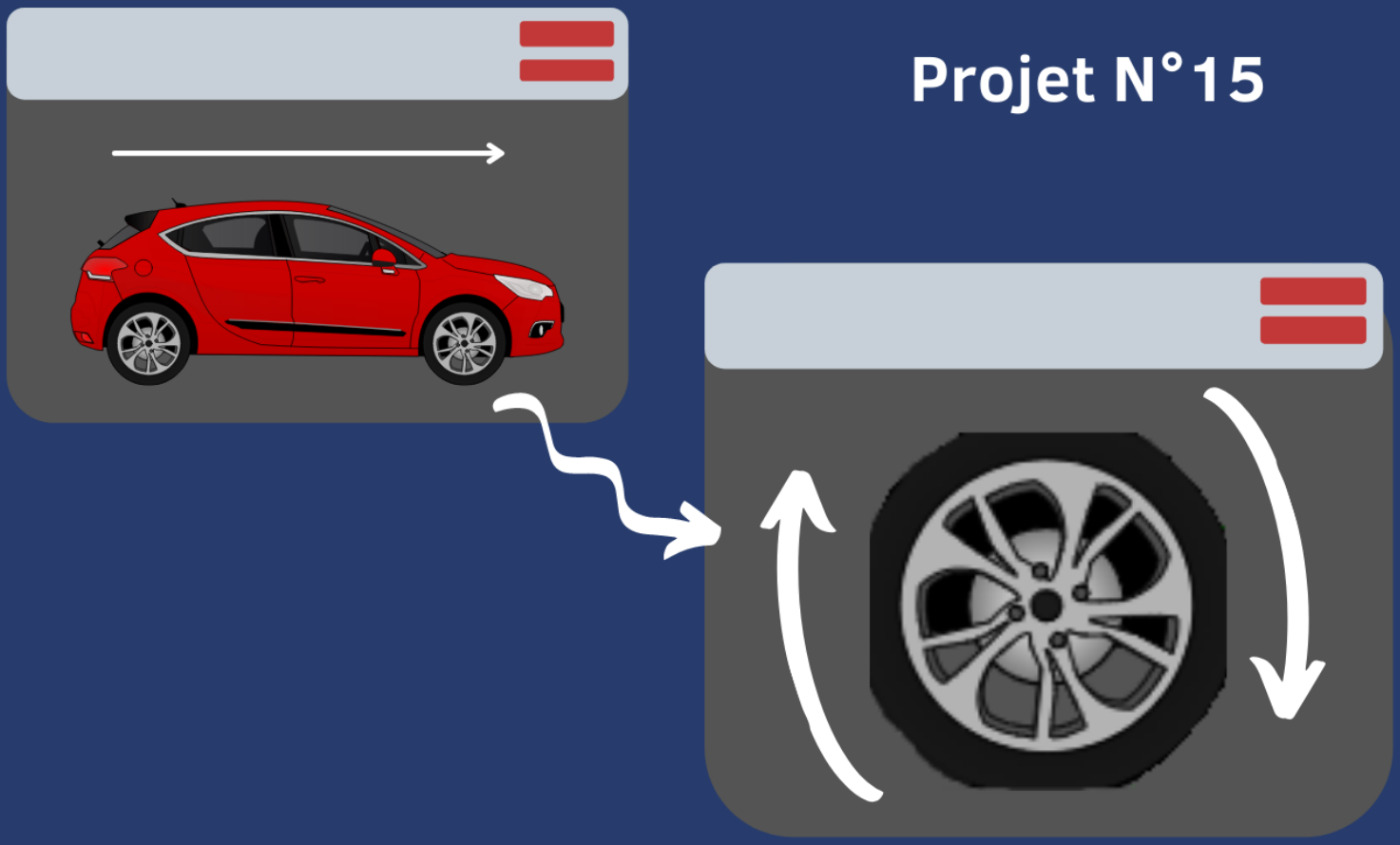
Projet N°15



Atelier d'animation 3D

Manuel d'utilisation

DUT Informatique 2021/2022 - S4

Andréas COTTET
Julien DERAMAIX
Quentin TOURNIER
Nathan VACHER

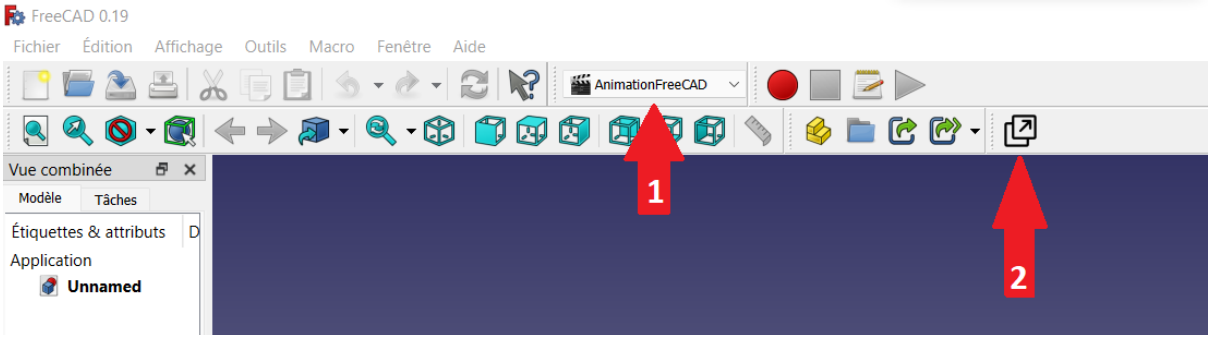
Tuteur & Commanditaire : Marc Dalmau

Accessing the workshop

How do I access the scenario construction window?

