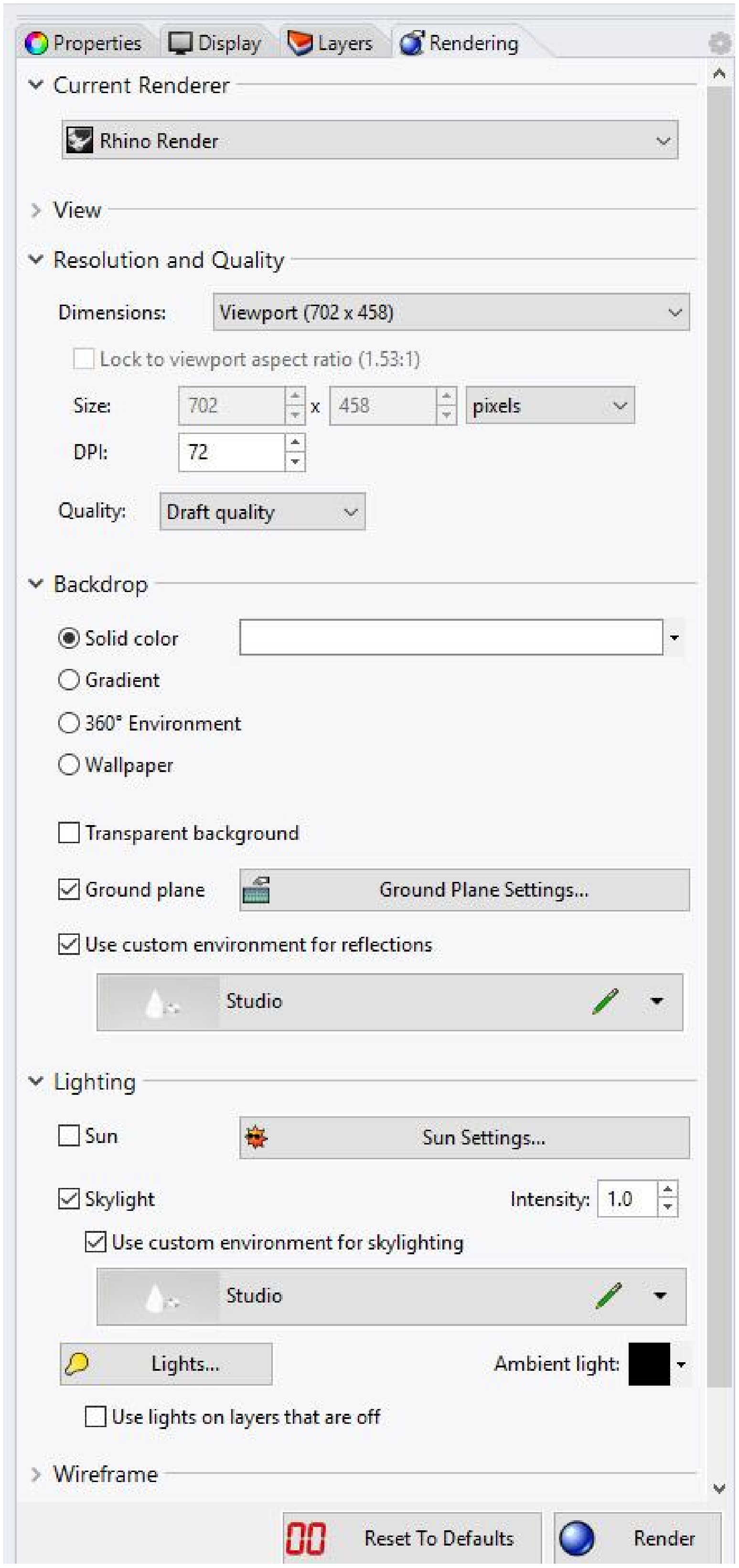
Ambient color: ■

- [ ] Use advanced GPU lighting

Clipping plane objects

Display models customization through Rhino Options

Rendering controls for the default Rhino Render as well as the interactive rendering display mode called Raytraced can be accessed in the Rendering panel. If you are using a separately installed rendering plugin in Rhino, refer to that plugin's documentation for UI locations. The Render drop down menu > Current Renderer flyout can be used to switch between rendering plugins.



Current Renderer

Rhino Render

View

Resolution and Quality

Dimensions: Viewport (702 x 458)

☐ Lock to viewport aspect ratio (1.53:1)

Size: 702 x 458 pixels

DPI: 72

Quality: Draft quality

Backdrop

● Solid color

○ Gradient

○ 360° Environment

○ Wallpaper

☐ Transparent background

☑ Ground plane — Ground Plane Settings...

☑ Use custom environment for reflections

Studio

Lighting

☐ Sun — Sun Settings...

☑ Skylight — Intensity: 1.0

☑ Use custom environment for skylighting

Studio

Lights... — Ambient light: ■

☐ Use lights on layers that are off

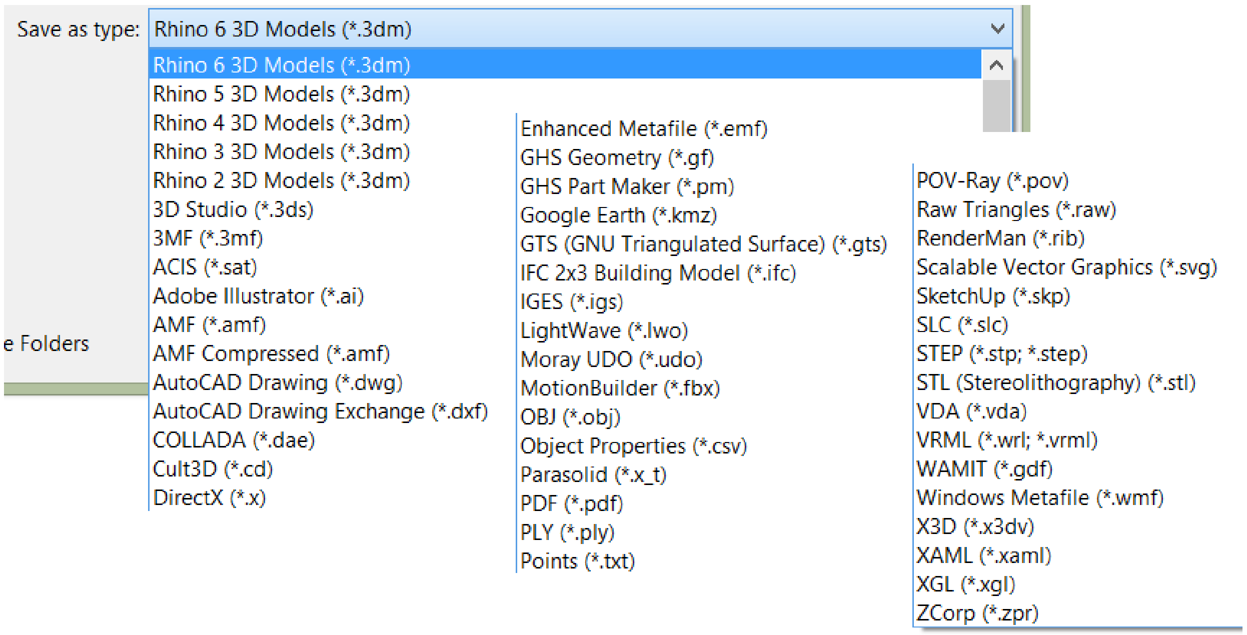
Wireframe

Rendering panel

Files

3D modeling applications save geometry and data to a file specific format. When this file format is 'open', other applications can support importing data from that file format. This helps promote interoperability among different applications. Rhino publishes its 3DM file format in OpenNURBS allowing other applications to fully support the format.

Some applications support a limited number of geometry types, or specialized object types. This makes interoperability more challenging, and some data is bound to be lost in the conversion. Rhino pays great attention to file I/O and supports many different import and export formats, making it possible to model in Rhino and then export your model to downstream processes, or import models from other software applications into Rhino. For a complete list of import and export file types refer to the Rhino Help > Contents > File I/O > File Formats.



Save as type: Rhino 6 3D Models (*.3dm)

Rhino 6 3D Models (*.3dm)
Rhino 5 3D Models (*.3dm)
Rhino 4 3D Models (*.3dm)
Rhino 3 3D Models (*.3dm)
Rhino 2 3D Models (*.3dm)
3D Studio (*.3ds)
3MF (*.3mf)
ACIS (*.sat)
Adobe Illustrator (*.ai)
AMF (*.amf)
AMF Compressed (*.amf)
AutoCAD Drawing (*.dwg)
AutoCAD Drawing Exchange (*.dxf)
COLLADA (*.dae)
Cult3D (*.cd)
DirectX (*.x)
Enhanced Metafile (*.emf)
GHS Geometry (*.gf)
GHS Part Maker (*.pm)
Google Earth (*.kmz)
GTS (GNU Triangulated Surface) (*.gts)
IFC 2x3 Building Model (*.ifc)
IGES (*.igs)
LightWave (*.lwo)
Moray UDO (*.udo)
MotionBuilder (*.fbx)
OBJ (*.obj)
Object Properties (*.csv)
Parasolid (*.x_t)
PDF (*.pdf)
PLY (*.ply)
Points (*.txt)
POV-Ray (*.pov)
Raw Triangles (*.raw)
RenderMan (*.rib)
Scalable Vector Graphics (*.svg)
SketchUp (*.skp)
SLC (*.slc)
STEP (*.stp; *.step)
STL (Stereolithography) (*.stl)
VDA (*.vda)
VRML (*.wrl; *.vrml)
WAMIT (*.gdf)
Windows Metafile (*.wmf)
X3D (*.x3dv)
XAML (*.xaml)
XGL (*.xgl)
ZCorp (*.zpr)

The Rhino export file formats

Part II: Modeling workflows

Workflows in architecture are typically unique to each building, and are highly dependent on the people and requirements involved. The overall flow contains specific tasks. The following sections introduce a series of tutorials that are task-based. They are grouped into four sections: Setting up, conceptual design, detailed design and prototyping workflows. All tutorials use mostly Rhino, but also Grasshopper and plugins.

Modeling set up

There are a few steps to take before each new project to ensure proper organization and settings. For this training, you will do the following:

- Create a folder for your model and give a descriptive name. As the project grows, you may add new folders and subfolders. Also, make sure that files are named to show in desired order, and describe the content for quick reference.
- Create a new template with appropriate document properties, options, layers and reference geometry.
- Create your model as close to the World origin as possible.
- Generate scaffolding or reference envelope, guides, and construction planes to match model orientation.
- Set up views and snapshots for effective visualization.

Project template

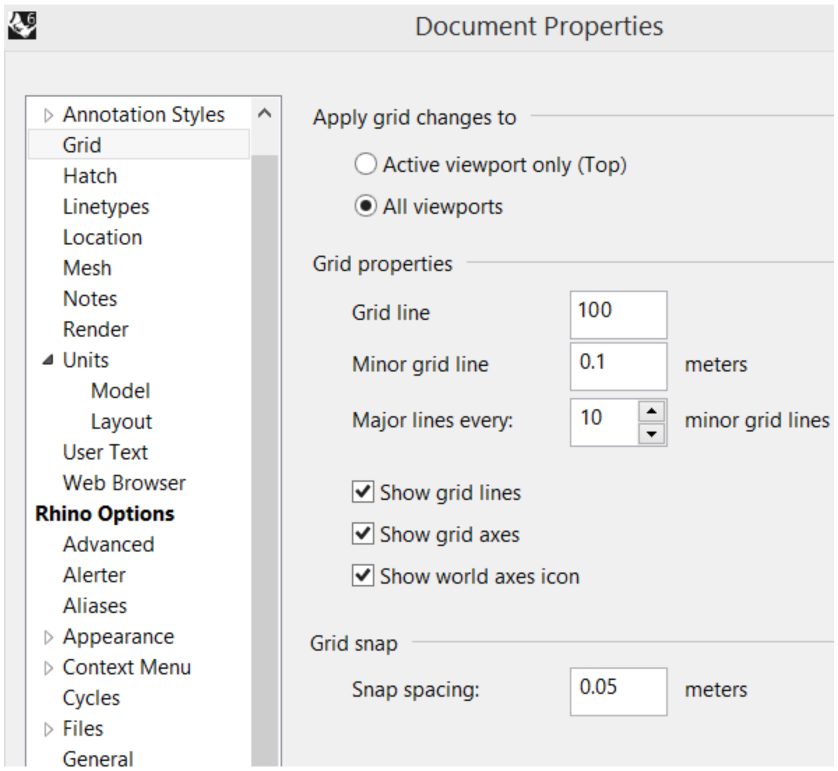
A template is a Rhino model file you can use to store basic settings. Templates include all the information that is stored in a Rhino 3dm file: objects, blocks, layouts, grid settings, viewport layout, layers, units, tolerances, render settings, dimension settings, notes, and any setting in document properties.

You can use the default templates that are installed with Rhino, or save your own templates to base future models on. You will likely want to have templates with specific characteristics needed for particular types of model building. The standard templates that come with Rhino have different viewport layouts or unit settings, but no geometry, and default settings for everything else. Different projects may require other settings to be changed. You can have templates with different settings for anything that can be saved in a model file, including render mesh, angle tolerance, named layers, lights, and standard pre-built geometry and notes. If you include notes in your template, they will show in the Open Template File dialog.

The New command begins a new model with a template (optional). It will use the default template unless you change it to one of the other templates, or to any other Rhino model file. To change the template that opens by default when Rhino starts up, choose New and select the template file you would like to open when Rhino starts, then check the Use this file when Rhino starts box.

To create a template:
1 Start a new model.
2 Select the Large Objects - Meters.3dm file as the template.
3 From the Render menu, click Current Renderer, and then click Rhino Render.

To set the Document Properties
1 From the File menu, click Properties.
2 In the Document Properties dialog, on the Grid page, change the Grid line count to to 100, Snap spacing to 0.05, the Minor grid lines every 0.1, and the Major lines to every 10.
3 On the Mesh page change the setting to Smooth & slower.
4 On the Rhino Render page, check Use lights on layers that are off.
5 On the Units page, Model units, change the Angle tolerance to 0.5. The end tangent normals will be determined by this setting. In Layout units, set units to Millimeter, and tolerance to 0.1, angle tolerance to 0.5.
6 On Location change to Los Angeles, CA USA.
7 On Linetypes set Model space linetype scale to 100 mm.
Click OK.



Apply grid changes to
( ) Active viewport only (Top)
(●) All viewports

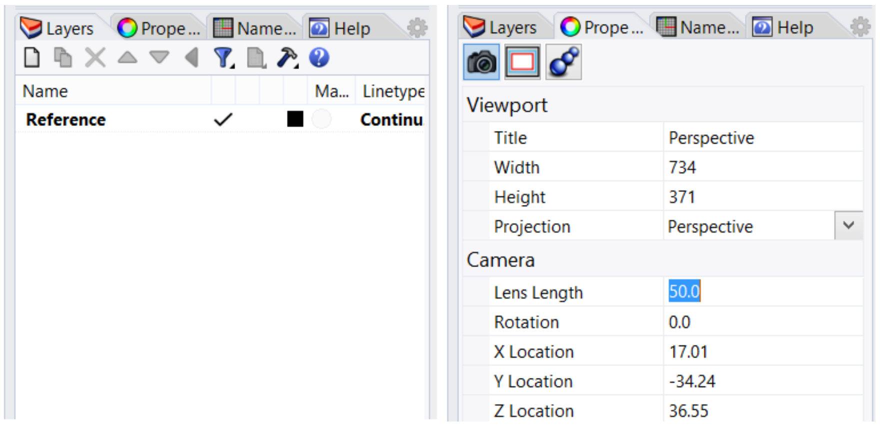
Grid properties
Grid line 100
Minor grid line 0.1 meters
Major lines every: 10 minor grid lines

✓ Show grid lines
✓ Show grid axes
✓ Show world axes icon

Grid snap
Snap spacing: 0.05 meters

Figure (63): Grid settings

To set up the layers and Properties
1 Open the Layers panel and rename Default to Reference and delete the rest of the layers.
2 Open Properties panel and set the camera Lens Length to 50
3 Open Named CPlanes and Help Panels and turn off all other panels



Name	Ma...	Linetype
Reference ✓	■	Continu

Viewport

Title	Perspective
Width	734
Height	371
Projection	Perspective

Camera

Lens Length	50.0
Rotation	0.0
X Location	17.01
Y Location	-34.24
Z Location	36.55

Figure (64): Set Camera lens

To set save notes
1 From the Panel menu, click Notes. Type the details about this template in the Notes panel. Make sure to start the line with //
2 From the File menu, click Save As Template. Name the template House_Decimal_Meters_0.001.3dm.

This file with all of its settings is now available any time you start a new model.

To set a default template
1 From the File menu, click New.
2 Select the template you want to use as the default template.
3 In the Open Template File dialog, check the Use this file when Rhino starts checkbox.
You should make custom templates for the kind of modeling that you do regularly to save set up time.

Resource 4: Templates
Template files in Rhino: https://vimeo.com/86730224

Productivity

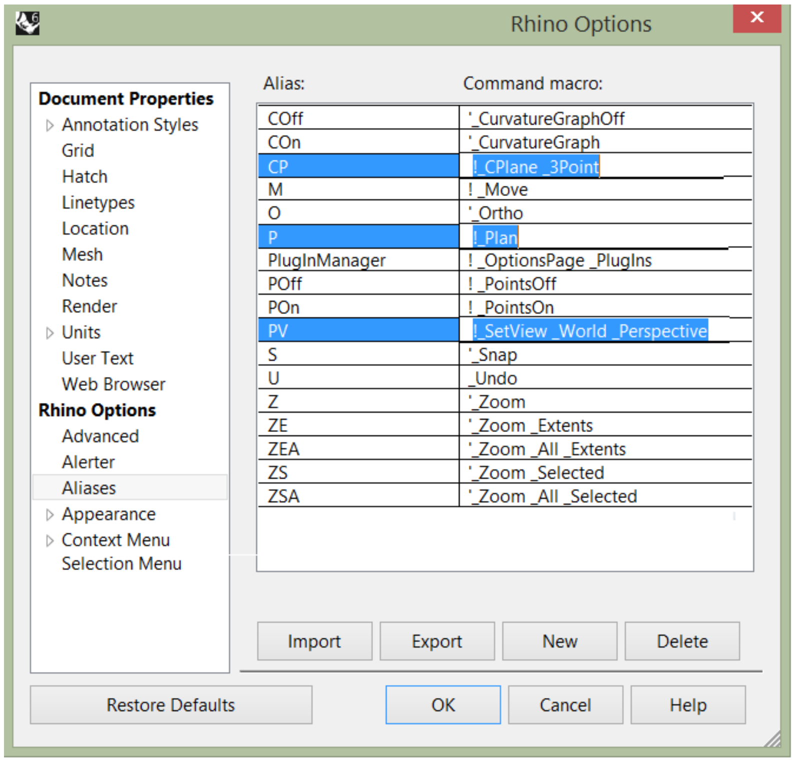
You can access existing commands and macros in a variety of ways in Rhino. The following table is a quick reference of the Rhino common macro key words and characters. For details, check the wiki article here and the Rhino macros help.

Macro character	**Meaning**	**Example**
(space)	All entries (command words and numerical inputs) need to be separated by a single space.	
Pause	Wait for user input	! Circle Pause 50
MultiPause	Wait for multiple input	! _Polyline _MultiPause
Enter	Simulate pressing "Enter" and is needed to end selection inside or to end commands	Polyline 1,1 1,9 6,9 Enter ;Same macro using relative coordinates to last point: Polyline 1,1 r0,8 r5,0 Enter
! (exclamation point)	Cancels the previous command. It is a good practice to use it before all macros (unless running nested command)	! SelAll
‘ (apostrophe)	Run as nestable command (while running another command). Note not all commands are nestable.	!Move ‘ SelAll Enter MultiPause
_ (underscore)	Runs command as English command name This is a good practice in case you pass the macro to a non-English user	! _Circle _3Point 0,0,0 1,1,0 0,3,0
- (hyphen)	Bypass dialog box (scripted command mode)	! _SelLast -_Rebuild _PointCount=10 _Degree=3 _Enter

Table (11): Rhino macro notation (partial list)

Here are few different ways to access macros quickly.

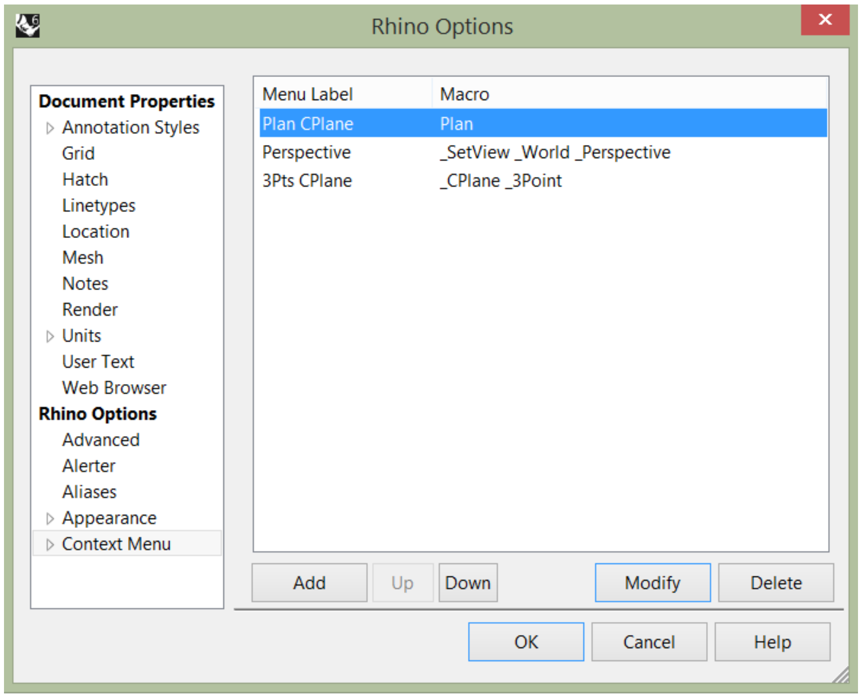
- Aliases are user-created commands for commonly used commands and macros. You can set your own aliases and access them through command line (type alias name). Notice the following three aliases:
    - **CP** _CPlane _3Point: create new CPlane from 3 points.
    - **P** Plan: switch to *Plan* view (parallel projection to active CPlane).
    - **PV** _SetView _World _Perspective: switch to World perspective view.



Alias:	Command macro:
COff	'_CurvatureGraphOff
COn	'_CurvatureGraph
CP	! _CPlane _3Point
M	! _Move
O	'_Ortho
P	! _Plan
PlugInManager	! _OptionsPage _PlugIns
POff	! _PointsOff
POn	! _PointsOn
PV	! _SetView _World _Perspective
S	'_Snap
U	_Undo
Z	'_Zoom
ZE	'_Zoom _Extents
ZEA	'_Zoom _All _Extents
ZS	'_Zoom _Selected
ZSA	'_Zoom _All _Selected

Figure (65): Set Aliases

- Context Menu macros can be added through Rhino Options. Add the following three macros:
  - **Plan CPlane** : Plan
  - **Perspective** : _SetView _World _Perspective
  - **3Pts CPlane** : _CPlane _3Point



Menu Label	Macro
Plan CPlane	Plan
Perspective	_SetView _World _Perspective
3Pts CPlane	_CPlane _3Point

Figure (66): Set Context menu

The context menu appears when you perform a right-click and hold:

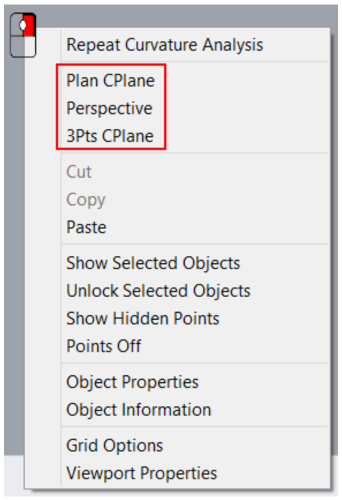
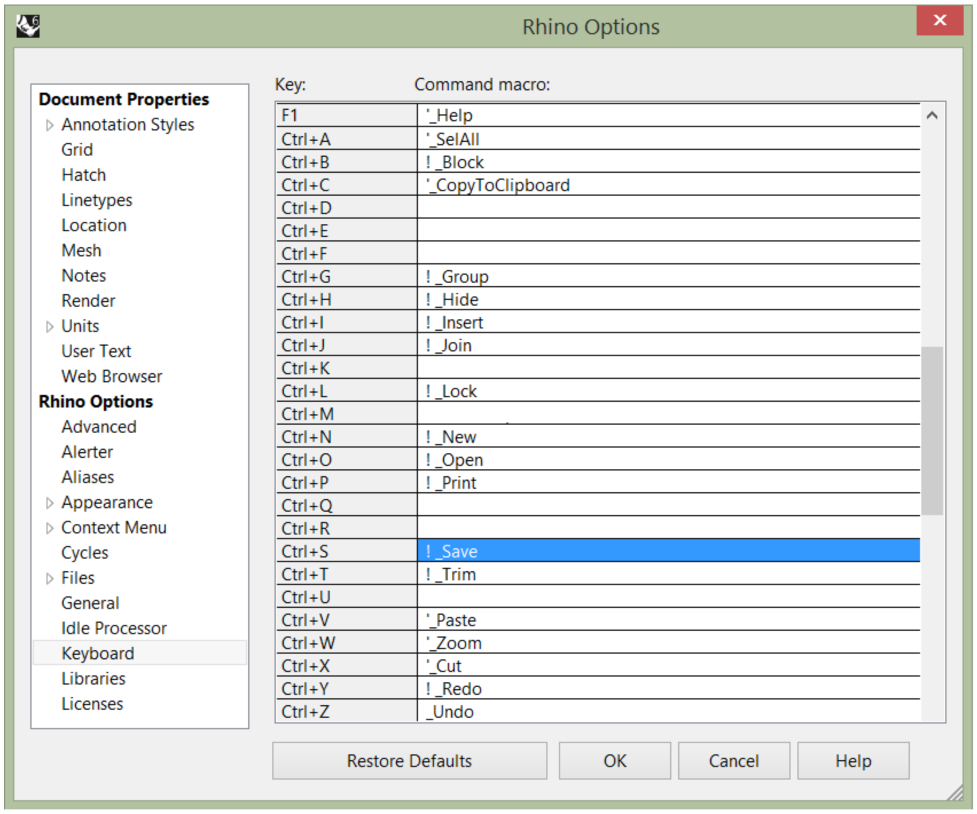


Figure (67): Activating context menu

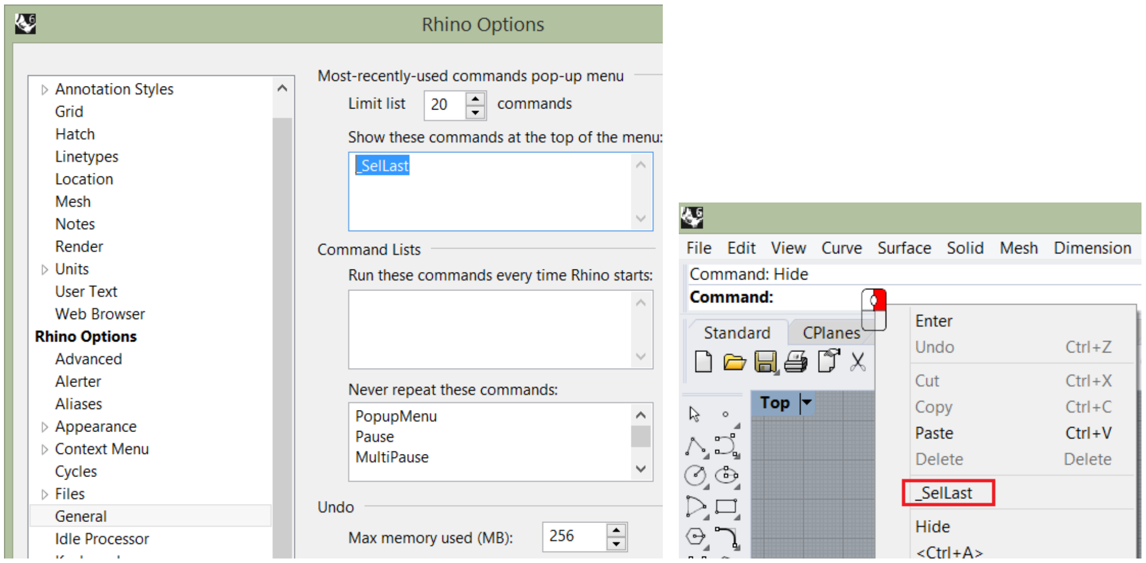
- Keyboard shortcuts: Rhino templates come with a predefined set of shortcuts that the user can modify and expand. Here are some of the more popular keyboard shortcuts:



Key:	Command macro:
F1	'_Help
Ctrl+A	'_SelAll
Ctrl+B	! _Block
Ctrl+C	'_CopyToClipboard
Ctrl+D	
Ctrl+E	
Ctrl+F	
Ctrl+G	! _Group
Ctrl+H	! _Hide
Ctrl+I	! _Insert
Ctrl+J	! _Join
Ctrl+K	
Ctrl+L	! _Lock
Ctrl+M	
Ctrl+N	! _New
Ctrl+O	! _Open
Ctrl+P	! _Print
Ctrl+Q	
Ctrl+R	
Ctrl+S	! _Save
Ctrl+T	! _Trim
Ctrl+U	
Ctrl+V	'_Paste
Ctrl+W	'_Zoom
Ctrl+X	'_Cut
Ctrl+Y	! _Redo
Ctrl+Z	_Undo

