

Beegraphy Editor Documentation

About

Welcome

BeeGraphy Editor is an online Node editor. It allows 3D Builders to create parametric models in a more comfortable environment by offering innovative tools.

We provide the ability to create and design a variety of parametric models that can be used in construction, furniture, clothing, jewelry, fashion, manufacturing, parts, AR \ VR & Gaming, and other areas.

Thanks to "BeeGraphy Editor" you can work with your team on one project where all team members can see each other in real-time.

We build models with unique curves called NURBS and with modern mathematical formulas, so they are the most accurate and technically upgraded parametric models.

Introduction

"Beegraphy Editor" is an online platform that allows 3D builders to study visual programming, solve problems and create their parametric models.

Let's have a journey with "Beegraphy Editor" and explore it.

What is "BeeGraphy Editor"?

"BeeGraphy Editor" is a wide-ranging Node Editor that allows the creation of 2D and 3D parametric models in any dimension. BeeGraphy Editor works on a web platform which makes it accessible to everyone. Building models with nodes is making construction simpler and more convenient. By connecting several Nodes in "BeeGraphy Editor" you can get different models. The process could take a long time without nodes.

Collaboration doc

"Beegraphy Editor" allows you to work on a project in real-time 24/7.

You can invite teammates to your team and create an accessible project for teamwork. The page is being updated automatically in real-time, so all the team members can see each

other's changes at once, which is making the teamwork more efficient. Team members are divided by colors, which allows you to separate the work of each teammate.

Hello Beegraphy Editor

Welcome to a web-based platform “BeeGraphy Editor” which provides 3D parametric models and space to build them.

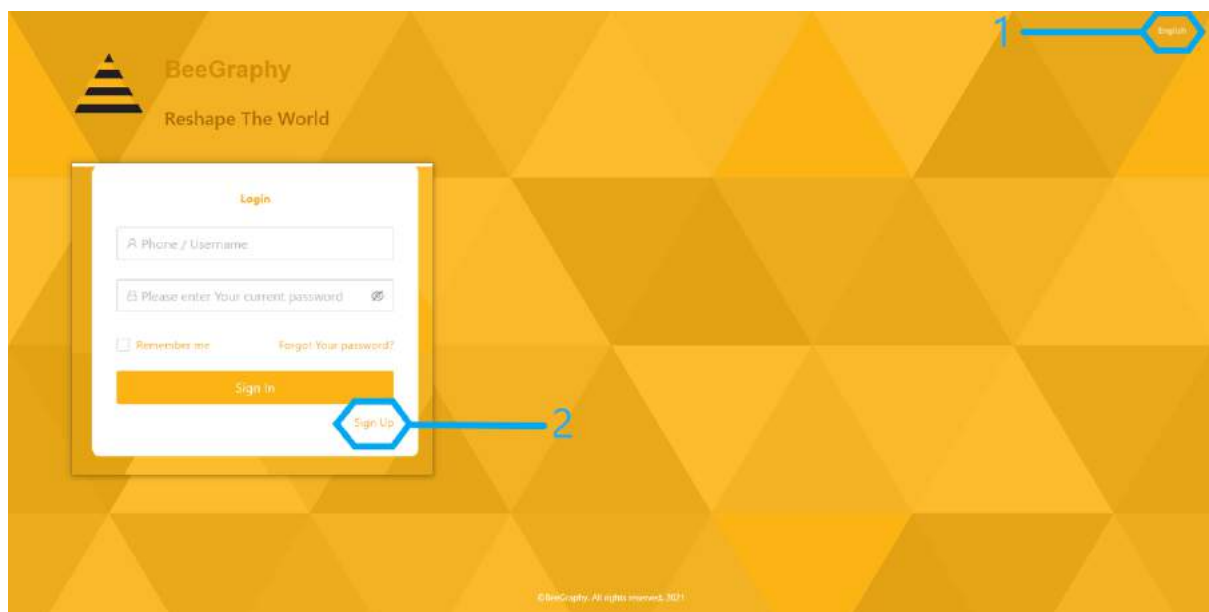
”BeeGraphy Editor” cloud-based platform gives you a space to create and edit 2D and 3D parametric models online with your team.

You can use it to develop a wide range of creative workflows.

Registration and Launching Beegraphy Editor

Let's register on the “BeeGraphy Editor” and get started by reviewing the key features of the interface.

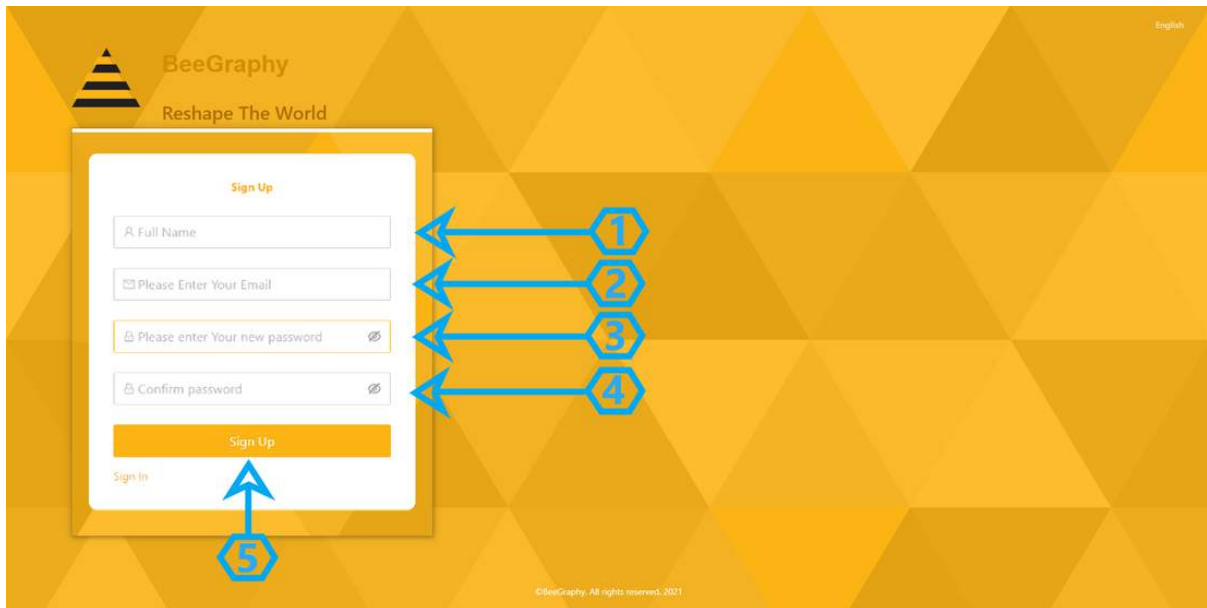
1. You will need to register for access to "BeeGraphy Editor".



You can choose the language on the website: Armenian or English

Click on the "Sign Up" button to go to the registration page

In the "Sign Up" section you need to fill in the following fields:



Name (1), Email (2), Password/ Repeat it (3,4).
Click the "Sign Up" button (5) to register and log in to "Beegraphy editor".

The User Interface

BeeGraphy Editor home page introduction.



There is a Menu button in the upper left corner, through which you can move: Home, Nesting, Product sections (1).

There are buttons for changing the language and personal page settings in the right corner. (2)

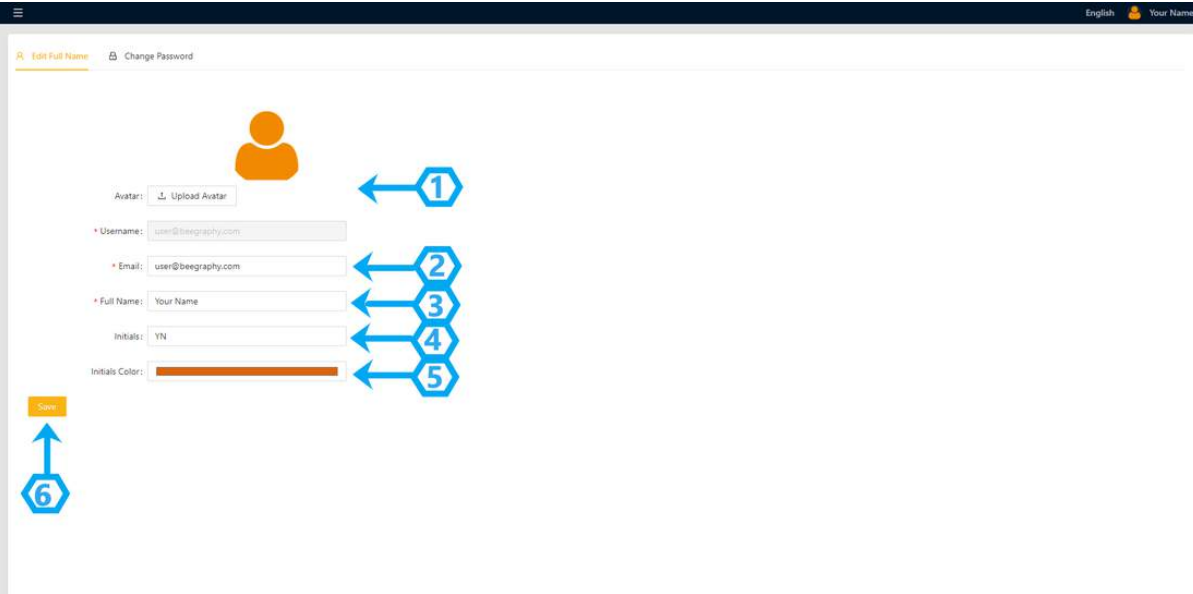
Profile, Transactions, Settings, and Logout buttons are visible when clicking on the user name. (3)

The teams are displayed on the left side of the page. The first team is created by default after registration. (4)

The teamwork field is in the center of page (5)

“BeeGraphy Editor” personal page settings

You need to click on the name and enter the profile section to make personal data changes.



The screenshot shows the 'BeeGraphy Editor' personal page settings interface. At the top, there is a navigation bar with a menu icon (1) and a user profile section with 'English' and 'Your Name' (2). Below the navigation bar, there are two tabs: 'Edit Full Name' (active) and 'Change Password'. The main content area contains a form with the following fields and controls:

- Avatar:** A placeholder image of a person (3) and an 'Upload Avatar' button (1).
- Username:** A text input field with the value 'user@beegraphy.com' (2).
- Email:** A text input field with the value 'user@beegraphy.com' (2).
- Full Name:** A text input field with the value 'Your Name' (3).
- Initials:** A text input field with the value 'YN' (4).
- Initials Color:** A color selection bar (5).
- Save:** A yellow button at the bottom left (6).

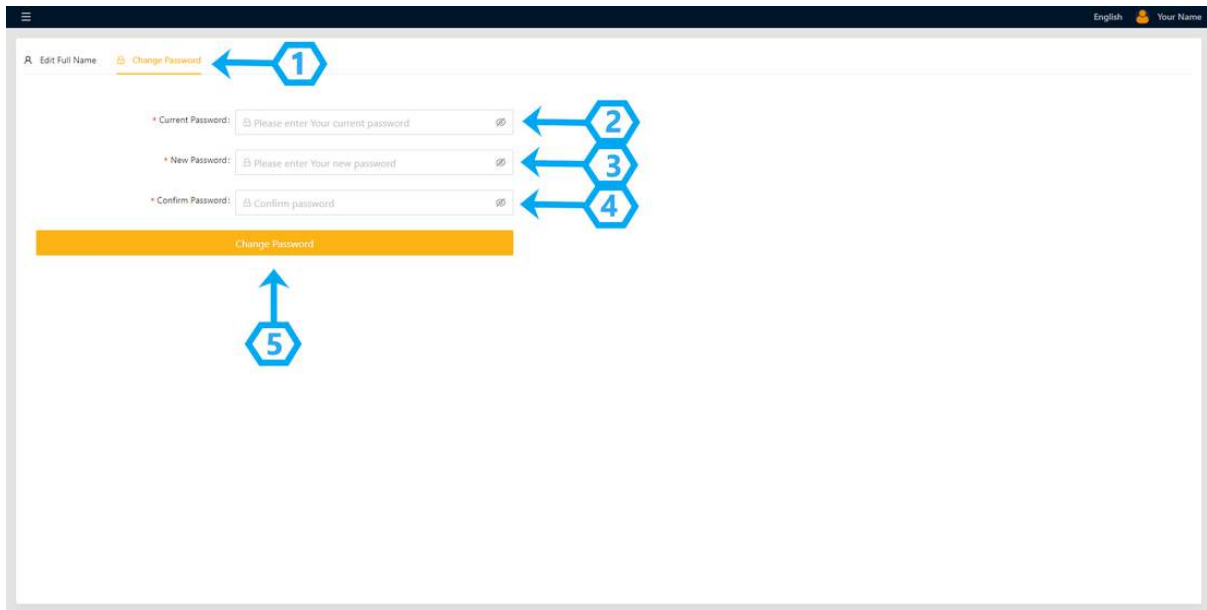
Blue arrows with numbers 1 through 6 point to these specific elements in the interface.

Here you can add a picture (1), or change it .

you can add an address (2), name (3).

You can also add initials, additional name (4), and color (5).

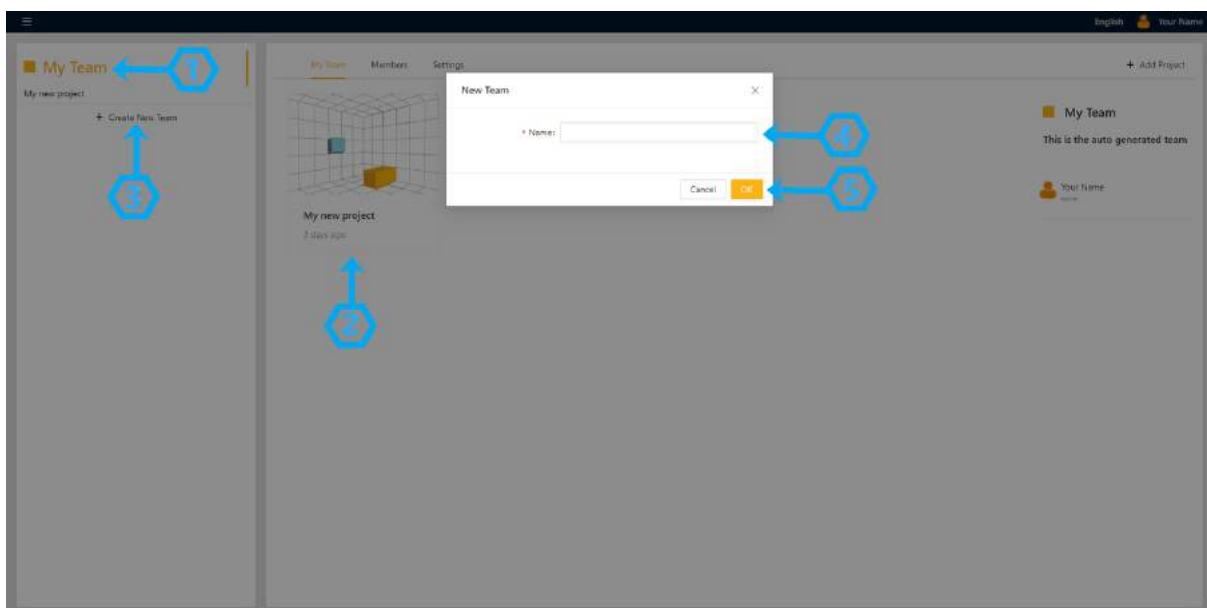
Click the "Save" button (6) to save the changes.



Click the "Change Password" button (1) to change your password. In the first field (2) you have to write your old password, and in the second and third fields (3,4) you should enter your new password. Click the "Change Password" button (5) to save the changes.

Create A Team, Project, and Resource

Create a team in “BeeGraphy Editor”



Your first team is automatically created under the name "My Team" (1) after registration, which contains the first project - "First Project" (2), which in turn is its first resource "First Resource".

