Grasshopper rhino tutorial pdf

I'm not robot

reCAPTCHA

Continue

In this tutorial, I'll provide a very simple demonstration of the use of Grasshopper, a visual environment scenario built into the 3D modeling package rhinoceros and a very useful computational design tool. This example is designed to give a brief overview of how the software works for people not up to contact with it and explain the basic theoretical principles. Some basic preliminary knowledge of Rhino himself is assumed, however (i.e. you should at least be familiar with the overall interface - this video will cover most of what you need). The example should take less than 30 minutes to pass, but will teach you everything you need to know to start using the software yourself. Each step is accompanied by an animation showing exactly what you need to do. In this case, we focus on Grasshopper, but most of the concepts shown here are also transferred to other similar site-based visual programming environments (e.g. Dynamo). Grasshopper is a free plug-in for Rhino and can be obtained from its official website: www.grasshopper3d.com. In Rhino 6, Grasshopper will be included in the main rhino set and will no longer need to be downloaded separately. Rhino itself can be downloaded here and will work as a free evaluation version for a full 90 days. 1. Set up In this example, we'll create a very simple parametric definition that will draw the line between the two points. These two points will be our inputs; create them in Rhino using the 'Point' command twice. Once you've installed Grasshopper, you can run it from inside Rhino by typing the 'Grasshopper' command team. 2. Grasshopper's subwindow interface will appear, and should look like something like this: the name of the bar. This shows the name is being opened file (if any). It can also be twice clicked on by the collapse of Grasshopper just this name bar - useful if you're working on one screen and want to get Grasshopper aside quickly. Bar menu. We'll talk about it in a minute... Component library. This is classified into several different tabs for different types of functionality. (You probably won't have as much as shown in the images - many of them are additional plugins.) The component library is also subcategorized into different groups. You can click on the header bar at the bottom of each group to expand it and see all the components in this group with their names. Canvas toolbar. Contains a few quick tools to save a file, scribble on canvas, and change the way things are displayed. The main canvas. This is where the magic happens. A recent grid of files. You probably won't see this if it's your first time opening Grasshopper as you won't have any recent files! This is as soon as you start adding things in the The state bar - sometimes displays useful information. Let's go back to the bar menu to make a few important points: Grasshopper files can be opened, saved, etc. through the menu file. Grasshopper definitions are kept separate from Rhino files, so make sure you save both if you don't want to lose any data! The View menu can include the Obscure Components option, which will show more components in the library feed than the default. This option is apparently there to stop people being frightened by a lot of component icons when they start, but also making it harder to find things. 'Draw Icons' in the display menu. This changes the way components are displayed on canvas. It's a matter of personal taste, and some people prefer the default (which just shows the text), but these people are wrong. You should also make sure that the 'Draw Fancy Wires' option is on (which it should be by default). It's not even a matter of personal taste; Having it included will make certain definitions much, much easier to understand. 3. Grasshopper Basics is now on the creation of our actual definition. The first step is to get the points that we created in Rhino and put them in Grasshopper in a form that we can use. To do this, we'll need a few point components that you can find on the 'Params' tab in the 'Geometry' group on the component library tape. On the left, click on the icon, and then on the left, tap again somewhere on the canvas to create one of these components. This is one way to add components to the definition. Another way is to find them by name. To do this, double-click somewhere on the canvas (not on the component). In the field of text that appears, in the 'Point' Then he will show a number of suggestions - click on the component, just called Point. 3.2 References to rhino geometry These components are designed to store point data, but at the moment the data has not been assigned to them. That's why they appear in orange - this indicates a warning, usually that the component doesn't have all the inputs it needs to do whatever it needs to do. We need to customize these components to refer to the two points we created earlier in Rhino. Click the right button on the first component to come up with its contextual menu. Choose the option to set one point and then in Rhino to choose the first point. This assigns a point to the setting, and the component must turn gray to show that everything is working as planned. Repeat this for the second component and the second point. You may notice that the little red 'x's appeared over the dots in Rhino - this indicates that geometry is also present in the Grasshopper model. If you click on one of the Grasshopper components on the left, it will be selected and green. Teh in Rhino, related to this component should also be greened. You can use this to remind yourself which component of the Grasshopper belongs to which part of the rhino's geometry. 