Figure (68): Set Keyboard shortcuts

- Startup commands: Right mouse click on the command area shows recently used commands. You can add your choice of command on top of the list by setting the Rhino options > General, then enter your macros in the top field as in the following:



Most-recently-used commands pop-up menu
Limit list 20 commands
Show these commands at the top of the menu:
_SelLast
Command Lists
Run these commands every time Rhino starts:
Never repeat these commands:
PopupMenu
Pause
MultiPause
Undo
Max memory used (MB): 256

Command: Hide
Command:

Figure (69): Set Keyboard shortcuts

- Toolbars: macros can also be placed inside toolbar buttons. If you prefer toolbar access, you can create your own custom toolbars to include repeated commands and macros. You start by creating a new toolbar:

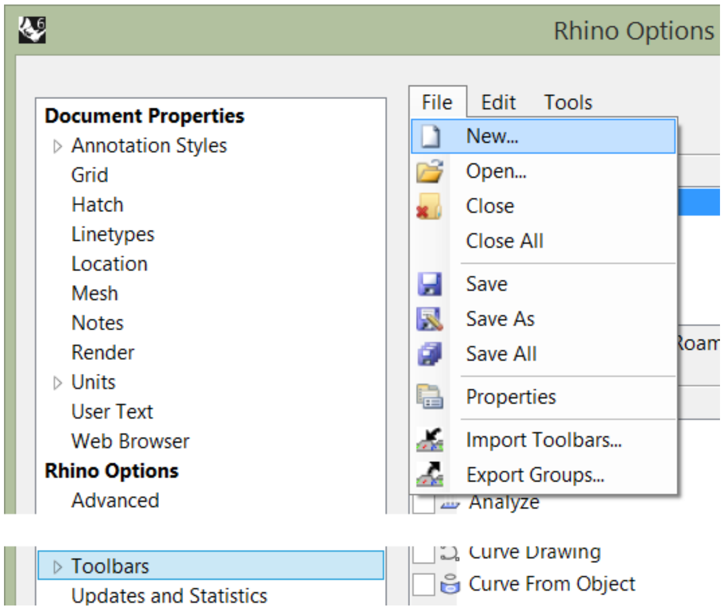
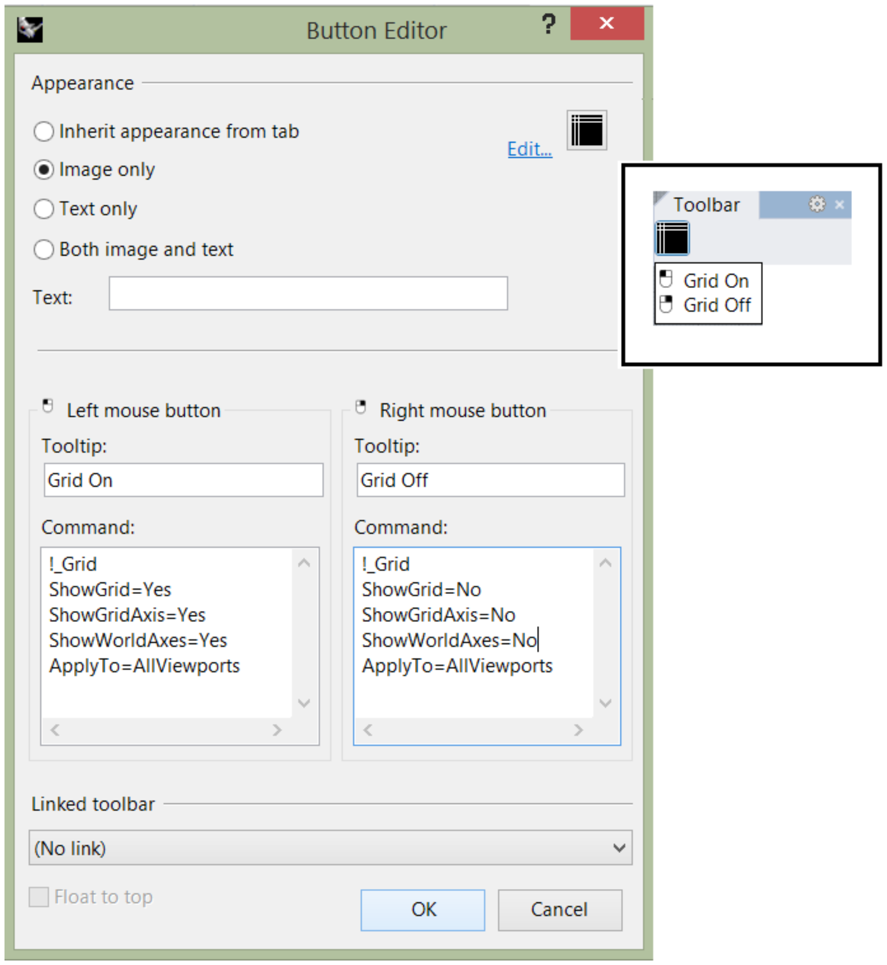


Figure (70): Toolbars

Then edit the button to include your macro:



( ) Inherit appearance from tab
(●) Image only
( ) Text only
( ) Both image and text

Text:

Left mouse button

Tooltip: Grid On

Command:

```
!_Grid
ShowGrid=Yes
ShowGridAxis=Yes
ShowWorldAxes=Yes
ApplyTo=AllViewports
```

Right mouse button

Tooltip: Grid Off

Command:

```
!_Grid
ShowGrid=No
ShowGridAxis=No
ShowWorldAxes=No
ApplyTo=AllViewports
```

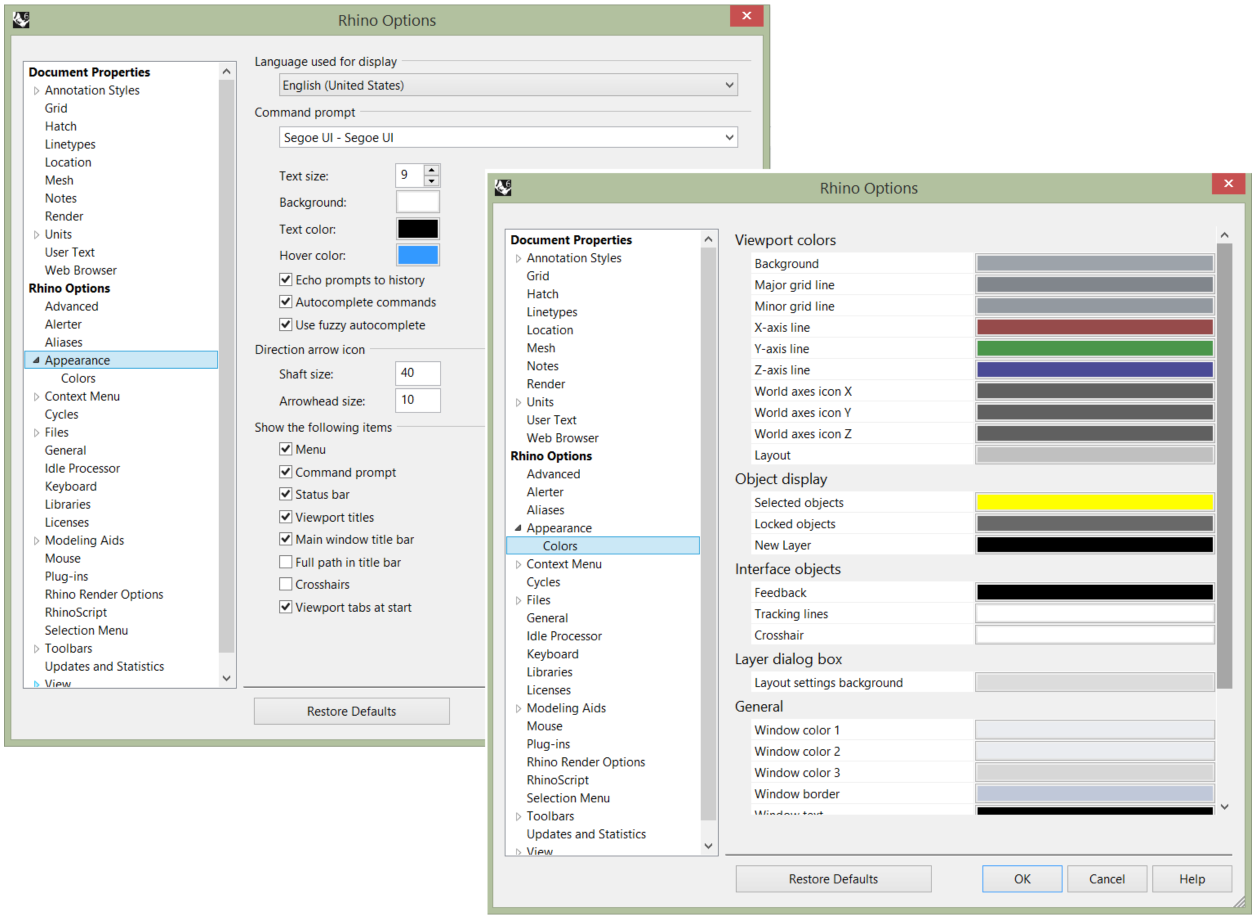
Linked toolbar: (No link)

☐ Float to top

Figure (71): Button editor

Appearance

You can also customize the color scheme and fonts in the Appearance and Colors under Rhino Options. You can always restore defaults to revert to original settings.



Language used for display: English (United States)

Command prompt: Segoe UI - Segoe UI

Text size: 9
Background: ⬜
Text color: ⬛
Hover color: 🟦

- [✓] Echo prompts to history
- [✓] Autocomplete commands
- [✓] Use fuzzy autocomplete

Direction arrow icon

Shaft size: 40
Arrowhead size: 10

Show the following items

- [✓] Menu
- [✓] Command prompt
- [✓] Status bar
- [✓] Viewport titles
- [✓] Main window title bar
- [ ] Full path in title bar
- [ ] Crosshairs
- [✓] Viewport tabs at start

Viewport colors

Background	⬜
Major grid line	⬜
Minor grid line	⬜
X-axis line	🟥
Y-axis line	🟩
Z-axis line	🟦
World axes icon X	⬛
World axes icon Y	⬛
World axes icon Z	⬛
Layout	⬜

Object display

Selected objects	🟨
Locked objects	⬛
New Layer	⬛

Interface objects

Feedback	⬛
Tracking lines	⬜
Crosshair	⬜

Layer dialog box

Layout settings background	⬜

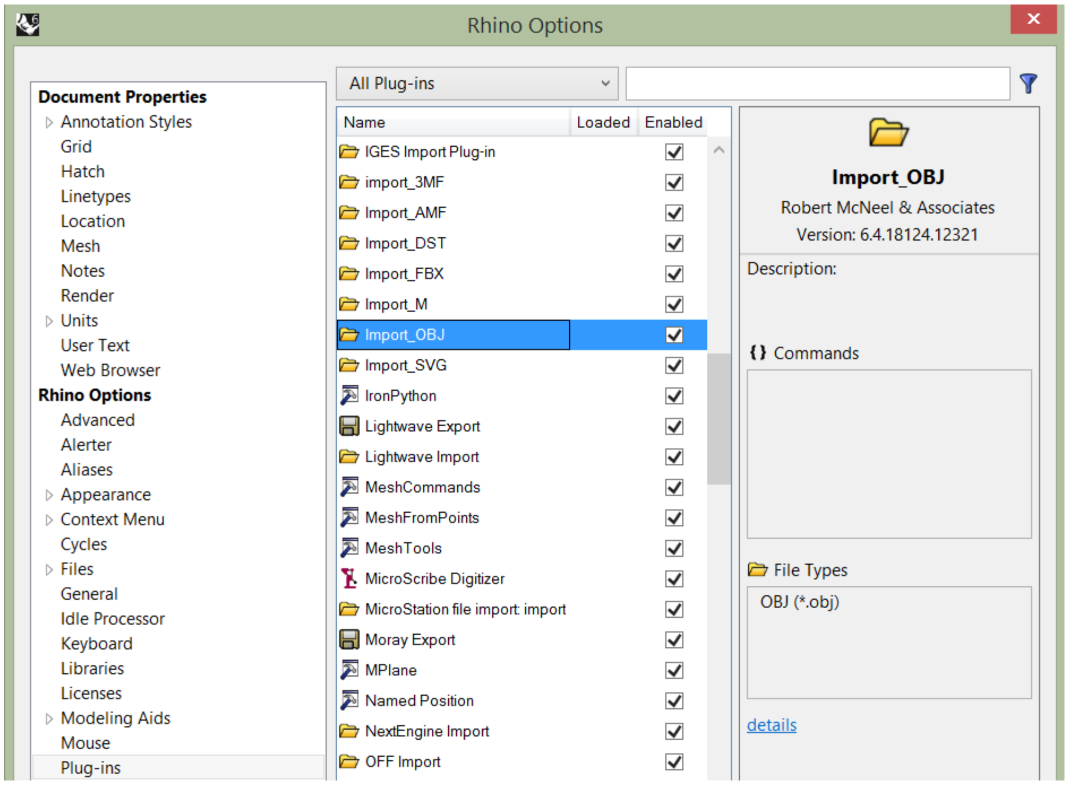
General

Window color 1	⬜
Window color 2	⬜
Window color 3	⬜
Window border	⬜

Figure (72): Customize appearance

Plugins

Professionals typically need specialized plugins to support the core Rhino functionality and improve productivity. Installed plugins are saved in a special McNeel folder and Rhino knows where to find them. Rhino ships with many plugins, especially those for file input/output. If you run the PluginManager command, you can see a long list of plugins


Rhino Options

Document Properties
- Annotation Styles
- Grid
- Hatch
- Linetypes
- Location
- Mesh
- Notes
- Render
- Units
- User Text
- Web Browser

Rhino Options
- Advanced
- Alerter
- Aliases
- Appearance
- Context Menu
- Cycles
- Files
- General
- Idle Processor
- Keyboard
- Libraries
- Licenses
- Modeling Aids
- Mouse
- Plug-ins

All Plug-ins

Name	Loaded	Enabled
IGES Import Plug-in		✓
import_3MF		✓
Import_AMF		✓
Import_DST		✓
Import_FBX		✓
Import_M		✓
Import_OBJ		✓
Import_SVG		✓
IronPython		✓
Lightwave Export		✓
Lightwave Import		✓
MeshCommands		✓
MeshFromPoints		✓
MeshTools		✓
MicroScribe Digitizer		✓
MicroStation file import: import		✓
Moray Export		✓
MPlane		✓
Named Position		✓
NextEngine Import		✓
OFF Import		✓

Import_OBJ
Robert McNeel & Associates
Version: 6.4.18124.12321

Description:

Commands

File Types
OBJ (*.obj)

details

Figure (73): Plugin manager

Most third party plugins come as **.rhi** files. Those are special zipped files that Rhino recognizes. The user can install .rhi files by double clicking on them, and following the wizard to install for one user or all users. Rhino places installed plugins in the proper folder and loads when Rhino opens next time. Rhino also recognizes if there is a toolbar that comes with the plugin and allows loading it by the user.

If there are plugins that you do not like Rhino to load, you can place load protection by un-checking the “Enabled” box for these plugins. Next time you open Rhino, they will not load. You can always check the box again any time.

Concept modeling

This involves generating site and building geometry of the concept. Concept design goes through multiple iterations and can be modified before reaching the final solution. It is a good practice to keep a record of the steps to be able to communicate and modify easily. Architectural projects typically start with a site geometry, then generate mass model, analysis and visualization. We will cover all of these aspects using step by step tutorials.

Site terrain and surroundings

Geographical information is documented digitally with multiple layers of information. This is usually public information. To be able to bring site data to the modeling environment, and use relevant information is an important part of design. Designers need to model their site terrain, surroundings, and place their building in site.

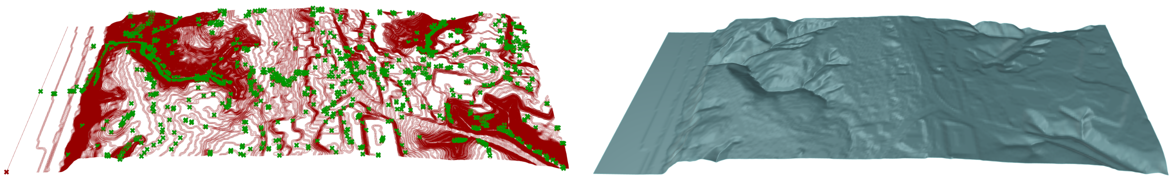
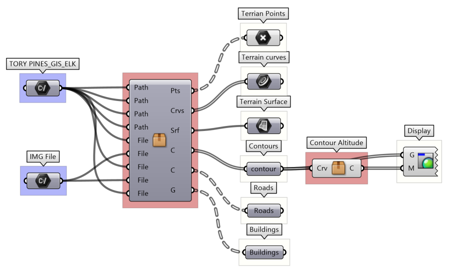
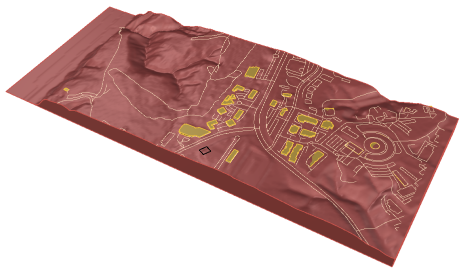


Figure (74): Import site GIS information

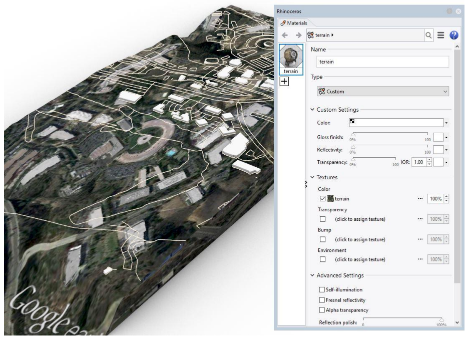
Resource 5: GIS data import using Grasshopper plugins
Workflow to import GIS data and generate site geometry: [Tutorial: import GIS information and model the terrain and surroundings]()

In the GIS tutorial we will be looking at how to acquire, parse and represent data from open source geospatial data.

Tutorial 401: GIS	
Import GIS data for the location with terrain, roads and buildings. Need to use specialized plugins to import GIS data from surveys or images. The following [link]()has a workflow that uses a Grasshopper plugin to turn GIS data into terrain surface and surroundings.	 Terrian Points Terrain curves Terrain Surface TORY PINES_GIS_ELK IMG File Path File Pts Crvs Srf C G Contours contour Contour Altitude Crv Display M Roads Buildings

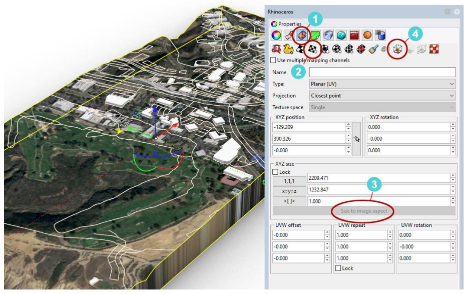
Place in Rhino approximate site near World origin	

To add the terrain photo as a texture map, drag and drop the Terrain.png image onto the model in any Rhino viewport. This will create and assign a new material in the Materials panel with the terrain image set to the color channel. Enter Rendered mode in the Perspective view to see the applied material.

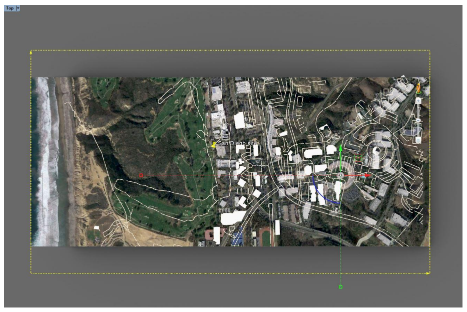

Name: terrain
Type: Custom
Textures: Color ☑ terrain

The terrain image is not sized and positioned correctly yet. Use the texture mapping section of the Properties panel to adjust it.

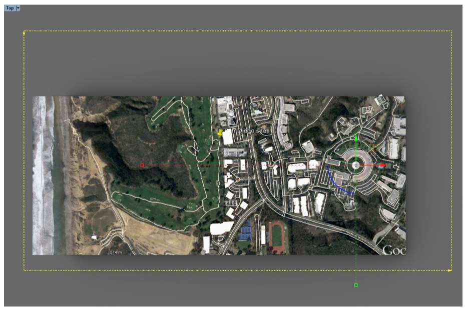
- Select the model and click the texture mapping icon in the Properties panel.
- Change the mapping method used to Planar. There will be several options prompted within the command line. Choose 'Bounding Box' for the sizing and enter through the others to accept their defaults.
- Next, click the 'Size to image aspect' button in the texture mapping settings.
- Then click the icon to show the planar mapping widget for adjustment.

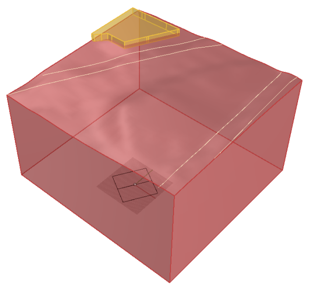

Type: Planar (UV)
Projection: Closest point
Texture space: Single
XYZ position: -129.209, 390.326, -0.000
XYZ rotation: 0.000, -0.000, 0.000
XYZ size: 2209.471, 1232.847, 1.000
UVW offset: -0.000, -0.000, -0.000
UVW repeat: 1.000, 1.000, 1.000
UVW rotation: 0.000, -0.000, 0.000

Enter the Top view and use Rendered mode to see the terrain texture. Select the mapping widget which is displayed as a dashed yellow rectangle. It will be easier to see the widget if you change the solid color backdrop in the Rendering panel to dark gray. With the Gumball enabled, either right click the center of it and choose Relocate Gumball, or use the command RelocateGumball. Position the Gumball origin to the center of the circular structure in the terrain image as shown. Use the Shift key while placing the X and Y axis directions to keep them aligned with the edges of the planar mapping widget.

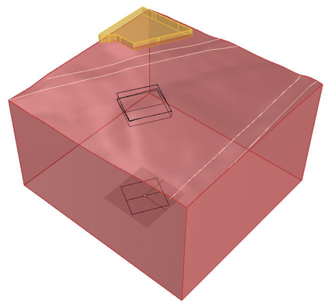
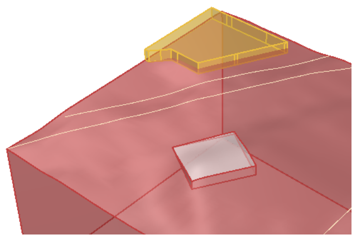
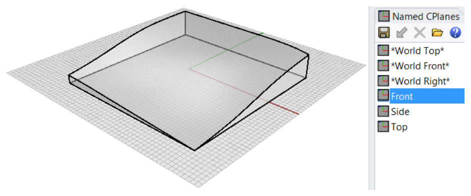
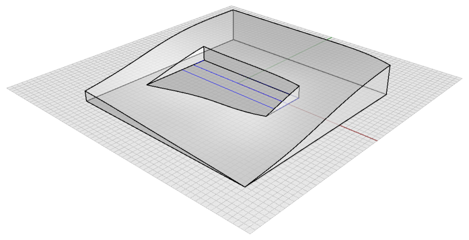


Using the Gumball, reposition the mapping widget so that the origin point is over the center of the circular structure in the model. Next, scale the mapping widget by clicking and dragging on either of the scale boxes at the end of gumball axis, hold the Shift key while dragging to maintain the aspect ratio of the image. Look at matching the white lines in the model to some existing roadways in the image as it's scaled. Once done, you can hide the mapping widget via Properties using the same icon that displayed it.



Cut out smaller part of the terrain and place near World Origin - Boolean out part of the site with buildings and roads. - Define site boundary in the flat XY Plane (40*40 tilted 30 degrees clockwise)	

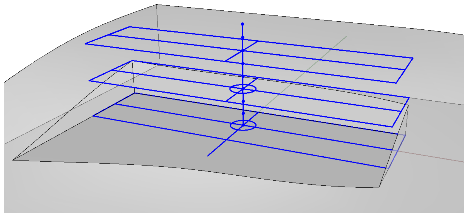
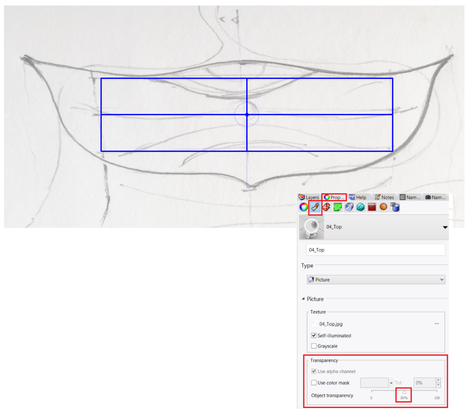
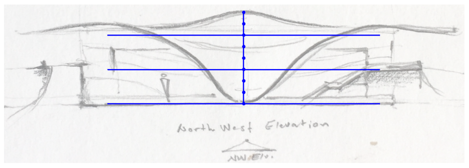
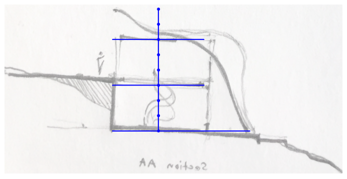
You can take part of the terrain closer to the site to model the concept. The following workflow shows how to take part of the terrain and define site boundaries and a cut region.

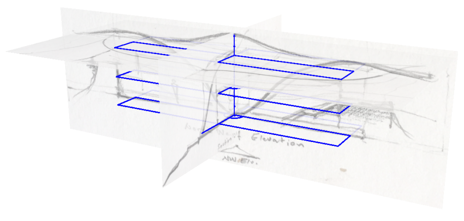
Tutorial 402: Site	
Site boundary and levels - Project the flat outline to the terrain. - Copy the flat outline to the bottom and lowest and highest points.	
Geometry of the site slope: - Extrude base curve and Cap - BooleanSplit extruded box by the terrain. - BooleanSplit the terrain by the extruded box. - Delete unwanted split parts	
Set CPlane to be at the center of the site and save three main CPlanes for quick reference. - CPlane > 3Point to set the CPlane to the Top, Front and Side relative to the site orientation - In Named CPlane panel, save the new CPlane as Top, Front and Side Note: you can set up the site CPlanes in the template file	 Named CPlanes *World Top* *World Front* *World Right* Front Side Top
Create the site cut for the building mass (24*6*6). Calculate the cut volume (~420 cubic meter). - Rectangle > Center > 24 > 6 to outline building base. - Box for the building mass across the site - BooleanSplit to cut the site	

Mass model

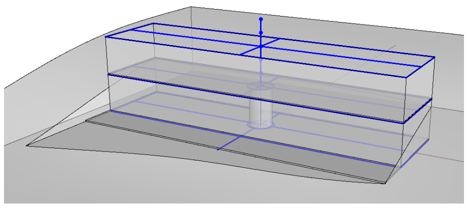
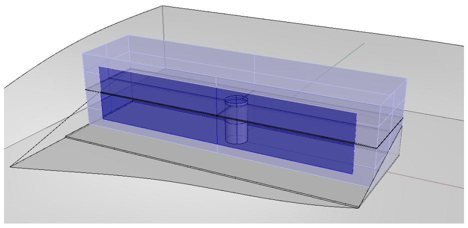
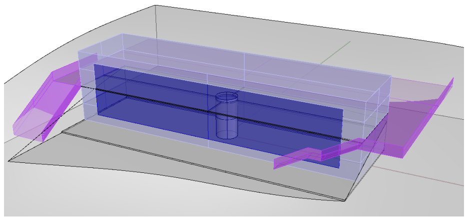
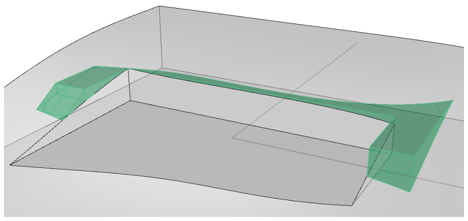
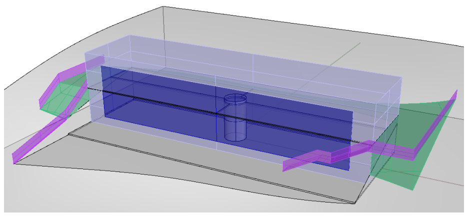
Generating building geometry involves creating scaffolding, placing sketches, and modeling the building’s main geometry elements.

Scaffolding is very useful to define reference geometry to help create the building. This geometry helps define things such as orientation, dimensions, and levels. Scaffolding geometry can be used as a reference when creating the building geometry.