You need to click the "Create New Team" button (3) to create a new team. Fill in the name of the team and (4) and click the "OK" button. (5)

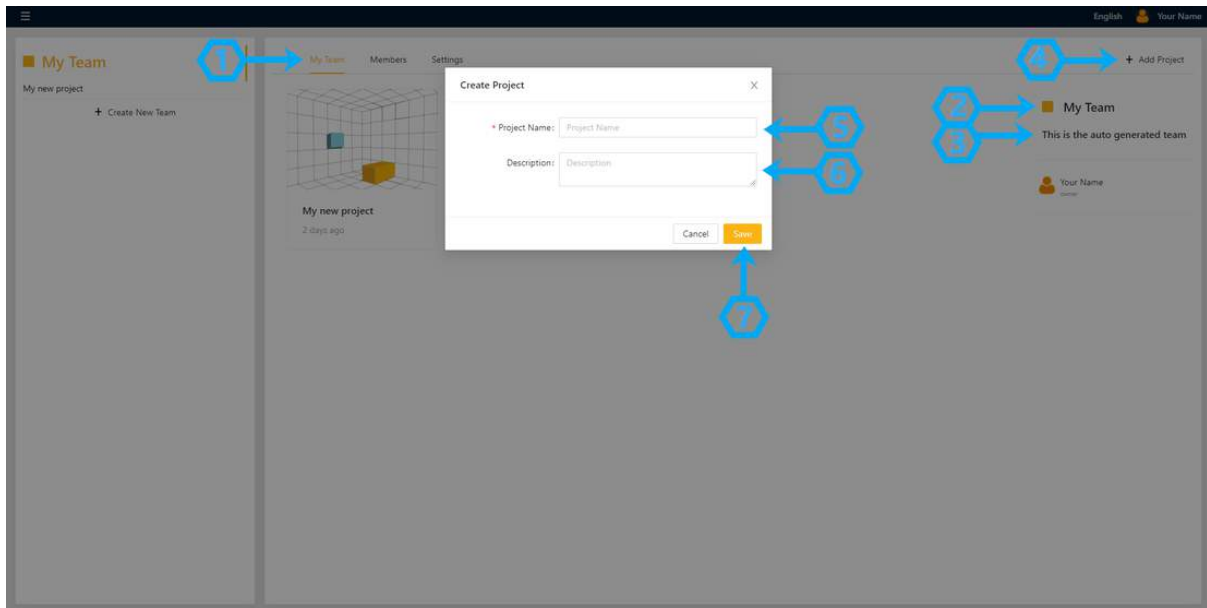


The team has the following sections: "Team" (6), "Members" (7), "Settings" (8).

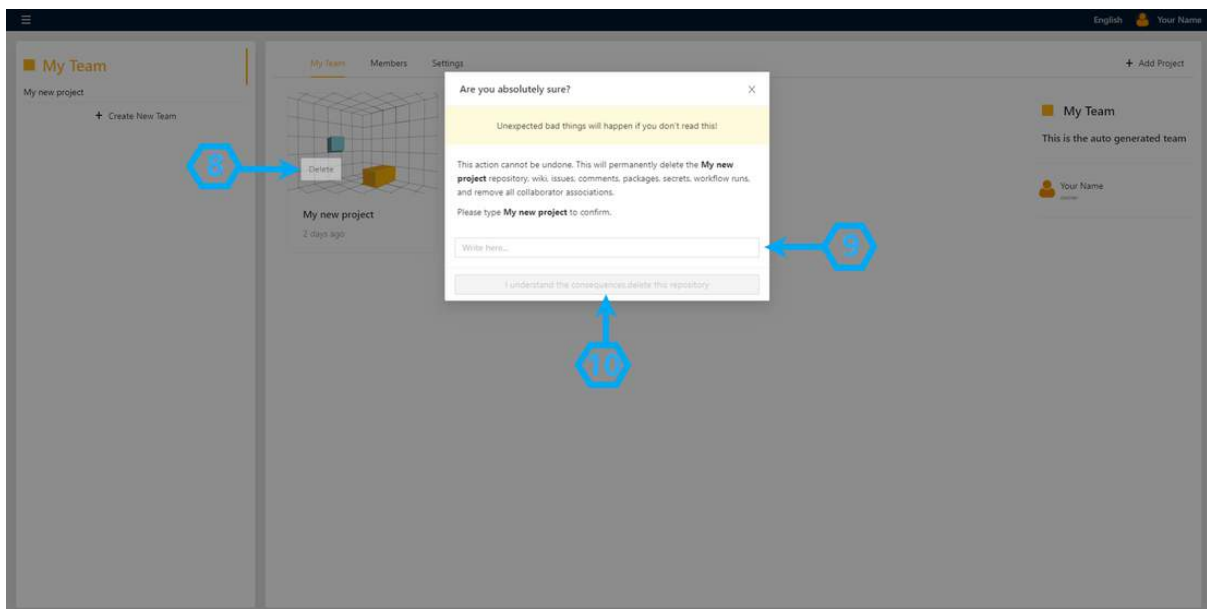
You can edit the team name (9) and comment (10) in the "Settings" section, as well as give it the preferred color (11). Click the "Save" button (12) to save the changes.

You can delete the team by clicking the "Delete Team" button (13), in the opened confirmation window fill in the team name and click to remove the team, therefore it will be removed forever.

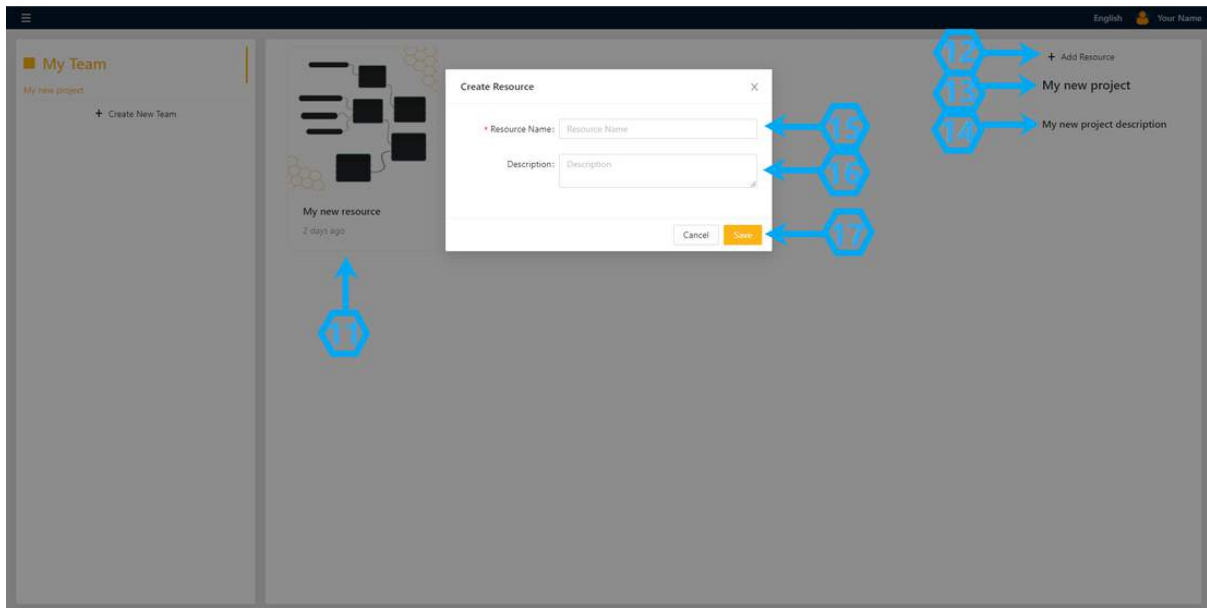
Create a project and resource in the “Beegraphy editor” team



In the "Team" section (1) you can change the team name (2) and comment it (3). Click on "Add Project" (4) to add a new project to the team. By filling in the project name (5) and explanation (6) and clicking the "Save" button (7) you create a new project.

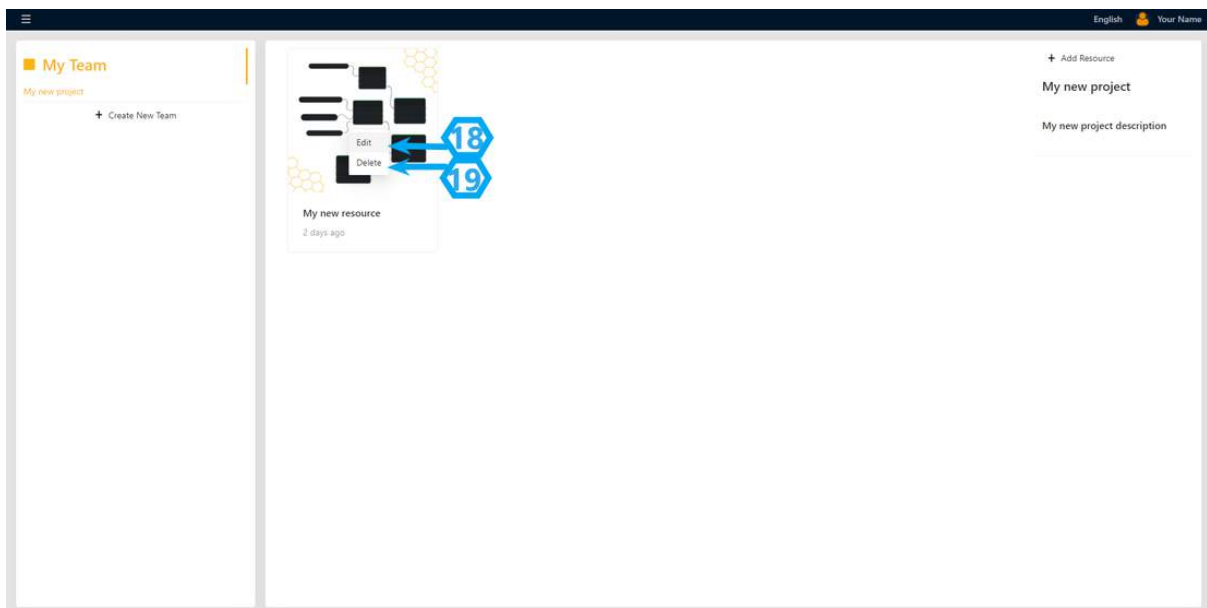


To delete the project, right-click on the project and click on the "Delete" button (8), fill in the project name (9) in the opened confirmation window, and confirm (10)

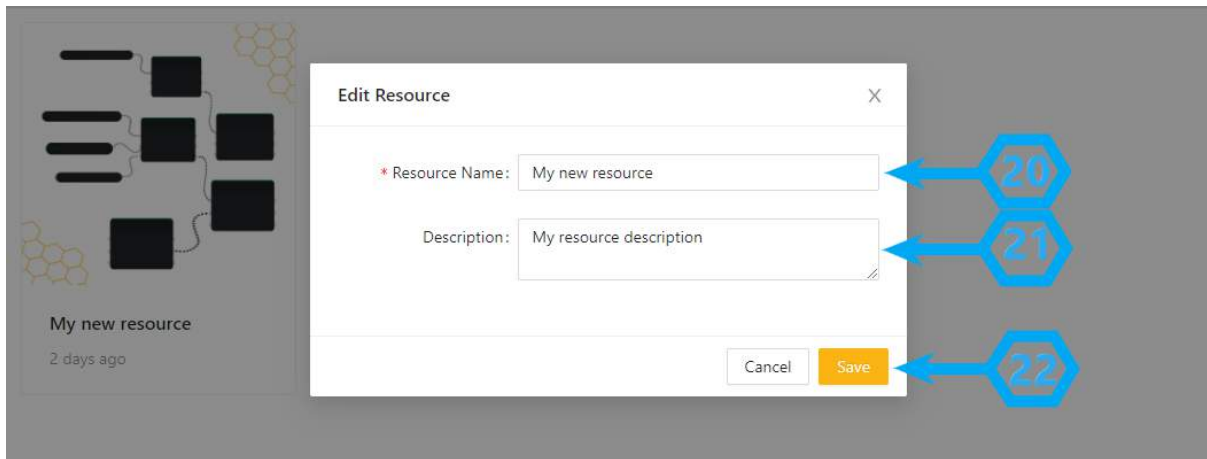


By clicking on the created project, you will see the resources in the project (11), here you can change the name of the project (13) and comment (14).

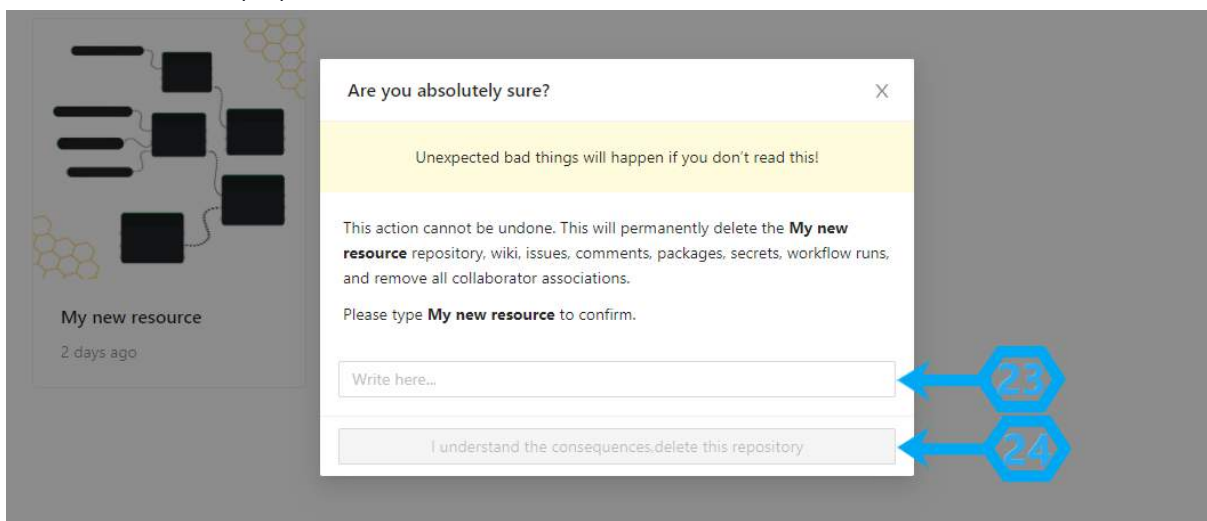
To add a new resource, just click on the "Add Resource" button (12), fill in the resource name (15) and comment (16), and save by clicking "Save" (17).



To edit or delete the resource, you need to right-click on the resource, select "Edit" (18) or "Delete" (19)

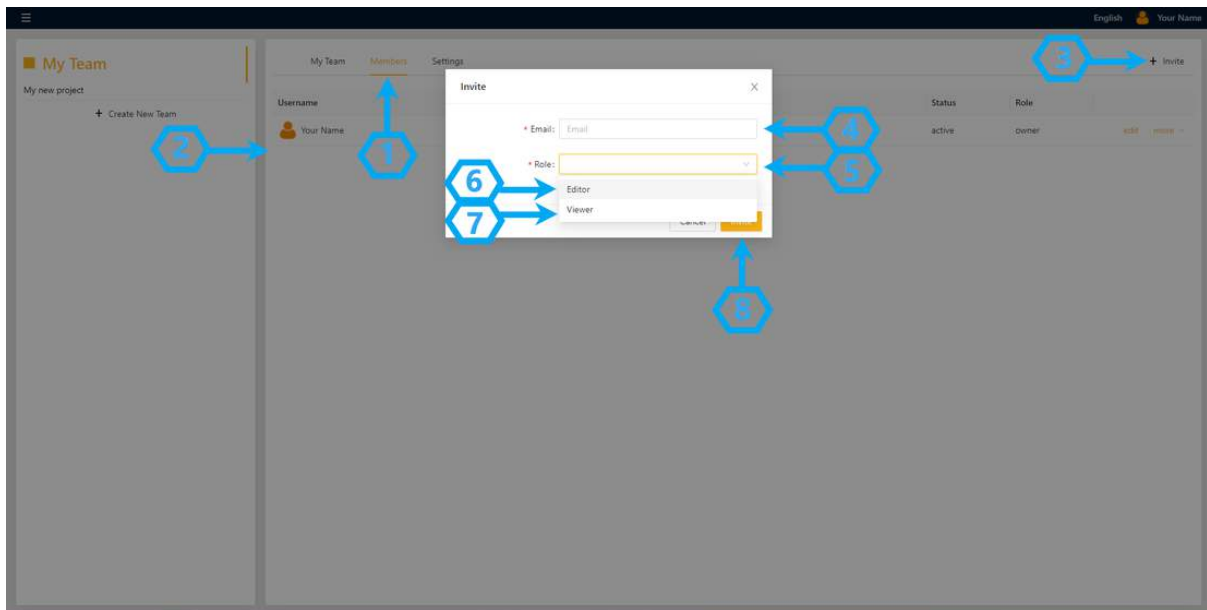


You can edit the name of the resource (20) and comment (21). Save the changes by clicking the "Save" button (22).