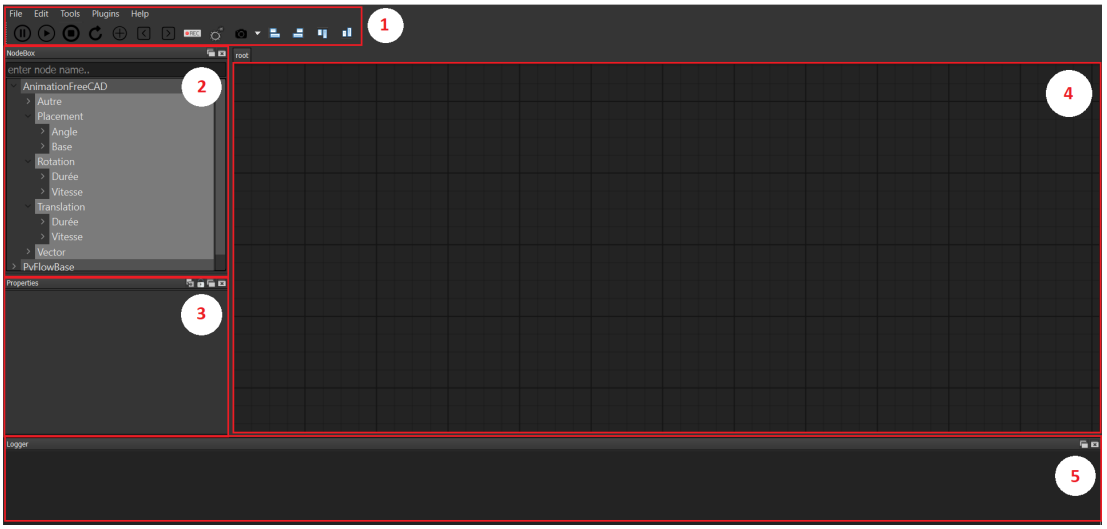
First of all, you will first need to create or open a project on FreeCAD. Then select "AnimationFreeCAD"(1) in the list of installed modules and finally click on the "Open PyFlow"(2) button.

Important: If the module does not appear in the list of installed modules, we invite you to review the installation of our 3D Animation Workshop



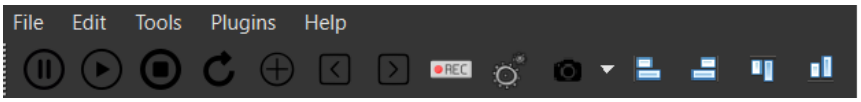
Organization of the interface of the workshop

As a reminder, our 3D Animation Workshop is based on PyFlow which gives access to a pre-made interface that you can see below. We notice that this interface is divided into several parts and that it is adapted for our Workshop.

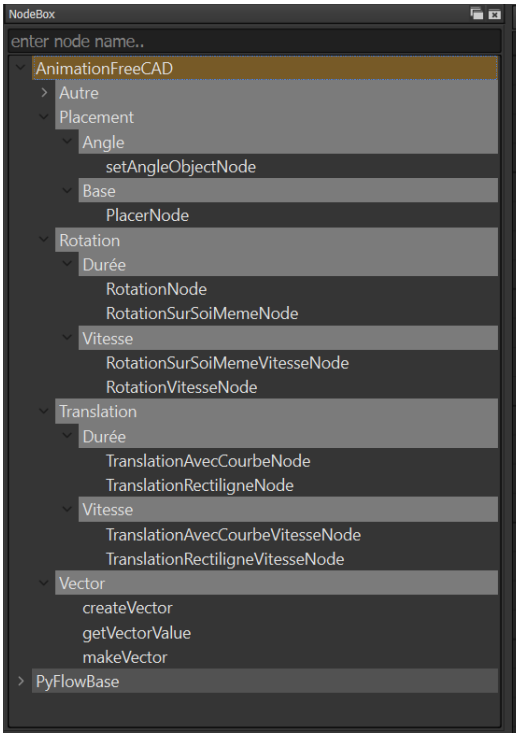


1. The PyFlow

menu This menu gives you access to default buttons of PyFlow such as the screen capture of the scenario creation area or to manage the alignments of the layout of the nodes and new buttons which will be detailed to you in the functionalities part of the guide.



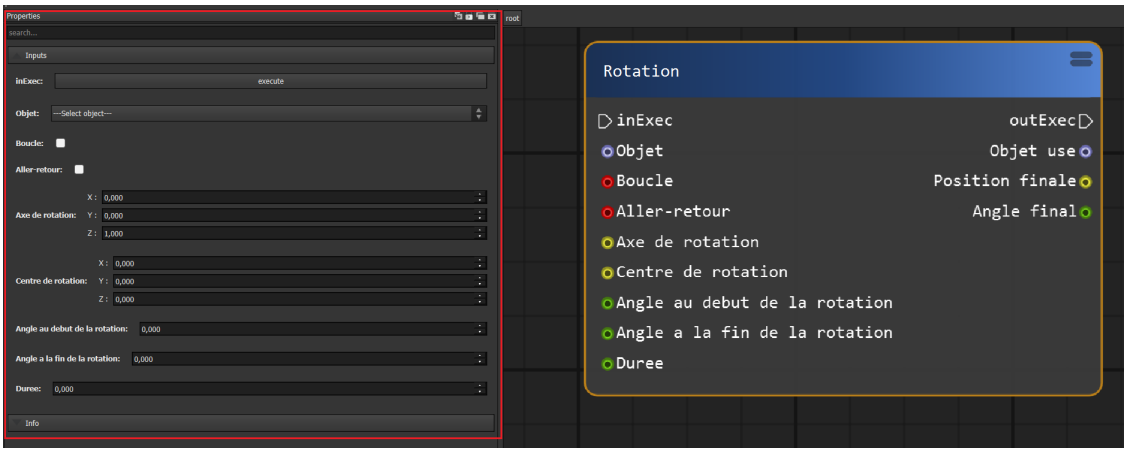
2. The nodes



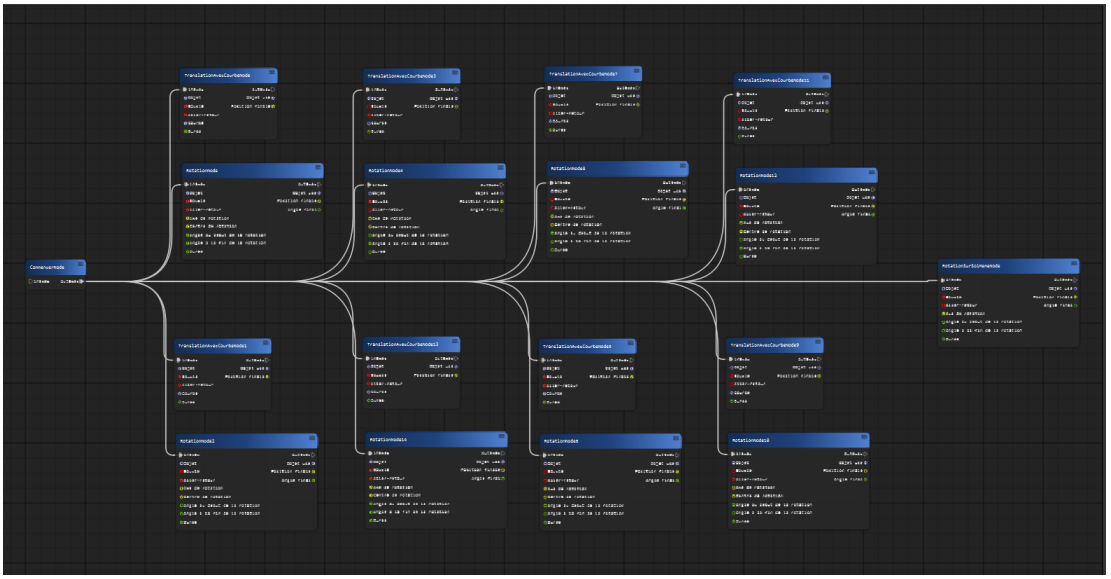
menu This drop-down menu gives you access to all the nodes available to create your scenario. It is available in the “Nodebox” window of Pyflow.