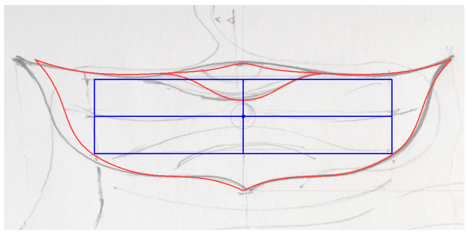
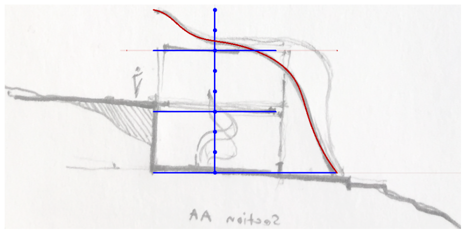
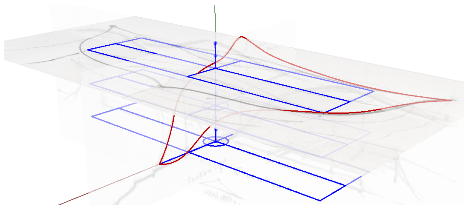
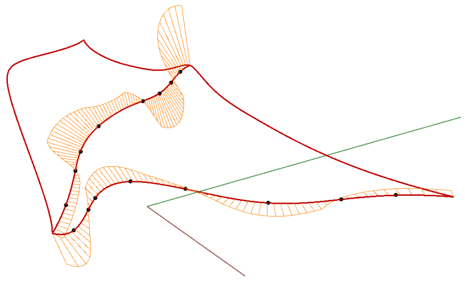
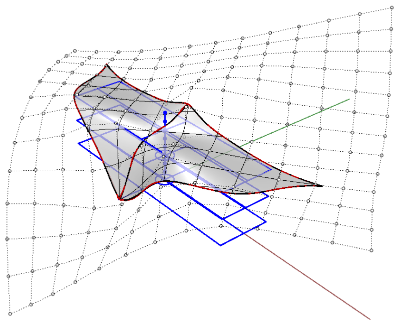
Tutorial 403: Scaffolding	
Create scaffolding for mass, levels and stairs: - Create new layer for Scaffolding geometry - Rectangle (24*6) - Draw Lines through the rectangle center - Draw Line > Vertical from center with 8m length - Divide the vertical line into 8 segments to create reference points that are spaced by 1m - Copy base rectangle and lines vertically to level 3 and 6 m - Create Line at the base 3 m long - Circle (radius=1) on lower and middle levels for staircase	
Place top sketch - Make SiteCenter the active CPlane (double click SiteCenter in the list of Named CPlanes panel) - Use Picture to place top sketch - Use Plan to align parallel to active CPlane - Move and Scale the picture to fit dimensions - Change transparency to 30% to see scaffolding: - Select picture - Go to Properties Panel > Materials > Picture > transparency and set to 30%	 Layers Prop... Help Notes Nam... Nam... 04_Top 04_Top Type Picture Picture Texture 04_Top.jpg Self-illuminated Grayscale Transparency Use alpha channel Use color mask Tol: 0% Object transparency 0 50% 100
Place elevation sketch - Create Site Front CPlane and save in Named CPlanes - Place elevation Picture and Move/Scale to fit dimensions - Change transparency to 30% (same steps for top)	
Place section sketch - Create Site Side CPlane and save in Named CPlanes - Place section Picture and Move/Scale to fit dimensions - Change transparency to 30% (same steps for top)	

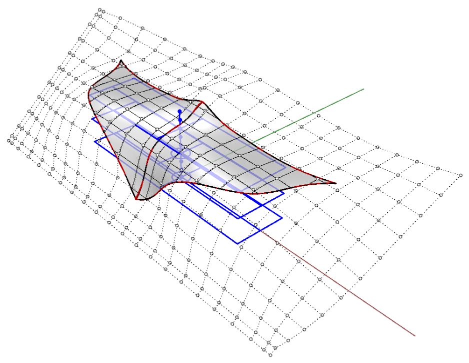
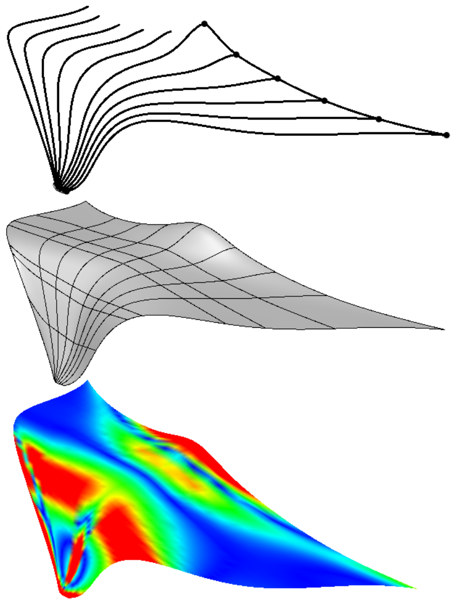
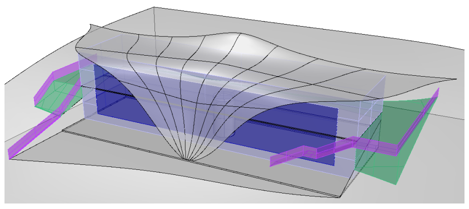
Go back to perspective view to verify that all sketched in place.	

Use scaffolding and sketches to create floor slabs and elevation and staircase surfaces.

Tutorial 404: Mass	
Create initial massing geometry - Switch to SiteCenter CPlane (in Named CPlanes panel) - Create a new Building layer and a Mass sublayer, and make Building>Mass current layer - Box to create first and second floor slabs - ExtrudeCrv to create walls surfaces - Cylinder to place staircase mass	
Add solid and void surfaces in elevation. - Calculate Area of solid and void (penings area = ~92 sqm, walls = ~260 sqr m)	
Add front and side platforms/balconies - Draw curves, ExtrudeCrv, then Join and MergeAllFaces - Use ortho angle = 30 to define stairs' slope (30-37 degrees is the recommended stairs slope)	
Generate fill - ExtrudeSrf platform surface in the -Z direction, then BooleanSplit with the site to generate the fill solid - Volume = ~47 cubic meter	
Clean up - Trim rails and platform surfaces	

Use reference sketches as reference to create main curves of the roof in each picture plane, then adjust curves to be placed correctly in 3D space. Use the curves to generate the roof surface.

Tutorial 405: Roof	
Trace roof outline in the top picture. Snap to other end curves and scaffolding when relevant. - Set CPlane to top picture - Use Curve to trace one side - Use ProjectToCPlane to ensure curves on plane (snapping can cause control points to go off the plane) - Use Mirror to reflect the curves and ensure symmetry	
Repeat the tracing process for the section	
Adjust top curves to match elevation contour - Need to adjust only half the curves, then mirror them - Set CPlane to Elevation. - Set selection filter to"ControPoints" only. - Move points Vertical - Adjust using side view as well - Mirror curves Test curves' smoothness with curvature graph for curves. The points mark significant locations on the curve where curvature changes from concave convex, or the middle of a long span.	 
Roof surface - Patch Having boundary curves and cross section curves might be enough to create a good patch surface. - Use Patch command and select roof 3D curves. Use Preview to adjust Patch settings - Pull curves to surface - Trim surface with pulled curves Notice that the surface parameterization (direction of isocurves) is not in any particular useful direction. Also the surface is not symmetrical.	

Now Patch with starting surface. Create a plane that goes through the three extreme points and make slightly bigger. Use that plane as a starting surface in Patch command. Repeat steps above to create the patch surface. Notice that parameterization followed the plane direction. Also when comparing the mean curvature of the two patches, the second comes out smoother (notice the transition from blue to red). Notice that both surfaces are trimmed surfaces.	
Roof surface - Loft surface Loft surfaces are typically easy to set up, construction curves are in one direction and are easy to adjust and match. History also allows more interactive editing until reaching a desirable solution. - Divide front curve into 5 segments. - TweenCurves between middle and end, then adjust curve ends to align with the points. Use PointsOn to move control points. - Mirror tween curves. - Loft curves with History on to be able to adjust the surface as needed. - Test mean curvature using CurvaturAnalysis	
Concept all put together	

Concept visualization

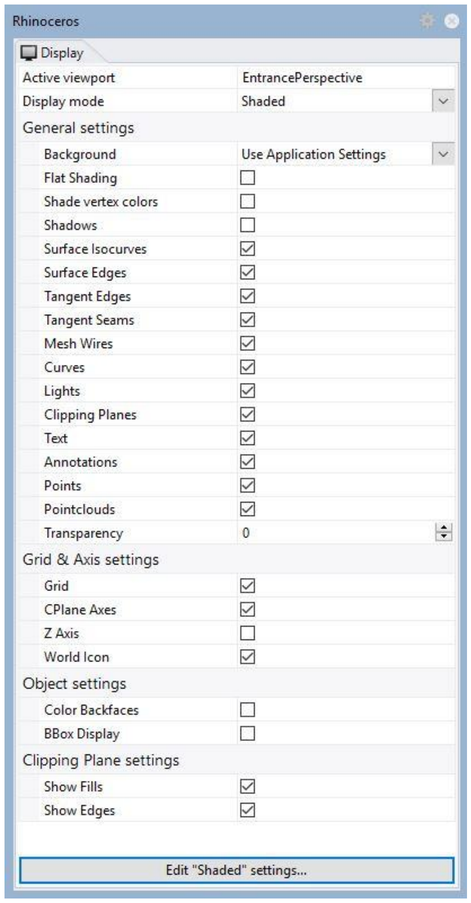
Visualization is an essential part of concept development and communication. Rhino has a rich set of tools for quick as well as elaborate visualization. Viewport display modes offer quick access to a variety of effects. They all are working modes where you can continue to edit and orbit your model. There are also tools to assign materials, create renderings and capture images.

Tutorial 406: Visualization

Display modes:

A variety of display modes are included with Rhinoceros and these are accessible in the Display panel which can be found in the Panels drop down menu if not already shown. Many commonly changed settings can be altered within this panel. Click the button at the bottom of the display panel to open the full options for the mode chosen.

Display mode settings are system specific and are not part of the 3dm file. To export and import customized display modes, use the Options command and navigate to View>Display Modes. Any existing display mode can be copied as the starting point for a new display mode.



Rhinoceros

Display

Active viewport	EntrancePerspective
Display mode	Shaded
General settings	
Background	Use Application Settings
Flat Shading	☐
Shade vertex colors	☐
Shadows	☐
Surface Isocurves	☑
Surface Edges	☑
Tangent Edges	☑
Tangent Seams	☑
Mesh Wires	☑
Curves	☑
Lights	☑
Clipping Planes	☑
Text	☑
Annotations	☑
Points	☑
Pointclouds	☑
Transparency	0
Grid & Axis settings	
Grid	☑
CPlane Axes	☑
Z Axis	☐
World Icon	☑
Object settings	
Color Backfaces	☐
BBox Display	☐
Clipping Plane settings	
Show Fills	☑
Show Edges	☑

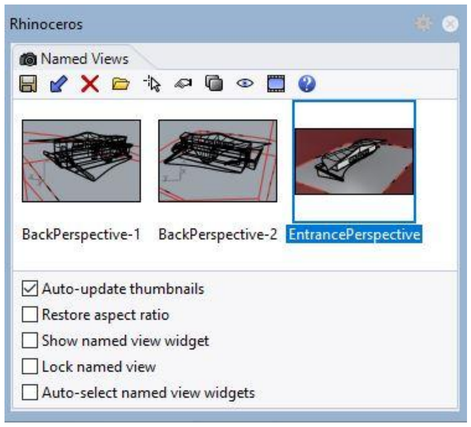
Edit "Shaded" settings...

Named Views:

Open the file '406_Visualization.3dm'. There are several saved named views which can be seen in the Named Views panel. You can show the Named Views panel via the Panels drop down menu if needed.

Double click any thumbnail image in the Named Views panel to restore it in the active viewport.

Use the icons at the top of the panel to access various named views functionality. Among these controls you can import named views from another 3dm, edit a named view by simply rotating the viewport as well as animate the transition between named views.

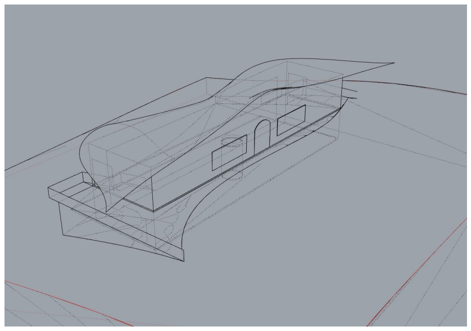
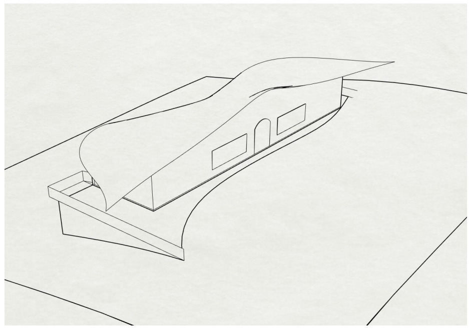
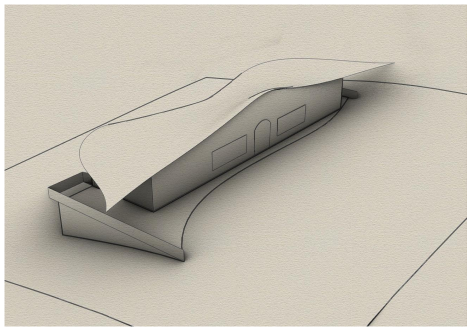
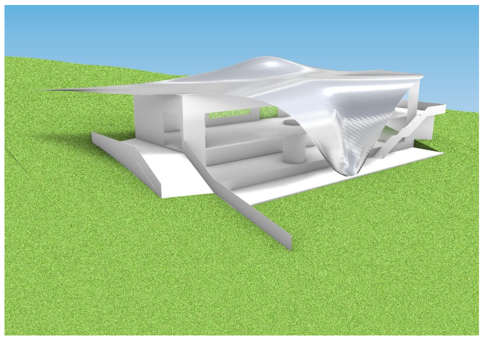


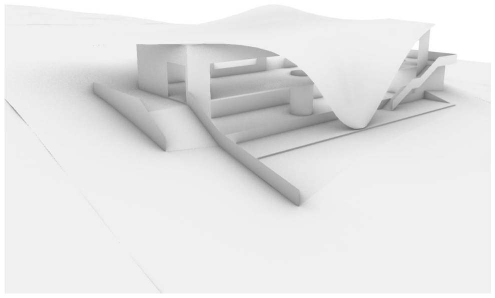
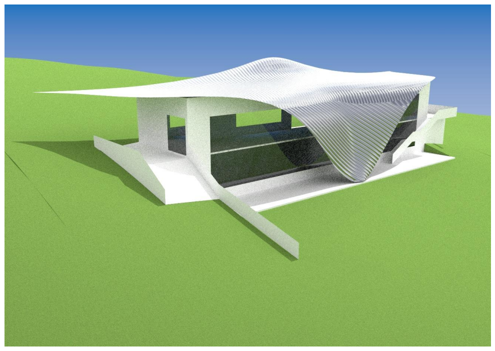
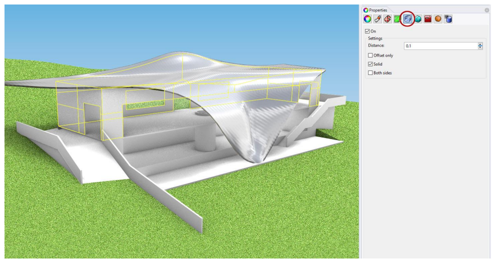
Rhinoceros

Named Views

BackPerspective-1 BackPerspective-2 EntrancePerspective

☑ Auto-update thumbnails
☐ Restore aspect ratio
☐ Show named view widget
☐ Lock named view
☐ Auto-select named view widgets

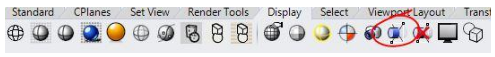
Display mode features: Some similarities can be found between certain default display modes and these are good to understand as you experiment with customization. **Technical, Artistic and Pen** are all derivatives of the Technical display mode. These modes require an initial calculation time dependent upon the complexity of the scene. Hidden lines and Silhouettes can be visualized in these modes making them unique.	  
Rendered mode* uses many of the settings found in the Rendering panel which can be shown via the Panels drop down menu if needed. Namely, the display of assigned materials and cast shadows distinguish this mode from a standard Shaded display. *See Advanced Visualization section 508 below for more details.	

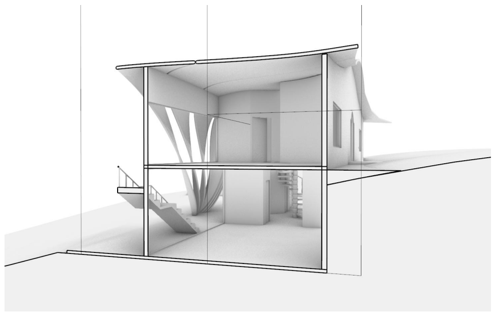
Arctic mode will override any material assignments and custom lighting to produce a soft shadowed grayscale display mode. Materials remain in the file and are still assigned. By default, transparent materials such as glass will remain transparent in the Arctic mode. This option is configurable within the display mode settings.	
Raytraced mode* is unique among the display modes in Rhino. This mode actually renders the model interactively in the viewport. Reflections between objects as well as indirect illumination (the bouncing of light) are calculated. *See Advanced Visualization section 508 below for more details.	
Render mesh modifiers in the Properties panel can be used to change the look of the model without impacting the actual geometry. One example is the Thickness modifier which when enabled will make any surfaces appear as solid in shaded modes.	 Properties [✓] On Settings Distance: 0.1 [ ] Offset only [✓] Solid [ ] Both sides

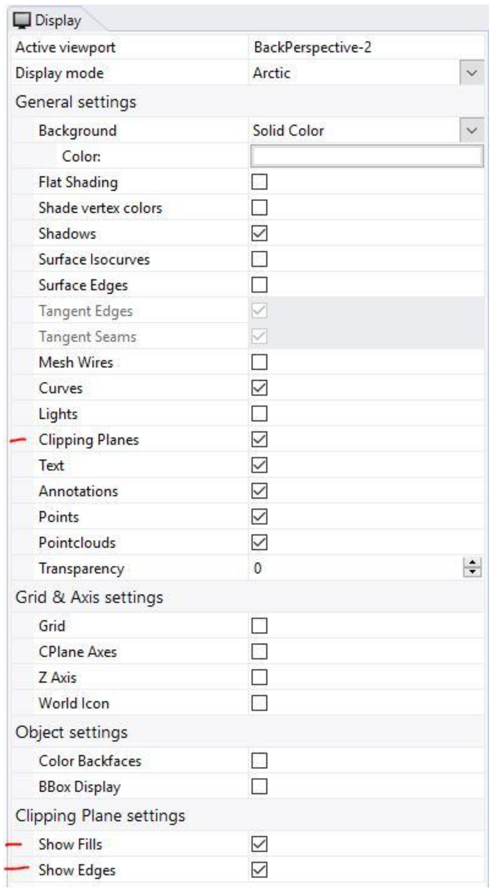
Another display feature that can be quite helpful in concept visualization is the use of ‘clipping planes’. You can create a clipping plane with the ClippingPlane command or by clicking the icon in the Display toolbar group.

Multiple clipping planes can be used together and the views they affect can be specified in the Properties panel > Clipping Plane section while selected.

Display modes also have the ability to set how clipping planes are displayed and whether the clipping plane widget is shown.







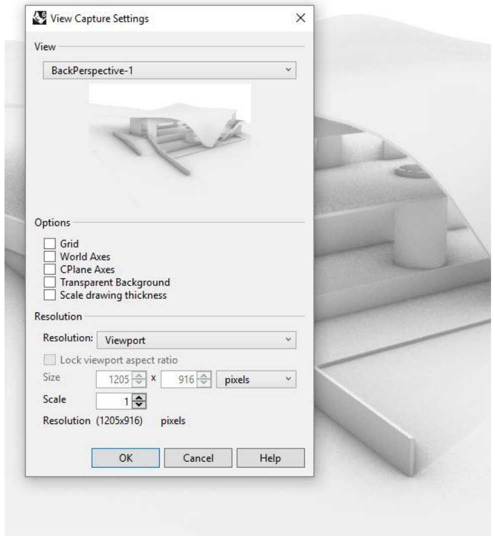
Display

Active viewport	BackPerspective-2
Display mode	Arctic
General settings	
Background	Solid Color
Color:	
Flat Shading	☐
Shade vertex colors	☐
Shadows	✓
Surface Isocurves	☐
Surface Edges	☐
Tangent Edges	✓
Tangent Seams	✓
Mesh Wires	☐
Curves	✓
Lights	☐
Clipping Planes	✓
Text	✓
Annotations	✓
Points	✓
Pointclouds	✓
Transparency	0
Grid & Axis settings	
Grid	☐
CPlane Axes	☐
Z Axis	☐
World Icon	☐
Object settings	
Color Backfaces	☐
BBox Display	☐
Clipping Plane settings	
Show Fills	✓
Show Edges	✓

To save an image of the viewport in any display mode, you can use the commands ViewCaptureToFile or ViewCaptureToClipboard. Both commands will offer additional options in a floating dialog. Here you can change the scale of the image to be greater than the monitor resolution*. This can be useful for instance if you wish to ultimately print the image as part of a high resolution poster.

Used in conjunction with Named Views, multiple display modes of the same view can be saved for external compositing. Here the Silhouette command was used over an Arctic view capture and combined with the same view captured in Rendered mode.

*Note that scaled Raytraced view captures will require a re-render of the viewport to account for the added pixels. Other modes, however, will scale up during capture relatively instantly.

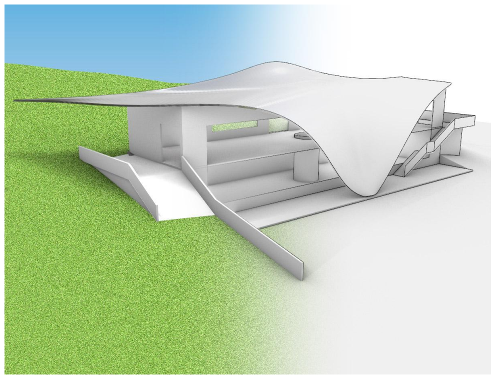


View Capture Settings

View: BackPerspective-1

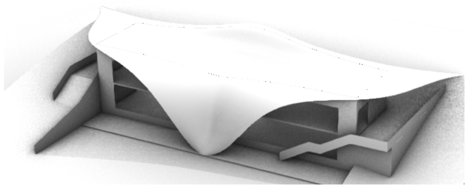
Options
- ☐ Grid
- ☐ World Axes
- ☐ CPlane Axes
- ☐ Transparent Background
- ☐ Scale drawing thickness

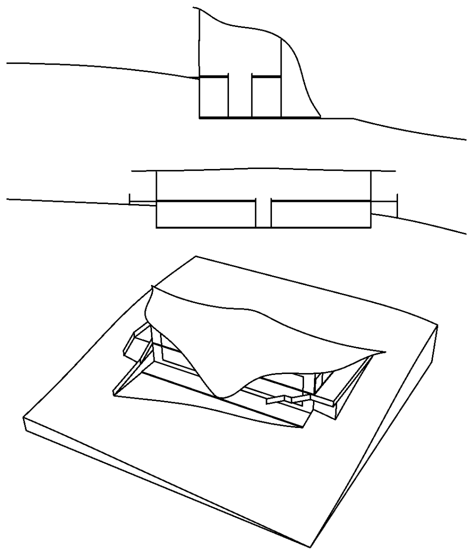
Resolution
- Resolution: Viewport
- ☐ Lock viewport aspect ratio
- Size 1205 x 916 pixels
- Scale 1
- Resolution (1205x916) pixels



Concept analysis

Analysis helps evaluate the design option and develop the concept. We have done some analysis above for the roof surface continuity, the overall volume of cut and fill in the site and building orientation. Rhino supports many workflows for concept analysis and the following includes some of those.

Tutorial 408: Concept analysis	
Shadows study - Use SetOneDaySunAnimation, then PlayAnimation to view and RecordAnimation to save.	

Basic scheduling: - Building areas of floors, glass, outdoor payments. - Add data to objects (names and other attributes) in the object properties panel.	
Basic line drawings - Use Section on Top view to create sections - Use Make2D of the current view	

Detailed modeling

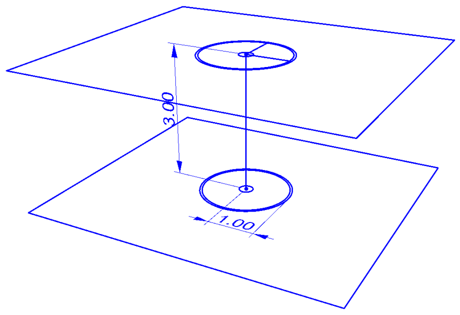
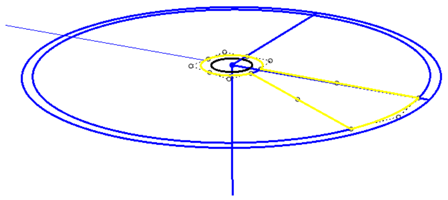
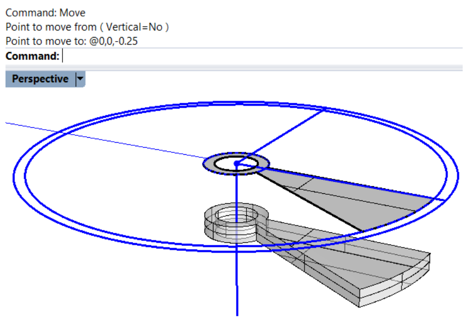
Advanced NURBS modeling

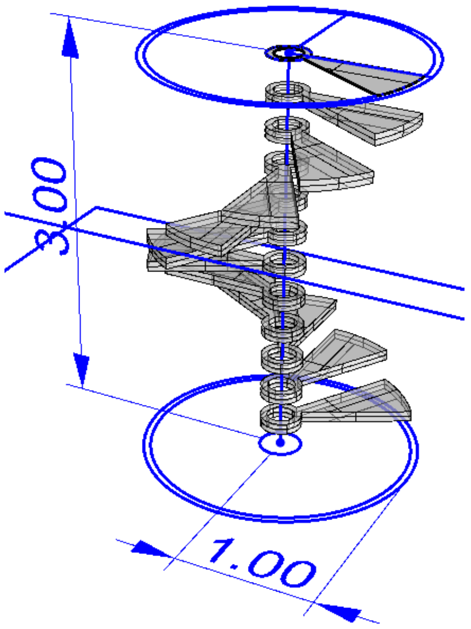
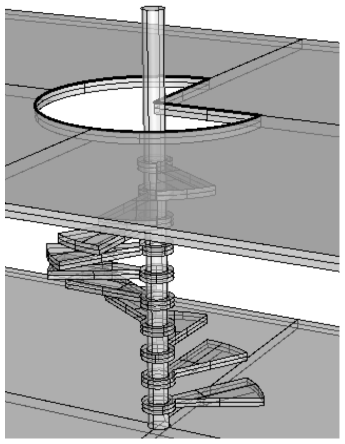
As the model progresses from concept to details, changes become more time consuming. Direct modeling favors a more linear approach where the overall concept is decided upon before starting to add more details. In Rhino, there are few parametric features, such as blocks and history, that allow for updates past the concept stage but they are typically more limited. Algorithmic modeling is more suitable for a nonlinear approach to design where details can be worked on before the concept is finalized. That is because it is easy to adjust at a later stage. Workflows past concept stages include advanced geometry, rationalization, visualization and documentation.

Building details

Building details include circulation, layout, materials, structure and floor plans. This work involves accurate modeling and considerations of building standards and material specification.