Press the "Delete" button and fill in the resource name (23) & confirm (24) in the confirmation window. This will irreversibly remove the resource.

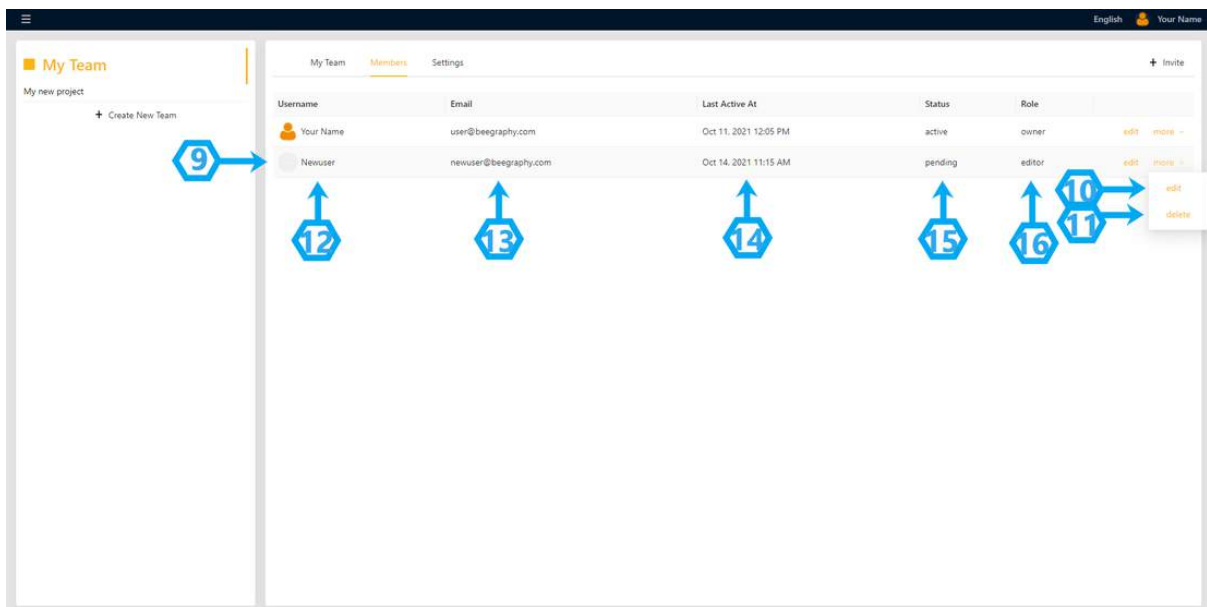
Add members to the “Beegraphy editor” team



In the "Members" section (1) you can see teammates (2) as well as add a new teammate.

Click on the "Invite" (3) button to add a new team member. Fill in the invited teammate's email address (4) and his position (5). The position can be an "Editor" (6) or a "Viewer" (7).

Click the "Invite" button (8) to add a new teammate to the list (9).

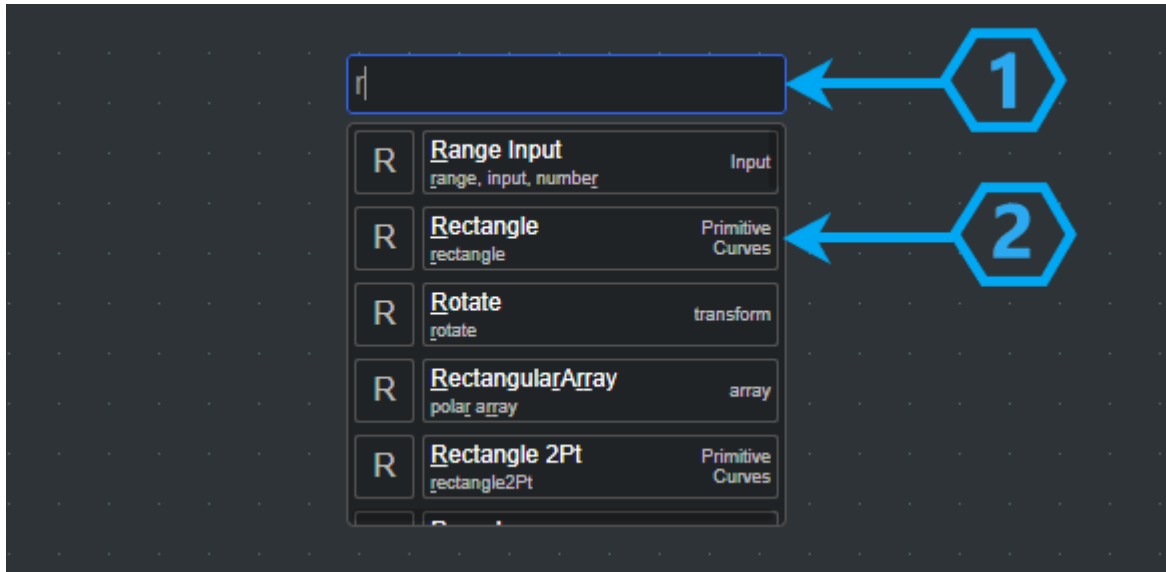


Click the "Edit" (10) or "Delete" (11) buttons to remove team members or to change their roles. Here you can also see the names of teammates (12), email address (13), last login time (14), status (15), and role (16).

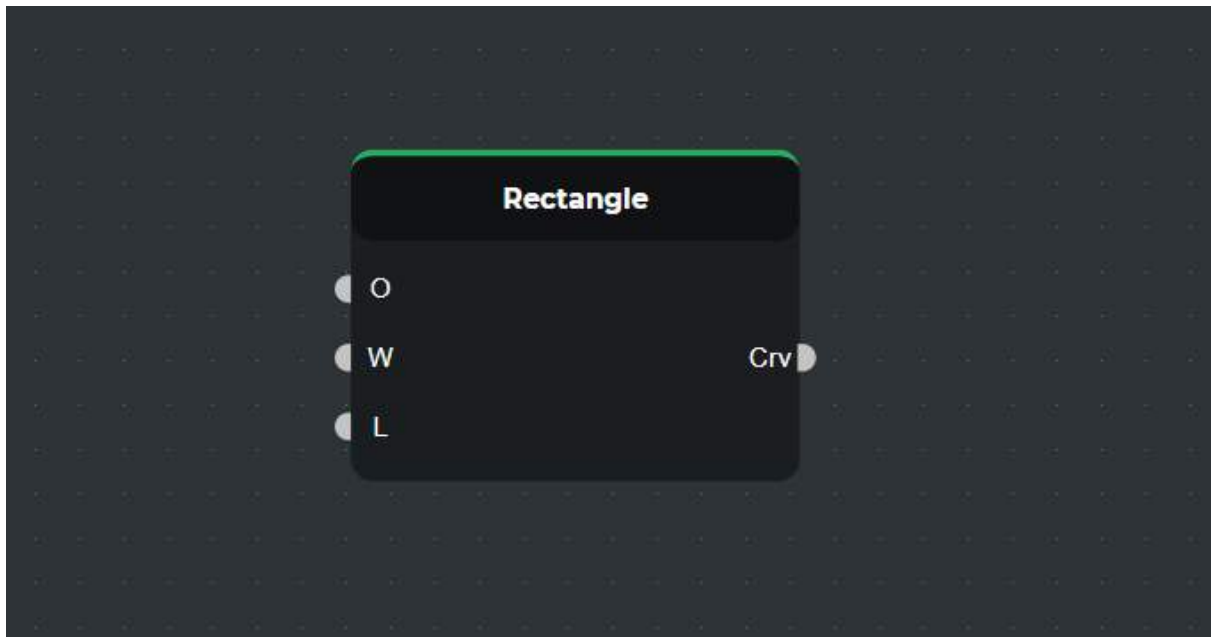
The Workspace

"Workplace" Introduction

The workspace is designed to work with nodes, through which it is possible to get 2D or 3D parametric models.



Double-click on the work area to add a node in the work field, You will see "search" and "selection" areas. Click in the search field (1) to search. Type the name of the node, the acronym, or the word associated with it and the search results will appear in the list (2). After finding the node, it will appear in the working field with one left mouse click.



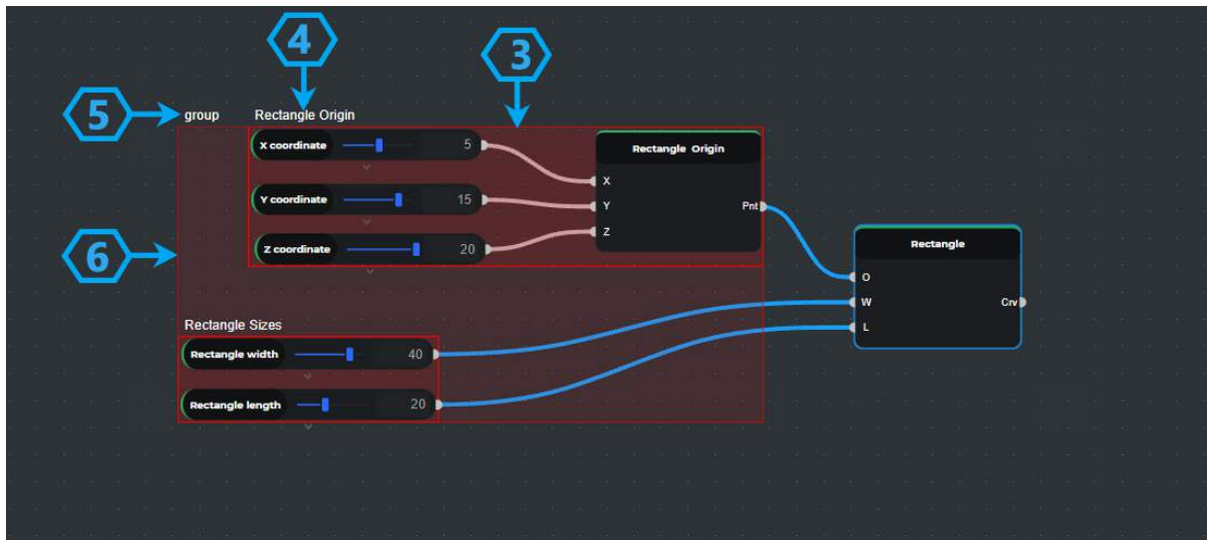
By holding down the left mouse button on the name of the node, you can change its position and place it anywhere you want.

Left-click on the selected node, click the "Delete" button to delete the node in the workspace

There are 3 options for selecting multiple nodes in the workspace.

1. Left-clicking on a space from left to right, you mark the nodes that are completely included in the indicated image.
2. Left-clicking on a space from right to left indicates the nodes that are partially or completely included in the specified image
3. Hold down the "Shift" button and left-click to select any of your nodes.

To remove nodes, you need to hold down the "Ctrl" button and left-click on the nodes to be removed or left-click to select those. Here also the right or left selection option is used.



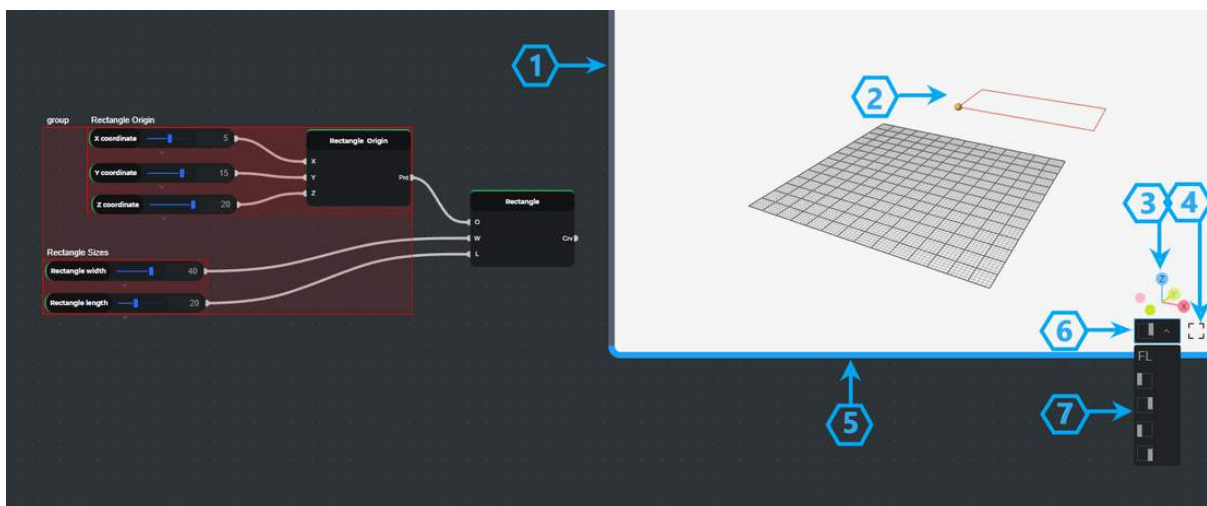
The selected nodes can be grouped. Press the "Ctrl + G" key combination and the selected nodes will appear in one group (3).

All the nodes in the group can be moved at the same time by holding down the left mouse button in the group.

You can change the group name (4) by clicking on the default group name (5) in the left corner of the created group. You can also group several created groups into one (6) by selecting the groups by pressing the "Ctrl + G" key combination.

In the workspace, there is the ability to change the scale, which you can do with the help of a mouse scroll.

“3D View” Introduction



To the right of the workspace is the working 3D view (1).

Here you can see the output values in the working field corresponding to the values of the displayed nodes, displayed in 3D space (2).

In the right corner, you can see the directions of the XYZ (3) coordinate plane axes, as well as the full-screen display mark (4), by clicking on it, the 3D view will take a full-screen view. To reduce it you have to click on the same mark again. To resize the workspace, hold down

the left mouse button and enlarge or reduce the window (5). By holding down the left mouse button, you can rotate the coordinate system and view the image from different angles. You can change the position of the 3D view on the screen by clicking the button to change its position (6), there you have 5 options (7). If you choose the first option, the 3D view will stand out in a separate window. The other 4 options are its position in the upper left-right corners, in the lower left-right corners. To change the 3D view, you need to move the mouse cursor to the sides of the working 3D view. When the cursor changes, click the resize icon with the left mouse button.

Get Started

Work in “Beegraphy editor”

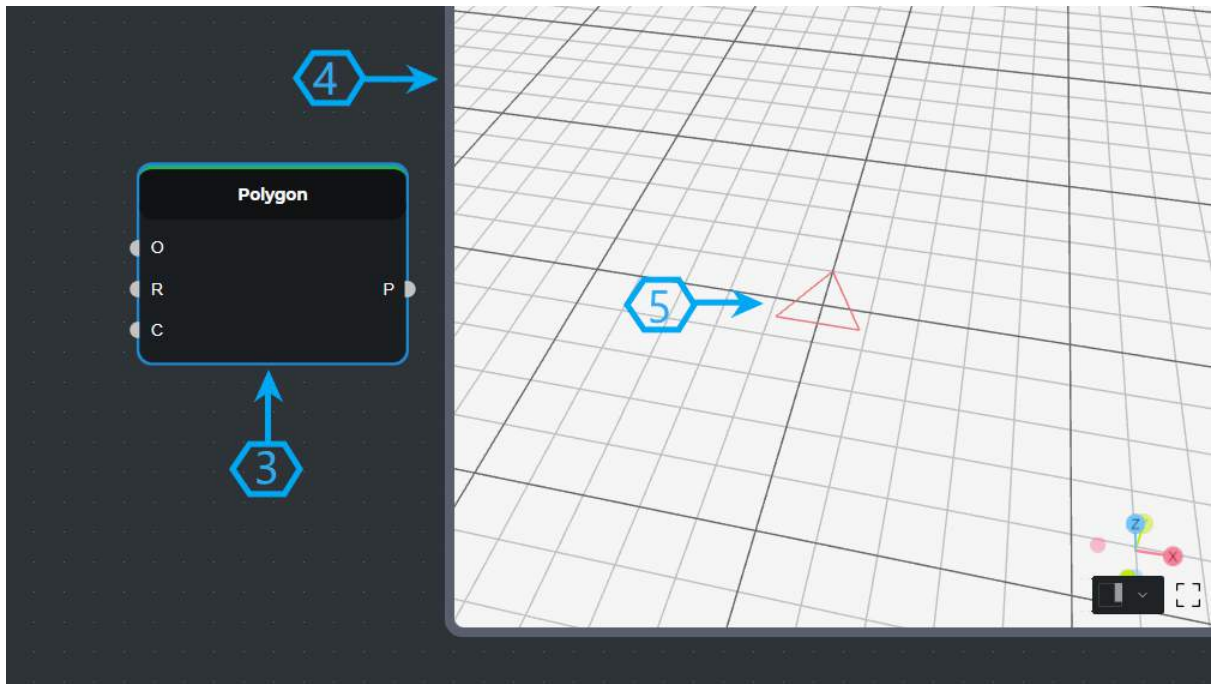
Now, as you have a general idea about "Beegraphy editor" you can get started.