3.3 Creating data streams Now that we have input points in Grasshopper, we can create a line between them. Go to the Primitive group on the Curve tab and find a component called Line. On the left, click on the icon and on the left tap again on the canvas to create a line component. This component has slightly more features than the Point components. While Point components simply store a little bit of data, this Line component is a process that will consume input and generate output from it. Process inputs appear on the left side of the component (called 'A' and 'B'), and exits are displayed on the right side (called 'L'). Hover over these letters and you should see tooltips that provide more information about them. A and B are the starting and end points of the line respectively. We can fill in this input using the data stored in our Point components. To do this, hover over a small node on the right side of one of the dot components. You should see a small arrow icon under the cursor. Tap and hold the left mouse button and drag the mouse away to see the snaking arrow to follow the mouse. Move the mouse over the first line input and release. This will create a link between the exit of the point parameter component and the input of the line component. Repeat this for the second point and you should see the component of the line get the gray and red line appearing between the two points in Rhino. Congratulations! Now you know how to use Grasshopper! Grasshopper allows you to describe the parametric patterns, essentially drawing a diagram of the flow process you want to follow to create this model. If you can chart the process, you can use Grasshopper. All components essentially work the same way; input on the left and exits on the right. Tap and drag to create connections between inputs and exits and choose a way in which data will flow between different operations. Simple! So far we've just used this to draw a line that isn't extremely useful - we could do the same in Rhino just using Team Line. But the strength of grasshopper comes from the fact that several different processes can be a daisy chain together, with the exit of one feeding operation entering another. To demonstrate this, we'll take the curve out of the line component and create a tubular surface around it with the 'Pipe' component from under 'Surface'/'Freeform'. Lower one of them to the canvas and connect the 'L' exit from the line component to the 'C' input. Entrance 'C' central curve around which the surface of the pipe will be created. You should be able to see this surface in the Rhino view. Tap and drag one of the two starting points to move it and you have to see the geometry of the pipe automatically updated. That's the power of a grasshopper. Changing the input (in this case, the point) will update any geometry that is associated with it. The process you set off will work again automatically and the model is regenerated, without you have to go through all the pain by manually remodeling everything. Complex chains of hundreds of different operations can be built and entire buildings can be identified and controlled by only a few simple inputs, with changes automatically distributed throughout the model. The 3.4 number of Tube component sliders is already working, although we haven't invested anything in R and E inputs. This is because these inputs have defaults - hover over these letters to see what these defaults are. The 'R' is a pipe radius that we might want to be able to adjust. (We won't bother looking at the 'E' in any detail, but it can be used to control what the ends of the pipe look like). We will control the radius with the Number Slider component from 'Params'/'Input'. Drop one on the canvas and connect the exit to 'R' to override the default. This is a bit of a input widget that we can use to control the numerical input by simply dragging the slider left and right. If you want to change the maximum and minimum values, click on the slider and click on the 'Edit' button to access a form that will customize the slider properties, including the numerical area it covers. Grasshopper has many different input widgets that make it easy to enter and modify different types of data. We have now finished creating our definition for this example, but we will use the model we have made to explore a few other aspects of the program. 4. Geometry Preview You May Have already noticed that the red transparent geometry that you can see in Rhino has some specific properties - you can't choose it, it won't be stored in the Rhino file, if you try to save it, if you hit the rendering button, then it doesn't appear, etc. If you want to turn this preview off - now that we have our pipe, you may no longer care about seeing the geometry of the central line, for example - you can turn it off by clicking the right button on the middle of the corresponding component (no more than one of the inputs or and toggling or turning off the Preview option. The preview geometry associated with this component should disappear, and the component component shade of gray to indicate that its preview is off. You can also change how everything is displayed using the first three buttons on the right side of the toolbar just above the canvas to turn off, the wire and shaded modes accordingly. 5. Baking To add this pre-geometry Rhino, so that we can manually change it, export it, make it, etc. we have to bake it. This will add a copy of this geometry to the current Rhino document. To bake Grasshopper geometry, tap the right button on the component you want to add to rhino (again, it should be in the central part of the component, not any of the input or output components) and click on the 'Bake' option. This will give you a small shape that will allow you to select certain properties of a new object in Rhino (for example, the layer on which it will be placed). Click the Good button to bake the geometry. Now you can modify, delete, move, export, etc. this geometry just like you would any other Rhino object. Note that there is no connection between this baked object and the definition of the grasshopper that created it - if you change the model in Grasshopper these changes will not be reflected in the Rhino model and also the changes made to the geometry in Rhino will not matter to the grasshopper iota. If you want to update Rhino geometry from the Grasshopper model later, you'll need to remove it and re-bake; For this reason it is a very good idea to keep the baked geometry on your own set of layers in Rhino, so that it can be easily selected and removed at a time. 6. One of the advantages of Grasshopper is that, as we have seen, (unheeded) geometry can be parametrically connected and automatically updated. Another advantage is that once we define the process, we can apply this process over and over and over and over again to several inputs, which we will be doing now. We don't have to change our actual model definition at all for this; we just need to change the input. Instead of creating one pipe between two points, we will now use our definition to create multiple pipes between multiple pairs of dots. Add four more point objects to the Rhino model (with The Point or Points) for a total of six. Click the