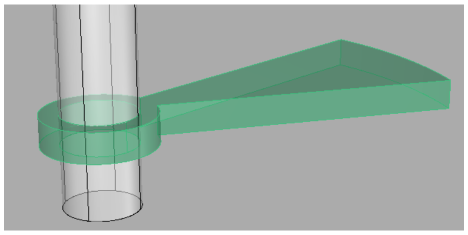
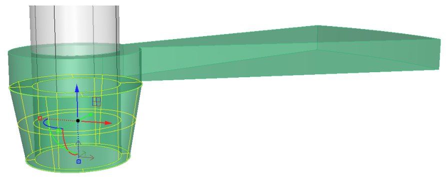
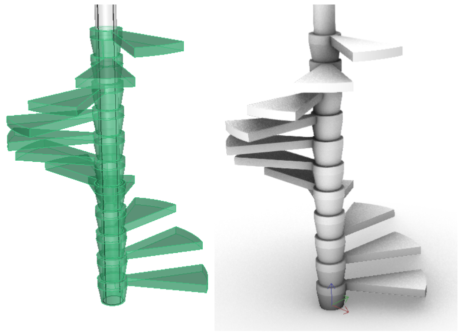
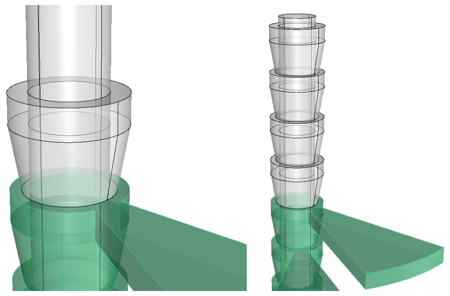
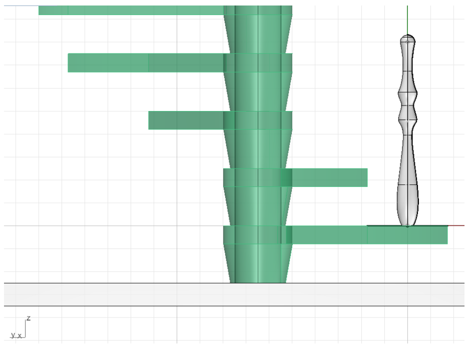
Tutorial 501: Spiral staircase

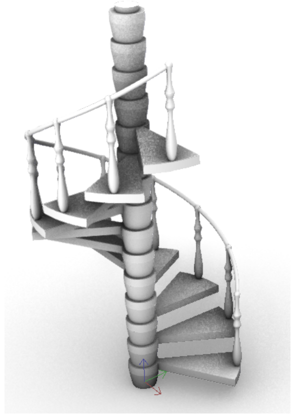
Create the stairs' scaffolding with the desired dimensions. - Locate the staircase center to be on top of the lower floor. Adjust the named CPlanes origins to be that center point. - Make circles with radiuses equal 0.15, 0.95 and 1m. Each step width equal 0.8m. - Floor to floor height equal 3m. - Use Distance and Dim commands to measure and mark dimensions. Note that you need to change the CPlane to be parallel to vertical center line before you can apply dimension.	
Create one step curves - Use the stairs centerpoint - Draw inner Arc. Angle = 360/number of steps per full rotation. For 12 steps, angle = 30 degrees. - Draw outer arc. Need to decide step width - Draw lines connecting endpoints of the step, then join all curves (use Line and commands). - Draw inner and outer circles of the step end - Delete inner curve, Trim then Join step curves join into 2 curves.	
Create step solid - Use PlanarSrf to create the step surface - ExtrudeSrf with "0.08" for the thickness - Calculate step tp step distance: floor to floor height is 3m and number of steps is 12 steps, so the distance from one step to the next is 0.25m - Move down 0.25	Command: Move Point to move from (Vertical=No) Point to move to: @0,0,-0.25 **Command:**

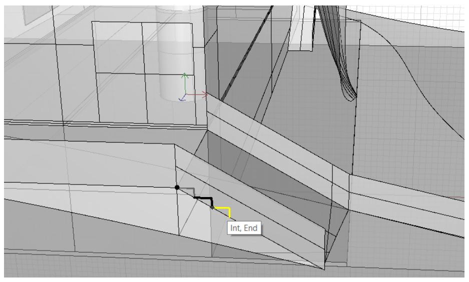
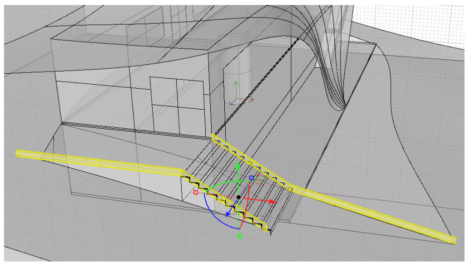
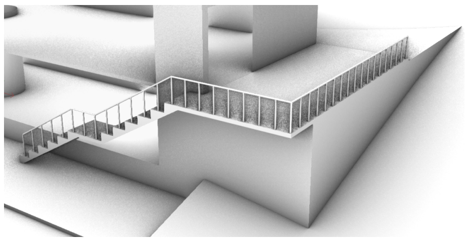
Create all steps. - Change CPlane so that Z direction points downwards. - ArrayPolar command with 11 steps StepAngle=30, offset option “ZOffset=0.25” to create all the steps	
Add staircase pole and upper floor slab - Use Cylinder - Use PlanarSrf and ExtrudeSrf (thickness=0.1)	

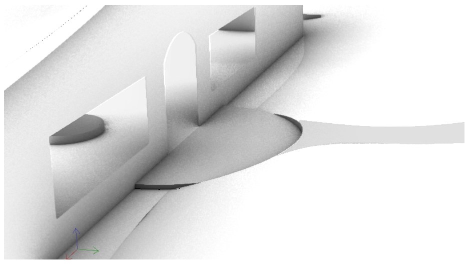
Repeated geometry, such as the step solid in the staircase tutorial, are best modeled as blocks. This helps keep model size smaller and allows designers to change step geometry and populate to all steps without remodeling. The following tutorial creates steps and railing as blocks.

Tutorial 502: Blocks

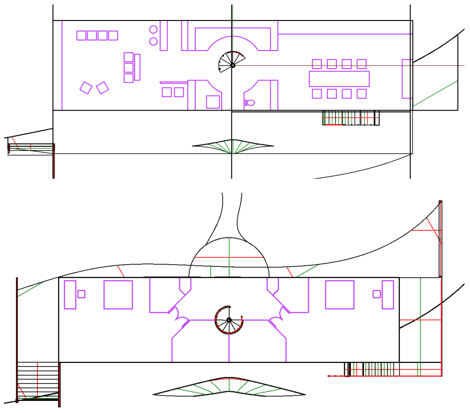
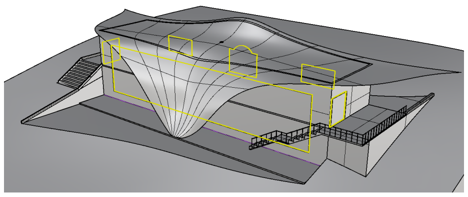
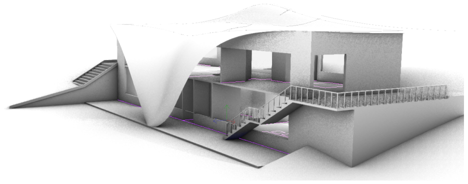
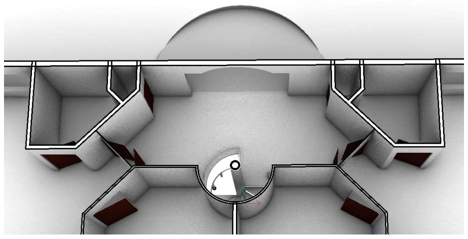
Create a block - Copy one of the steps into new layer - Select the step geometry, then run Block to define a new block. The selected step will turn into a block instant. - Run What command to confirm that it is a block instant.	
Edit the block - Create additional geometry to add to the step block and put in new layer. - Run BlockEdit to edit, remove and add the new geometry to the block	
Recreate the Steps (run ArrayPolar)	
Create new block to cover the rest of the pole - Create new Layer for the pole block. - Copy step added geometry InPlace - Move to be on top of the top step - Extrude top cap by 0.08m. - Run Block to create a new block - Array (or Copy) the pole block vertically to cover the whole pole.	
Add another block for the spindles - Change CPlane to Front, then bring origin to the spindle center (CPlane, and set origin point). - Change to Plan view - Create profile curve - Use RevolveCrv to create the surface - Cap to create the solid - Block the solid with reference point for the handrail - Change CPlane to Top, locate one spindle on the first step, then run ArrayPolar with the same options as the steps.	

Add the handrail - Use InterpCrv to connect the tops of the spindles. - Pipe through the curve for circle cross section (r=0.02), or Sweep for custom cross section. - Cap the ends of the pipe	

Tutorial 503: Outdoor details	
Create outdoor stairs - Change CPlane to align with the stairs profile - Use Polyline to draw one step - Copy to cover all steps - Join all polylines - ExtrudeCrv to create the steps - Create solids for the staircase slab when needed	
Create stringer - Adjust stringer concept surfaces to appropriate height - ExtrudeSrf to create the solid	
Create rails - Create cylinder on the first step, then turn into a block. - Copy rails. - Polyline to create the curve for the handrail - Pipe the handrail	

Create entrance surfaces - Draw entrance shape - Extrude to intersect with terrain - Create pavement path and project to terrain surface - Copy terrain surface, then trim with pavement curve to create pavement surface	

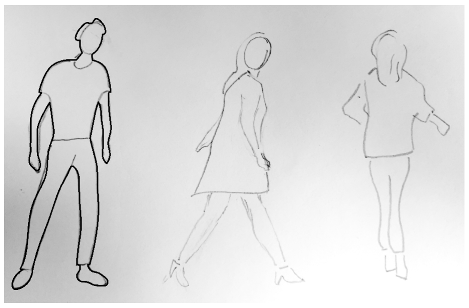
Tutorial 504: Floor layout

Sketch floor layout - Set CPlane to each floor - Use ClippingPlane to be able to view and work on each level. - Use Plan view - If you need to use other viewports, use SynchronizeCPlanes - Draw general floor layout	
Add walls and windows - Draw rectangle of the floor, then use Slab to create the exterior wall. Make it high enough to intersect the roof, and Trim with the roof surface - Draw openings and use MakeHole to create holes in the wall slabs - Move openings outline curves inwards, and give thickness	
Add interior partitions - Use Slab to create interior walls - Trim walls with the roof surface using BooleanSplit	
Add doors. The general steps are: - Align CPlane to the wall, draw the door rectangle, then MakeHole. - Offset all sides of the rectangle, except the bottom, then close the curve to create the frame outline. Then ExtrudeCrv into a solid. - PlanarSrf the original rectangle and move mid frame	

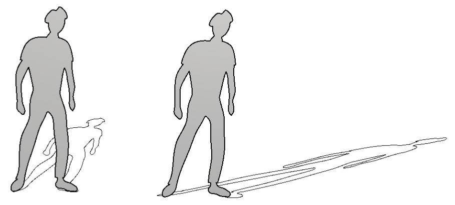
Tutorial 505: Custom Figures

Create figures outlines in Rhino

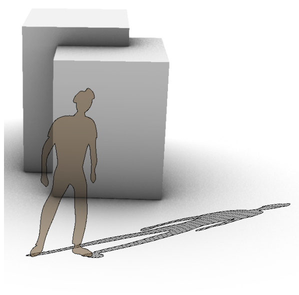
- Insert the picture into the top viewport (use Picture command).
- Lock the image (use the Lock command) and you may need to turn off GridSnap
- Trace the figures using the control point curve command (InterpCrv or use Curve command). Note that: two points in a row will give you a tangent on the endpoint, while three points in a row will allow you to get a portion of the control point straight.
- If you end a curve and need to continue from where you left, then use ContinueInterpolateCrv.
- Make sure it is a valid closed curve (check with What command).

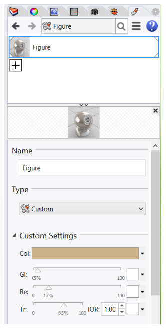


- Copy and rotate the figure (use alt + rotate 90° on the gumball widget).
- Select the vertical figure and create a surface (use PlanarSrf command)
- Scale the horizontal figure in one direction mimicking the sunset long shadow (use Shear or Scale1D).



- Create a Hatch inside the horizontal curve
- Choose your prefered Hatch pattern and rotate it at 90° using the pattern rotation in the Hatch dialogue.
- Hide your curves
- Add transparency to your figure (use the material Editor).





Name: Figure

Type: Custom

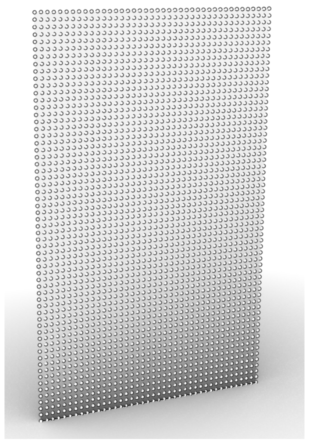
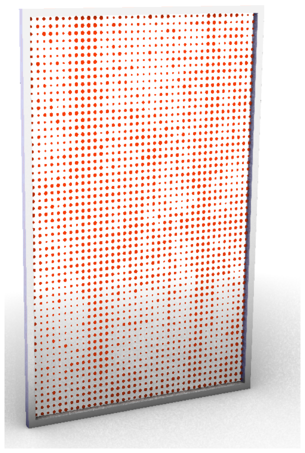
Custom Settings

IOR: 1.00

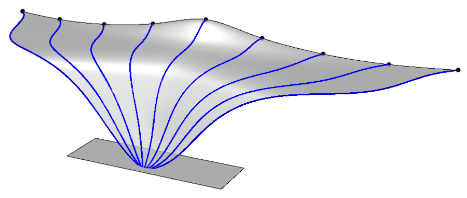
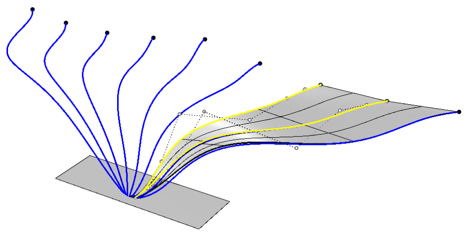
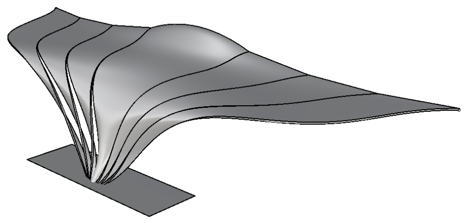
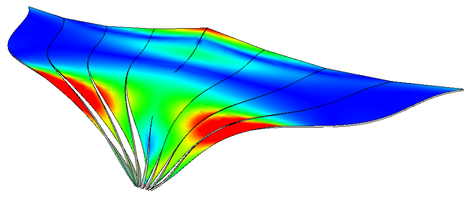
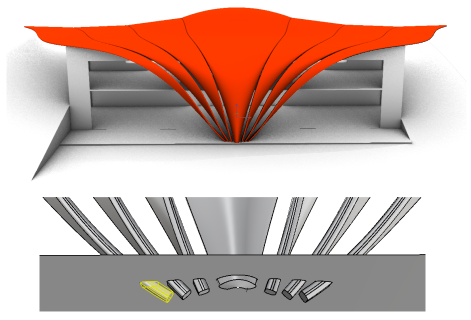
Rationalization

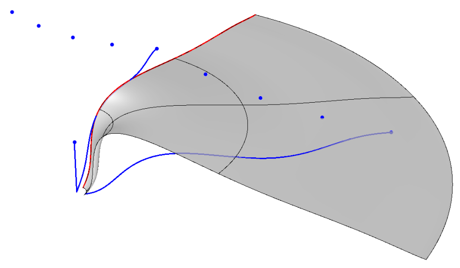
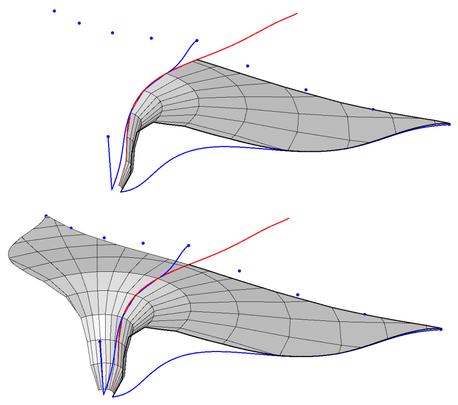
The concept of rationalization is closely tied with building and fabrication processes. Understanding the materials and building workflows is essential to turn models into buildings. Sometimes designers start with abstract form with little consideration about buildability and cost. This can potentially lead to compromising design intent. It is very useful that designers be informed about fabrication processes and workflows to improve communication with other building professionals. The following tutorials go over a couple of common workflows. One to generate custom perforation, and the other is to rationalize free-forms.

Tutorial 506: Custom paneling

Use an image to print on a screen. This image is taken in Torrey Pines, San Diego.	
Create the interior partition with perforation based on regional picture - Install PanelingTools plugin from **[here]** - Create partition surface (1.8x2.9) - Create paneling grid using ptGridSurfaceDomainNumber with 40x60 u and v spans Note: to increase the amount of detail in the partition, you can use more dense grid	
Create custom perforation - Run ptPanelGridCustomVariable with Scale and Bitmap options. Use a Circle with center point for the pattern. - Split the surface with the curves and put circle surfaces in one layer and the remaining surface with holes in another - Change layer material for the surface with holes to be glass, and the circular surfaces to be paint. - Create frame (Offset outline square, PlanarSrf the 2 squares, then ExtrudeSrf)	

With complex free forms, it is usually a compromise between form and cost. For the roof, we will first create curved panels, then explore strategy to build it with flat panels (more economic, but might involve changing the form).

Tutorial 507: Roof panels	
Twist roof segment to vertical louvers - Extract vertical IsoCurves on equal intervals including end edge curves - Duplicate inner curves - Draw lines at the base	
- Each 2 adjacent curves, fit control points to the base line. - Rotate corresponding control points right above the base by 90 degrees, and the ones above them by 45 degrees (use inner most control point as rotation base). - Loft the curves - Note: try to extend the end of the curve below the platform to be able to cut evenly after adding thickness.	
- Repeat for all 4 bays - Mirror to complete	
Perform curvature analysis to check continuity across the fins.	
- Add thickness and FilletEdge to soften the form and add joints between the bays - Trim out geometry below the platform level with BooleanSplit.	

Tutorial 507: Flat panels	
Redefine roof surface so that it can be rationalized into planar panels - Edit center curve - Create revolve surface to cover half the roof	
Create roof panels (use PanelingTools plugin) - Create the grid with ptGridSurfaceDomainNumber - Use ptPanelGrid with Faces output - Split faces with the outline and remove outside panels - Mirror panels to complete the coverage	

Advanced visualization

After concept visualization there may be a need to more realistically represent a design for further review or marketing purposes. The term ‘rendering’ is often used here, but the definition of what this means can vary. In this section, we’ll define rendering as the assignment of real world materials and the accurate calculation of lighting, reflections and refractions.

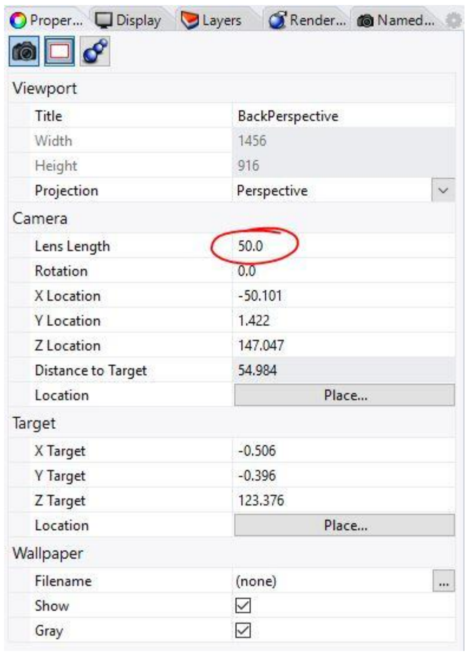
Rhino includes two rendering ‘engines’, Rhino Render and the Raytraced display mode. Both of these will utilize the material library, camera controls, lights, texture mapping and most rendering settings equally with some exceptions.

Tutorial 508: Visualization

Cameras/Views:

Open the “508_DetailedVisualization.3dm” file. There are a couple of named views already present in this model. You’ll already be viewing the model in the one called “BackPerspective”. With nothing selected, look in the Properties panel. Here you can see the lens length used for this named view’s camera. By default the lens length Rhino uses is 50, which matches the perspective you’d have looking at something on a desk in the real world. Let’s change the lens length for this named view to 35 instead, which will give the view more of an architectural perspective.

Then go into your named views panel and save the named view again using the same name to replace it.



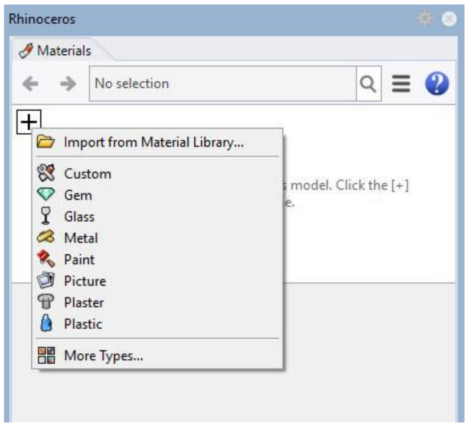
Viewport	
Title	BackPerspective
Width	1456
Height	916
Projection	Perspective
Camera	
Lens Length	50.0
Rotation	0.0
X Location	-50.101
Y Location	1.422
Z Location	147.047
Distance to Target	54.984
Location	Place...
Target	
X Target	-0.506
Y Target	-0.396
Z Target	123.376
Location	Place...
Wallpaper	
Filename	(none)
Show	✓
Gray	✓

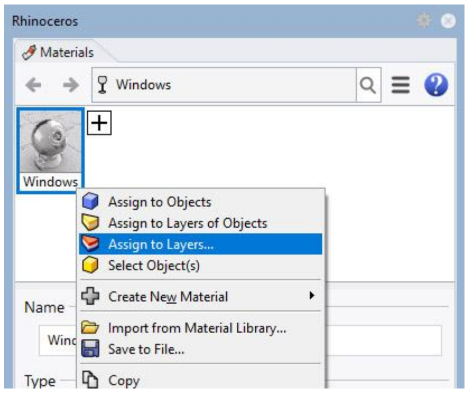
Materials and texture mapping:

To assign materials to the model we first need to create some in the Materials panel. Click the + symbol to load a standard material type like Glass first.

You can rename the Glass material to ‘Windows’ as soon as it’s created by simply typing. Then right click the material thumbnail and choose ‘Assign to Layers…’. Choose the ‘Glass’ sub layer under Openings to assign the new glass material.

Drag and drop of a material swatch onto objects in the viewport or pre-selection of objects followed by ‘Assign to Objects’ in the right click menu of the material thumbnail are alternatives for applying materials.

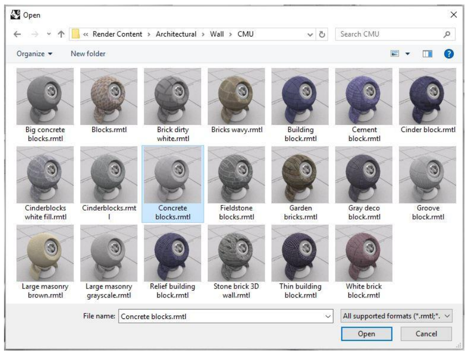


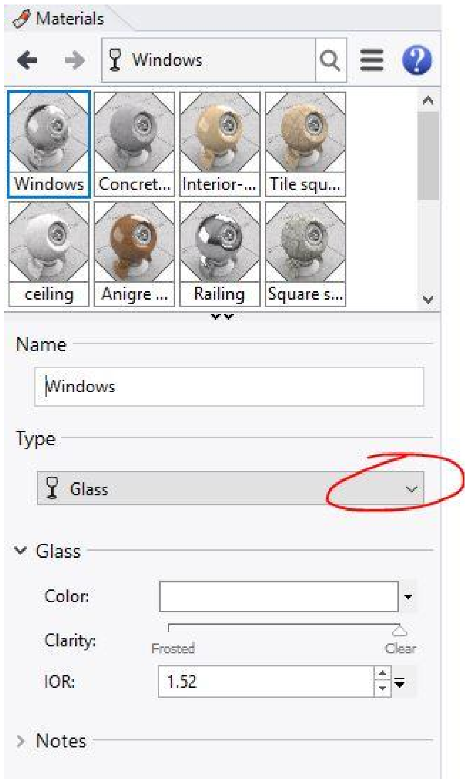
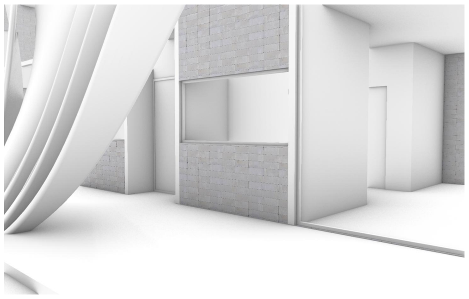


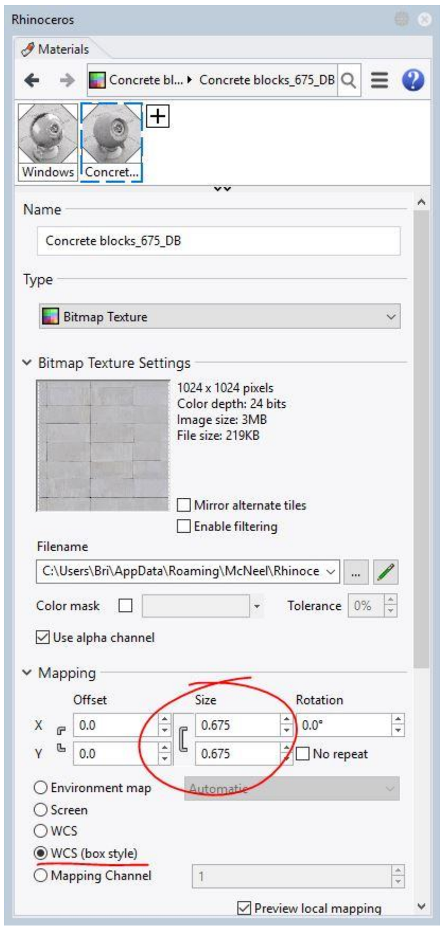
Next, use the ‘Import from Material Library’ option when making a new material in the Materials panel. In Architectural > Wall > CMU select ‘Concrete blocks.rmtl’ and click Open.

The Rhino material library contains many materials that utilizes images also known as texture maps. These are downloading on demand when opening a material for the first time. If you will be offline and want access to all the Rhino material library textures, use the command DownloadLibraryTextures beforehand.

Also note, the window that opens when importing materials is a standard Windows file browser which allows for thumbnail viewing and searching with text filters.



Standard material types like glass, metal and paint can be converted into "custom" materials if and when you want greater control over their look. Simply click the type drop down menu in the Materials panel to change a material's type.	 Materials · Windows · Name: Windows · Type: Glass · Glass · Color: · Clarity: Frosted / Clear · IOR: 1.52 · Notes
Assign the Concrete blocks material to the ExteriorWalls layer. Then zoom in to see the scale of the texture map. By default, any imported materials from the Rhino material library will be using what is called "World Coordinate System" mapping or WCS for short. This mapping method uses a set real world size for the image textures within the material. Click the texture name in the color channel for the material to see the settings for the texture map. Note that the size of the texture is set to .675, the model is using Meters as the unit of measurement so this is the scale of the texture shown in the preview once applied to the model. The texture is also mapped onto objects using WCS and the box style of projection. This is great for linear objects like walls. If you want to change the texture scale you can do so globally here by altering the size. If there is more than one texture in the material, you will need to adjust each one independently. This will impact all objects with the material assigned. To use non-WCS mapping methods such as planar, cylindrical or a custom unwrap for more complex forms, choose 'Mapping Channel' in the texture's settings and edit the selected objects texture mapping properties in the Properties panel.	


Rhinoceros

Materials

Concrete bl... ▸ Concrete blocks_675_DB

Name: Concrete blocks_675_DB

Type: Bitmap Texture

Bitmap Texture Settings

1024 x 1024 pixels
Color depth: 24 bits
Image size: 3MB
File size: 219KB

☐ Mirror alternate tiles
☐ Enable filtering

Filename: C:\Users\Bri\AppData\Roaming\McNeel\Rhinoce

Color mask ☐ Tolerance 0%

☑ Use alpha channel

Mapping

	Offset	Size	Rotation
X	0.0	0.675	0.0°
Y	0.0	0.675	☐ No repeat

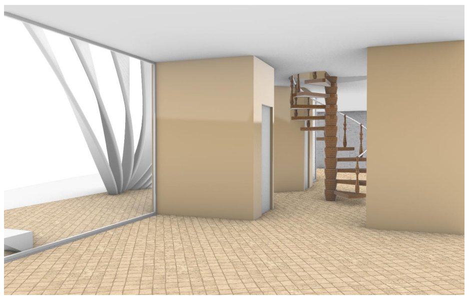
○ Environment map — Automatic
○ Screen
○ WCS
● WCS (box style)
○ Mapping Channel — 1

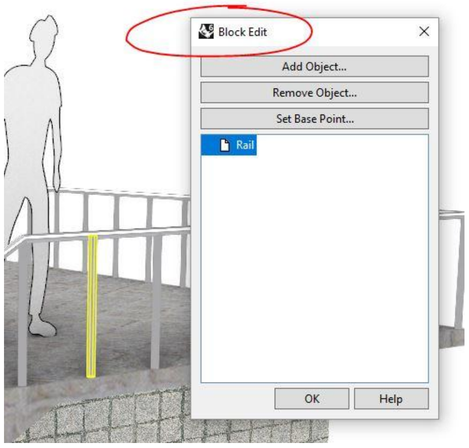
☑ Preview local mapping

Experiment with adding more materials to the model from both the standard material types and the library. Here I've added a Paint material to the interior walls layer and a Tile material from the library's Architecture > Floor category.

It is sometimes necessary to assign different materials to the same object. The slab for the second floor for instance in this model also represents the ceiling for the ground floor. To assign a separate material to the bottom surface of this polysurface, Hold the Shift + Ctrl keys to sub-object select just that surface. You may then right click any material swatch to assign that material only to the selected sub-object. A default Plaster material will work well here as a flat ceiling white paint.

If you are assigning materials to block instances, such as the balusters for the exterior railing, you must do those assignments within the block editor. The advantage of using blocks here is that assigning the material once during an edit will also update the material of all other instances of that block in the scene.