Let us first consider the method of polygon construction.

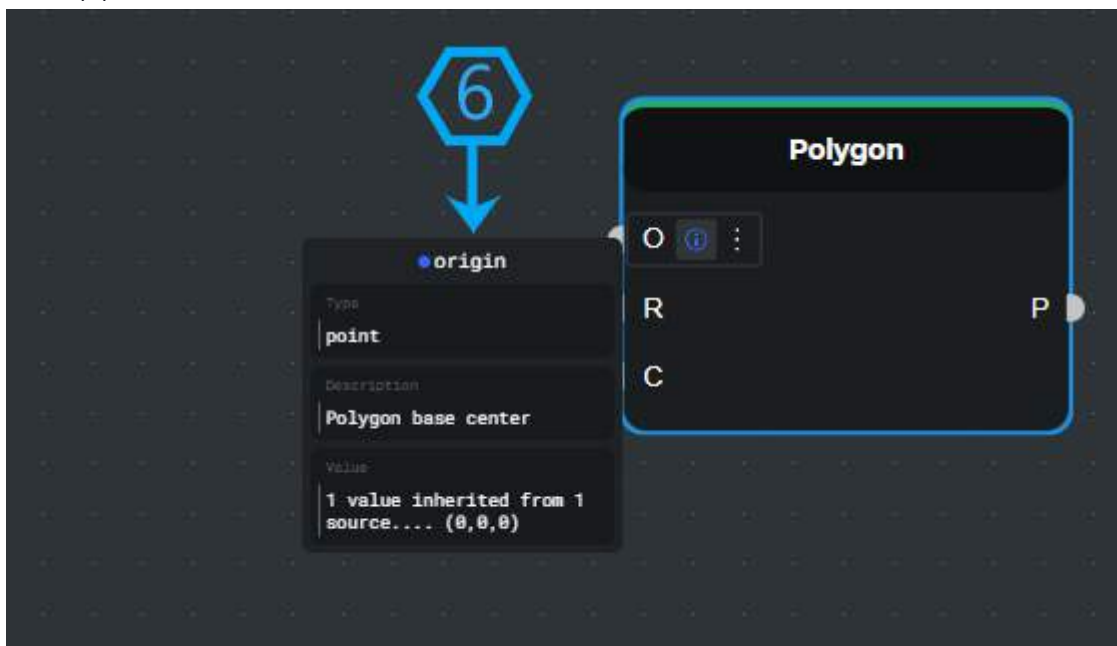
To build a polygon, you need to find the appropriate node from the list and place it in the working field of the "Beegraphy editor".



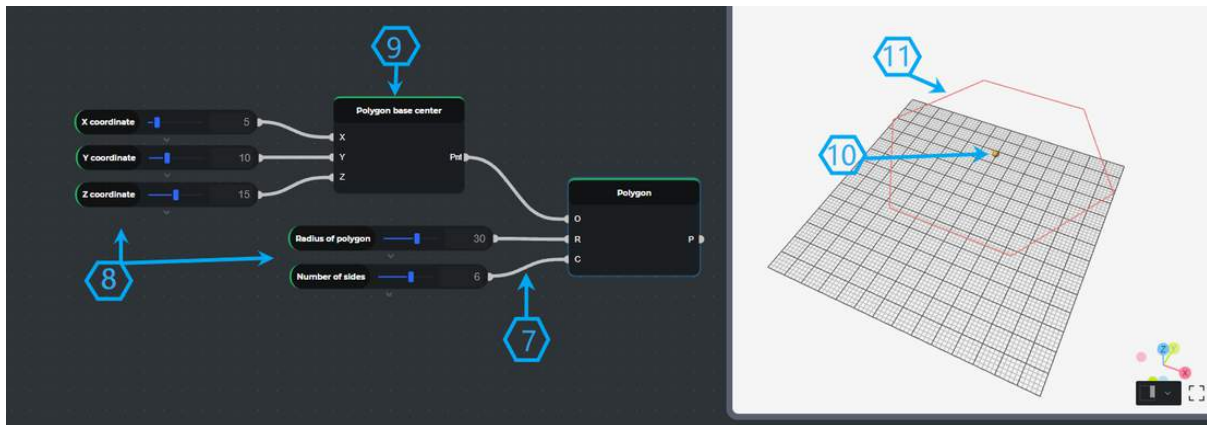
After the left mouse double-clicks, you can search the needed node in the opened window (1). By clicking on the node name (2), it will be placed in the working field (3), at the same time in the working 3D view (4) you will see the output image output from the default polygon input values (5).



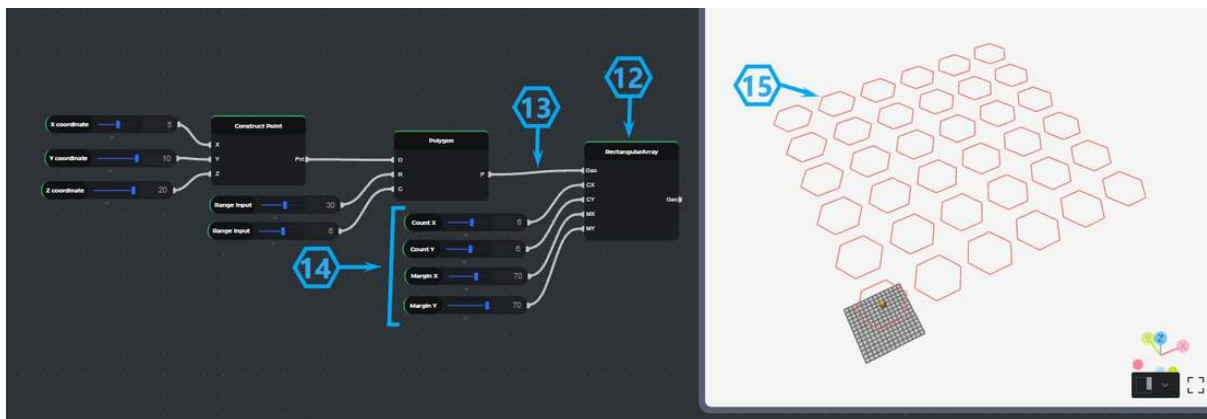
Now let's try to change the input values of the node. For that, it is necessary to connect the appropriate type of nodes to the input nodes with the help of information about the input node (6).



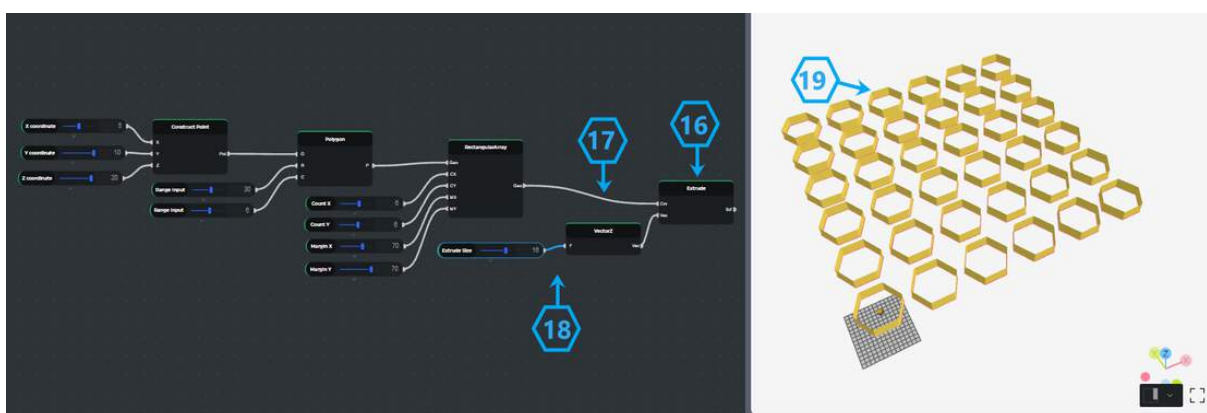
Place the necessary nodes in the working field and connect the corresponding nodes. For that, you need digital ranges (Range Input) (8) and point (Construct Point) (9). If the connections are made correctly and you have no problems with the input data types after connecting the input values, then the data reaches the node input through connected paths (7), the output receives a value or image corresponding to the input data, in this case, you will notice in 3D view. Here you see the created point (10) and the polygon (11). Thus, by performing a few highly simplified operations, you easily obtained a parameterized polygon where the coordinates of its center, the radius of the circle drawn, and the number of sides are parametric.



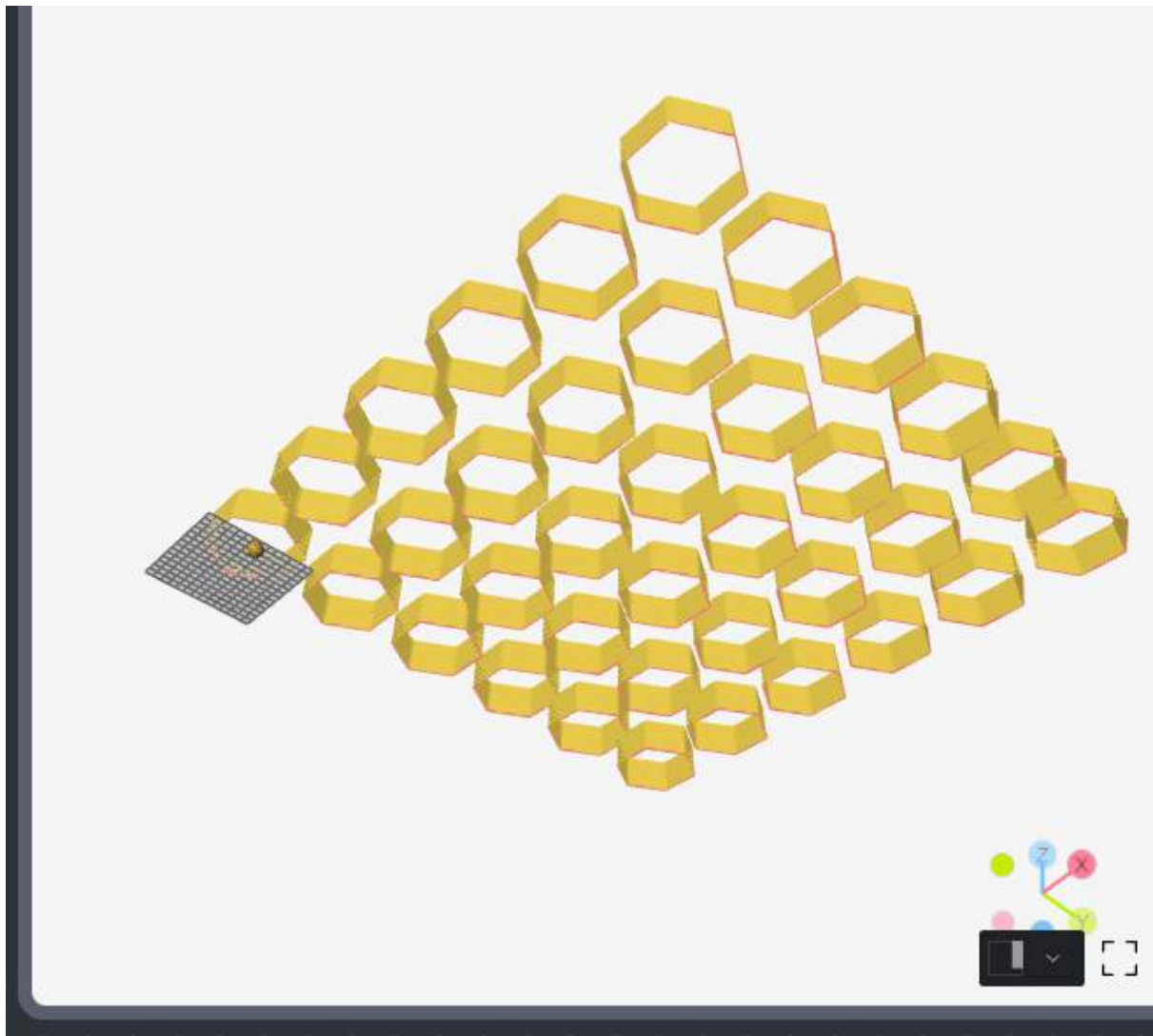
Let's complicate the problem a bit. Take the polygon you created, multiply it by arranging it horizontally and vertically using a very simple tool "Rectangular Array" (12), which gets the image (13) that you need to multiply a few parameters (14). OX and OY are the numbers of images in the direction of the coordinate planes. By introducing these values to the working space, you will see the image you get (15).



Here's how to get a new, more sophisticated model with one simple tool. And now give the surface to the already received model and raise it along the OZ axis. Mark the tool "Extrude" (16), at its input, add the image (17) and the direction (18) in which direction you should raise the surface of the image. After making the connections you see the obtained image (19).



In the working space, you can use the mouse to change the position of the image and see it from different angles.

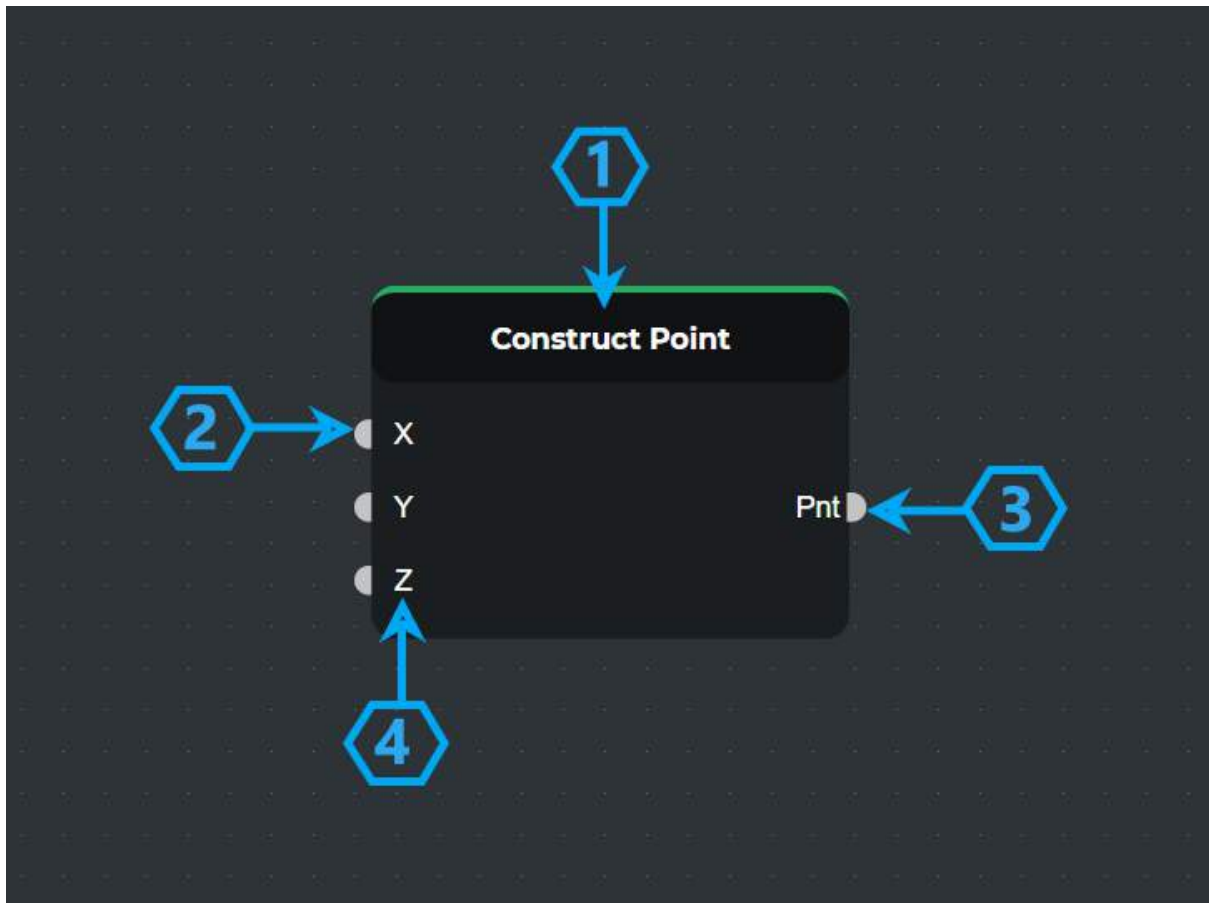


Nodes

A node is a toolkit that performs a specific action depending on the entered values and returns a certain result. In some cases, the result is displayed in the view.