3. The properties windowThis

window allows you to view all the parameters to be entered for a node. In this example we must enter an object to move, an execution mode either an execution or a loop execution or a round trip execution, an axis and a center of rotation, a start and end angle and finally a duration.



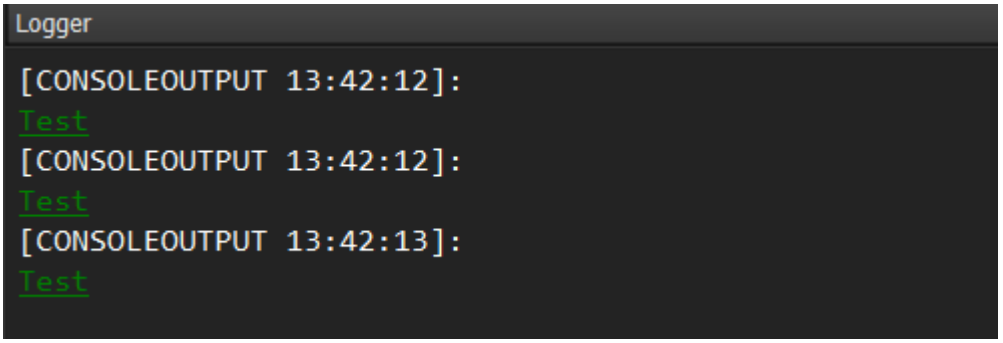
4. Scenario creation



area This area allows you to add nodes to your scenarios, to make them interact with each other.

5. The Logger

This window allows you to see the logs made by the framework, that is to say the error messages and any messages displayed by the execution of the code.

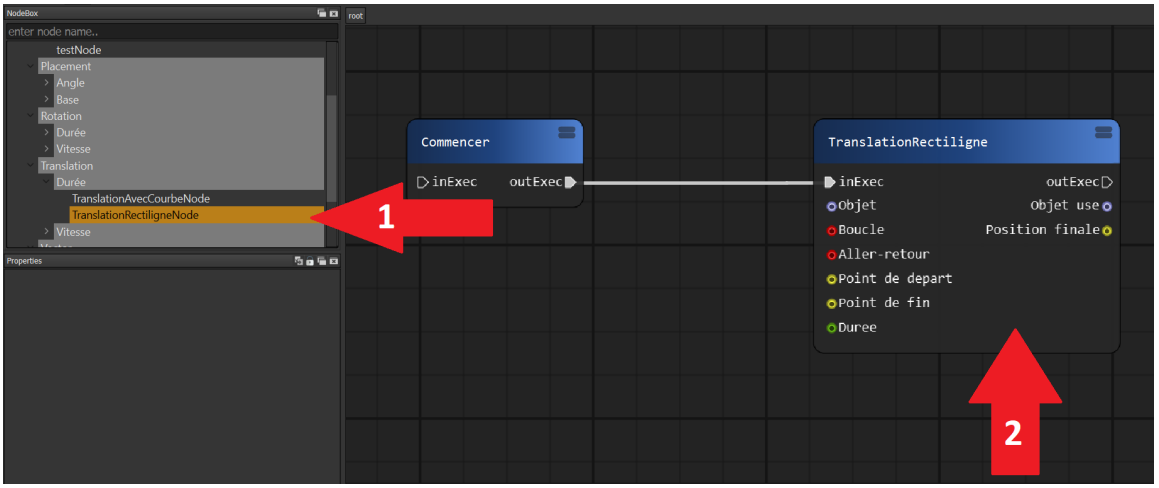


Features

Create a scenario

To create a scenario you simply have to add the visual blocks (nodes) you want in the scenario creation area.

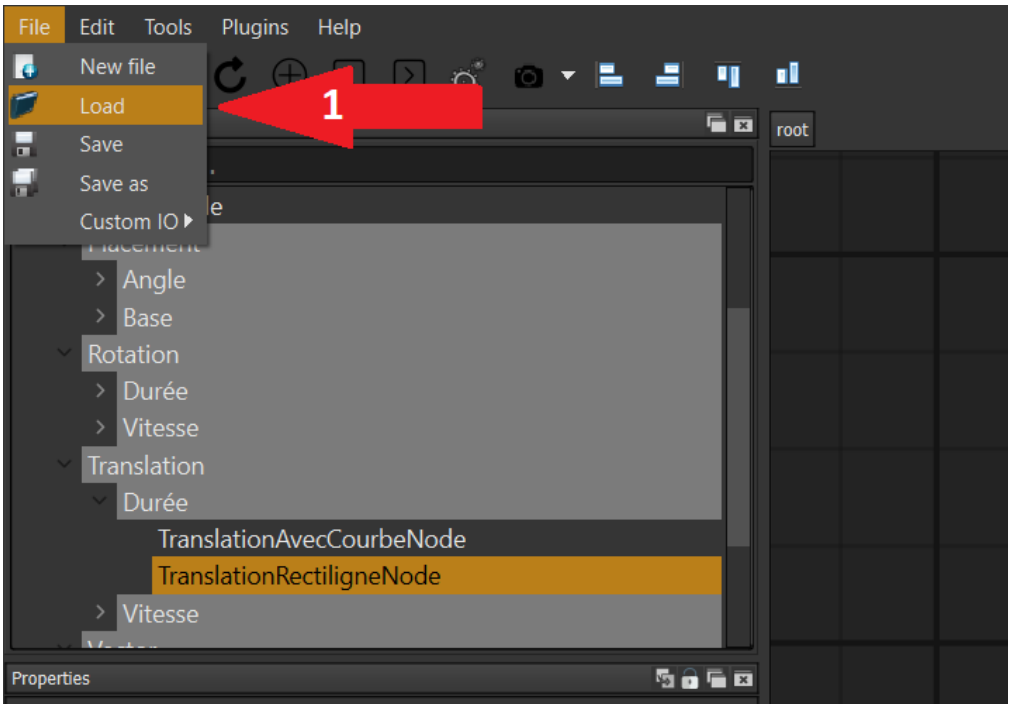
1. Select a node desired
2. Move the node in the scenario creation area