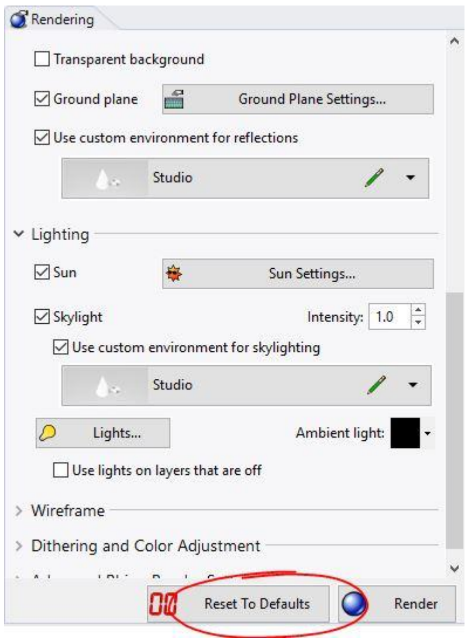




Lighting / Exteriors:

If a model is created in Rhino 6 it will automatically use a collection of new default lighting settings. These can be seen in the Rendering panel and include a default skylight environment for both the color of the lighting as well as reflections.

If a model has been imported to Rhino 6 from an earlier version of Rhino, you can click the 'Reset to Defaults' button at the bottom of the Rendering panel to use the Rhino 6 defaults.


Rendering

☐ Transparent background

☑ Ground plane — Ground Plane Settings...

☑ Use custom environment for reflections — Studio

Lighting

☑ Sun — Sun Settings...

☑ Skylight — Intensity: 1.0

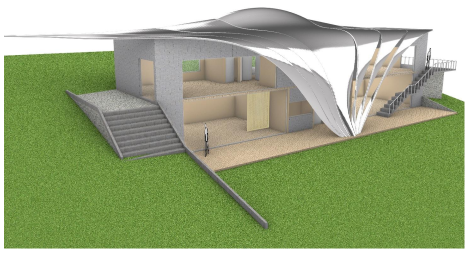
☑ Use custom environment for skylighting — Studio

Lights... — Ambient light: ■

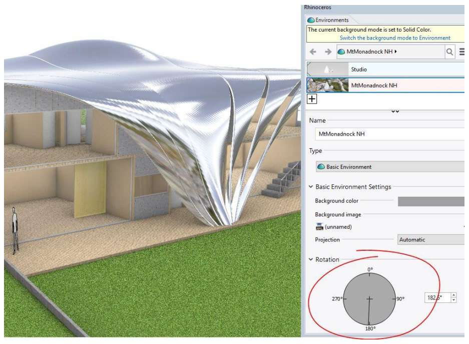
☐ Use lights on layers that are off

Exteriors are much easier to set up and will calculate faster than interior shots. Let's first look at exterior views of the model to render. Additional materials have been applied as before using sub-object materials where needed such as the interior wall surfaces of the exterior wall polysurfaces. The Grass bright material from the Organic > Grass category of the material library has been used for the basic ground cover.

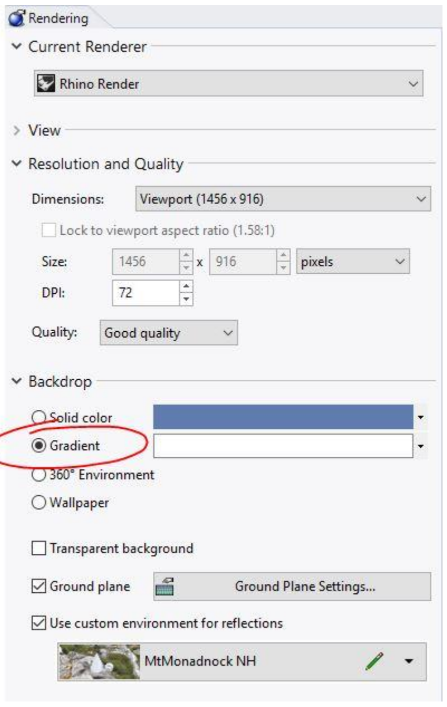
For the roof, a standard metal material with the hatch bump texture selected was used. The reflection of an environment sky becomes important with a reflective roof material. In the Rendering panel, change only the reflective environment to one from the library by using the drop down menu and choosing 'use new environment' >'Import from environment library', the Mt. Monadnock one for example should work well. Open the Environments panel to adjust the rotation of the environment reflections.



In the Rendering panel you can use the reflection environment as the visible backdrop but let's use a simple gradient instead to keep the focus on our building.

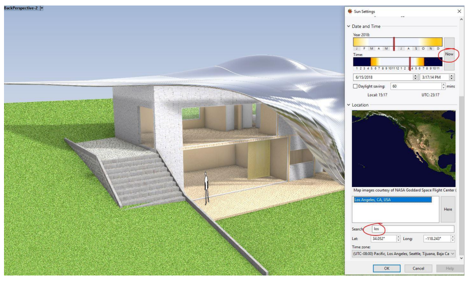


Rhinoceros
Environments
The current background mode is set to Solid Color.
Switch the background mode to Environment
MtMonadnock NH
Studio
MtMonadnock NH
Name: MtMonadnock NH
Type: Basic Environment
Basic Environment Settings
Background color
Background image: (unnamed)
Projection: Automatic
Rotation: 182.5°

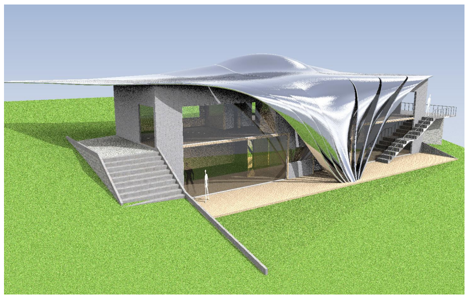
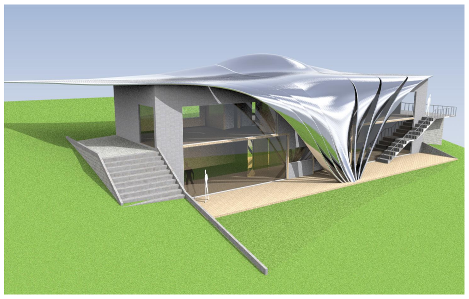
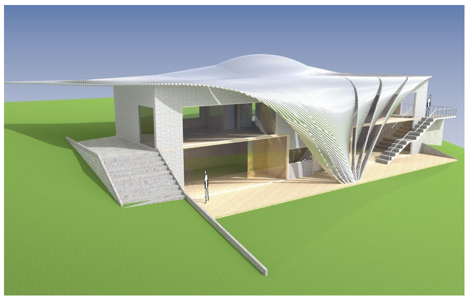
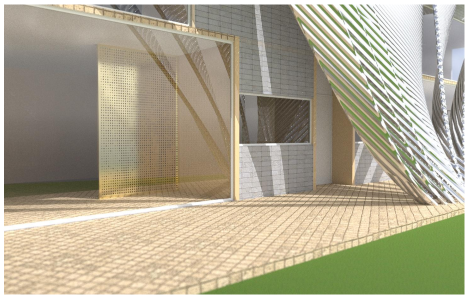


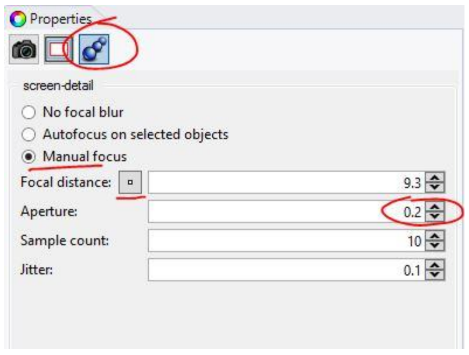
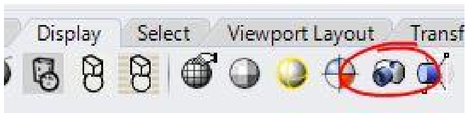
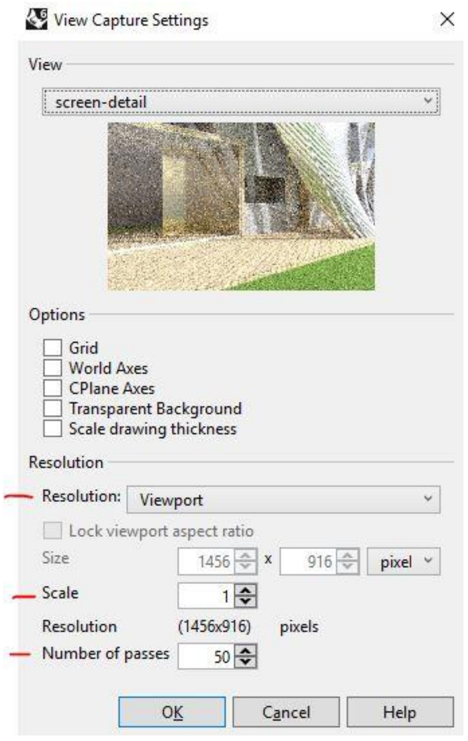
Rendering
Current Renderer: Rhino Render
View
Resolution and Quality
Dimensions: Viewport (1456 x 916)
Lock to viewport aspect ratio (1.58:1)
Size: 1456 x 916 pixels
DPI: 72
Quality: Good quality
Backdrop
Solid color
[✓ Gradient]
360° Environment
Wallpaper
Transparent background
✓ Ground plane — Ground Plane Settings...
✓ Use custom environment for reflections
MtMonadnock NH

Exteriors are more realistic when the Sun is enabled as well. In the Rendering panel > lighting section, turn the Sun on and open the Sun settings to set the location and time. Our site is in San Diego so either enter the exact lat/long or use Los Angeles which will come up in the search.



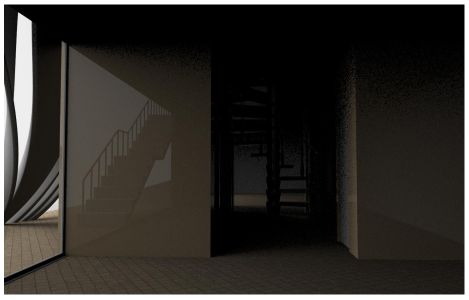
Sun Settings
Date and Time
Year 2018:
Time:
6/15/2018 3:17:14 PM
Daylight saving: 60 mins
Local: 15:17 UTC: 23:17
Location
Map images courtesy of NASA Goddard Space Flight Center (
Los Angeles, CA, USA
Search: los
Lat: 34.052° Long: -118.243°
Time zone: (UTC-08:00) Pacific, Los Angeles, Seattle, Tijuana, Baja Ca

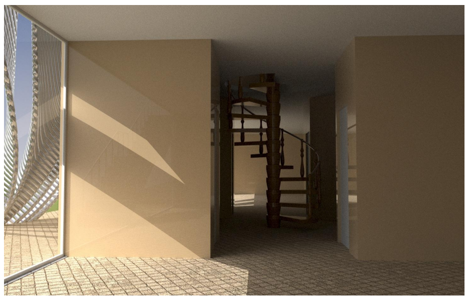
With Rhino Render set in the Rendering panel, use Low quality and do a test render to check overall lighting and contrast. Shadows, reflections and bump maps will appear different from the Rendered display mode. The speed of the rendering is determined by several factors. The pixel dimensions of the image rendered and the quality set in the Rendering panel being the most significant. Here's a comparison of low quality taking 3 mins. versus good quality taking 33.	 
Another option for Rendering in Rhino 6 is the Raytraced display mode which interactively uses a rendering engine called Cycles. You can pick the devices in your computer to calculate the rendering in Rhino Options > Cycles. If you have an Nvidia GPU with Cuda in your computer, your speeds can increase greatly. Multiple GPUs can also be used in unison. Here's the same scene rendered to 500 samples in 3 mins in Raytraced mode. Note that since Raytraced mode uses Cycles, color saturation and materials won't look exactly the same as Rhino Render and overall will produce a different look.	
The Properties panel has focal blur settings when nothing is selected and that view is active. These focal blur settings get saved in any named view. Raytraced mode (shown here) as well as Rhino Render can use this feature. Focal blur helps direct the viewer's eye to an area of interest in the composition.	

	 Properties screen-detail No focal blur Autofocus on selected objects [✓ Manual focus] Focal distance: 9.3 Aperture: 0.2 Sample count: 10 Jitter: 0.1
To save a Raytraced image use the [ViewCaptureToFile]() or [ViewCaptureToClipboard]() commands. Make certain that the settings for the resolution are set to Viewport, that the scale is set to 1 and that the samples field does not have a higher value in it then the number of samples achieved in the Raytraced viewport prior to running the command. If these settings differ, a reprocessing of the Raytraced view will begin.	 Display Select Viewport Layout Transf  View Capture Settings View screen-detail Options Grid World Axes CPlane Axes Transparent Background Scale drawing thickness Resolution Resolution: Viewport Lock viewport aspect ratio Size 1456 x 916 pixel Scale 1 Resolution (1456x916) pixels Number of passes 50 OK Cancel Help

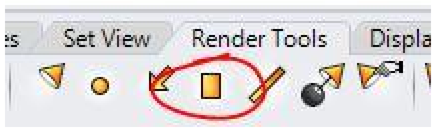
Lighting / Interiors:

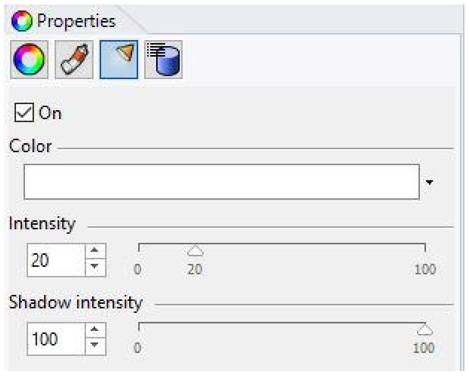
Rendering interior scenes often requires the addition of light objects which makes set up a bit more difficult and will also increase rendering times. You may have already noticed in experimenting that Rhino Render and Raytraced mode differ in terms of contrast with interiors. This is because the Raytraced mode / a.k.a. Cycles, calculates what is called indirect illumination or more simply the bouncing around of light. Rhino Render does not. Here are two views without any additional lights for a comparison of Rhino Render and Raytraced.





You can add a variety of lights in the Render Tools toolbar group. Follow the prompts in the command line and edit any lights via the Properties panel > Light section when a light is selected.





☑ On

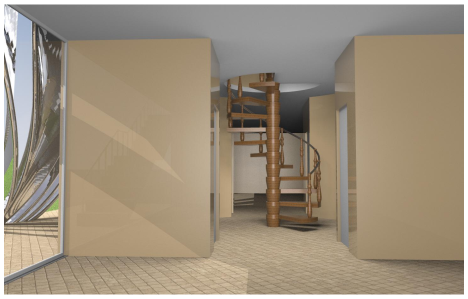
Color

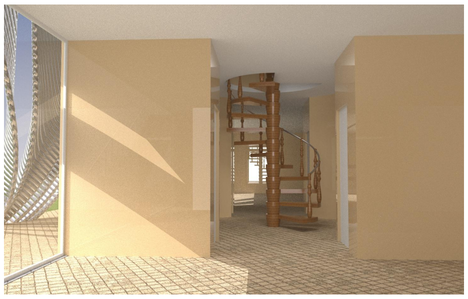
Intensity: 20 (0 – 20 – 100)

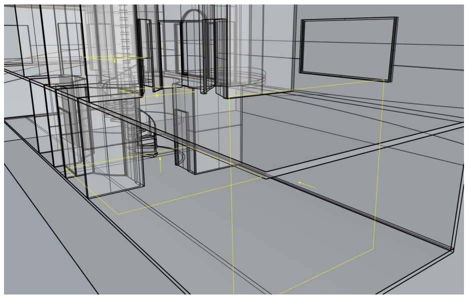
Shadow intensity: 100 (0 – 100)

In these two renders, the first with Rhino Render and the second with Raytraced, four additional rectangular lights have been added. Due to Raytraced calculating the bounced light or indirect illumination, the intensity of the lights was lessened to 20% from the intensity of 60% used for Rhino Render.

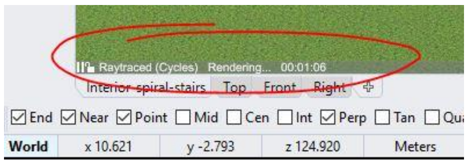
Here are the locations of the rectangular lights highlighted in the scene.







An additional note on the Raytraced display mode. At the bottom of the viewport using Raytraced there will be a set of controls for pausing the calculation, locking the viewport from accidental rotation and at the right end the sample count which can be clicked and edited directly while running Raytraced. There's no target sample count that means "done" for all scenes but in general 500 to 1000 samples will produce a high quality image.

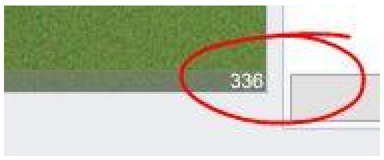


Raytraced (Cycles) Rendering... 00:01:06

Interior-spiral-stairs Top Front Right

☑ End ☑ Near ☑ Point ☐ Mid ☐ Cen ☐ Int ☑ Perp ☐ Tan ☐ Qu

World x 10.621 y -2.793 z 124.920 Meters



Drawings

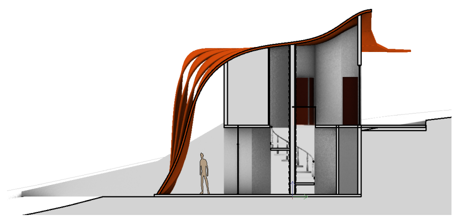
Extracting 2D data out of the 3D model is useful design and documentation workflow. You might need to slice through your model to extract useful reference geometry, or use to laser cut your form, but the more common use of 2D drawings is for documentation. Delivering 2D drawings is a requirement in most architectural projects. Documents might be needed to issue building permits and to share with contractors for construction. Rhino is essentially a 3D modeler with viewports that represent 3D space. It also has paper space that can be set up to organize 2D data and print to scale. The workflow involves two main parts. The first is extracting the 2D drawings, and the second is to lay them out to scale in paper space. The following tutorials show a workflow for extracting and printing drawings.

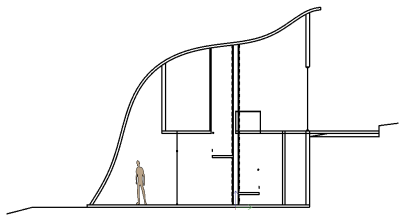
Resource 6: 3D printing tutorials
Rhino Layout workflow document: https://wiki.mcneel.com/rhino/layouts5

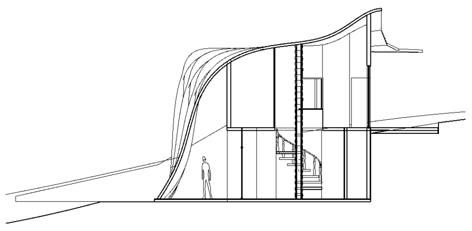
Tutorial 509: Sectioning

You can use Rhino built-in commands to extract sections and elevations using Section, Contour and Make2D commands in Rhino, and view using ClippingPlanes. While very effective for quick snapshot of the 2D views, there are some limitations:
- Extracted sections are static (they don't change when the model changes),
- Extracted Sections and Contours need some work to project or reorient the sections from the true 3D location to and other flat plane that the drawings are placed on (usually World Y-Plane).
- Can extract only curves, and need extra work to clean curves, join and create caps, hatches, etc.

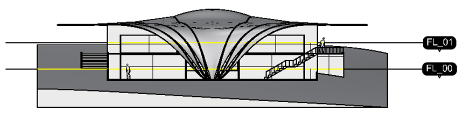
Using Rhino drawing commands:
- Run ClippingPlane and draw at desired location. You can render the view with the clipping plane active. This gives raster image of the model.
- Run Section command to go through the model in the desired direction, then put output in new layer. Output is vector drawing (can print with high resolution or export to other vector based applications).
- Run Make2D to extract objects outlines behind the section. Note that Make2D output is placed on WorldXY. You can set one of the views to be World-Top to view output. Note that the Make2D output is vector based.







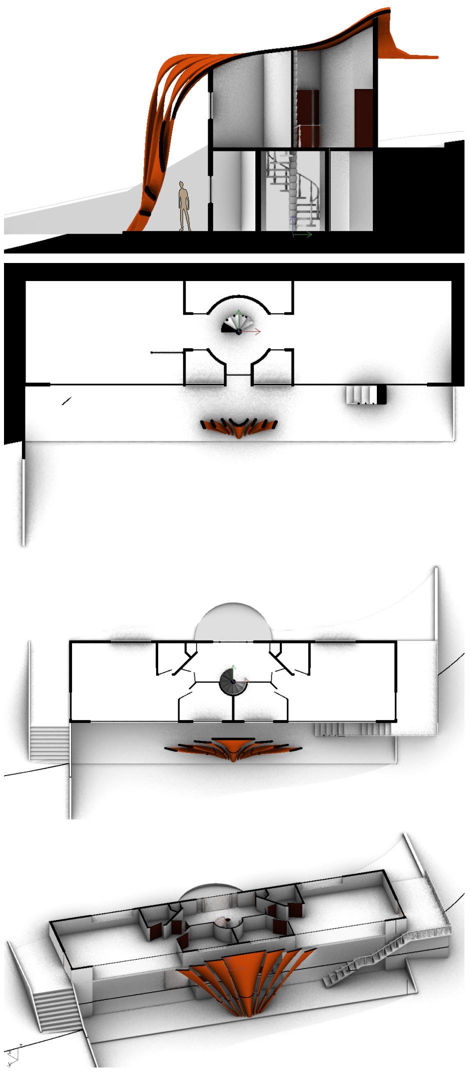
SectionTools plugin supports dynamic sectioning that updates with model changes. It also resolves some of the workflow limitations described above.
- Install SectionTools plugin for Rhino 6
- Use stCreate, and select objects to section (you can press "Enter" to section through ALL visible objects. This is a good option if you know you need to update the model).



View sections in 3D space
- Run stViewSections, select the section and set options to "Yes". Select the view to view the section in.
- Use stClearSectionViews to clear the clipping from a selected viewport

Edit sections
- stEditSections to reset options
- stEditSectionsObjects to select specific objects to section, or section through all visible objects
- stEditSectionsHint to edit what part of the section hint to view
- stMoveSections to move sections

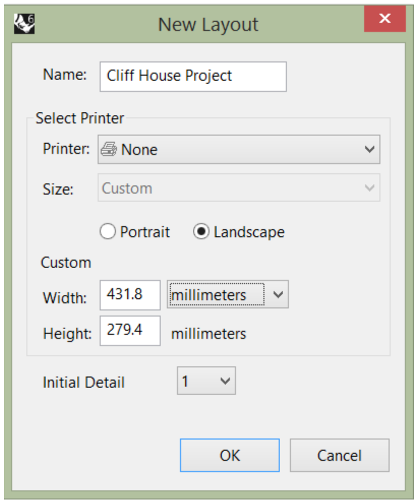
Save desired views as NamedView to be able to set to when needed in the paper viewports



Tutorial 509: Layout

Create a new layout:

- Use Layout command to set up a new layout.
- Enter the layout name, paper dimensions and the number of initial details.
- You can use any units to set paper size. That will not affect the actual units of your layout space (which we have set up to millimeters).



New Layout

Name: Cliff House Project

Select Printer

Printer: None

Size: Custom

Portrait [✓ Landscape]

Custom

Width: 431.8 millimeters

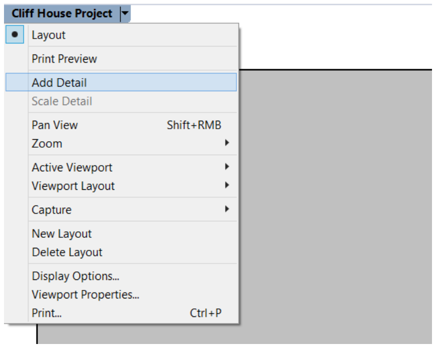
Height: 279.4 millimeters

Initial Detail 1

OK Cancel

Add new Detail:

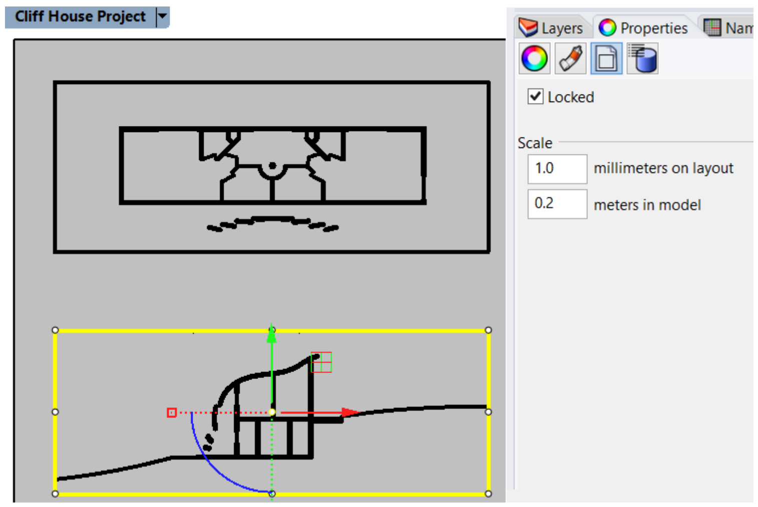
- Press the arrow in the Layout title, and select “Add Detail”
- Drag a window to define the boundary of the detail.
- Double click inside the detail to activate the viewport and be able to Pan and Zoom to the part of the model you need.



Set the detail to scale:

- Select the detail outline
- In the Detail properties, set the scale (1:200) and check the Locked box. Once you lock, you cannot zoom or pan the view to ensure that the scale remains consistent.

Note that you can set different scale for each detail.

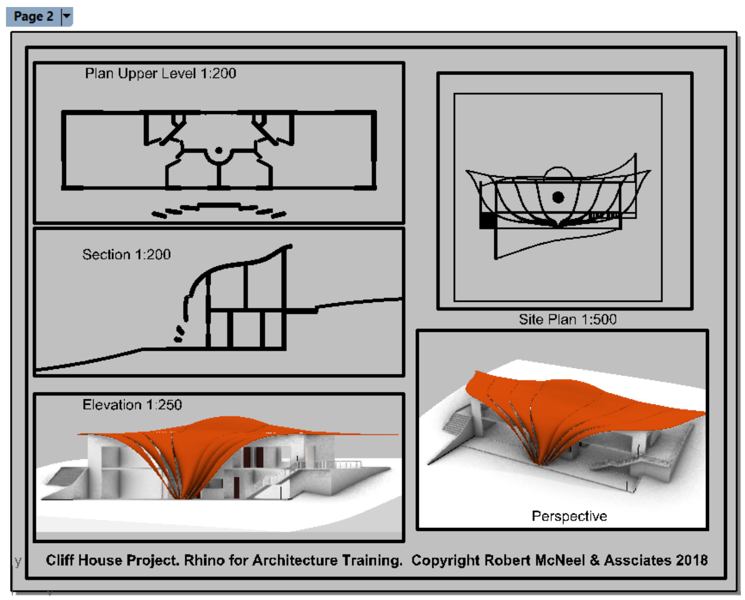


[✓] Locked

Scale

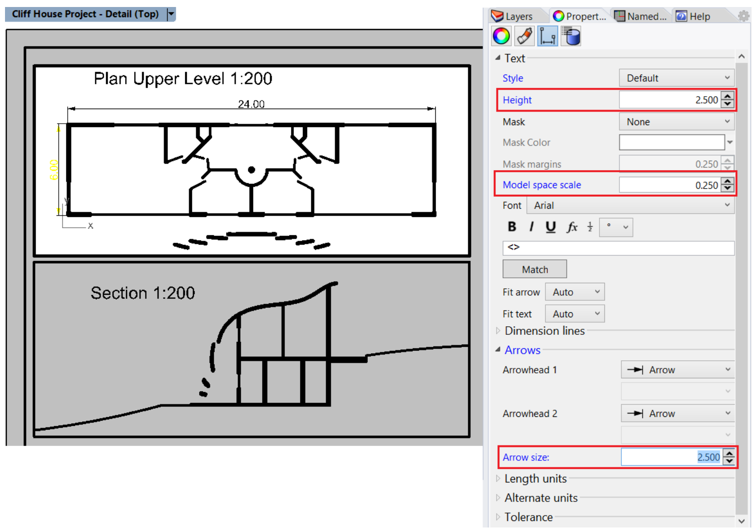
1.0 millimeters on layout

0.2 meters in model

Rendered views and labels: - Enter other details and set the view and projection. - You can use any display mode for the details. - Add Rectangle as a border for the layout - Inset labels using Text command	

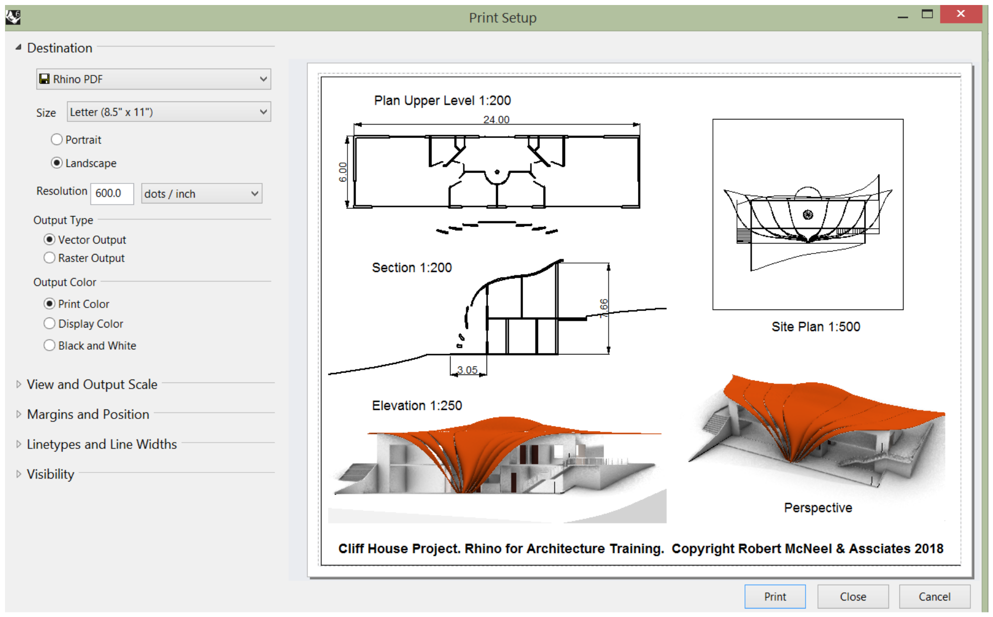
Page 2
Plan Upper Level 1:200
Section 1:200
Elevation 1:250
Site Plan 1:500
Perspective
Cliff House Project. Rhino for Architecture Training. Copyright Robert McNeel & Assciates 2018

Tutorial 509: Annotation

It is recommended that you do dimension in Layout view, and you put them in separate layer. This way you can control their visibility. Note that dimensions <u>DO NOT</u> update dynamically with model changes. - Create a new layer for Dimensions. - Create dimensions using Dim command. The default scale might be tiny. To adjust the scale: - Select the dimension, and in properties. Set Height to bigger number of units (e.g. 2.5 mm). - Pumping the paper space scale of the dimension results in bigger size dimensions in model space. Set "Model space scale" to be a fraction of the paper space (0.25) - Make sure to set the Arrow scale to match the Height.	