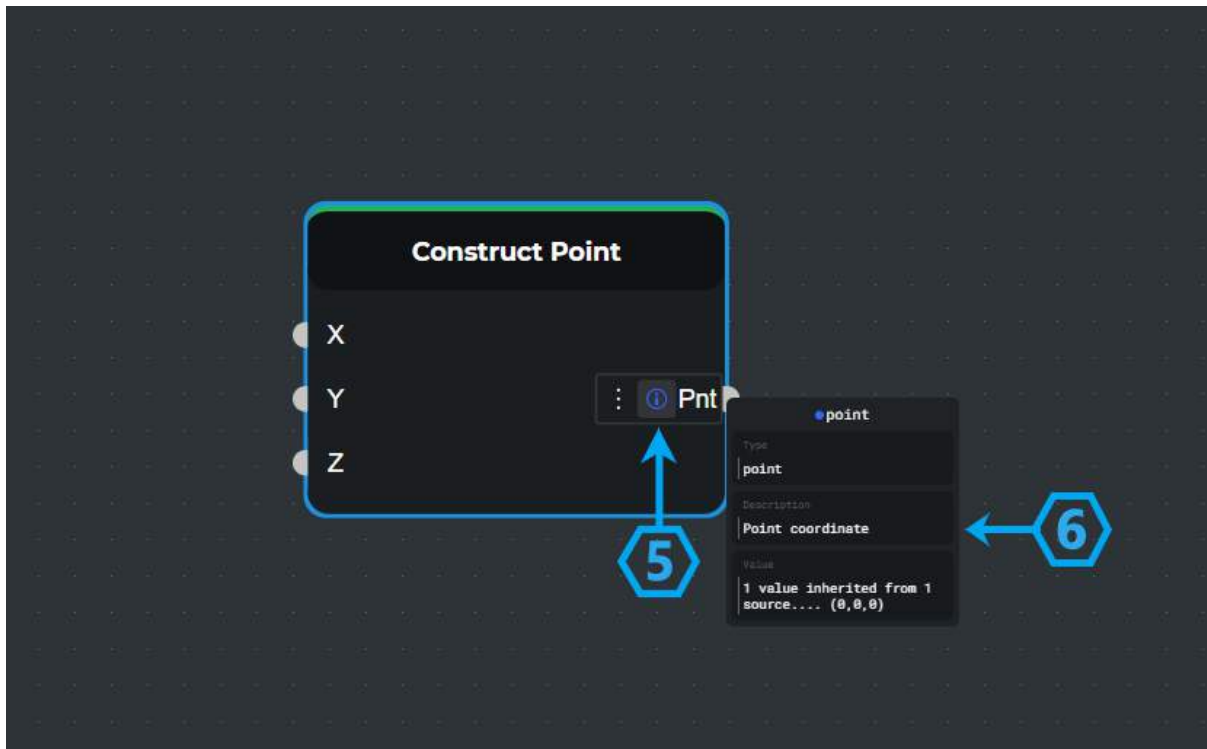
Node Anatomy

All the nodes have names (1) and icons.

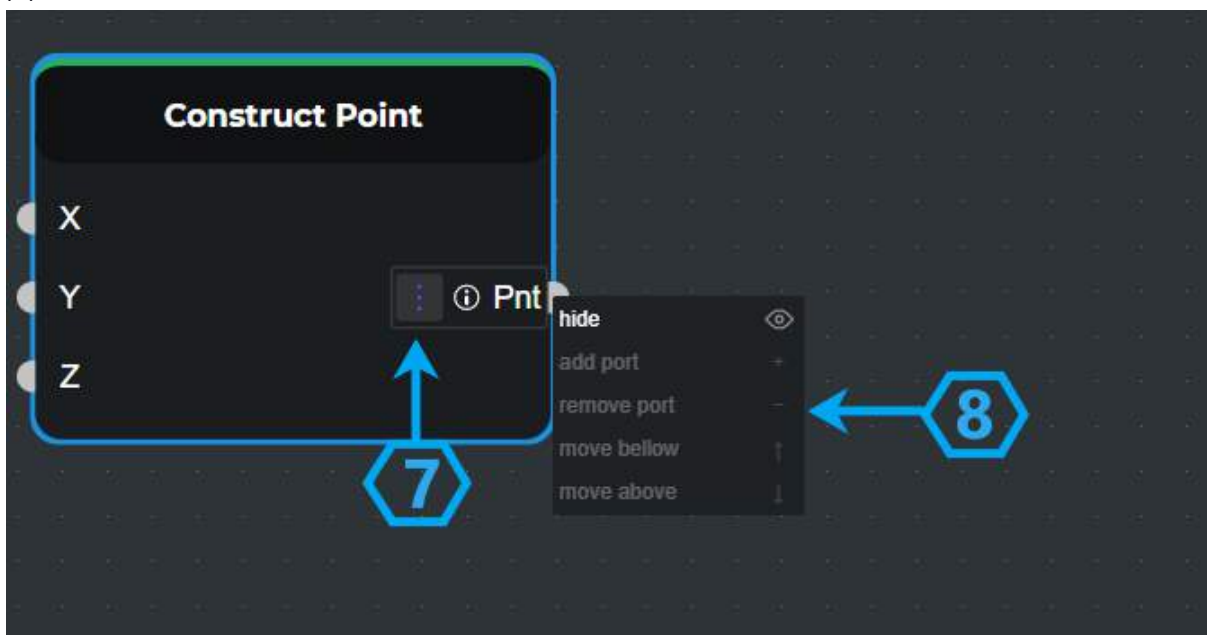


Node name and icon design describe the action to be taken when adding a given node. The node has "input port" (2), "output port" (3), and each node has its label (4), which describes its type. The node has "input port" (2), "output port" (3), and each node has its label (4), which describes its type.

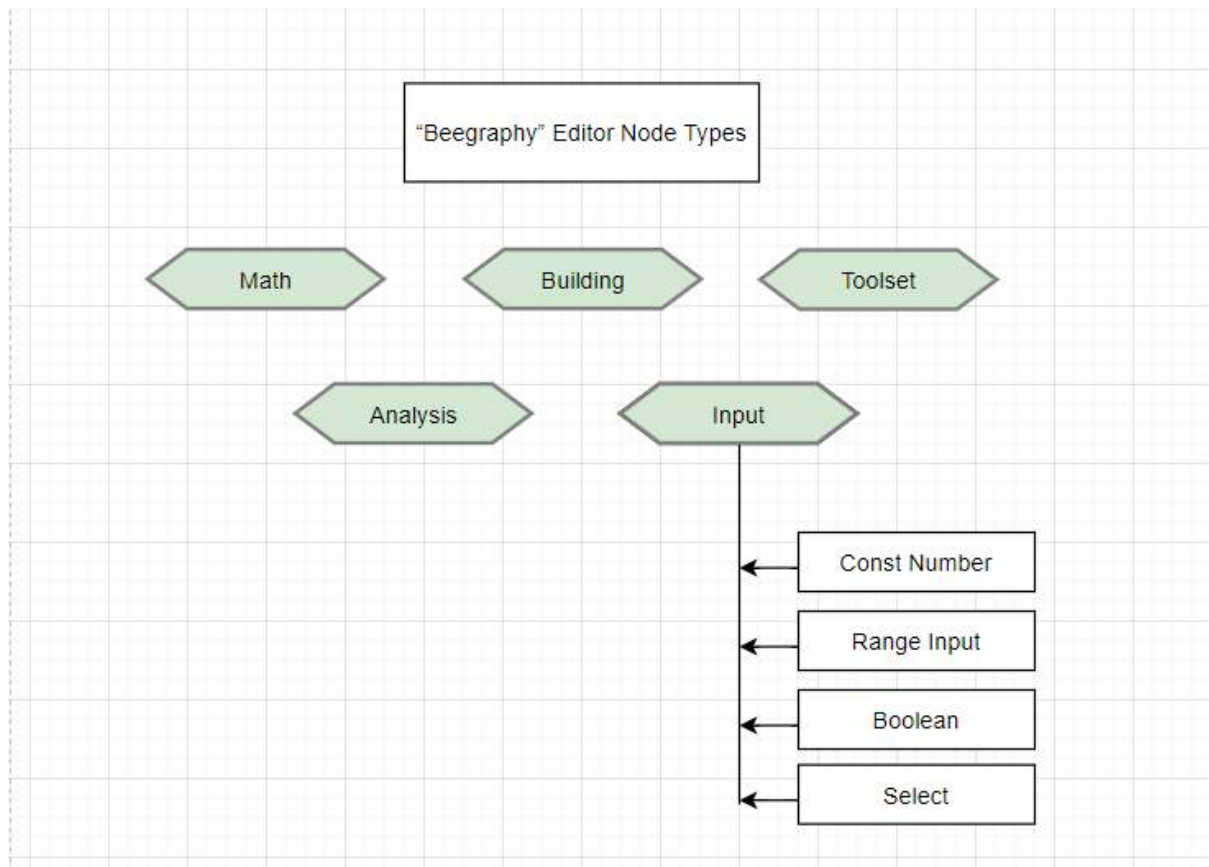
Nodes have an information button (5) that provides detailed information about the node (name, value type given, comment, value of node, or output) (6).



All the nodes have a menu button (7), with which you can perform various node operations (8).



According to their actions, nodes are divided into the following types



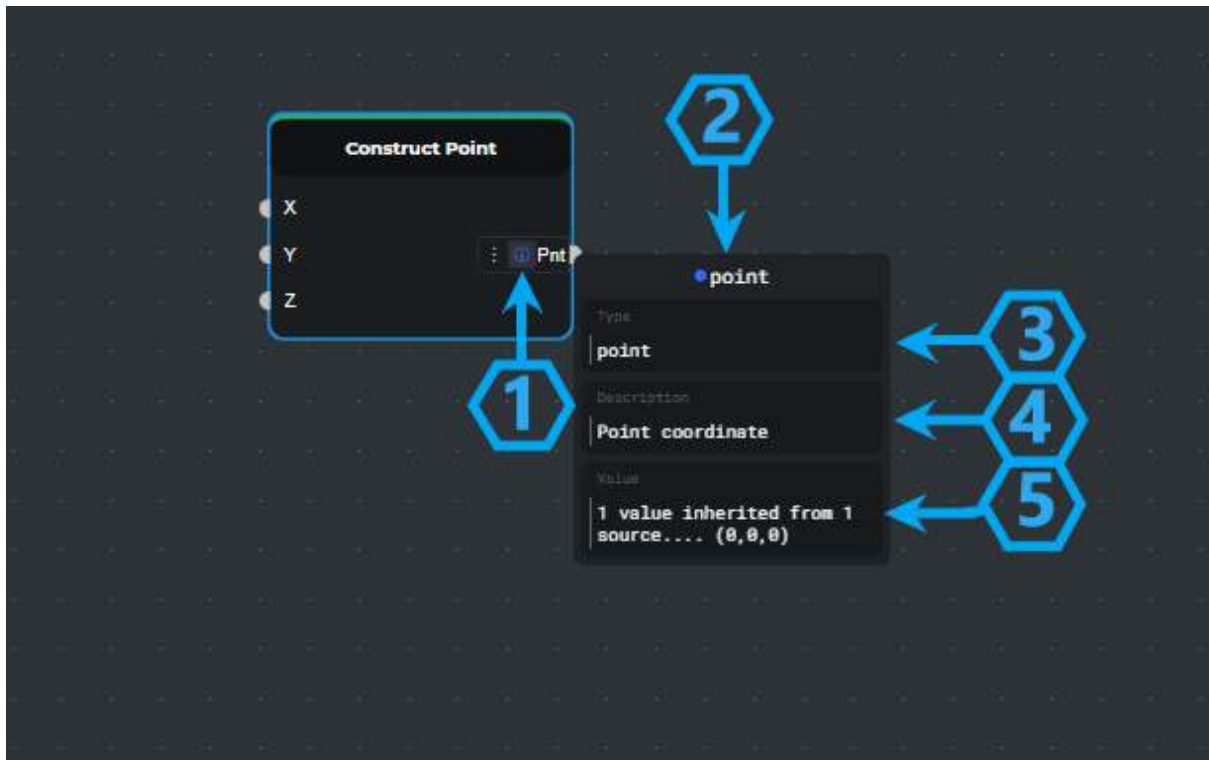
Ports

The nodes are connected by ports. Ports are input-output. Each node has its type and it must be connected to the corresponding type of input value. To connect the nodes, the output port of the connected node must be connected to the input port of the input node. Only the corresponding types need to be connected for the proper working node. If the data does not match, the node turns red, indicating that there is a mismatch between the types: the node cannot work with the given data.

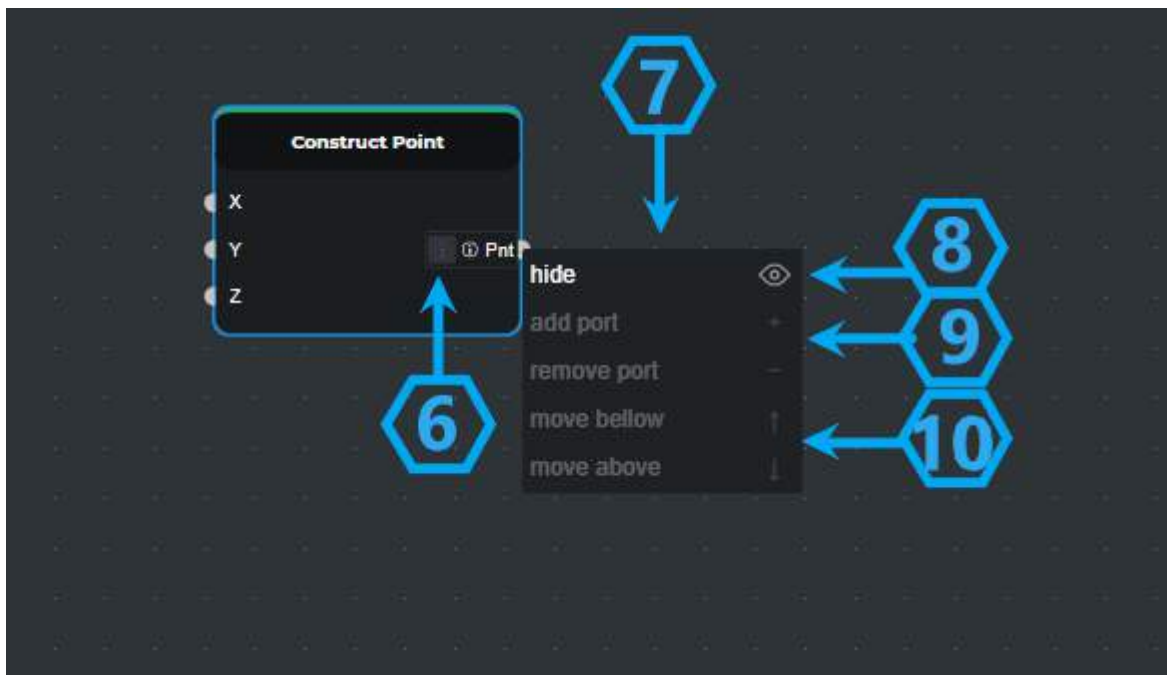
More than one input data can be connected to the input port and that data will be processed by the input node based on the connection sequence. When transferring some data to the input port, it is necessary to hold down the "Shift" button on the keyboard to bridge the connections to that port with the left click of the mouse. An infinite number of output data can be taken from the output port, which can be used in the corresponding input ports of different nodes. To connect one node to another, bridge the connection to the other port with the left click of the output port.