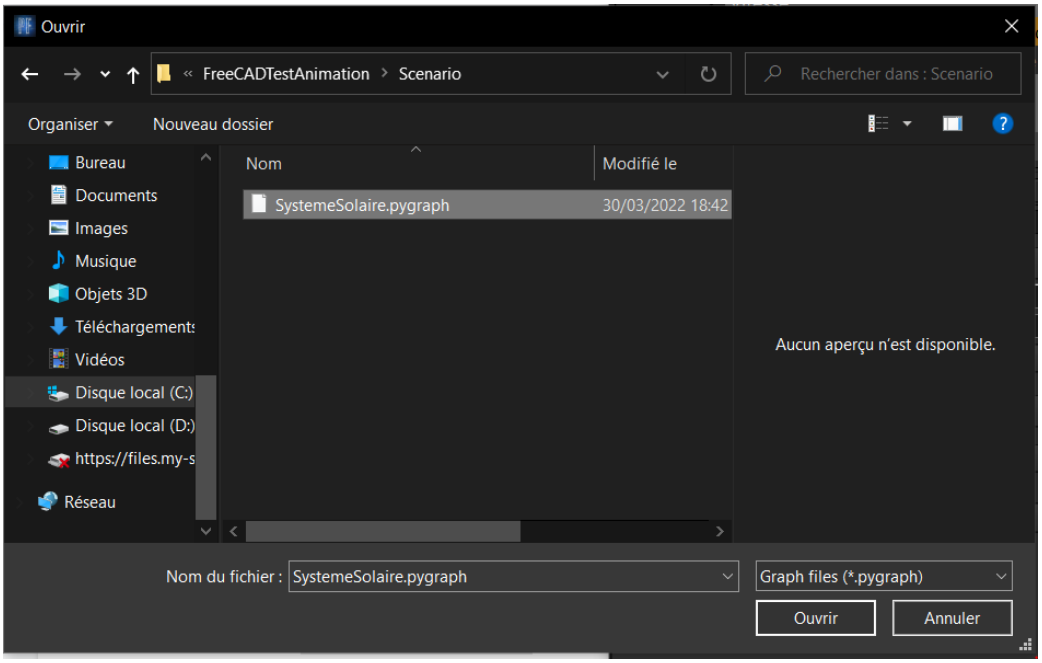
Open a scenario

To open an already existing scenario, simply go to the menu dedicated to the functionalities and click on File>Load.

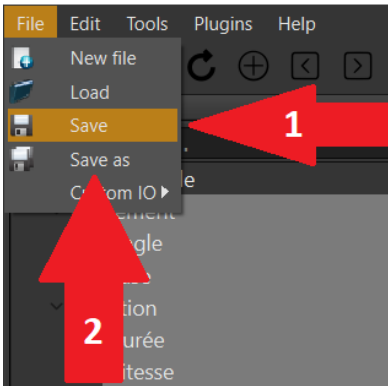
1. Click on the “File” tab
2. Then choose the “Load” option



3. Finally select your scenario



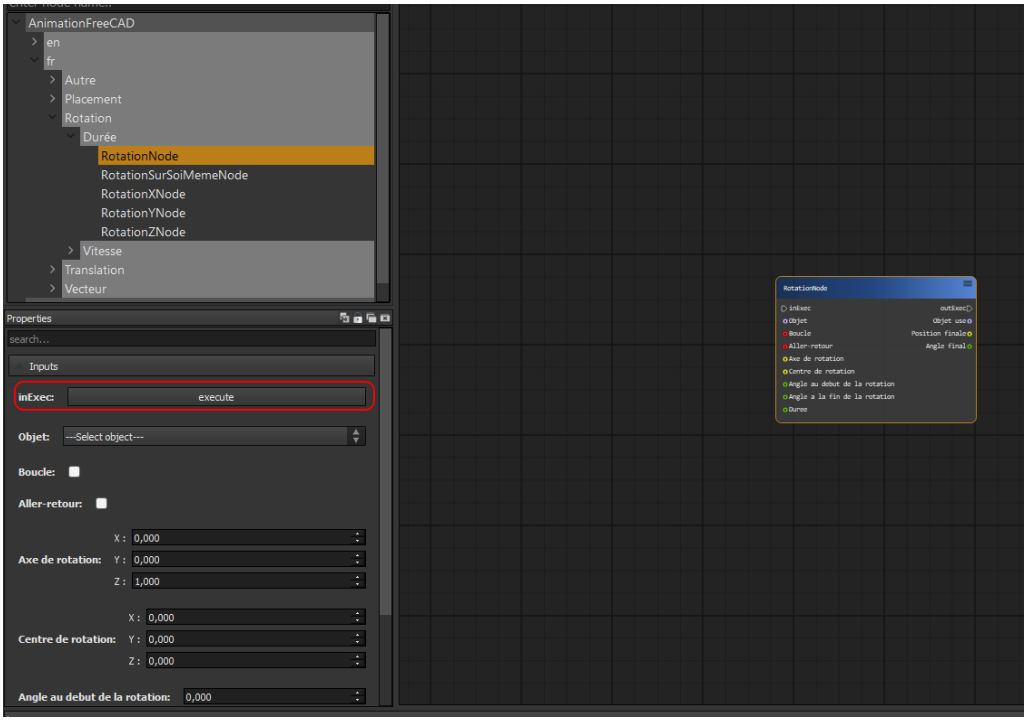
Save a scenario



To save a scenario, you must first have created it. Then you have two backup choices:

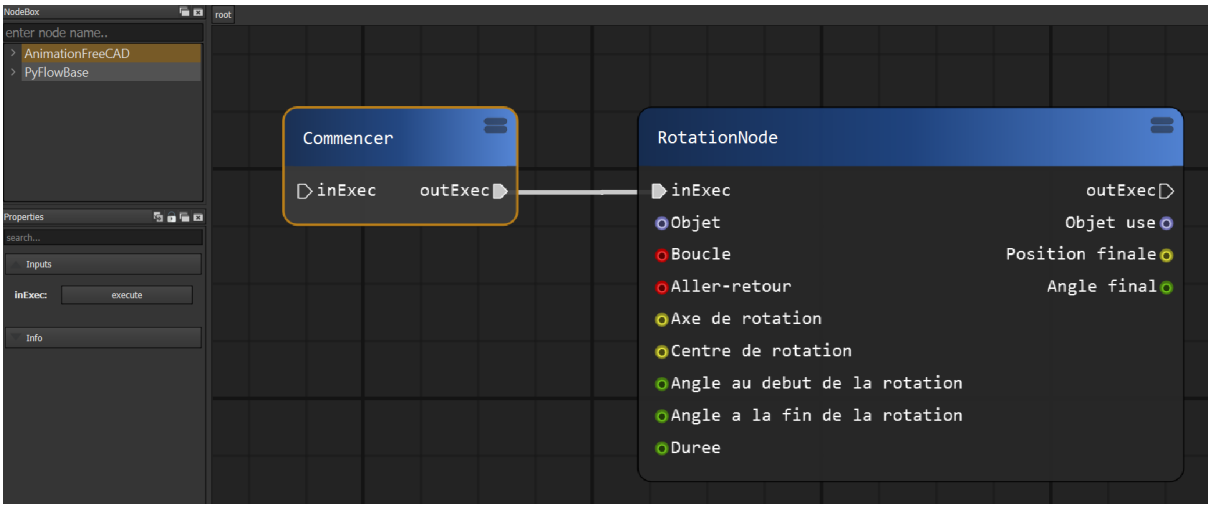
1. Save the current scenario file
2. Save the scenario in a new file

Run a node



There are several ways to run a node with our module:

- Directly run the Node by pressing its button "Executed".
- Connect the input of the node with another node, when the first node has finished its execution it will call the execution of the second.



The buttons

Pausing the animation

During an animation in progress, you can pause it at any time using this button.

Resume animation

You can resume an animation after pausing it via this button.

Stop the animation

After launching an animation in progress, you can stop the animation at any time using this button.

Reset the position of the animation

After a finished animation you have the possibility of being able to reset the position of all the objects at the time of the opening of our 3D animation workshop via this button.

Adding a step

During the animation you have the possibility of being able to record a particular moment via this button. This step remembers the placement of the objects. This feature will allow you to resume at a specific time that you saved.

Go to the previous/next step

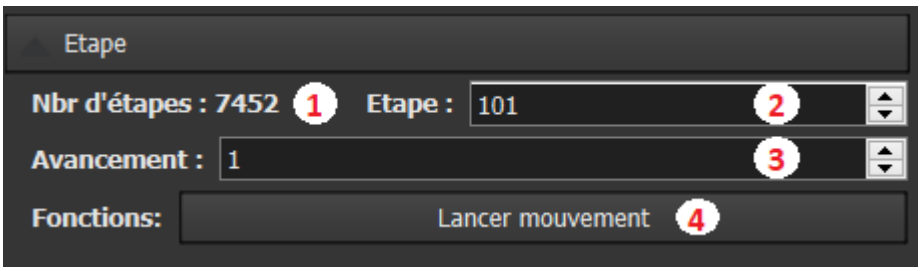
After having recorded the steps during the animation, our workshop offers you the possibility of being able to browse all the recorded steps via two buttons (previous/next) available via these buttons.

[on the left] Previous step
[on the right] Next step

Export the animation to video

After having finished your scenario, you can at any time decide to export the animation to video format via this button.

Animation Debug Tool



This interface allows you to manipulate the animation directly from its steps. So you can go directly to a specific step in the animation and be able to see if a part of the animation does not perform as expected.

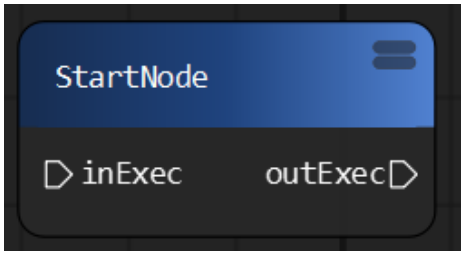
The interface has 4 parts:

- The first indicates the number of steps that the movement contains.
- The second represents the current position of the object, it can be modified in order to move the object. It automatically positions the object when the step is changed.
- The third which allows to increase the step of the small button of the drop-down list n°2
- And the last part is the button which makes it possible to launch a movement starting from the step located in the drop-down list °2.
-

Visual blocks (nodes)

Other

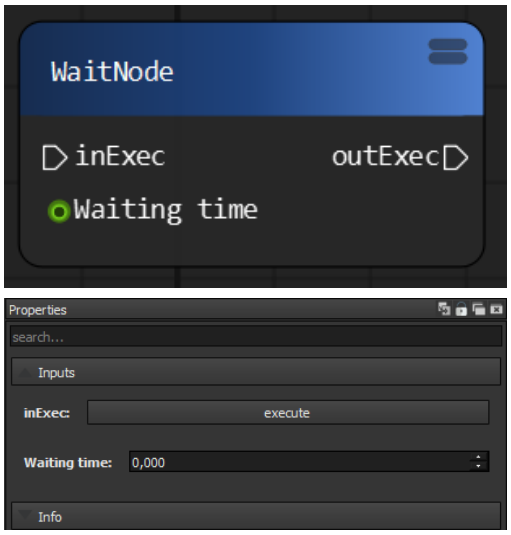
Start



Description
node “Start” sends a signal to all the nodes that are connected with it. This block is useful for launching a set of simultaneous movements from the start of the animation.
Parameters

Input (Input)	Output (Output)
- inExec: (optional) link with another block	- outExec : (optional) link with another block

Wait

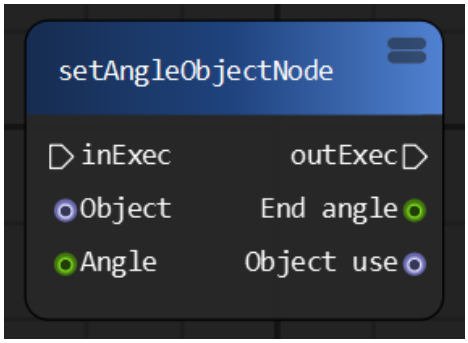


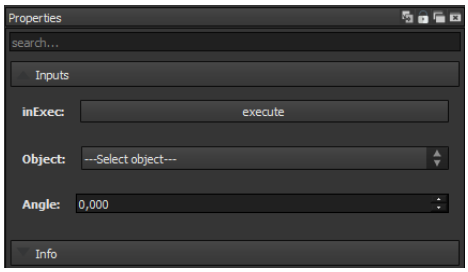
Description
The “Wait” node is used to trigger an execution after a waiting time.
Parameters

Input (Input)	Output (Output)
- inExec: (optional) link with another block - Waiting time: float	- outExec : (optional) link with another block