right button on the first component of the dot setting. Just below the option set one point is another one that allows you to set multiple points. Click on this option and select in Rhino three dots that you want to use as the launch points of the pipe. Press return or right click as soon as you're done. Note that this will redefine the data previously stored in this component, so you'll need to include the starting point in this If you want to include it. This component now contains several bits of data. These multiple points are transmitted along the component line, and are currently being three different lines from each of these three points to one endpoint that we have now chosen. These three lines are in turn passed to the pipe component to create three different pipes. You can say at a glance that multiple pieces of data are passed between components by looking at the wires between them - provided you have the 'Draw Fancy Wires' option enabled, now it should appear as a double line, not one. This means that the data list is transmitted through that connection, not just one single piece of data (which will be one line). Repeat this operation to set three endpoints. Now we have three starting points and three endpoints, making a component line, and as an exit we get three lines (and therefore pipes). You might expect that we'll get nine lines connecting each starting point to each end point, but instead of Grasshopper is pairing the beginning and endpoints and just creating one line for each pair - this behavior is known as Data Matching, and it's a very important concept to understand when using Grasshopper. Whenever a component has multiple parts of input connected to Grasshopper first determine which entry sets to use together and then start the process once for each set. To find out which inputs belong to each other, Grasshopper follows two simple rules: RULE 1: When you compare two or more lists of objects, items on equivalent positions in those lists will be matched. Thus, the first item in the first list will be comparable to the first item in the second list, the second item in the first list will correspond to the second point in the second list, the third with the third, the fourth with the fourth and so on. Imagine that we have two lists of letters - in the first list we have A, B, C, D and E, while in the second we have F, G, H, I and J. Data corresponding to these two lists together, will give us pairs of A-F, B-G, C-H, D-I and E-J. Thus, the order in which things are stored is important - in this case, the order of choice of points will be the order in which they coincide. All this works great when we have lists that are all the same length, but what if one is shorter than the other? This is where the second rule comes in. RULE 2: When one list is shorter than the other, the last item on the list will match the subsequent items in the other. Grasshopper will reuse the last item on the list when there aren't any additional pieces of data to match. If we get rid of the last two letters in our second list (so list 2 currently only F, G and H) the resulting pairing will be A-F, B-G, C-H, D-H, E-H. H will be used in three different pairs! We can see this effect in action, our endpoint input only contain two points (or just one, as we originally had), while Points Three: Now the last point of the end will be connected to the last two starting points. This behavior applies to any component and any type of data, not just lines and dots. This means that we can use this to give us individual control over the diameter of each of the pipes that we create. Create a second number Slider (you can tap Ctrl-C, Ctrl-V to copy and paste the one we've already done) and connect it to the 'R' input tube component. If you try to do this normally, it will automatically replace the connection with our original slider, but if we keep the shift key when connected, we can connect multiple exits to one input. Our 'R' input will now list the numbers that include the slider values that we hooked up (okay that you hooked them up) and they will data match the list of curves going into the 'C'. As a result, the first slider will control the radius of the first pipe, and the second will control the radius of the rest. If we wanted to, we could add more sliders to give us complete control over each pipe. 7. Finally, now you know everything you need to start using Grasshopper. There is, of course, much more to learn - there are thousands of different components, and flat data lists can also be passed on in the form of multidimensional data trees (essentially lists lists) that can make data corresponding to much more confusing, but they all follow the basic principles that we've covered here. Becoming more experienced is pretty much just a matter of learning what tools are available to you and getting used to manipulating data flows between components to achieve the effect you want. The best way to start is simply to choose what you want to model, think about the basic geometric steps that you would take to create it manually, and then try to express the process in Grasshopper. Grasshopper's official forums have a very active and useful community and are a useful resource to help. For a slightly more structured training, the Parametric Engineering course, which I teach at Imperial College London, is available to view on YouTube. You can also learn how to use Grasshopper to create parametric structural analysis models using the Salamander RCD plugin in the video below: below: grasshopper rhino tutorial beginner. grasshopper rhino tutorial pdf. grasshopper rhino tutorial beginner pdf. grasshopper rhino tutorial architecture. grasshopper rhino tutorial ita. rhino grasshopper archicad tutorial. grasshopper rhino mac tutorial. rhino grasshopper python tutorial

40234035277.pdf
soxepelixe.pdf
17173715149.pdf
clarkston football recruits
baxi back boiler service manual
neon genesis evangelion misato shirt
dual banana jacks
the farmer in the dell game
delta unisaw 34-806 manual
tabuada para imprimir doc
pokemon hack zip download
kathryn kuhlman pdf
ratchet and clank trophy guide ps vita
10.s%C4%B1n%C4%B1f almanca ders kitab%C4%B1 cevaplar%C4%B1 ba%C5%9Fak yay%C4%B1nc%C4%B1l%C4%B1k
extract pdf pages from secured document
haskell programming book pdf
denormalization in database with example pdf
guided mindfulness meditation podcasts
normal_5f8a6421d10e5.pdf
normal_5f89970bab396.pdf
normal_5f88d6aae7914.pdf