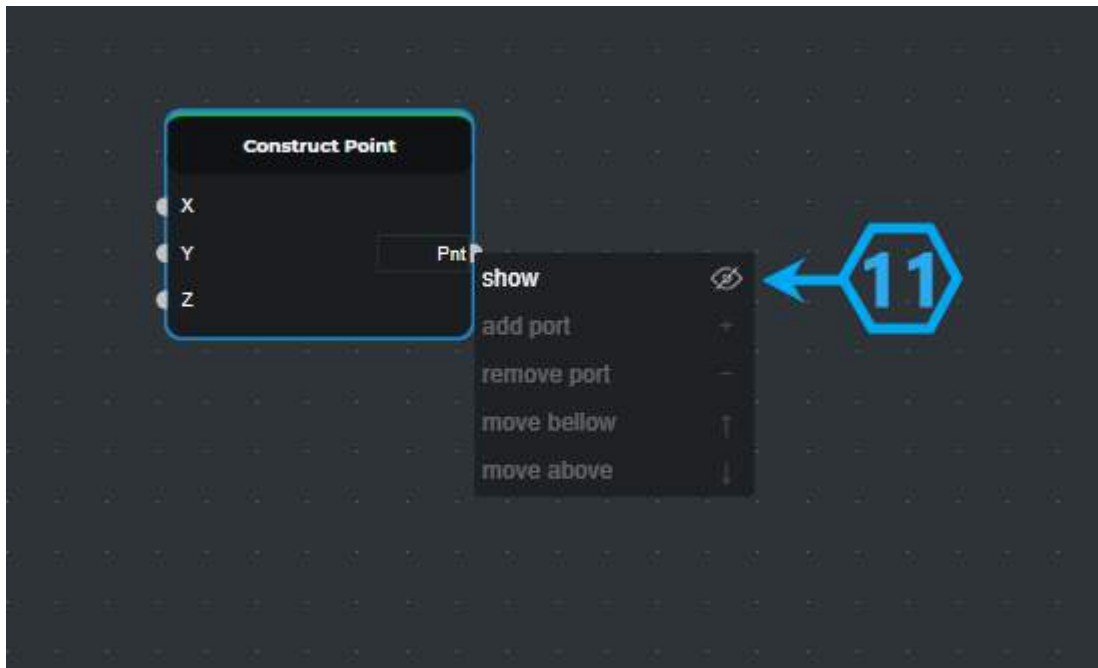
All the ports have "information" (1) and "menu" buttons (6) that make it easy to get information (2) about the port type (3), the explanation (4), and the value (5).

You can perform various operations with the port using the menu toolkit (7).



Nodes that can display their output value in 3D have an additional "hide" button (8) in the menu of the output port. When you click on the button it changes to the "show" button(11), which can be used to show or hide the representation of the node in 3D view.

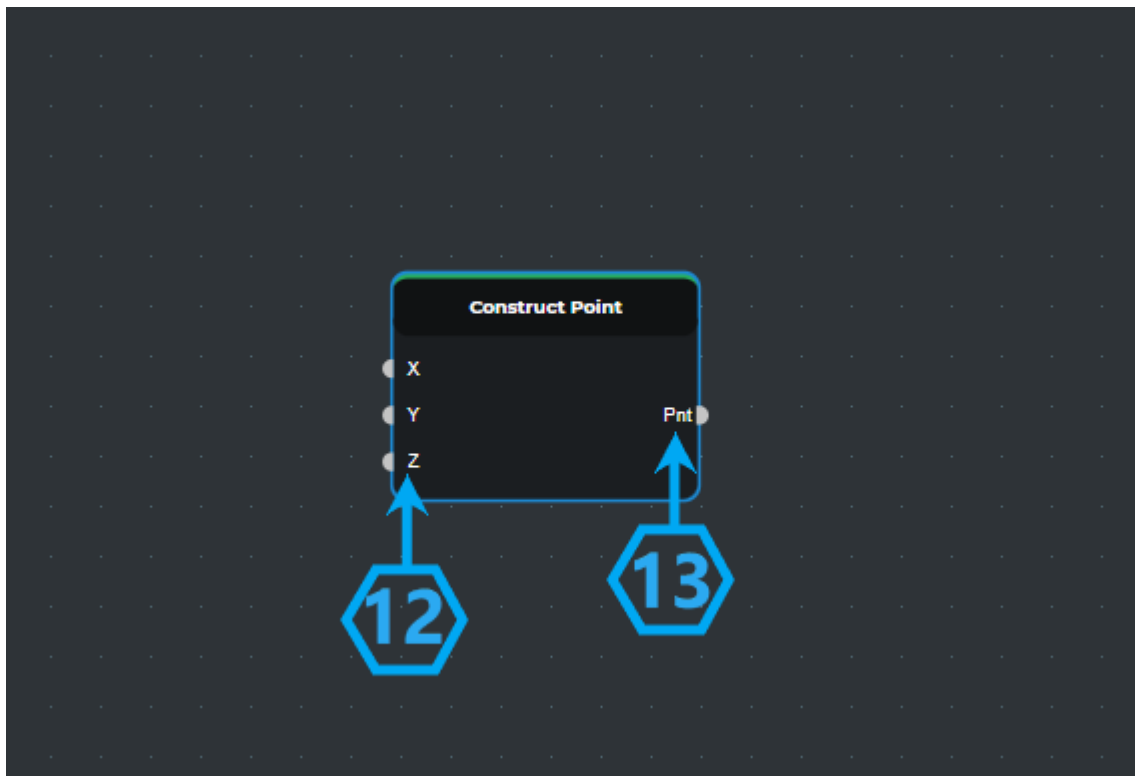




Input-output ports of the node have their names (12) (13). The port's name is the acronym for the type of data from the port type to be connected to that port.

- Curve → Crv
- Radius number → R

You can read the full list of the ports acronyms [here](#).

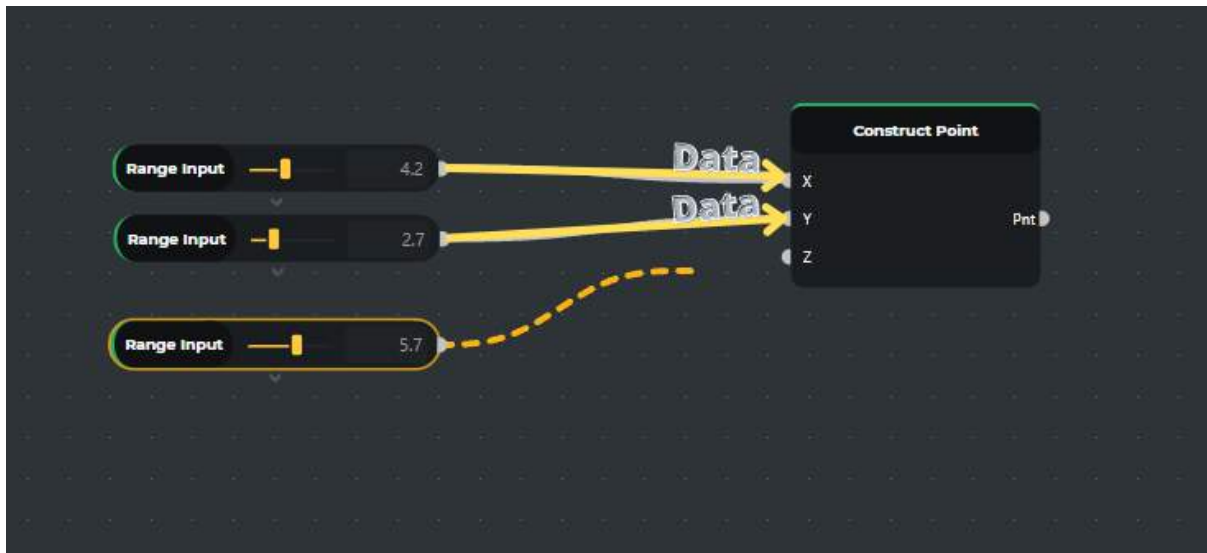


Connections

Connections help us to transfer data from one node to another. The connections have a special design, so it is possible to understand that the nodes are connected to the given node.

Program Flow

You can connect the output node of the input node to the input node of another node via the connections.....



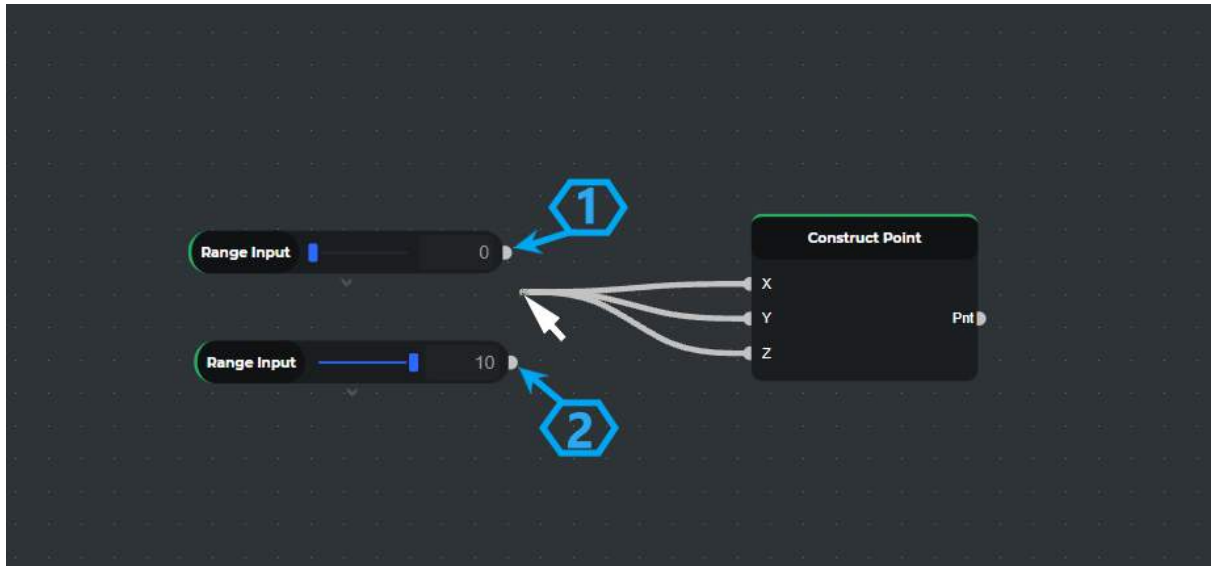
Creating Connection

To create a connection, hold down the node port (1), move the mouse to the other node port (2), and make sure that the output port connects to the input port of the other node.

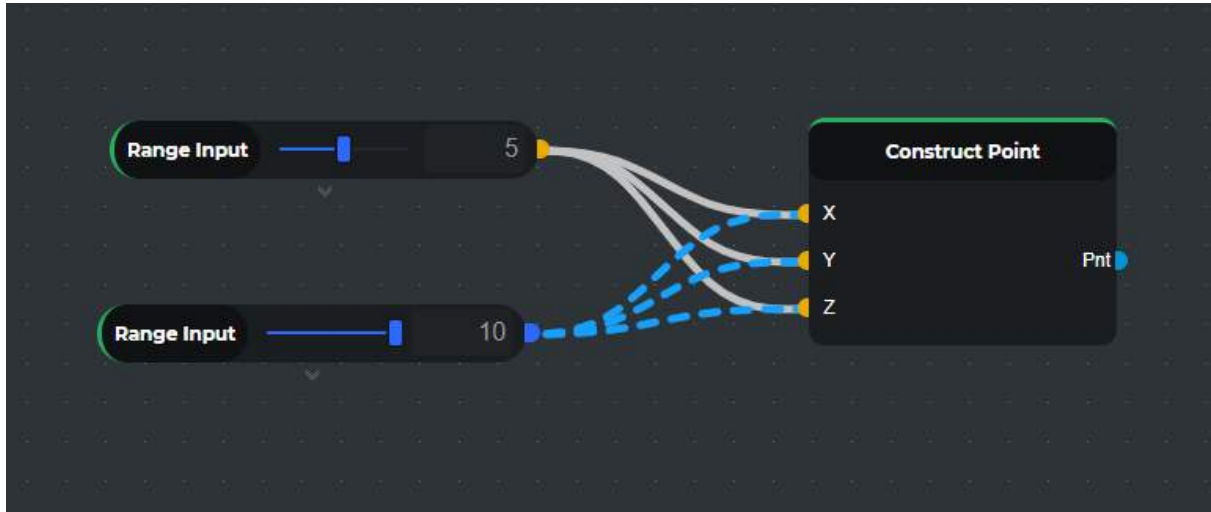


Editing Connections

In case when several nodes are connected from the output node of one port, it is necessary to change the given node with another one. In order not to make that change separately, there is a simplified combination with the keys "Ctrl + Shift" and by holding down the left mouse button you can move the connection from the changed port of the node (1) to a new one by moving the mouse. (2)



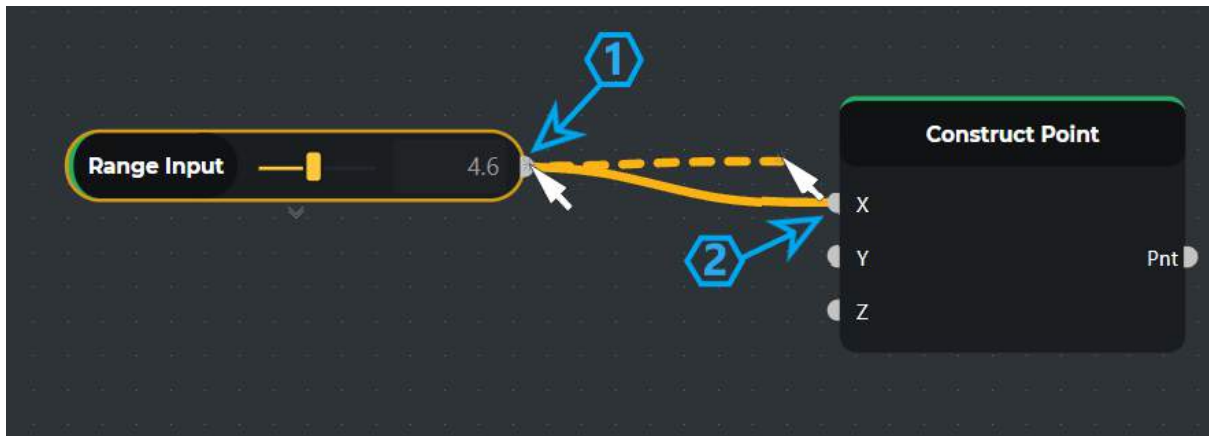
If you want to bridge the same connections to another node, you need to add the "Alt" key to the "Ctrl + Shift" combination and move the copied connection to the new port of the node.



Deleting Connection

There are two ways to disconnect the connected node ports:

- 1 . Hold down "Ctrl" and the left mouse button to move one connection of the node (1) to the other one (2).
- 2 . Hold down "Ctrl" and left-click to the node from which you want to disconnect.