Style: Default
Height: 2.500
Mask: None
Mask Color
Mask margins: 0.250
Model space scale: 0.250
Font: Arial
<>
Fit arrow: Auto
Fit text: Auto
Arrowhead 1: Arrow
Arrowhead 2: Arrow
Arrow size: 2.500

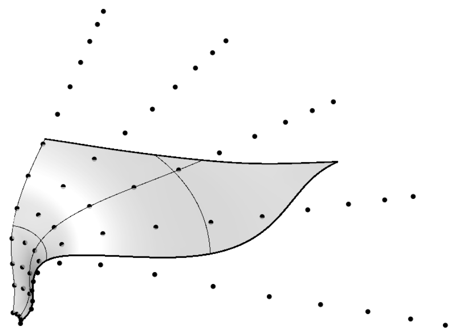
Tutorial 509: Printing

Print PDF using Print command. Set the resolution and other options.	 Print Setup Destination Rhino PDF Size: Letter (8.5" x 11") ( ) Portrait (●) Landscape Resolution: 600.0 dots / inch Output Type (●) Vector Output ( ) Raster Output Output Color (●) Print Color ( ) Display Color ( ) Black and White View and Output Scale Margins and Position Linetypes and Line Widths Visibility Plan Upper Level 1:200 Section 1:200 Site Plan 1:500 Elevation 1:250 Perspective Cliff House Project. Rhino for Architecture Training. Copyright Robert McNeel & Assciates 2018 Print · Close · Cancel

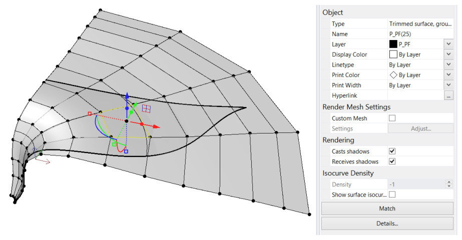
Prototyping

Prototyping is an integral part of concept development and detailed design. It is as much a design tool as it is for communication and presentation. The nature of digital modeling with virtual screen based representation is bound to miss important aspects that are only noticed through physical modeling. Prototyping uses CAD/CAM systems such as laser cutters, CNC machines and 3D printers. We will discuss two workflows using 3D printing and laser cutting, and CNC routing.

Laser cutting

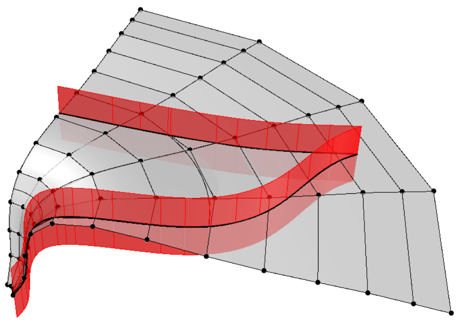
Tutorial 510: Laser cutting	
Use rationalized roof panels (flat panels). For this tutorial, we will rationalize with reduced number of panels. - Project the outline of the original roof onto the revolve surface and trim with it (without shrinking the surface). - Divide the surface into 4x12 spans using ptGridSrfDomainNumber (PanelingTools plugin)	

Create the panels
- Use ptPanelGrid to generate the panels as Faces
- Notice that faces are grouped and each face has a unique name

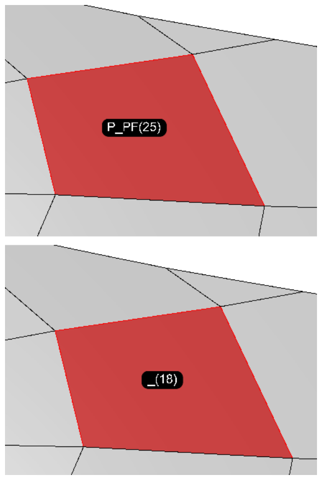


Object	
Type	Trimmed surface, grou...
Name	P_PF(25)
Layer	■ P_PF
Display Color	□ By Layer
Linetype	By Layer
Print Color	◇ By Layer
Print Width	By Layer
Hyperlink	...
Render Mesh Settings	
Custom Mesh	☐
Settings	Adjust...
Rendering	
Casts shadows	✓
Receives shadows	✓
Isocurve Density	
Density	-1
Show surface isocur...	☐

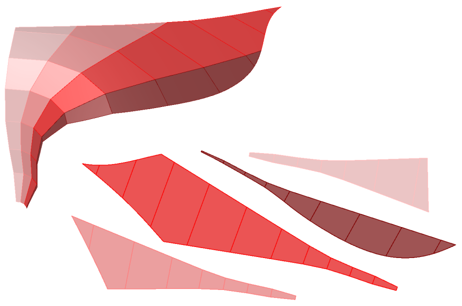
Split panels
- Extrude outline curve
- Use Split to split all panels
- Delete the part of panels outside the boundary



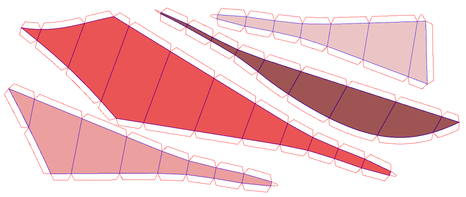
Labeling
- If the autolabel for panels is not desirable, you can relabel it using ptSerializeObjectsName
- To tag objects with their name, use ptTageObjects



Unrolling: you can unroll, tab and glue individual panels, but also look for a more efficient way if the form allows it.
- Join each step of panels, and make sure the polysurface normal faces outwards.
- Unroll together using UnrollSrf
- Change the color of steps to make it easier to distinguish



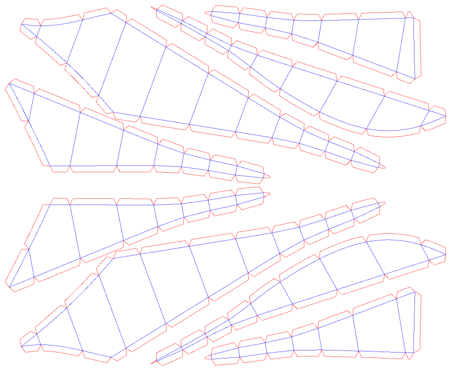
Create tabs
- Create new layer and choose color RED and make it the current layer (many laser cutting programs designate the red color for cutting and blue for etching)
- Arrange your strip to fit in the laser cutter bed (check your laser cutter dimensions)
- Use ptTabs with Recess option to create the cut outlines



Extract etch curves

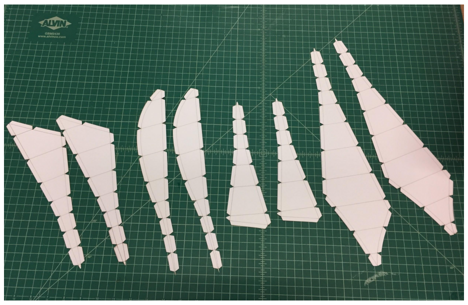
- Explode trips, the DupBorder to get the curves
- Explode the curves then remove duplicates (SelDup, then Delete)
- Put curves into a new layer and make its color BLUE
- Mirror all curves for the second half of the roof
- Select all cut and etch curves, and Export using file format suitable for your laser cutter

Note that if you need to bend panels in 2 different directions, then use a dashed line red cut curves for etching instead of blue continuous curves.
Also, take into consideration the final scale of the printed part.



Cut panels using the laser cutter

Put it all together by gluing corresponding tabs

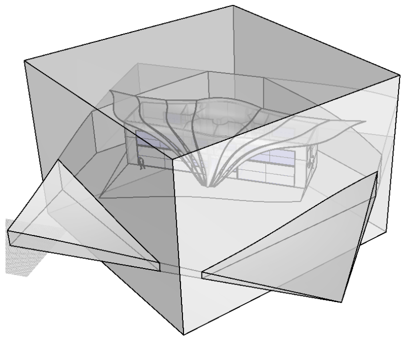


3D printing

Tutorial 510: 3D printing

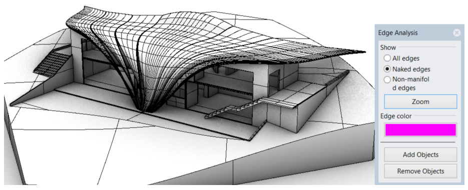
Check your printer tolerances. 3D printers have a minimum thickness that they can print successfully.

- Set model scale to fit the final print size. Usually set units to millimeter.
- Make sure all thicknesses are within tolerance. Recreate the solids to have proper thickness even if it is a little different from the design intent.

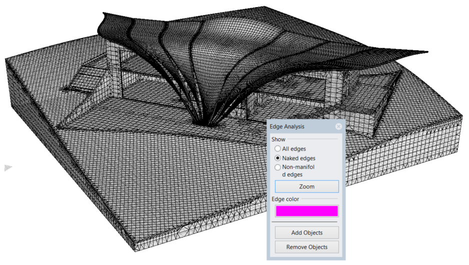
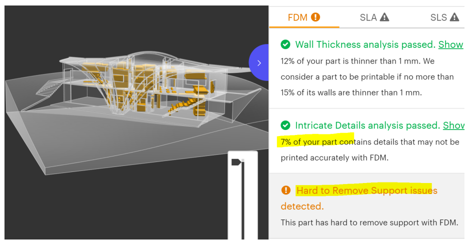
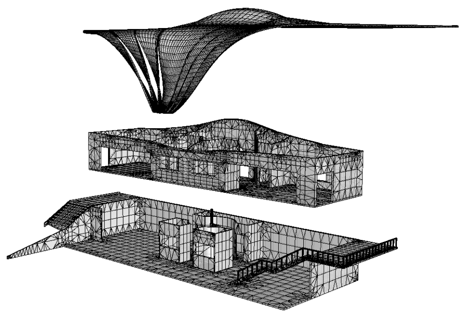


- BooleanUnion as many polysurfaces as you can. Ideally end up with exactly one polysurface.
- Make sure you have CLOSED polysurfaces.

Note that you may need to do extra operations to Boolean successfully and make thickness within printer tolerance. You may use OffsetSrf, Move objects to overlap, etc.

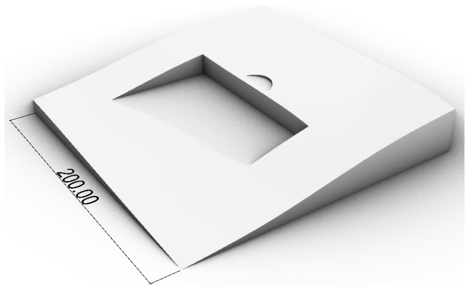


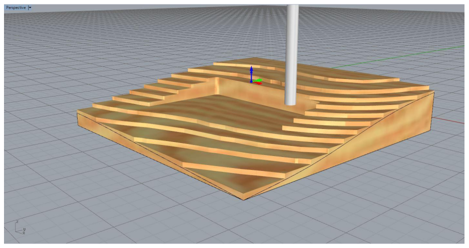
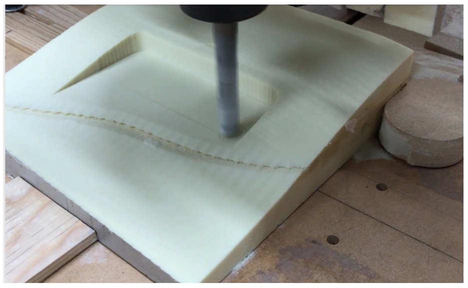
Edge Analysis
Show
All edges [✓ Naked edges] Non-manifold edges
Zoom
Edge color 🟪
Add Objects
Remove Objects

- Extract a mesh with reasonable face count. 3D printers have a limit of how many faces they can print, and minimum size. - Check the mesh to make sure it has no naked edges or holes. Repair when needed using MeshRepair tools and other Rhino mesh commands - Export using proper format acceptable by your 3D printer. Usually as STL or OBJ	 Edge Analysis Show All edges Naked edges Non-manifold edges Zoom Edge color Add Objects Remove Objects
- It is good practice to supply the Rhino file with the original polysurface as well as the mesh to the 3D printing service. This way if there is a scale or thickness issue, they can modify the original model and export for better resolution.	 Wall Thickness analysis passed. Show 12% of your part is thinner than 1 mm. We consider a part to be printable if no more than 15% of its walls are thinner than 1 mm. Intricate Details analysis passed. Show 7% of your part contains details that may not be printed accurately with FDM. Hard to Remove Support issues detected. This part has hard to remove support with FDM.
- Total volume and support material can be an issue for cost and profitability of the parts. - For the final print with Resin, the option to break the model into 3 parts gave best results in terms of cost and ability to print with min support. The cost was reduced from ~$500 to $75.	

Resource 7: 3D printing tutorials
3D printing workflow document: https://wiki.mcneel.com/rhino/3dprinting 3D print tutorial for a product design example: https://vimeopro.com/rhino/preparing-to-3d-print

CNC routing

Tutorial 510: CNC Routing	
Scale the model and adjust model to work well with your router tolerances Check for angles and adjust undercuts and angles when necessary depending on your CNC machine specifications and tools Depending on the CAM tool used, you might need to export as a mesh. Note that the mesh does not need to be closed like in 3d printing. CNC routing model is very forgiving. However, you need to know you machine to adjust angles and scale.	

CNC settings and simulation using RhinoCAM	 ***[Show simulation video]***
Final cut site model using CNC machine	 ***[Show milling video]***

Part III: Modeling methods

Architectural design involves creating ideas using a language of expression such as geometry and materials; and a medium of representation such as drawings or modeling. The media of representation tends to influence our design thinking and methodology. This is particularly true in the digital medium of design. Different digital tools are designed to support the different ways of modeling.

To understand modeling methods using digital means, let us start with a simple example. Suppose you want to create a composition out of circles with varying radii. You might start with drawing on paper, cut and experiment with different compositions interactively. This method uses a very familiar "tool" (pen and paper), and a basic understanding of geometry (circles have a center and circular curve). One also needs a bit of practice to come up with good hand sketched circles.

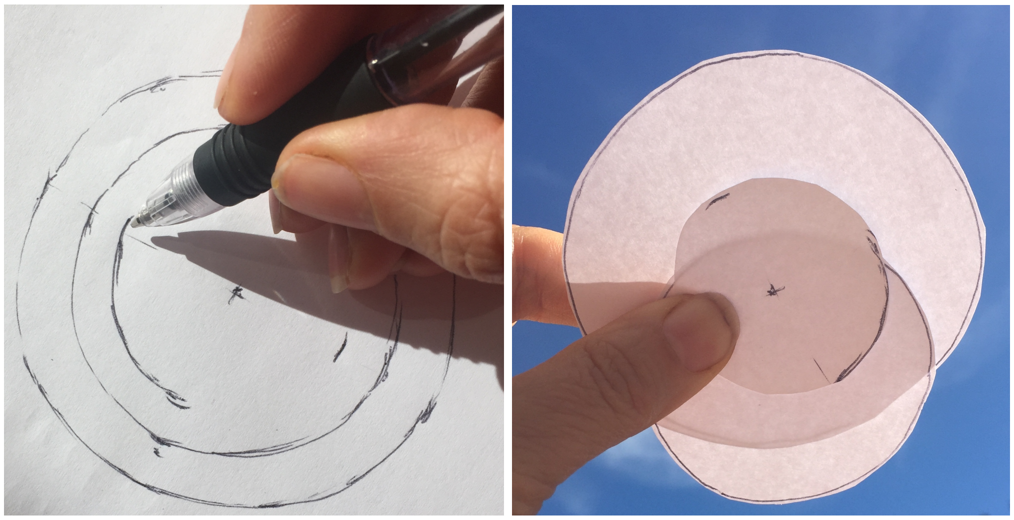


Figure (1): Hand drawing circles

Benefits of using pen and paper to represent geometry are plentiful. Drawings are cheap, available, easy to pass around, and we all learn how to draw with pencils from a very early age. Drawing medium is also great at keeping a record of all the attempts. However, there are some disadvantages, for one, it is hard to draw accurate circles by hand. Also, drawings inked on paper are hard to change. Designers have invented many tools and methods to help with drawing challenges. For example, a compass to improve accuracy, tracing paper to layer sketches, and so on. Just like drawing, digital representation of design ideas has its advantages and challenges. To fully understand those, we will examine a range of digital design methods.

Digital modeling supports a wide range of workflows for different stages of design; from intuitive modeling with vague ideas about the final result, to more structured and well-defined designs.

For example, you might start with some surface, have not decided yet if it is a wall, a ceiling, or simply a guiding geometry for something to follow. At that stage, you need flexibility to shape and mold your ideas. If your tools force you to decide on the material or building part, then your flexibility to change your mind and morph your initial thoughts into new ones will be hindered. On the other hand, if your design has matured and you need to deliver your model to engineers for evaluation, you need to have made detailed decisions about building parts and materials.

Creators of digital tools examine these needs and workflows closely and create digital tools that suit different stages and modes of design. In general, the creators of digital tools tend to support one of four modeling approaches: direct, algorithmic, object-based or parametric. Some of these overlap to a certain degree, especially the parametric one which we will address in a separate section. The main characteristics of the first three approaches are summarized in the following.

Direct Modeling	**Algorithmic Modeling**	**Object-based Modeling**
Uses abstract geometry	Process driven	Uses well-defined objects such as building parts that embed information about geometry materials, and other attributes
No restriction on the forms created or processes used to create them. Highly adaptable to designers preferences	Incorporates mathematics, logic and algorithmic processes to define forms and relations	Uses a library of parts to assemble a model
Suitable for intuitive conceptual design	Desirable for parametric design	Efficient when modeling specific building types and styles, and for production.
Abstract geometry is usually highly portable across digital tools, which favors its inclusion in many design workflows	Can communicate with external tools, but is dependent on the environment it is developed with.	Rigid. You are usually limited to the workflow designed by the environment itself and its family of tools. There are some industry standards to help interoperability.
Easy to understand and manipulate across team members	Well formed and clear algorithms is key to collaboration. It is easy to generate unreadable scripts that are hard to understand and manipulate.	Relying on standard objects is a significant advantage to help consistent modeling among team members.

Table (1): Comparison of digital modeling methods

The Rhinoceros core modeling environment supports direct modeling through its intuitive geometry creation methods and the rich set of tools to manipulate, analyze and share geometry. The Rhinoceros core is very easy to extend into specialized functionality through plugins. It has an open source file format (openNURBS), and much of its core is built using the same development tools available to all third party developers. Grasshopper, which started as a Rhino plugin and now ships with Rhino, has become a standard tool for algorithmic design. Its intuitive visual programming method, coupled with the powerful Rhino geometry engine make it very popular among designers and building professionals. Many plugins for Rhino and Grasshopper support specialized workflows. For example VisualARQ uses object-based modeling with standard libraries of building parts. ArchiCAD, which is a stand alone object-based tool, has a plugin linking Rhino and Grasshopper in real time.

Tangent 1: VisualARQ, BIM plugin for Rhinoceros and Grasshopper
VisualARQ makes it very easy to work with building parts (walls, windows, etc.). It is tightly related to both native Rhino geometry types (easy to transition between the two), and the algorithmic environment of Grasshopper (for example, it can define blocks algorithmically, and build relational models of the building). For more information about VisualARQ, go to http://www.visualarq.com

The accessibility and ease of adding to Rhino and Grasshopper resulted in a growing ecosystem containing hundreds of specialized tools for analysis, interoperability, robotics, visualization, and others. Most of these tools are shared for free and present an incredible resource. The inclusive nature of the Rhino and Grasshopper and its affordability make it a great choice in research and practice.

Tangent 2: Food4Rhino lists plugins for Rhinoceros and Grasshopper
Hundreds of plugins for Rhinoceros and Grasshopper can be downloaded from the food4Rhino website. Most are free: http://www.food4rhino.com

Direct modeling

Direct modeling is very comparable to pen and paper. After you familiarize yourself with Rhinoceros user interface (2 hour tops), you can start modeling with simple geometry such as curves and surfaces. For example, creating circles in Rhinoceros involves simple steps:

- Find a computer that has Rhinoceros installed,
- Run Rhinoceros and start a new file,
- Run “Circle” command (find it in toolbars, menus, or type the word “circle” in the command line)
- From here, you are guided through the steps (instructions are typed in the command line). You’ll be asked to specify a center, radius, all with some nice preview to see what you’ll get before you commit or accept.
- Run “Circle” command a couple more time snapping to the same center point, but with different radius.
- For now, ignore the many different ways you can create a circle that are offered by the “circle” command, or else you’ll be faced with more things to learn and decisions to make.

You end up with something like the following:

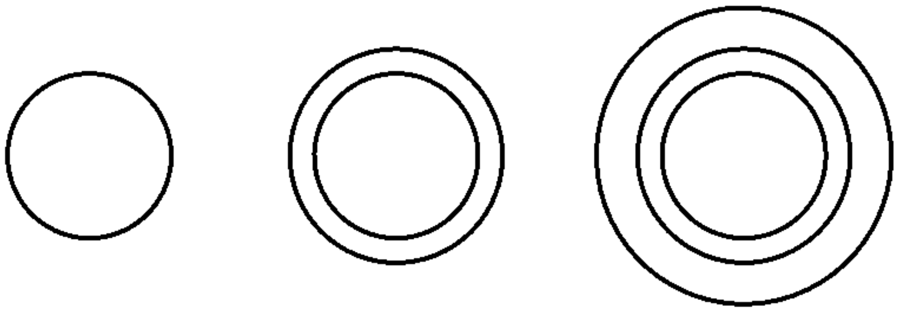


Figure (2): Modeling circles in Rhino

Designers usually get the hang of direct modeling fairly quickly for few reasons:

- The general workflow is similar to using pen and paper.
- There is not too much upfront work that the designers need to do other than opening the application and start drawing, albeit digitally.
- Actions (or commands) are typically intuitive and easy to remember. You can guess many of them (type a couple of letters, and the smart autocomplete will pull a list for you to choose from). All tools are also grouped in a logical arrangement in menus and toolbars.
- Once you run a command, it usually guides you through the process step by step.

Once you're comfortable making circles with Rhino, you'll find that there are many advantages that come with direct digital sketching over the good old pen and paper. For example, it is trivial to interactively scale until you are happy with the circles! But beware, if you are not disciplined, you can easily lose your early iterations. Unlike on paper, no trace is left on screen. You can always "undo" but that will only take you a step or two back.

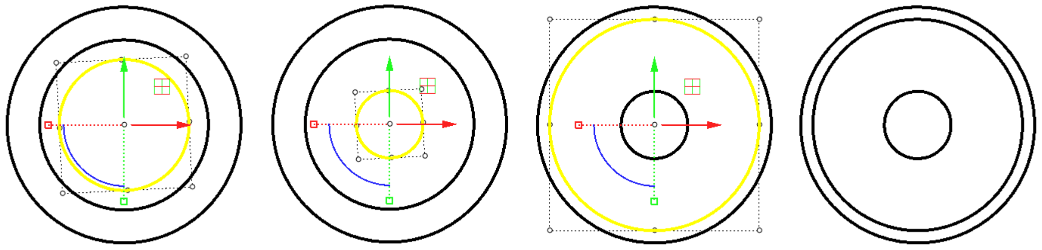


Figure (3): Editing circles in Rhino

Both methods mentioned above (hand drawing, and direct digital modeling) have limitations:

- Remodel every time you need to create the same thing.
- Hard to keep and compare variations.
- Hard to define dependency or relationships between the parts.
- Hard to embed knowledge about the process or the logic of design.
- Involves more work when transition to detailed design and documentation.

If you hit these limitations frequently at your work, then you should consider using algorithmic or object-based modeling.

Algorithmic modeling

Algorithmic modeling requires clear articulation of design problems and the steps to reach the solution. Once you wrap your head around the algorithmic design workflow, you will be able to overcome many of the limitations inherited in other modeling methods. The main advantage is the ability to work on projects early because it is fast to change input requirements and make new updates. It also helps with scalability and redoes of similar problems. There is, however, an upfront cost to learn algorithmic modeling. Knowledge of basic math, geometry, and logic design is essential. More importantly, developing algorithmic design skill requires discipline and commitment to learn. The good news is that you can acquire the skills of algorithmic design with practice and time, and once you get it, it is hard to lose.

It is best to explain algorithmic modeling workflow through an example. We will use Grasshopper to show a typical workflow. Let us start by exploring few different ways you can "define" a circle, or a bunch of them using Grasshopper. Notice I used the word "define" instead of "draw" or "model". The reason will become clear later. First, we will define one circle with a center point and radius. Then we move on to constructing different ways to define concentric circles.

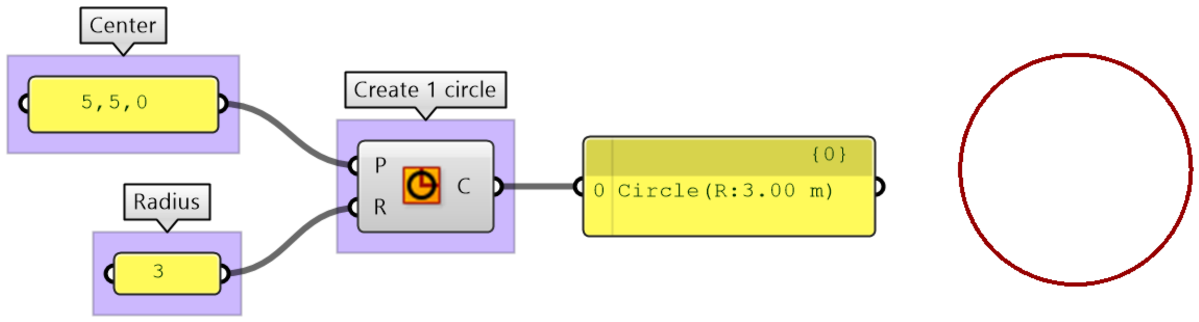


Figure (4): Define one circle with fixed (constant) center and a fixed radius

Tangent 3: How input is processed in the Grasshopper component
Grasshopper supports many ways to define "input" (the values on the left that feed into the circle battery). The input can be one or multiple values, and hence the circle may execute any number of times (one for each input combination). In the above example, the "circle" runs exactly one time using the one (center, radius) combination. The output (coming out of "C") at the right side of the component, show that we have created exactly one circle. A preview of the circle is drawn on the Rhino viewport.