Placement

Set the angle of an object

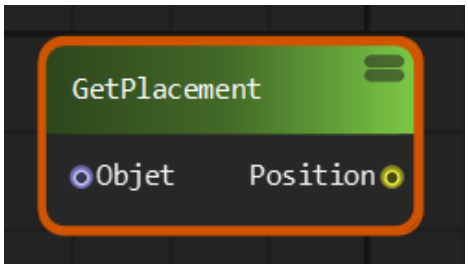




Description
The "SetAngleObject" allows you to define the angle of an object by entering an object and thus the desired angle. After execution the node returns the end angle as well as the object used.
Parameters

Input (Input)	Output (Output)
- inExec: (optional) link to another - Object block: FreeCAD.Object - Angle: float	- outExec : (optional) link to another - End Angle block: float - Object use: FreeCAD.Object

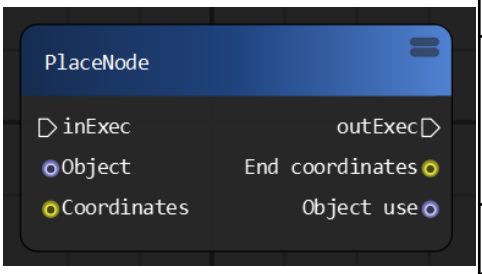
Get location of an object

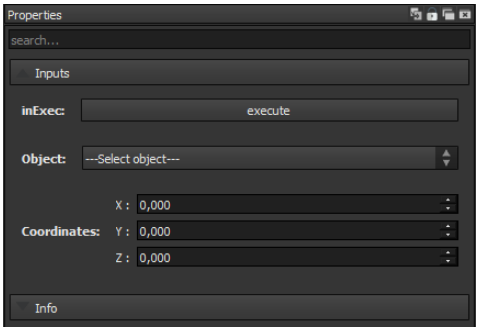


Description
The "GetPlacement" used to retrieve the position of an object passed as a parameter.
Parameters

Input (Input)	Output (Output)
- Object: FreeCAD.Object	- Position: float

Place an object

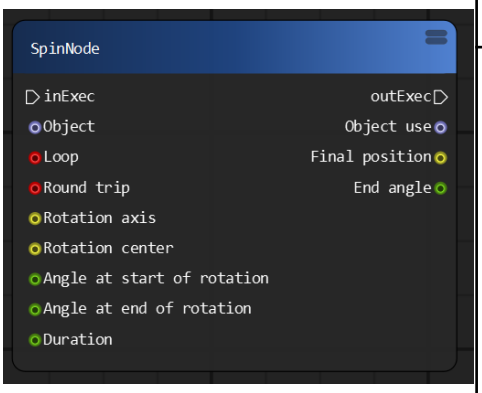


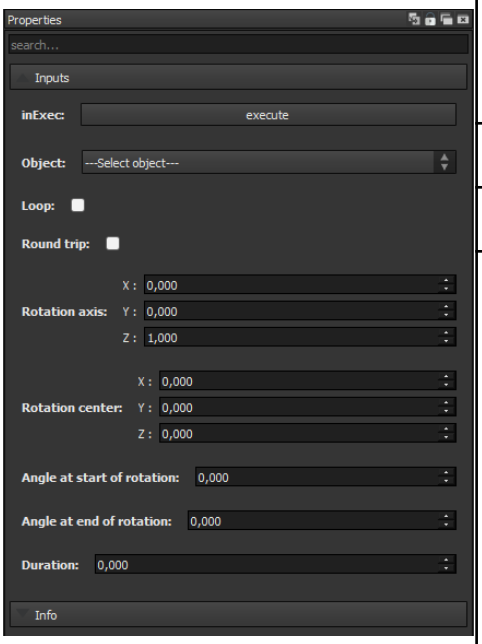


Description	
The "Place" node allows you to define the new coordinates of an object by entering an object as well as the coordinates (x,y,z) wanted. After execution the node returns the end coordinates and the used object.	
Parameters	
Input (Input)	**Output (Output)**
- inExec: (optional) link with another block - Object: FreeCAD.Object - Coordinates: float vector	- outExec : (optional) link with another block - Coordinates end: vector float - Object used: FreeCAD Object

Rotation

Rotation





Description

There are 2 variants of rotation a rotation where we will pass a speed as a parameter and a second where we will pass the duration of the rotation.

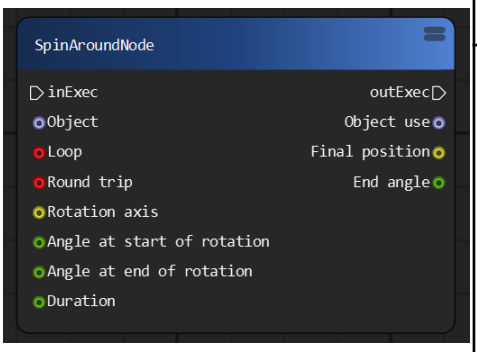
The “Rotation” node allows you to rotate an object by entering an object and several parameters. It is possible to activate additional execution modes (Loop/Round trip). You must enter the axis of rotation, its center of rotation, the start angle, the end angle and the duration or the speed of the rotation depending on the desired execution choice.

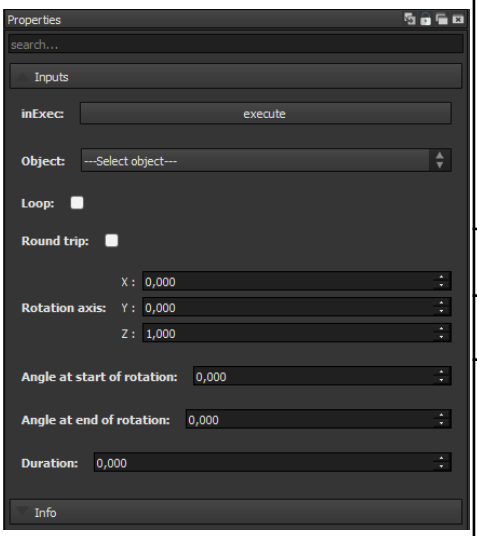
After execution the node returns the final position and the final angle.

Parameters

Input (Input)	Output (Output)
- inExec: (optional) link with another block - Object: FreeCAD.Object - Loop: bool - Round trip: bool - Axis of rotation: vector float - Center of rotation: vector float - Start angle: float - Angle of end: float - Duration/Speed: float	- outExec : (optional) link with another block - Final position: float - Final angle: float - Object use: FreeCAD Object

Rotation on itself





Description

There are 2 variants of the rotation on itself a rotation where we will pass in parameter a speed and a second where we will pass the duration of the rotation to it.

The node “RotationOnSelf” allows an object to be rotated on itself by entering an object and several parameters.