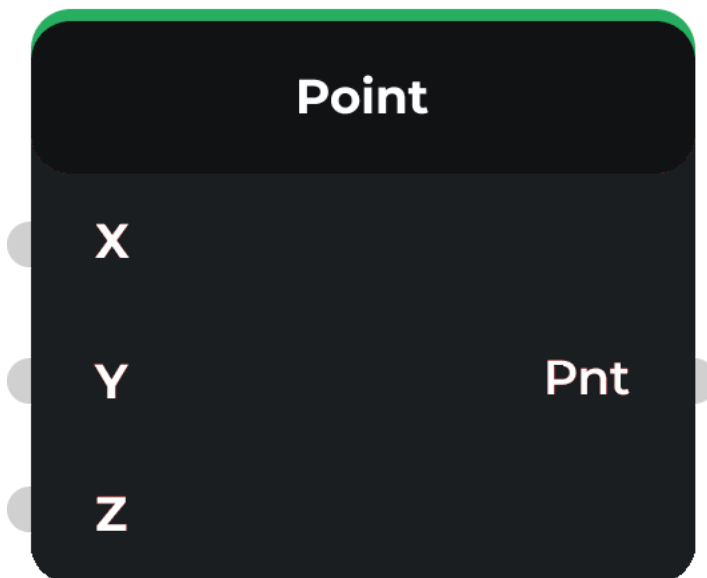
States

To make the work with nodes simpler, we have classified the nodes according to status.

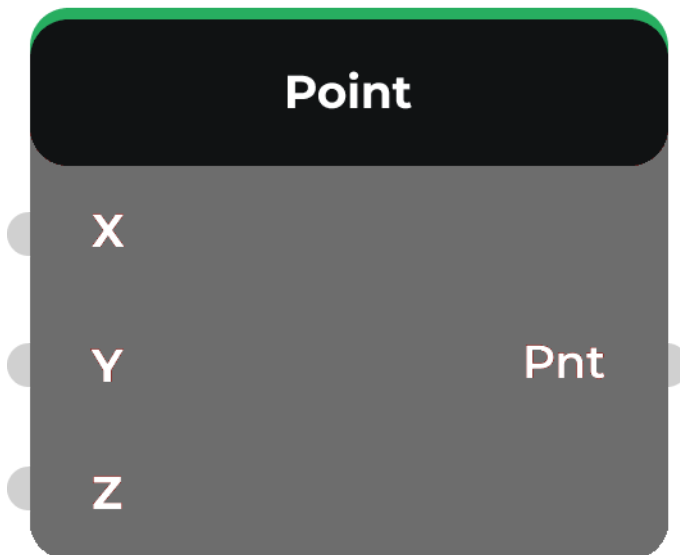
Node color indicates the status of the node.

Node statuses are as follows:

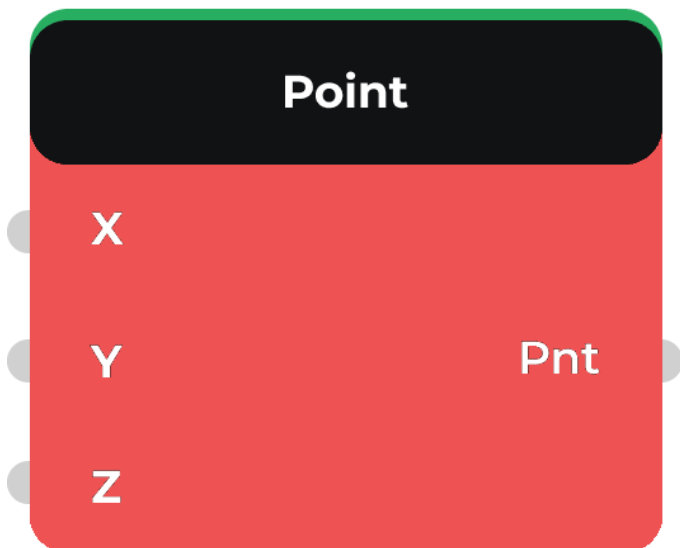
1. Active - Nodes with a Dark Grey Name background are well-connected and have all of their inputs successfully connected



2. Hidden - If the node is colored a light gray background, the node is hidden.



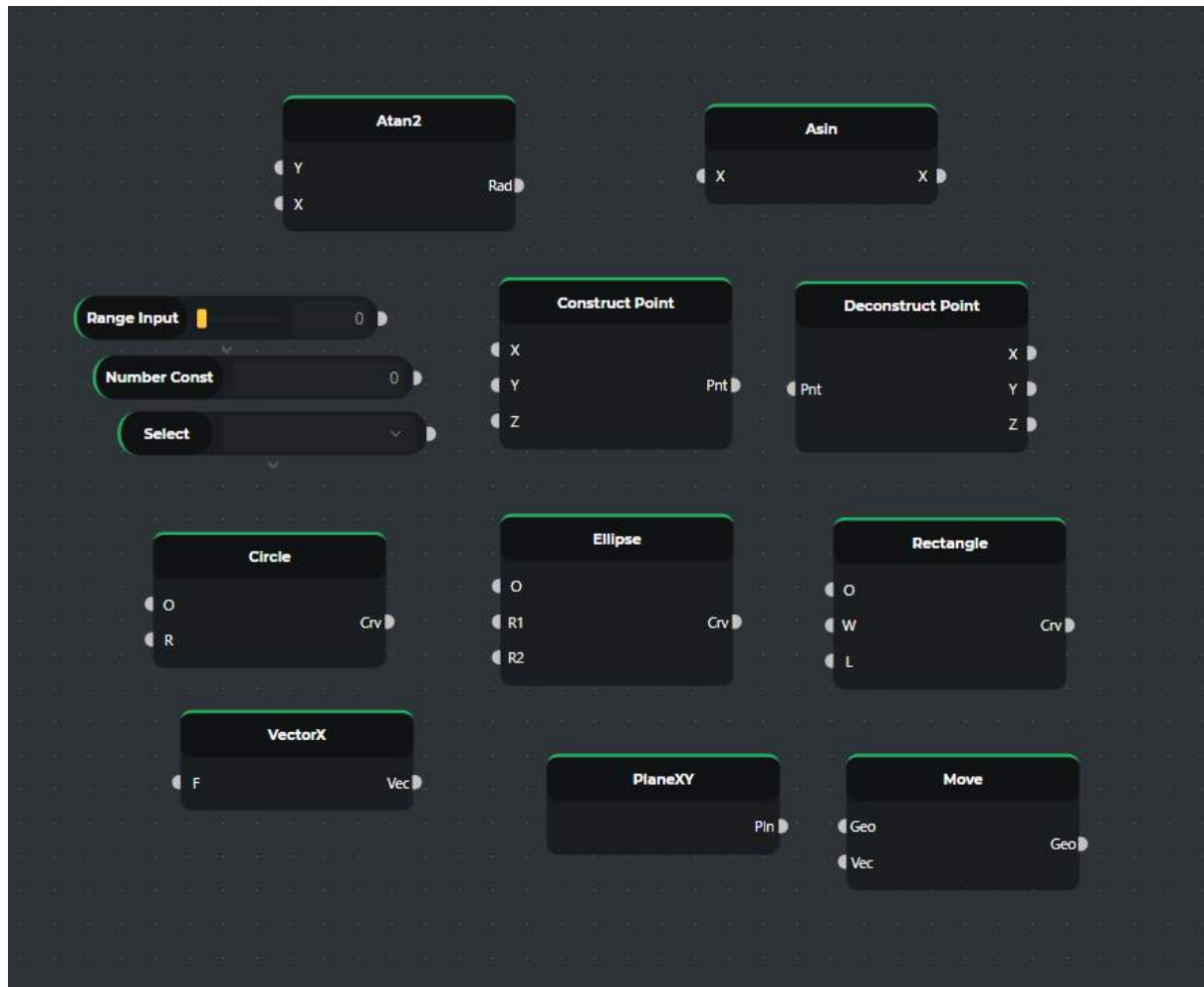
3. Error State - Red indicates that the Node is in an Error State



4. Marked - If the node is colored yellow borders, the node is marked.



The building Nodes Of Editor



Input

Input group nodes are designed to modify and transmit numeric and letter values.

Input nodes are as follows:

1. Number Const
Number Const is for transmitting a fixed number that can be used by other nodes as input.

Range Input returns a number in the range we entered, we can also enter the size of the number change step in the range.

1. Select

Select node returns a variable of type "string". Through it, we can create a list of values. The input field has two columns where we can enter the values of the variables and names.

Selecting the required name from the drop-down list after filling in will return the corresponding value of the name.

1. Boolean

The Boolean node returns a true or false value. When the switch is on, it returns a true, otherwise false value.

Primitive Curves

The Primitive Curves group is designed to produce geometrically simple, linear figures.

The nodes of Primitive Curves are as follows:

1. Line

The line node returns a segment whose start and endpoints are the points given at the node input. The Line node displays the return value in View.

1. Circle

The circle node returns a circle whose center u radius is the values given at the input of the node. The Circle node displays the return value on the View.

1. Circle 3Pt

The node Circle3Pt returns a circle that passes through the 3 points given at the node input. The Circle 3Pt node displays the return value in View.

1. Ellipse

The ellipse node returns an ellipse whose center and two radii are the values given at the node input. The Ellipse node displays the return value in View.

1. Rectangle

Returns the rectangle node to a rectangle whose starting point and both sides are the values given at the input of the node. The Rectangle node displays the return value in View.

1. Rectangle 2Pt

The Rectangle 2Pt node returns a rectangle whose points at both angles are the values given at the node input. The Rectangle node displays the return value in View.

1. Polygon

The polygon node returns a polygon whose center, radius, and several sides are the values given at the node input. The Polygon node displays the return value on the View.

1. Star

The star node returns a star-shaped polygon whose inner and outer radii and several sides are the values given at the node input. The Star node displays the return value on the View.

1. Polyline

The polyline node returns a curve that is constructed by connecting the points given at the node entrance. The other input of the node is given a boolean value, depending on whose value is true or false, the curve will be open or closed. The Star node displays the return value on the View.

Point group

1. Construct Point

The Construct Point node returns a point in a three-dimensional coordinate system whose x , y u z coordinates are the values given at the input of the node. The Construct Point node displays the return value in View.

1. Deconstruct Point

The Deconstruct Point node returns the x , y u z coordinates of the point received at the node input.

Vector group

1. VectorX

Vector X returns an OX axis oriented vector whose length will be the value given to the node input. Vector X does not have a visual display.

1. VectorY

Vector Y returns an OY axis-oriented vector whose length will be the value given to the node input. Vector Y does not have a visual display.

1. VectorZ

Vector Z returns an OZ axis-oriented vector whose length will be the value given to the node input. Vector Z does not have a visual display.

1. Vector XYZ

The VectorXYZ node is designed to obtain a vector starting at point O on a three-dimensional coordinate system whose direction & length depends on the input values x, y, and z. VectorXYZ has no visual representation.

1. Unit Vector

Unit Vector returns the unit converted by the vector received at the node input into a unit vector.

1. Deconstruct Vector

Deconstruct Vector returns the x, y, z coordinates of the vector received at the node input.

Curve Analysis

Closed

The closed node returns a value of type boolean, which indicates whether the curve obtained in the node's key is open or closed. If the curve is closed, it returns true, otherwise false.

Control Point

The ControlPoint node returns the Control Points, knots & weights of the input curve.

End Points

The EndPoints node returns the start & endpoint of the input curve.

Math

1. Addition

The Addition node receives numbers at the input and returns the sum of the numbers received at the given input nodes through the output node.

The Addition node in the "menu" has an "add port" feature that allows you to add new add-ons.

2. Subtraction

The subtraction node receives numbers at the input. Subtracts the input number "A" from the input number "B" and returns the result with the output node.

3. Multiplication

The multiplication node receives numbers at the input & returns the product of the numbers received at the given input nodes through the output node.

The multiplication node in the "menu" has an "add port" feature that allows you to add new manufacturers.

4. Division

The Division node receives numbers at the input. Dividing the number received at input "A" by the number entered at input "B" returns the result with the output node.

5. Absolute

The Absolute node receives the input node and returns the modulus of the received number with one output node.

6. Power

The power node performs the mathematical operation of raising the degree. It gets two numbers from the input node. Port "A" is the base and "B" is the pointer. Returns the result of the operation with the output node.

7. Square Root

The Square Root node performs the square root math operation. Receives one number through the input node. Returns the result of the operation with the output node.

8. Cube Root

The Cube Root node performs the cube root math operation. Receives one number through the input node. Returns the result of the operation with the output node.

9. Opposite

The Opposite node receives a number at the input node and returns the opposite of that number.

10. Minimum

The minimum node receives numbers at the input nodes and returns the smallest of the numbers in the output node.

The minimum node in the "menu" has an "add port" feature that allows you to add new comparators.

11. Maximum

The maximum node receives numbers at the input nodes and returns the largest of the numbers through the output node.

The maximum node in the "menu" has an "add port" feature that allows you to add new comparators.

12. Ceil

The Ceil node gets a number at the input node. The node rounds the number upwards. Returns the entire number obtained through the output node.

13. Floor

The floor node gets a number at the input node. Node rounds the number. Rounds the argument to the nearest whole number. Returns the entire number obtained through the output node.

14. Round

The round node gets a number at the input node. Node rounds the number. If the fractional part of the number is greater than or equal to 0.5, the argument will be rounded to the nearest whole number. If the fractional part of the number is less than 0.5, the argument will be rounded to the nearest whole number.

15. Fround

The ground node gets a numeric value and returns the exact value close to that number.

16. Hypot

The Hypot node receives numbers in the input ports and returns the square root of the sum of the squares of those numbers through the output node.

17. Sine

The sine node at the input port receives the angle with radians and returns the numeric value between -1 and 1, which represents the sine of the resulting angle.

18. Cosine

The cosine node at the input port receives the angle with radians and returns the numeric value between -1 and 1, which represents the cosine of the resulting angle.

19. Tangent

The tangent node in the input port receives the angle with radians u and returns a numeric value that represents the tangent of the resulting angle.

20. SineH

The SineH node at its input port receives the angle with radians and returns a numeric value that represents the resulting angular hyperbolic sine, which can be expressed as e .

21. CosineH

The cosineH node at the input port receives the angle with radians and returns a numeric value that represents the resulting hyperbolic cosine of the angle, which can be expressed as e .

22. TangentH

The TangentH node receives the angle with radians at the input port and returns a numeric value that represents the resulting hyperbolic angle tangent.

23. ArcSine

The ArcSine node in the input port receives the number u and returns the corresponding arcinus by radians.

24. ArcCosine

The ArcCosine node receives a number at the input port and returns the corresponding arccosine by radian.

25. ArcTangent

The ArcTangent node in the input port gets the number u and returns the arctangent of that number (in radians).

26. ArcSineH

The ArcSineH node in the input port gets the number u and returns the inverse hyperbolic sine of that number.

27. ArcCosineH

The ArcCosineH node receives a number at the input port and returns the inverse hyperbolic cosine of that number.

28. ArcTangentH

The ArcTangentH node in the input port gets the number u and returns the hyperbolic arctangent of that number.

29. ArcTangent2

The ArcTangent2 node gets two numbers "Y" and "X" in the input ports. It returns a numeric value from $-\pi$ to π that represents the angle of the point (x, y) . It is the counterclockwise angle, expressed in radians, from the positive x axis to the point (x, y) . Note that the y coordinate is passed to the first node and the x coordinate to the second node.

30. Logarithm

The logarithm node gets a number and returns the natural logarithm (base e) of a number.

31. Logarithm2

The Logarithm2 node returns the base 2 logarithm of a number.

32. Log10

The Log10 node returns the base 10 logarithm of a given number.

33. Log1p

The Log1p node returns the natural logarithm (base e) of 1 + a given number

34. Exponential

The exponential node returns e^x , where x is the given number, and e is Euler's number, the base of the natural logarithms.

35. Expm1

The expm1 node returns $e^x - 1$, where x is the given number, and e the base of the natural logarithms.

36. Imul

The Imul node returns the result of the C-like 32-bit multiplication of the two parameters.

37. Sign

The Sign node returns either a positive or negative +/- 1, indicating the sign of a number passed into the given number.

38. Trunc

The Trunc node returns the integer part of a given number by removing any fractional digits.

39. Count Leading Zeros 32

The Count Leading Zeros 32 node returns the number of leading zero bits in the 32-bit binary representation of a given number.