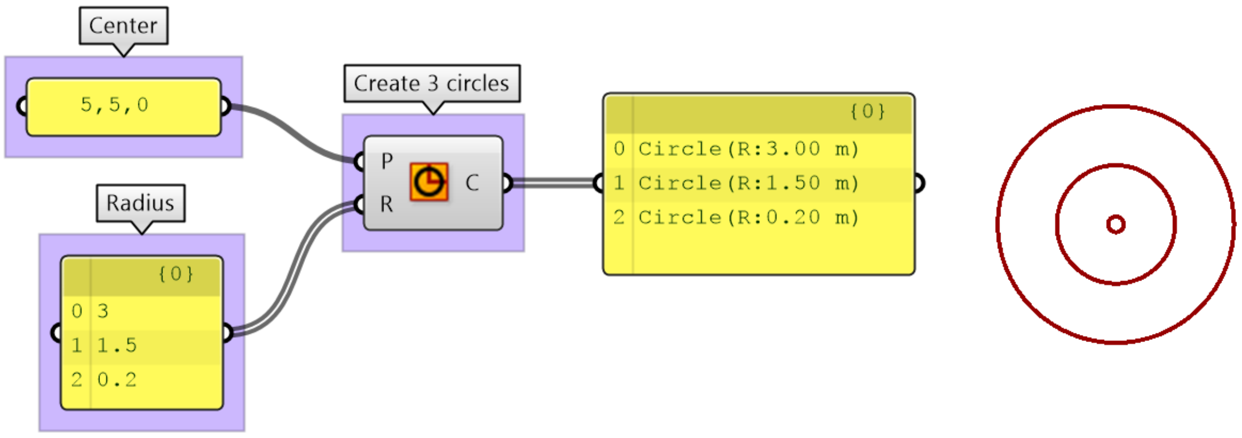


Figure (5): Define three circles with a constant center and constant list of radia

Grasshopper supports multiple ways to define or generate input values. You can directly supply them, but you can also generate them. This is where the power of algorithmic methods shows. For example, if you need 10 circles starting at radius equal 0.2, and increasing by 0.1 all the way to the tenth circle with radius equal 1.2, you can do that in many different ways. You can create a separate definition for each circle with one radius (similar to example 4). You can also supply a list of typed radia (similar to example 5). These two methods are correct, and they mimic the "direct modeling" method of repeating each circle. This is tedious to change and does not exploit the power of algorithmic design. A better way is to generate the list using a starting radius, step size and number of radia. This way if you like to change the initial radius or step size, then you can do that efficiently by changing only one value. You'll notice that definitions resemble building blocks connected to each other to generate the desired output. First, create the list:

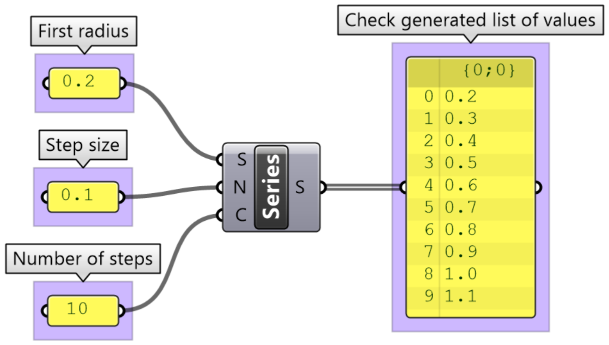
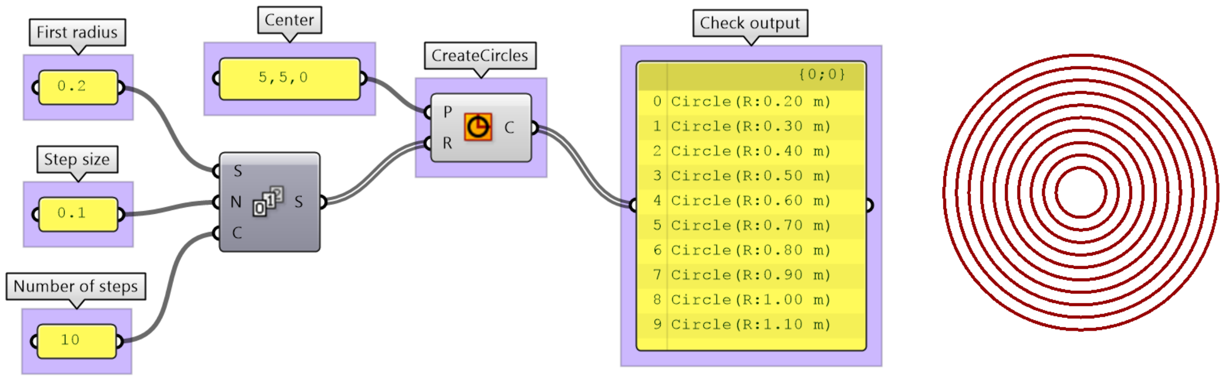


Figure (6): Use Series to generate a list of values

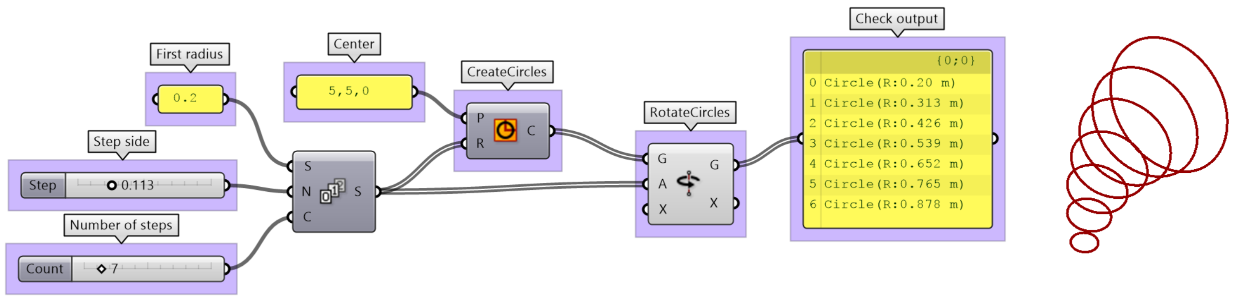
Then use the generated list as input to the circle radius:



```
{0;0}
0 Circle(R:0.20 m)
1 Circle(R:0.30 m)
2 Circle(R:0.40 m)
3 Circle(R:0.50 m)
4 Circle(R:0.60 m)
5 Circle(R:0.70 m)
6 Circle(R:0.80 m)
7 Circle(R:0.90 m)
8 Circle(R:1.00 m)
9 Circle(R:1.10 m)
```

Figure (7): Generative circles

One big advantage of using algorithmic modeling is that it naturally support parametric design. For example, instead of one value, you can define an input parameter as a range of values. You can then interactively change input and observe the effect on your output.



```
{0;0}
0 Circle(R:0.20 m)
1 Circle(R:0.313 m)
2 Circle(R:0.426 m)
3 Circle(R:0.539 m)
4 Circle(R:0.652 m)
5 Circle(R:0.765 m)
6 Circle(R:0.878 m)
```

Figure (8): Parametric circles

Tangent 4: What do components use as input when lists are involved?

Grasshopper has a specific way to generate the list of combinations from input. This is a very advanced topic and has to do with data management, but it is worth mentioning here to give you the heads up about how it works.

The general case is easy to understand. If a single value is supplied to each of the input, then those make one combination, and the component is executed once (as in figure 4). If one of the input is supplied with a list of values, then the component executes once for each value on the list and is combined with the same other single value input. For example, the radius input in figure 5 consists of three values and only one point for the center. Subsequently, the circle component executes or runs three times using the following combination for (center,radius): ((5,5,0), 3), ((5,5,0), 1.5), ((5,5,0), 0.2). The output coming out of "C" at the right side of the component shows three circles.

Now if a list is supplied for each input, the values of the same index are combined. If one list is shorter, the last value on the short list continues to be used.

For more control, Grasshopper makes many tools to manage how data is matched. For example, there are ways to combine each value in a list with **all** other values from the other list, but we can leave this for another time.

At this point, it is useful to define some terms commonly used in algorithmic modeling:

Data	Includes all the values (numbers, text, geometry, color, etc.) that are processed to create the output. These values take two forms: **variables** (fixed or certain) and **parameters** (change, uncertain). Data containers in Grasshopper are all called "parameters".
Data structures	Grasshopper organizes data within three structures: **single**, **list** and **tree.** Properly representing data structures and managing them is a big part of algorithmic modeling.
Functions, methods, and operations	Those typically take input, perform some operation, and produce output. In Grasshopper operations are encapsulated inside what is called "components".
Algorithms	They are the steps (recipe) that define the sequence of operations. It has three main parts: input, steps, and output. Grasshopper files contain one or more algorithms, which are commonly referred to as "definitions".
Parametric design	A method to create design solutions through a set of logical steps (**algorithms**) and values that can be changed (**parameters**) to help efficiently generate design variations. Parametric design is highly supported by algorithmic modelers.

Table (2): Algorithmic modeling concepts and their definitions

So far, we learned that Rhinoceros supports direct modeling familiar to most designers, while Grasshopper is an algorithmic modeling tool that helps articulate the design logic using algorithms. Here is a summary of what Rhinoceros and Grasshopper are good at, and why you might want to use them.

Rhino (direct modeling)	**Grasshopper (algorithmic modeling)**
Captures the intuitive workflow of traditional design medium of pen and paper	Based on computer programming principles, but is made intuitive through visual, rather than text-based scripting
Uses NURBS to represent and manipulate geometry accurately. But also support other types of geometry such as meshes	Rhino geometry commands are embedded in the components, combined with data management tools to support visual algorithmic modeling
Design decisions can be implicit and reflective	The workflow forces explicit definition of all modeling steps
In large, designers do not have to deal with or manage geometry data.	Understanding data types and data structures is essential. Designers have to be aware of data and actively manage them
Offers direct interaction with geometry. Typically work directly in model space.	There is separation between logic and geometry. The design logic is created in a separate space from that where the geometry is displayed.
Making changes may involve remodeling. It is not easy to generate and compare design variations.	The ability to change designs and create variations is perhaps the most pronounced advantage.
Hard to leverage mathematics and algorithmic logic.	Mathematics and dependencies are in the nature of this

Also hard to build model dependencies.	method.
Affordable, stable, accessible in Windows and Mac operating systems.	Fully integrated inside Rhinoceros in both Windows and Mac.
Flexible user interface. Customizable tools using macros. Also, supports many scripting languages (RhinoScript and Python) and plugin development (C++ and DotNet framework).	Provides access to the platform scripting and development libraries and functions. All can be accessed through text-based editors (Python, VB, and C#). Grasshopper functionality can also be extended with compiled Add-ons using the DotNet framework.

Table (3): Characteristics of Rhino and Grasshopper

Object-based modeling

Rhinoceros does not support object-based modeling as part of the core application. However, there are a few plugins that are integrated tightly with Rhino that do support this modeling method. One example is VisualARQ.

Object-based modeling almost never deals with abstract geometry. Designers typically model with objects such as walls, windows, stairs, and roof. These objects behave in predictable way and hold information about materials, cost, 2D representation, etc. Designers use standard objects or create their own. The main advantages of using an object-based modeling application such as VisualARQ can be summarized in the following:

1. IFC is the industry standard format to store objects and their properties. Many object-based applications use this format; therefore, there is good interoperability between them. Many other building applications for analysis and construction support IFC format which makes it more convenient to exchange files.
2. Objects can embed a lot of information about materials, 2D representation, cost, etc. This makes it more straightforward to generate 2D documentation and bills of materials.
3. Consistency in using standard styles and building parts across an organization or in future projects. Once these libraries are established, it is very productive to work with object-based modelers, especially for similar style projects.

Parametric modeling

Parametric modeling is a specialized modeling mode that is gaining popularity. It overlaps with all other modeling methods. In algorithmic modeling tools such as Grasshopper, all input values can be turned into parameters. Object-based modeling is also parametric by nature where changing specifications of any style (height, thickness, material, etc.) leads to updating all instances that use that style. It is less obvious how parametric design is supported in direct modeling tools. That is said, the Rhino core does support parametric design in three ways. The first is embedded in the very nature of the NURBS geometry, the second through blocks and finally when recording command history.

Tangent 5: Transformations and parametric design
Certain parameters are common in all digital tools. All modeling tools allow rotating, scaling, stretching and changing the geometry location. Such operations are called “transformations”. The angle is the parameter for rotation, the scale factor is the parameter for scaling, and so on. Rhino supports all basic transformations, but also a rich collection of advanced ones. Geometry can be stretched, twisted, bent, and flown along a curve or a surface. All Transformations are listed under the Transform menu.

NURBS geometry is parametric

One of the main appeals of NURBS modeling is that it defines geometry using parametric curves and surfaces that are easy to define and manipulate. One of the parameters that describes NURBS curves is called “control points”. Once you create a curve, these points can be dragged to modify the curve intuitively. The same thing is true for NURBS surfaces.

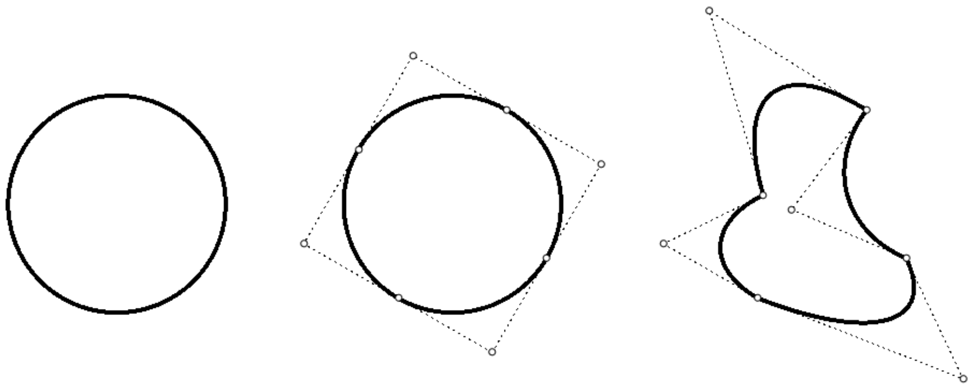


Figure (9): Control points in NURBS geometry as parameters

Tangent 6: What is NURBS geometry?
NURBS stands for non-uniform rational b-splines. A detailed description of parameters curves including NURBS is described in the “Essential Mathematics for Computational Design”: http://developer.rhino3d.com/guides/general/essential-mathematics/parametric-curves-surfaces/

Blocks as parameters

Rhinoceros supports blocks. Once a block is defined, any number of instances can be placed in the model. They all refer to the original block geometry and update when the block is changed.

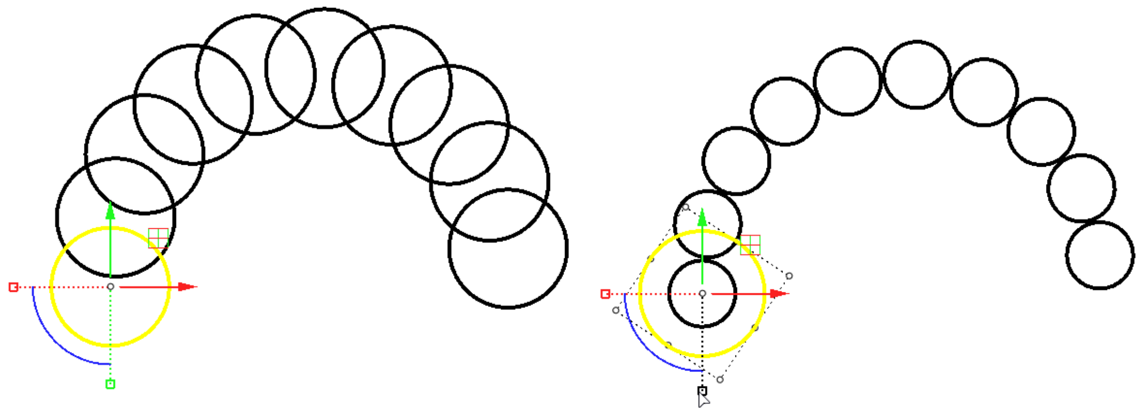


Figure (10): Blocks as parameters

Command history and parametric workflow
Most Rhinoceros commands support history. History is recorded only when you choose to *record history*. Input geometry becomes parameters. For example, if you record history before extruding the base curve of the tower, you can change the base curve to control the extruded form.

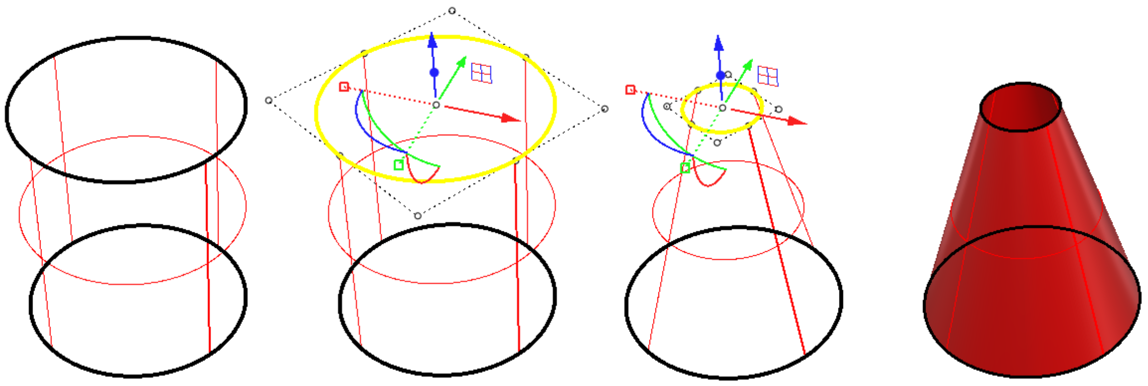


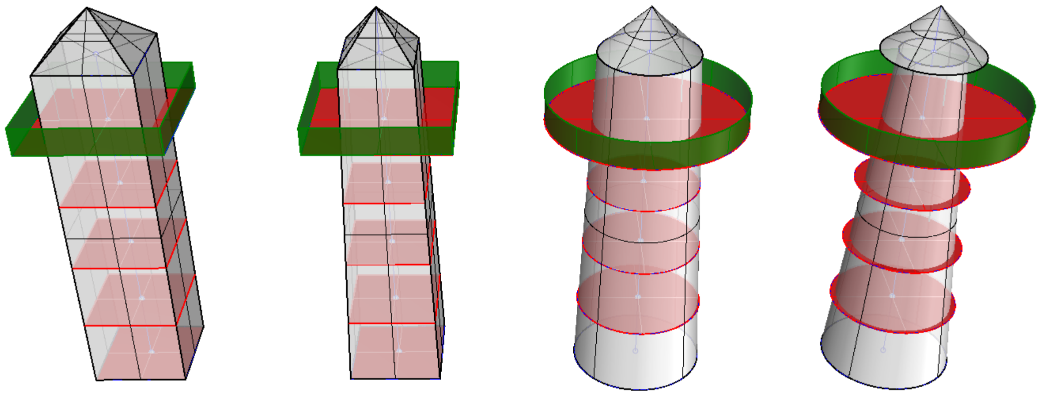
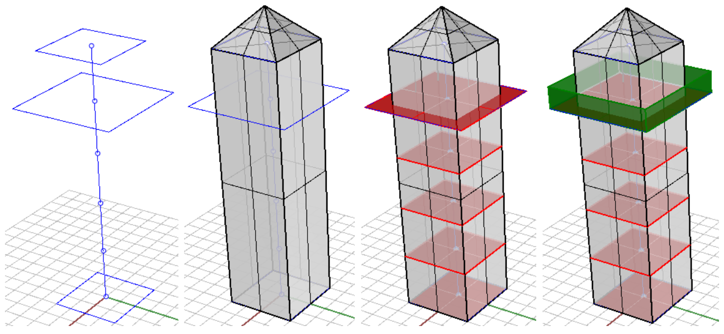
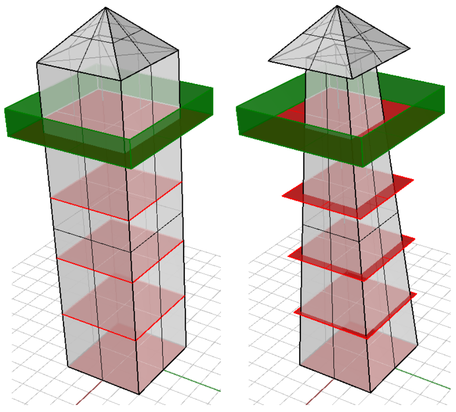
Figure (11): History supports parametric design

Resource 1: History recording in Rhino
History tutorial: https://vimeo.com/261535716

Comparative modeling

All modeling methods are widely used in architectural design, and they can complement each other. Each has its strengths and utility within the design and building workflow. Designers well versed in all methods are usually more productivity and competitive.

To gain an appreciation of all three modeling methods, we will model a simple lighthouse using Rhino, Grasshopper and VisualARQ, all side by side.

Direct Modeling using Rhino	
Explore scaffolding, concept design and detailed design workflows using Rhino core direct modeling workflow. Explore conceptual and parametric considerations. 	
Modeling steps	
Open: 01_ModelingMethods>05_ComparativeModeling>01_RhinoDirectModeling>**01_RhinoDirectModeling.3dm** Create reference geometry for the building shape and levels. Reference is very useful to set the scale the model and define rough outline	
Initial form can be quickly created with commands such as Rectangle, Box, Plane and ExtrudeCrv. Changing the form involves some remodeling No detailed decisions need to be made about materials, building specifications and others	

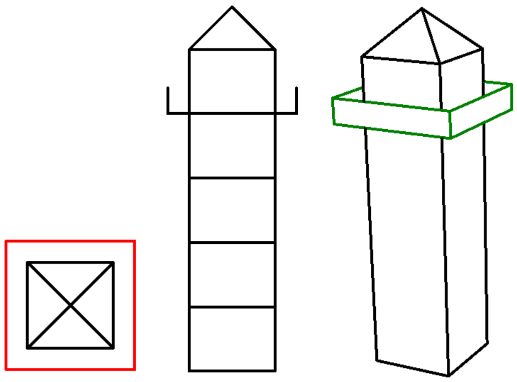
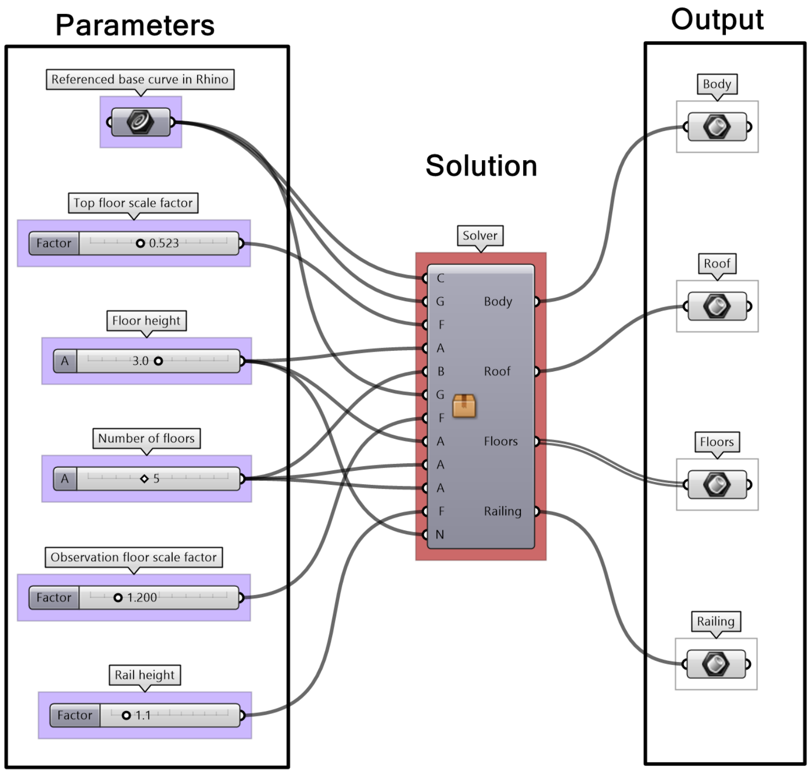
Can extract basic 2D drawings using Section and Make2D commands. Detailed models and documentation that includes wall thicknesses, openings and materials require additional modeling steps.	

Table (4): Direct modeling using Rhino

Algorithmic Modeling using Grasshopper

Explore parameters, concept design and detailed design workflows using Grasshopper algorithmic modeling workflow.

Explore conceptual and parametric considerations.



Modeling steps

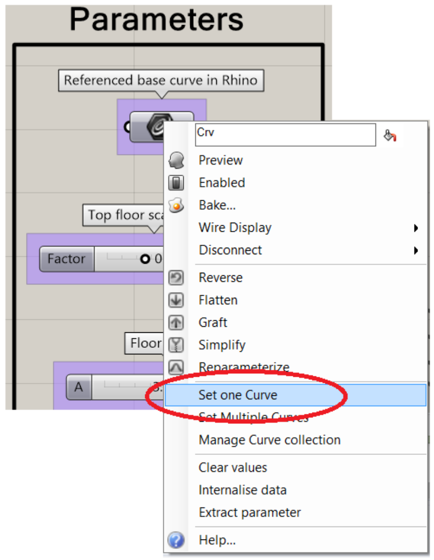
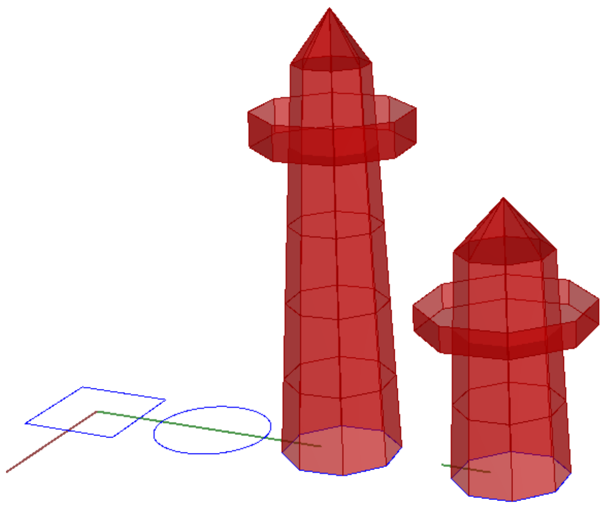
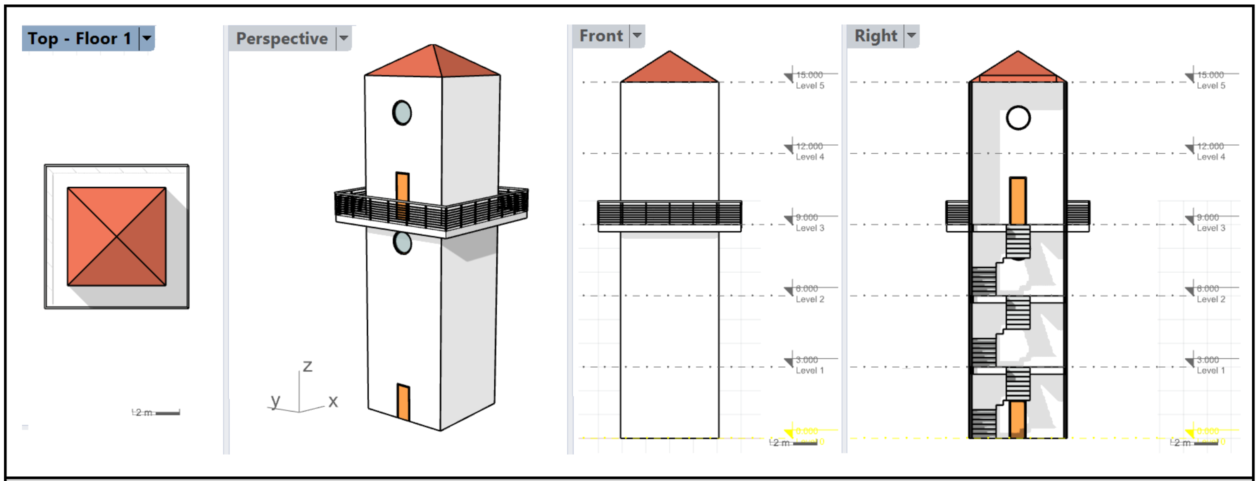
Open: 01_ModelingMethods>05_ComparativeModeling>02_GrasshopperAlgorithmicModeling>**02_GrasshopperAlgorithmicModeling.3dm** Run Grasshopper, and open **02_GAM_DefinitionCluster.gh** Set base curve for the tower with right mouse click on the curve component, then click "Set one Curve". Select the base curve in RHino. Make sure it is drawn in xy-plane.	 Parameters Referenced base curve in Rhino Top floor sca Factor Floor A Crv Preview Enabled Bake... Wire Display Disconnect Reverse Flatten Graft Simplify Reparameterize Set one Curve Set Multiple Curves Manage Curve collection Clear values Internalise data Extract parameter Help...
Experiment with changing the base curve and parameters to create design variations. All geometry is created in display and they do not exist in Rhino. You can Bake output to create Rhino geometry.	

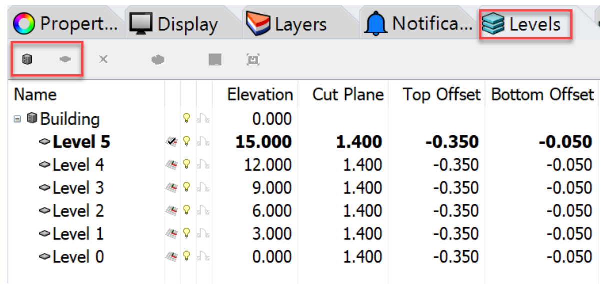
Table (5): Direct modeling using Rhino

Object-Based modeling using VisualARQ
Explore parameters as part of objects, concept design and documentation



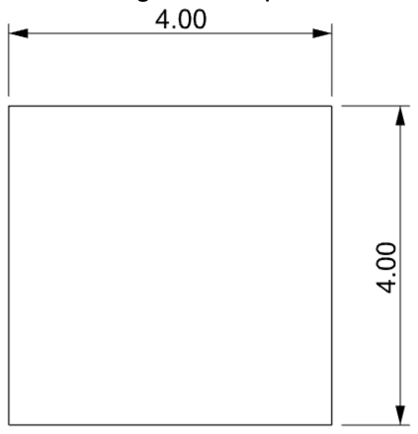
Modeling steps

Open a new VisualARQ Template - Meters.
Go to the Levels Panel.
Start adding Levels, make sure the building layer is highlighted.
Each level has its own CPlane and elevation height.