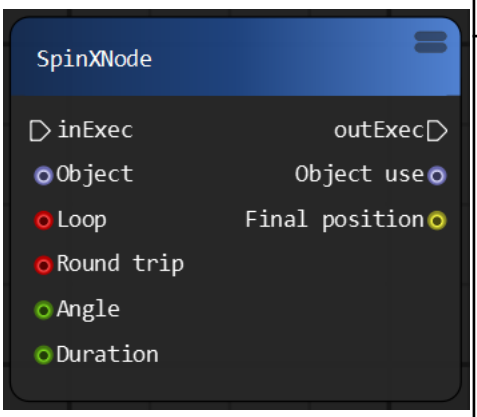
It is possible to activate additional execution modes (Loop/Round trip).
You must enter the axis of rotation, the start angle, the end angle and the duration or speed of the rotation depending on the desired execution choice.

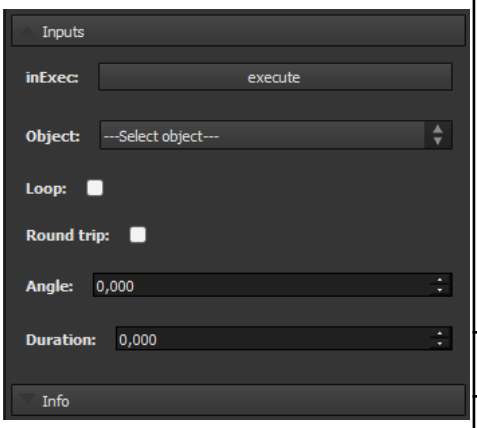
After execution the node returns the final position and the final angle.

Parameters

Input (Input)	Output (Output)
- inExec: (optional) link with another block - Object: FreeCAD.Object - Loop: bool - Round trip: bool - Rotation axis: vector float - Start angle: float - End angle: float - Duration/ Speed: float	- outExec : (optional) link with another block - Final position: float - Final angle: float - Object used: FreeCAD Object

Rotation on an axis (x or y or z)



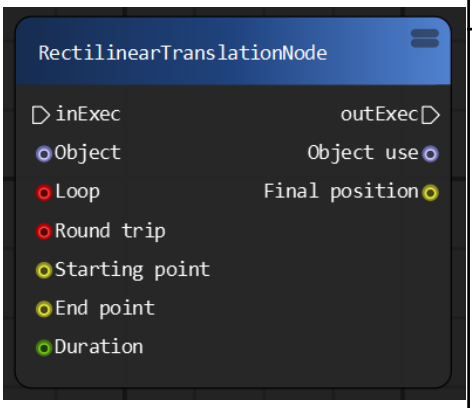


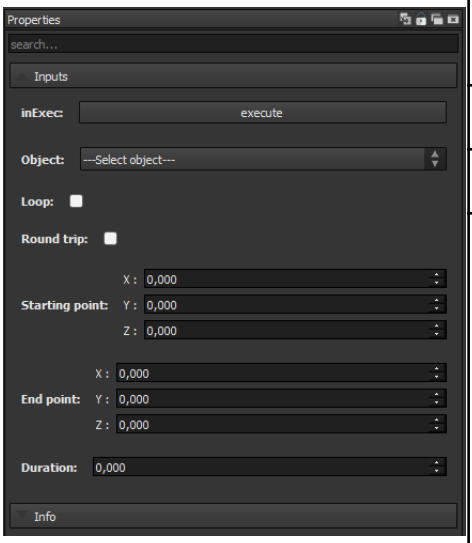
Description
The node “Rotation(X or Y or Z)” allows you to rotate an object on an axis that you want by entering an object and several parameters. It is possible to activate additional execution modes (Loop/Round trip). You must enter the angle and the duration or the speed of the rotation depending on the desired execution choice. After execution the node returns the object and the final position of the object. It should be noted that this node is more useful than the previous ones because it modifies the angle of an object relative to its current position, unlike the two previous ones which are based on the object's creation plane. The only weak point of this mode is that it is not yet operational for very precise axes of rotations, only the axes of rotation of space are defined there (x, y and z).

Parameters	
Input (Input)	**Output (Output)**
- inExec: (optional) link with another block - Object: FreeCAD.Object - Loop: bool - Roundtrip: bool - Angle: float - Duration/Speed: float	- outExec : (optional) link with another block - Final angle: float - Object used: FreeCAD Object

Translation

Rectilinear translation





Description

There are 2 variants of rectilinear translation: a translation where we pass a speed as a parameter and a second where we pass the duration.

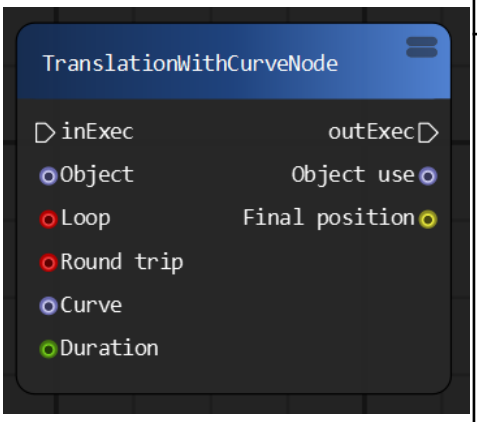
The node “TranslationRectilinear” is used to translate an object by entering an object and several parameters.

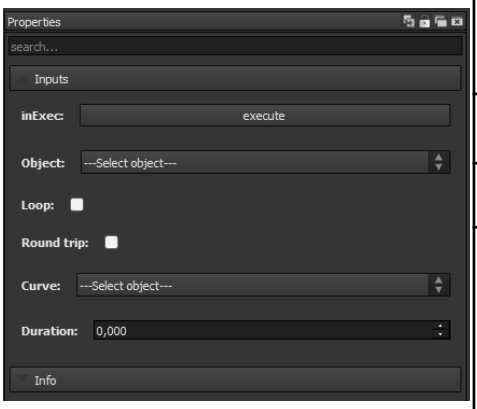
It is possible to activate additional execution modes (Loop/Round trip).
You must enter the starting point, the end point and the duration or speed of the transaction depending on the desired execution choice.
After execution the node returns the final position.