40. DegToRad

The DegToRad node receives the value in degrees at the input and it converts it to radians.

41. RadToDeg

The RadToDeg node receives the value in radians at the input and converts it to degrees.

42. StringToNumber

The StringToNumber node gets a string value at the input and converts it to a number type.

43. SQRT1_2

The SQRT1_2 node returns the square root of 1/2 which is approximately 0.707.

44. SQRT2

The SQRT2 node returns the square root of 2, approximately 1.414.

45. E

The E node returns Euler's number, the base of natural logarithms, e, which is approximately 2.718.

46. LN2

The LN2 node returns the natural logarithm of 2, approximately 0.693.

47. LN10

The LN10 node returns the natural logarithm of 10, approximately 2.302.

48. LOG2E

The LOG2E node returns the base 2 logarithm of e, approximately 1.442.

49. LOG10E

The LOG10E node returns the base 10 logarithm of e, approximately 0.434.

50. PI

The PI node returns the ratio of the circumference of a circle to its diameter, approximately 3.14159.

Toolset

1. List Length

The list length node returns the length of the list received at the input.

2. List Item

The list item node returns the value corresponding to the index of the received list.

Geometry Introduction

Beegraphy editor is an environment that allows you to create data processing. In addition to numerical data, geometric data can also appear here. Geometric data can be used to obtain aesthetic, high-precision, detailed models. Let's explore the different geometric objects available for use in the Beegraphy editor.

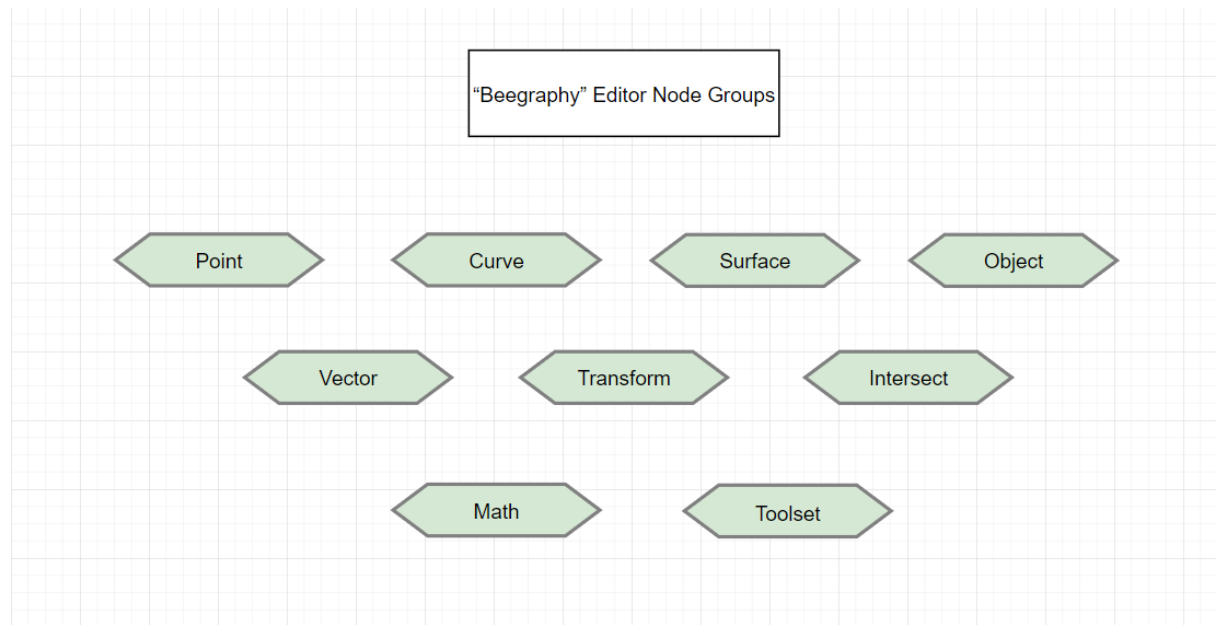
The Basics

Geometry, traditionally defined, is the study of shape, size, the relative position of figures, and the properties of space. We use it to make complex geometric models. Geometry has a hierarchy, it can be imagined in the following way: you need at least two points to create a Curve, you need at least three points to make a Polygon or a Mesh Face, and so on. If there is a curve in the plane, then the latter is characterized by certain points, by the formula given by them. With the help of these points, we can smoothly change the curve. Let's take a moment to look at the relationship between the Abstract and Hierarchical descriptions of Geometry. Because these two concepts are related. we can quickly arrive at a conceptual roadblock once we start developing deeper workflows or models.

1. A **Point** (defined by coordinates) doesn't have any dimensions to it - it's just numbered describing each coordinate
2. A **Line** (defined by two points) now has *one* dimension - we can "walk" the line either forward (positive direction) or backward (negative direction)
3. A **Plane** (defined by two lines) has *two* dimensions - walking more left or more right is now possible
4. A **Box** (defined by two planes) has *three* dimensions - we can define a position relative to up or down

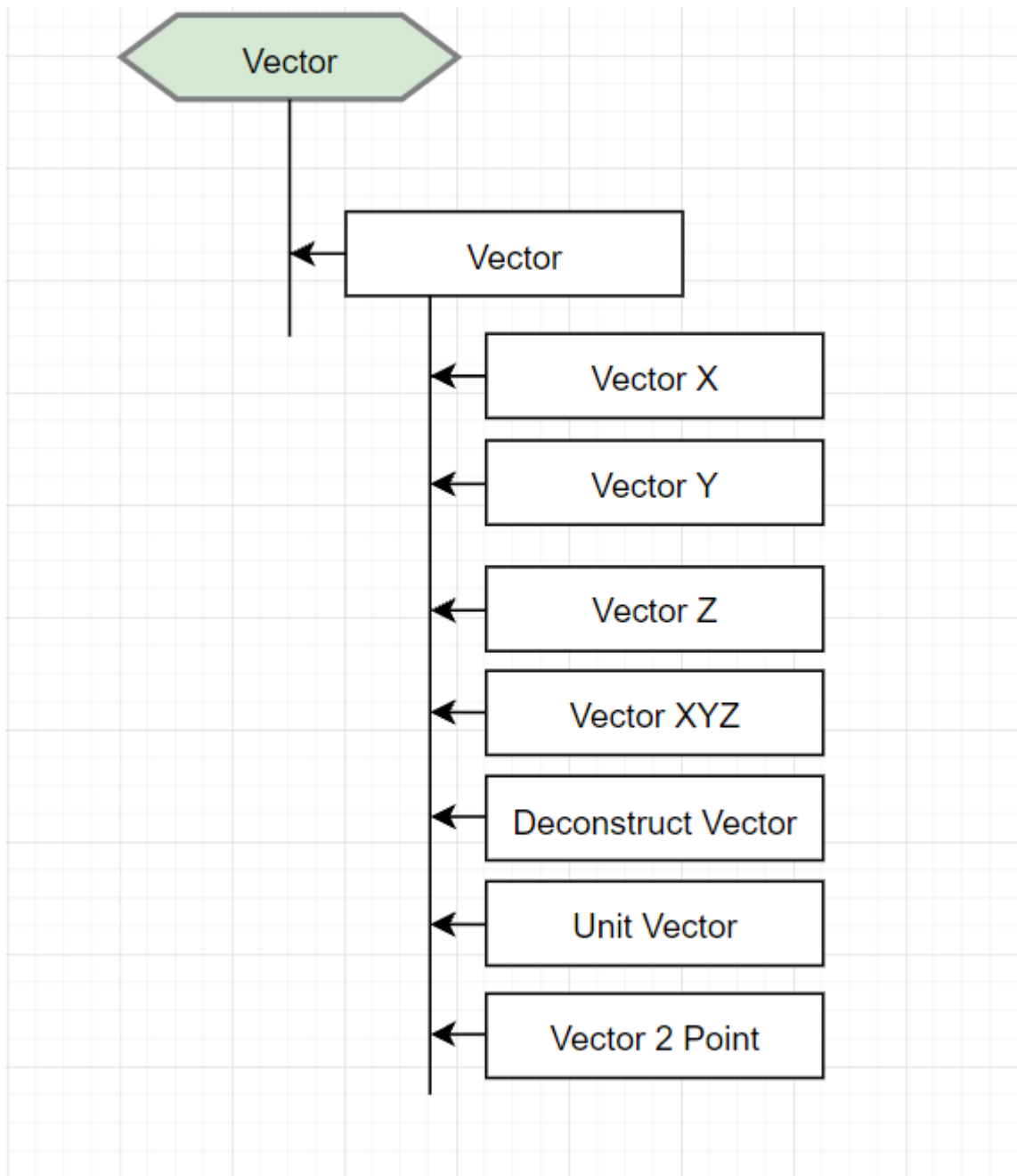
Dimensionality is a convenient way to start categorizing Geometry but it's not necessarily the best. After all, we don't model with only Points, Lines, Planes, and Boxes. Furthermore, there is a whole other category of Geometric types that are completely abstract and they define properties like vectors, different types of curves, works in different planes.

Beegraphy editor has such a hierarchical structure:



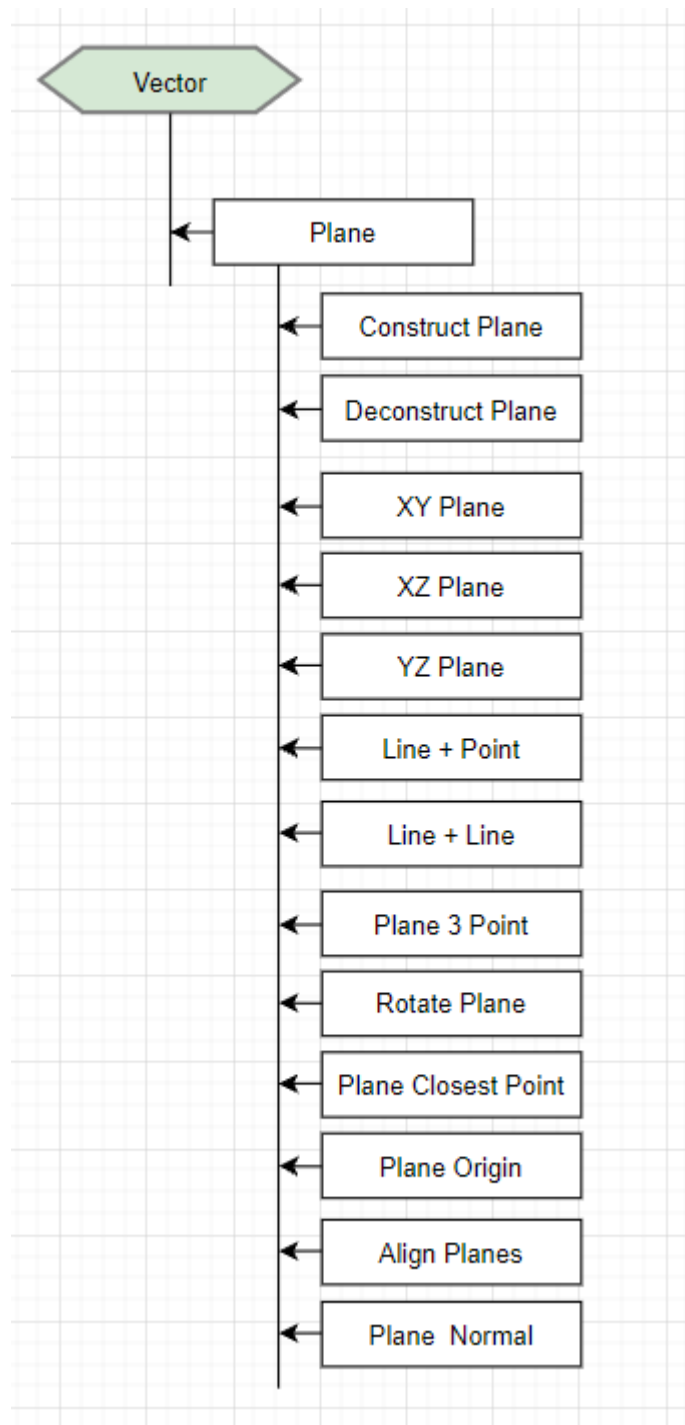
Vector

A vector is a geometric quantity describing Direction and Magnitude. It has a starting point, an ending point. Vectors are a very important node for building models in Beegraphy editor. And since they are aids in building, they do not reflect on the way we work when we create them in the field.



Plane

The plane can be imagined as a rectangle, which can be infinitely magnified in all directions. There are several ways to construct a plane, for example, with three points, two lines, a point and a line. By creating the plane, we can build the model we need on that plane.



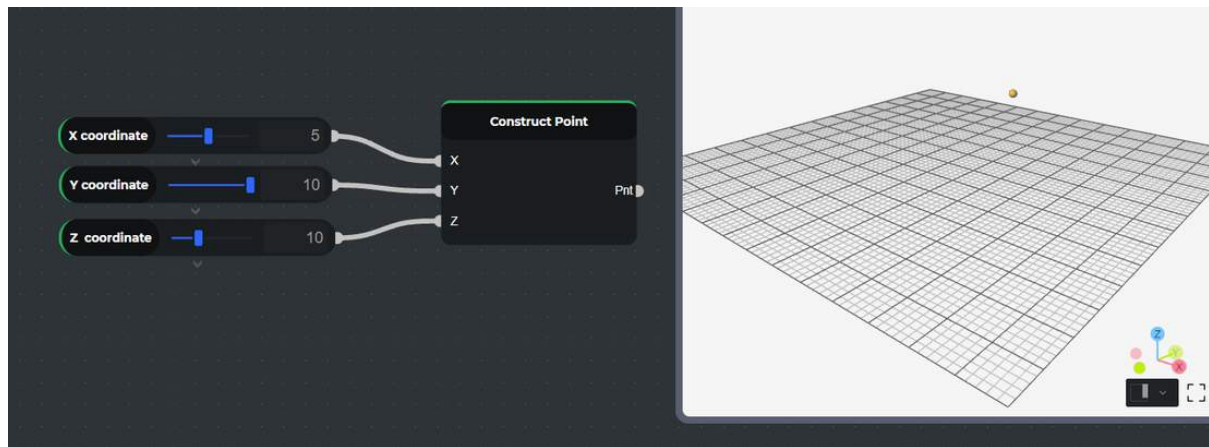
Coordinate System

If we are comfortable with Planes, we are a small step away from understanding Coordinate Systems. The XYZ coordinate system consists of 3 component planes (XY, YZ, XZ) that characterize it.

Points

Points are the foundation upon which all other geometry is created - we need at least two points to create a Curve, we need at least three points to make a Polygon or a Mesh Face,

and so on. Defining the position, order, and relationship among Points (try a Sine Function) allows us to define higher-order geometry like things we recognize as Circles or Curves. A Point is defined by nothing more than one or more values called coordinates. In the Cartesian three-dimensional coordinate system, a point is defined by three values: [X, Y, Z].



Curves

The term **Curve** is generally a catch-all for all different sorts of curved (even if straight) shapes. Capital "C" Curve is the parent categorization for all of those shape types - Lines, Circles, Splines, etc. It has control points, if all these points are in one place, then the curve is smooth, and if the beginning point and the endpoint of the curve are the same, then the curve is closed. The section is the simplest form of the curve, the construction of which is the part between the two points. Combining several parts, we will get a polyline. If the polyline is closed, it is polygonal.

NURBS

NURBS (Non-uniform Rational Basis Splines) are mathematical representations that can accurately model any shape from a simple two-dimensional Line, Circle, Arc, or Rectangle to the most complex three-dimensional free-form organic Curve. Because of their flexibility and precision, NURBS models can be used in any process from illustration and animation to manufacturing.

Degree: The Degree of the Curve determines the range of influence the Control Points have on a Curve; where the higher the degree, the larger the range. The Degree is a positive whole number. This number is usually 1, 2, 3, or 5, but can be any positive whole number.

Control Points: The Control Points are a list of at least Degree+1 Points. One of the easiest ways to change the shape of a NURBS curve is to move its Control Points.

Weight: Control Points have an associated number called a weight. Weights are usually positive numbers. When a Curve's Control Points all have the same weight (usually 1), the Curve is called non-rational, otherwise, the curve is called rational.