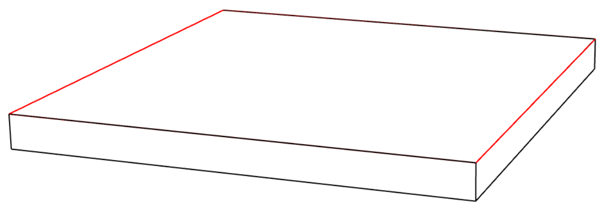


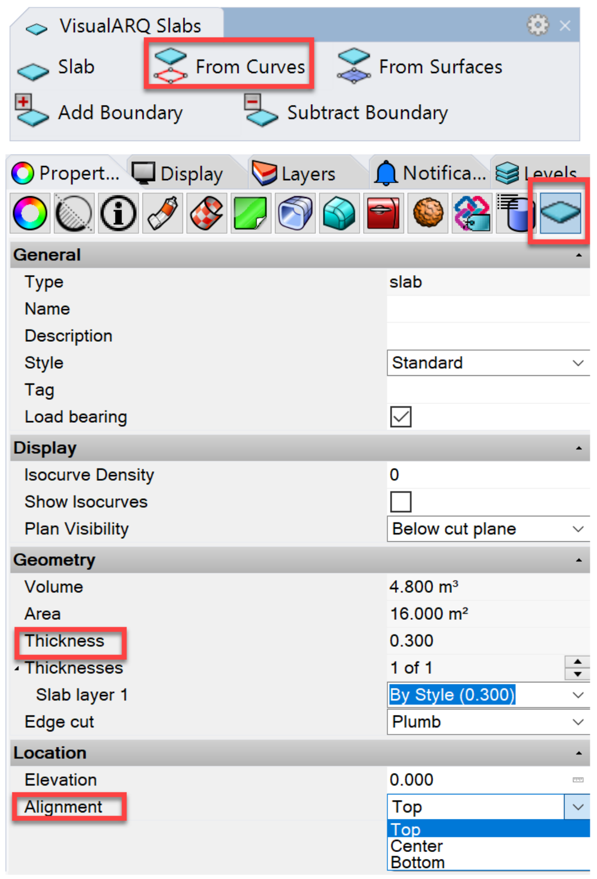
Name	Elevation	Cut Plane	Top Offset	Bottom Offset
Building	0.000			
Level 5	**15.000**	**1.400**	**-0.350**	**-0.050**
Level 4	12.000	1.400	-0.350	-0.050
Level 3	9.000	1.400	-0.350	-0.050
Level 2	6.000	1.400	-0.350	-0.050
Level 1	3.000	1.400	-0.350	-0.050
Level 0	0.000	1.400	-0.350	-0.050

Make sure Level 1 Cplane is active.
Draw the lighthouse plan.



Now select the plan curve and choose (vaWall) and choose (From Curve) option on the command line.





General

Type	slab
Name	
Description	
Style	Standard
Tag	
Load bearing	✓

Display

Isocurve Density	0
Show Isocurves	☐
Plan Visibility	Below cut plane

Geometry

Volume	4.800 m³
Area	16.000 m²
Thickness	0.300
Thicknesses	1 of 1
Slab layer 1	By Style (0.300)
Edge cut	Plumb

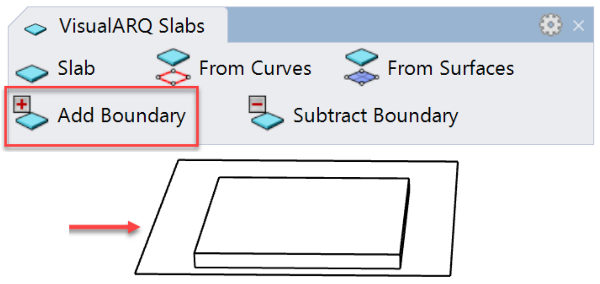
Location

Elevation	0.000
Alignment	Top

Top
Center
Bottom

Copy the slab to all four levels.
On the fourth level offset the curve 2m.

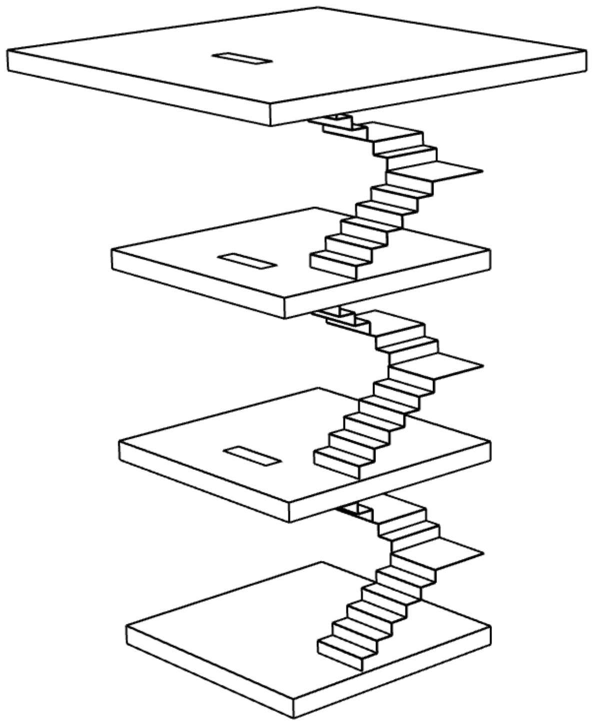
From the slab menu, choose (Add boundary) this will extend the slab to the desired curve.

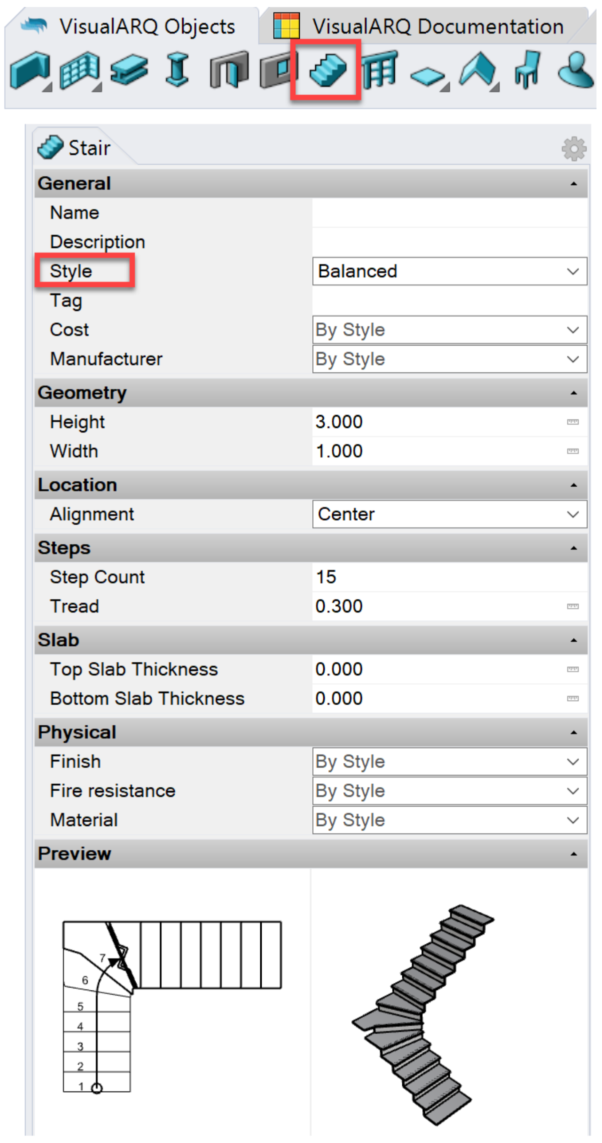


Click on the stairs from the VisualARQ tab. Follow the command line to complete the stairs.
Note: You can always turn the control points on to change the stair's insertion point.

From the properties panel, you can change the stairs' style.

Copy the stairs to all floors.





General

Name	
Description	
Style	Balanced
Tag	
Cost	By Style
Manufacturer	By Style

Geometry

Height	3.000
Width	1.000

Location

Alignment	Center

Steps

Step Count	15
Tread	0.300

Slab

Top Slab Thickness	0.000
Bottom Slab Thickness	0.000

Physical

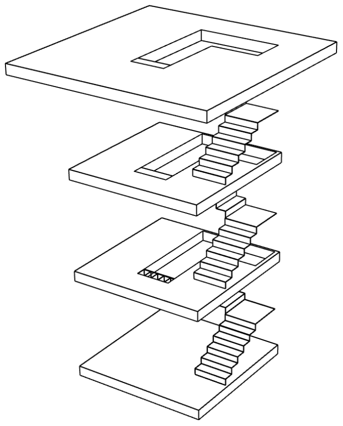
Finish	By Style
Fire resistance	By Style
Material	By Style

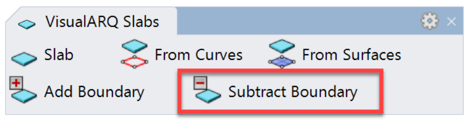
Preview

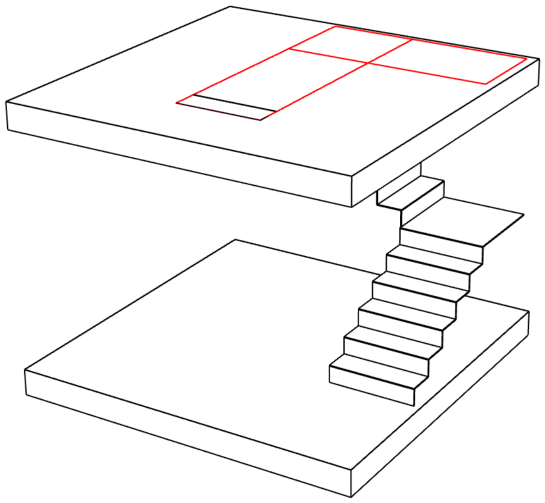
Now we need to make an opening in the slab. Draw a planar curve with the shape of the opening to make the cut.

From the slab menu select (Subtract Boundary), choose the slab then the planar curves to make the cut.

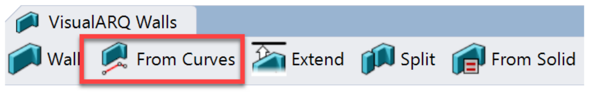
Repeat the same steps for all slabs:

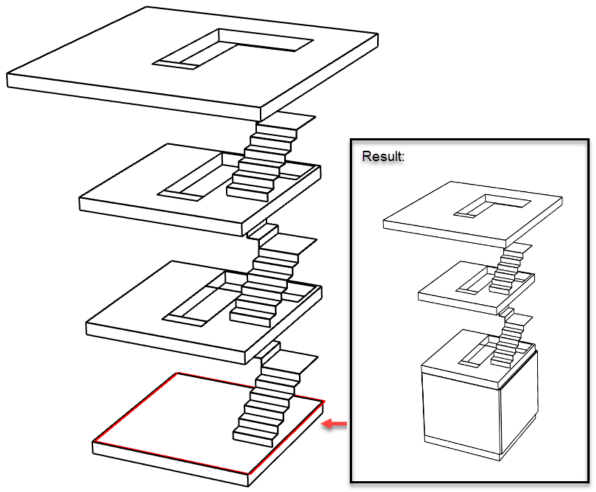


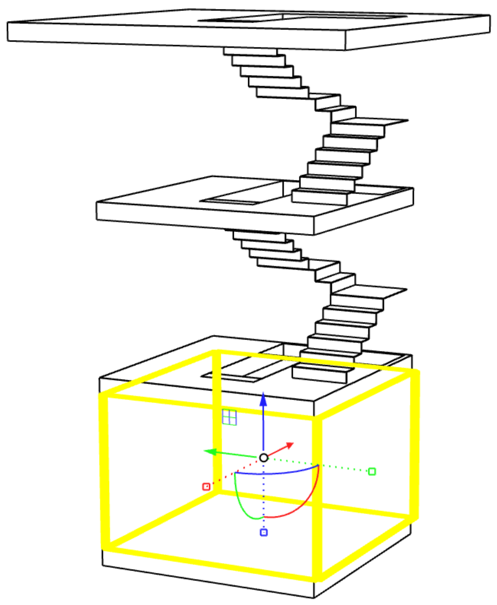
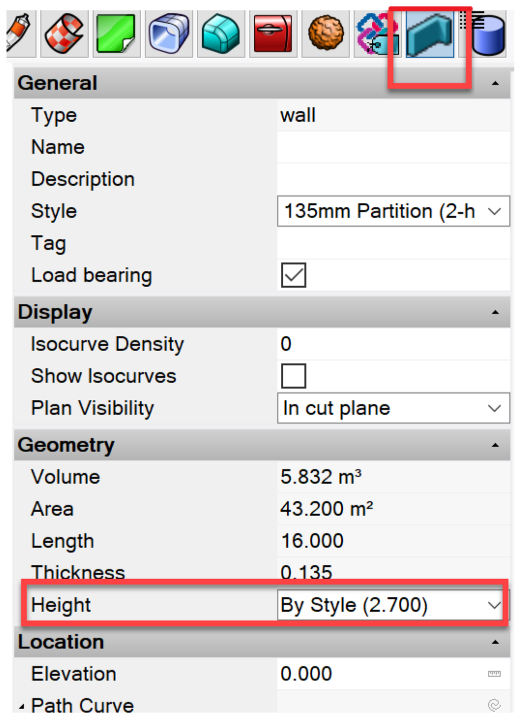
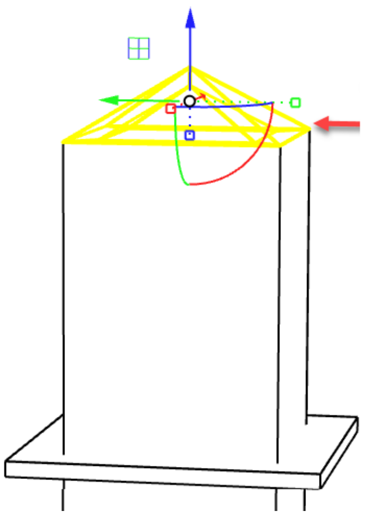
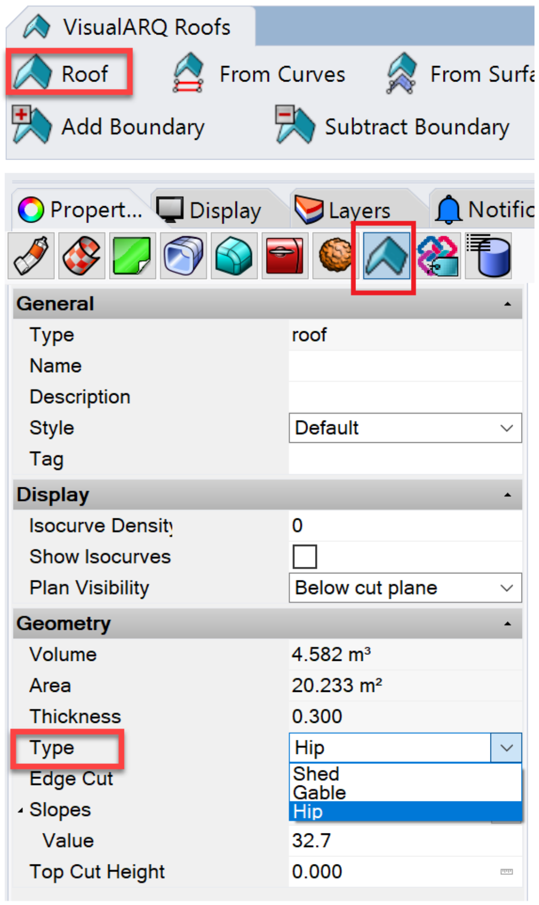




Select the plan curve from Level 0.
Click on (vaWall) Wall from curves.





Now highlight the walls and change their height from the wall properties to 15m. 	
Now create the roof. From the roof menu, select (Roof) then start selecting the corners of the walls. 	

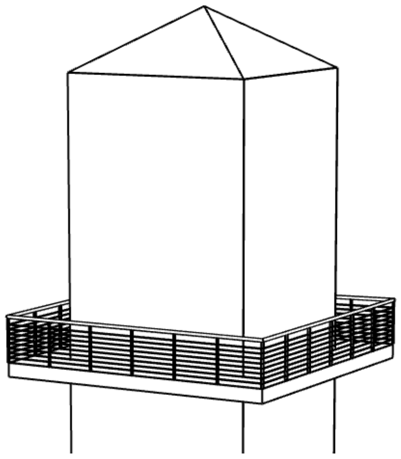
Screenshot 1 (wall properties):

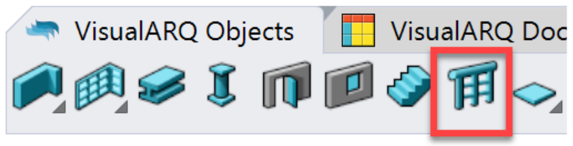
General	
Type	wall
Name	
Description	
Style	135mm Partition (2-h
Tag	
Load bearing	✓
Display	
Isocurve Density	0
Show Isocurves	☐
Plan Visibility	In cut plane
Geometry	
Volume	5.832 m³
Area	43.200 m²
Length	16.000
Thickness	0.135
Height	By Style (2.700)
Location	
Elevation	0.000
Path Curve	

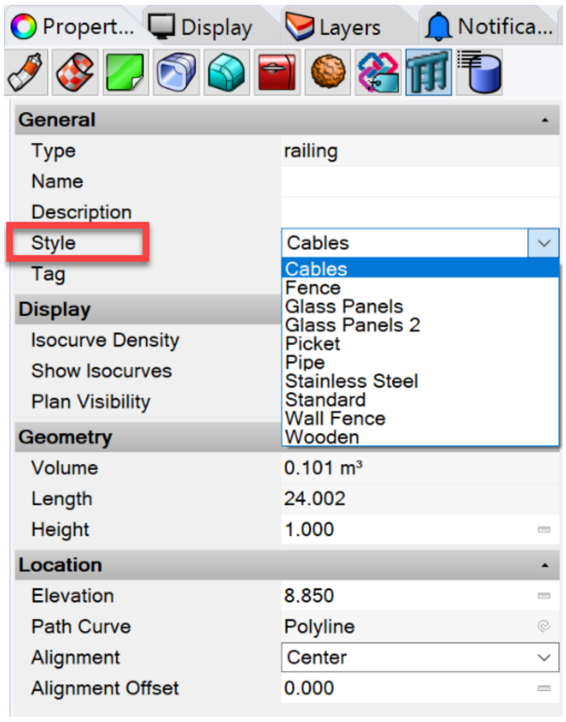
Screenshot 2 (roof properties):

General	
Type	roof
Name	
Description	
Style	Default
Tag	
Display	
Isocurve Densit	0
Show Isocurves	☐
Plan Visibility	Below cut plane
Geometry	
Volume	4.582 m³
Area	20.233 m²
Thickness	0.300
Type	Hip (Shed / Gable / Hip)
Edge Cut	
Slopes	
Value	32.7
Top Cut Height	0.000

Add the railings.
Select the railing option then start selecting the corners of the slab.
From the properties panel, change the railing style.





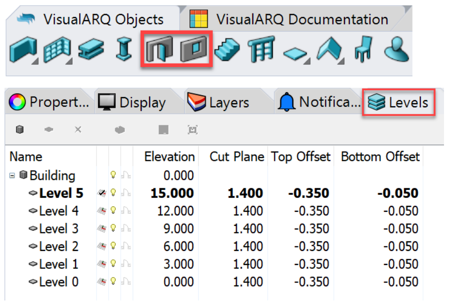


General	
Type	railing
Name	
Description	
Style	Cables
Tag	
Display	
Isocurve Density	
Show Isocurves	
Plan Visibility	
Geometry	
Volume	0.101 m³
Length	24.002
Height	1.000
Location	
Elevation	8.850
Path Curve	Polyline
Alignment	Center
Alignment Offset	0.000

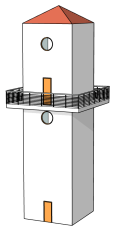
Style options: Cables, Fence, Glass Panels, Glass Panels 2, Picket, Pipe, Stainless Steel, Standard, Wall Fence, Wooden

Select the door option.
Make sure you are on Level 0 then add the door in the center of the wall.
Next, select the direction you want the door to open.

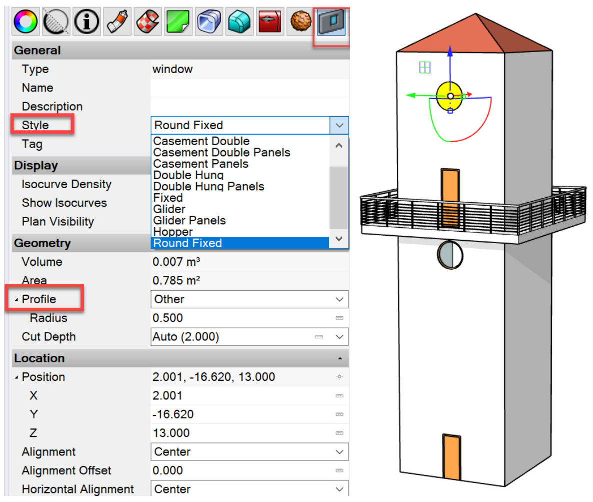
Now do the same for the windows.



Name	Elevation	Cut Plane	Top Offset	Bottom Offset
Building	0.000			
Level 5	**15.000**	**1.400**	**-0.350**	**-0.050**
Level 4	12.000	1.400	-0.350	-0.050
Level 3	9.000	1.400	-0.350	-0.050
Level 2	6.000	1.400	-0.350	-0.050
Level 1	3.000	1.400	-0.350	-0.050
Level 0	0.000	1.400	-0.350	-0.050



Click on the door or windows and change their styles and profiles from the default drop-down menu.

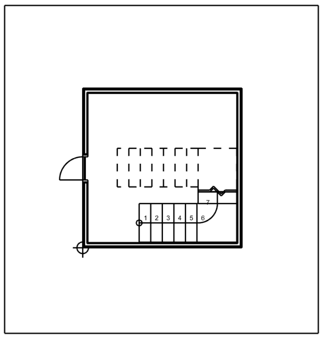


General	
Type	window
Name	
Description	
Style	Round Fixed
Tag	
Display	
Isocurve Density	
Show Isocurves	
Plan Visibility	
Geometry	
Volume	0.007 m³
Area	0.785 m²
Profile	Other
Radius	0.500
Cut Depth	Auto (2.000)
Location	
Position	2.001, -16.620, 13.000
X	2.001
Y	-16.620
Z	13.000
Alignment	Center
Alignment Offset	0.000
Horizontal Alignment	Center

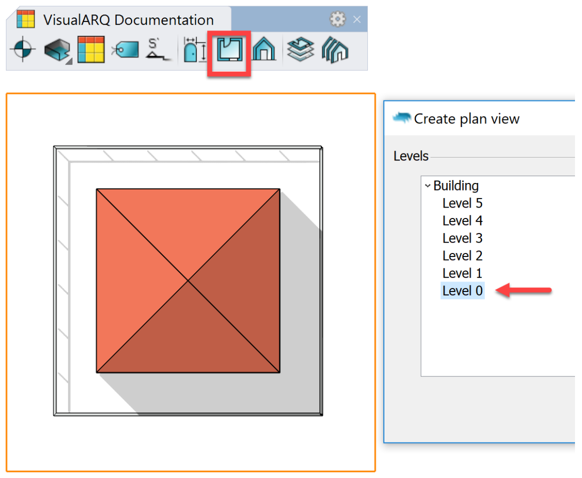
Extract 2D documentation.

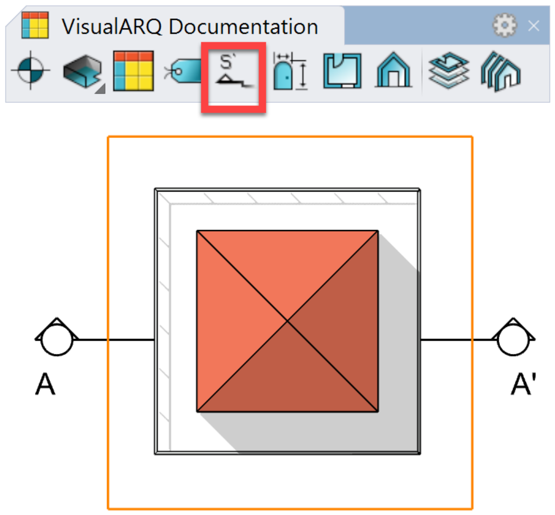
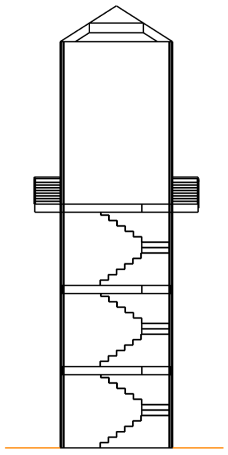
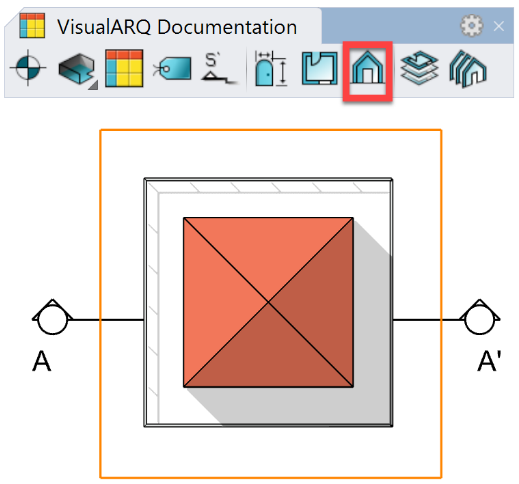
Plan:

From VA Documentation tab, Select Plan view, specify style and level.
Drag a rectangular around the building, then click for an insertion point.
Do the same for all Levels.



Level 0

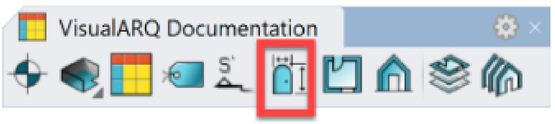


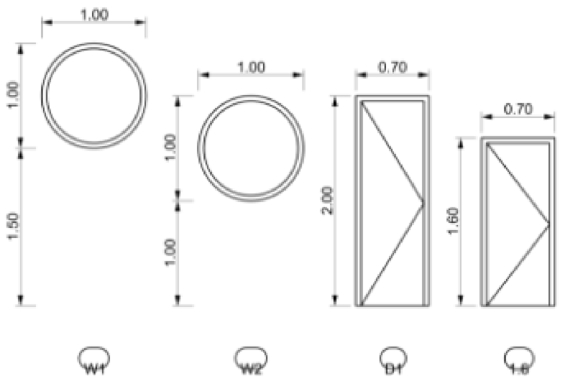
Section: Under the same tab select Section, then pick a style (Arrow) in this case. Then specify two points for the section cutting line. Type in a name for the section on the command line then hit enter.	
Now for VA Documentation Tabs click on Section View. Pick the Section arrow for the previous step and select an insertion point.  A - A'	

Openings Elevation:

Select (Opening Elevation) from the VA Documentation tab.
Select all openings, then specify and start and end point.

*Note: you can drag a window around the model to select all openings.

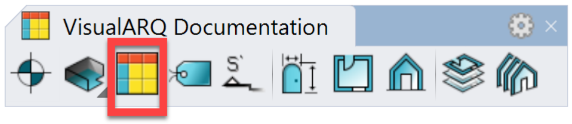


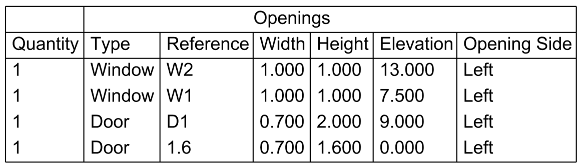


Tables:

Select (Table) under the VA Documentation tab.
Choose a table style, I choose openings for this exercise.

Now select all the openings in the model.
Specify where you want to insert the table.





	Openings					
Quantity	Type	Reference	Width	Height	Elevation	Opening Side
1	Window	W2	1.000	1.000	13.000	Left
1	Window	W1	1.000	1.000	7.500	Left
1	Door	D1	0.700	2.000	9.000	Left
1	Door	1.6	0.700	1.600	0.000	Left

Table (6): Object-based modeling tutorial using VisualARQ for Rhino

Resources

Support:

- Rhino Support Forum: https://discourse.mcneel.com/ - the best place to ask questions and meet with other Rhino users.
- Rhino email support: tech@mcneel.com
- Rhino support page: https://www.rhino3d.com/support
- Plugins for Rhino: http://www.food4rhino.com/

Workflows:

- Digital Fabrication tutorials for Rhino: https://www.rhino3d.com/tutorials#digital_fabrication
- Architecture Digital Fabrication: https://wiki.mcneel.com/rhino/architecture/home#digital_fabrication
- Rhino Fab Studio: http://www.rhinofablab.com/
- Using Rhino with Revit: https://wiki.mcneel.com/rhino/architecture/bim/rhino-to-revit
- Nick Senske Computational Design Courses YouTube : https://www.youtube.com/user/nsenske
- Jose Sanchez would like to share a series of online videos on Rhino modeling in architecture: Modeling the LF-ONE by Zaha Hadid : https://www.plethora-project.com/education/

Rhino in Architecture:

- Many Articles, Books, Workflows: https://wiki.mcneel.com/rhino/architecture/home
- ArchDaily blog covering Rhino: https://www.archdaily.com/tag/rhino
- The latest news about Rhino in AEC: http://blog.rhino3d.com/search/label/AEC
- Many videos on Rhino in Architecture: https://www.youtube.com/results?search_query=rhino+architecture
- Gallery of a few buildings done with Rhino and Grasshopper: http://www.grasshopper3d.com/page/architecture-projects