Parameters

Input (Input)	Output (Output)
- inExec: (optional) link with another block - Object: FreeCAD.Object - Loop: bool - Round trip: bool - Starting point: vector float - Ending point: vector float - Duration/Speed: float	- outExec : (optional) link with another block - Object used: FreeCAD Object - Final position: vector float

Translation with curve

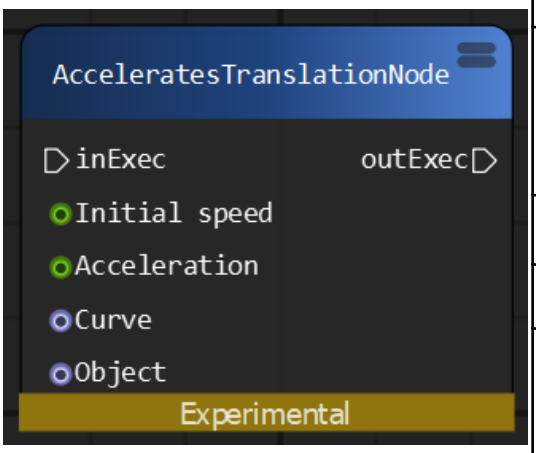


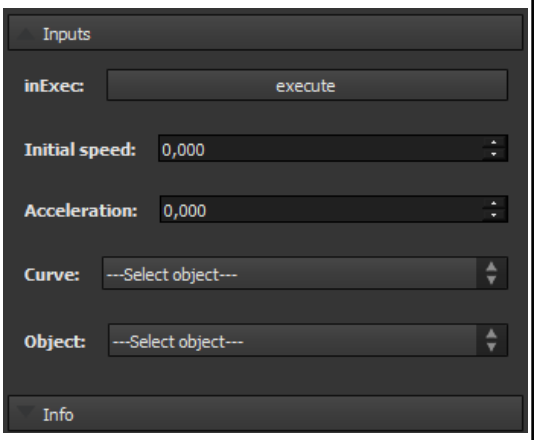


Description
There are 2 variants of the translation following a curve: a translation where we pass a speed as a parameter and a second where we let him pass the time. The node “TranslationAvecCourbe” allows an object to undergo a rotation on itself by entering an object and several parameters. It is possible to activate additional execution modes (Loop/Round trip). You must enter the tracking curve and the duration or the speed of the translation depending on the desired execution choice. After execution the node returns the final position.

Parameters	
Input (Input)	**Output (Output)**
- inExec: (optional) link with another block - Object: FreeCAD.Object - Loop: bool - Roundtrip: bool - Curve: object - Duration/Speed: float	- outExec : (optional) link with another block - Object used: FreeCAD Object - Final position: float vector

Accelerated Translation

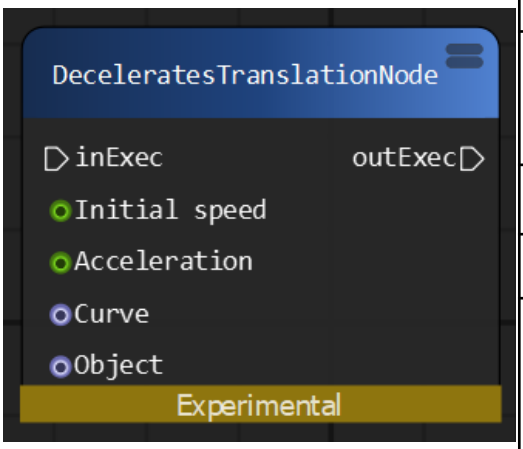


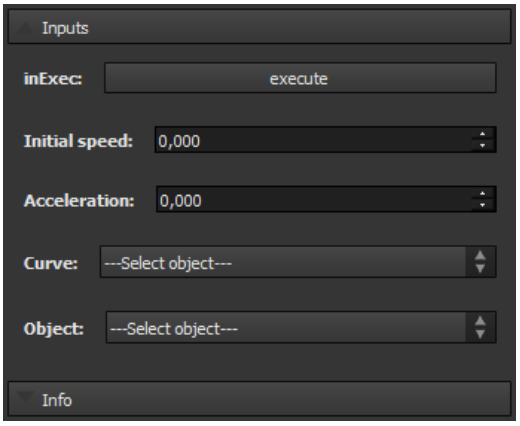


Description
The "TranslationAccelerate" node allows an object to move along a curve by accelerating its speed, depending on a base speed and an acceleration coefficient .
Parameters

Input (Input)	Output (Output)
- inExec: (optional) link with another - InitialSpeed block: float - Acceleration: float - Curve: FreeCAD.Object - Object: FreeCAD.Object	- outExec : (optional) link with another

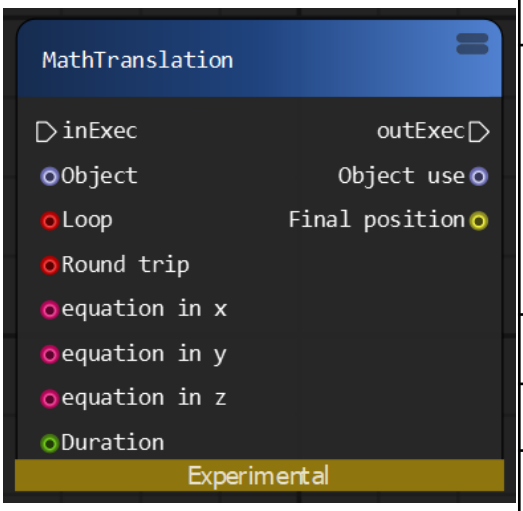
Decelerated Translation

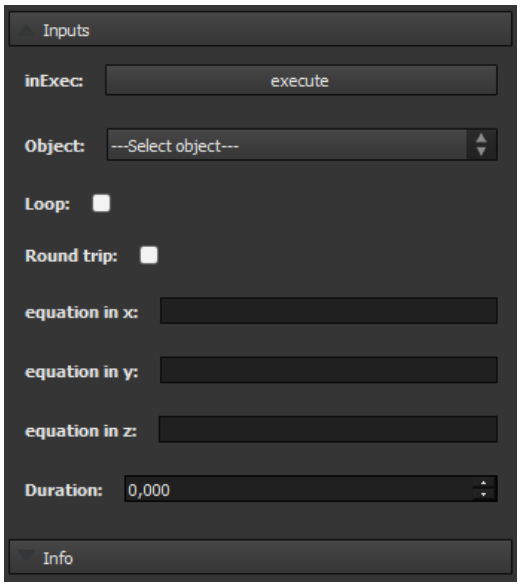




Description	
The node "TranslationDecelere" makes it possible to apply an acceleration to a movement, according to a base speed and a deceleration coefficient.	
Parameters	
Input (Input)	**Output (Output)**
- inExec: (optional) link with another block - InitialVelocity: float - Acceleration: float - Curve: FreeCAD.Object - Object: FreeCAD.Object	- outExec : (optional) link with another block

Mathematical Function





Description

The node “Function” is used to define a movement of an object according to 3 mathematical expressions: one on each axis.

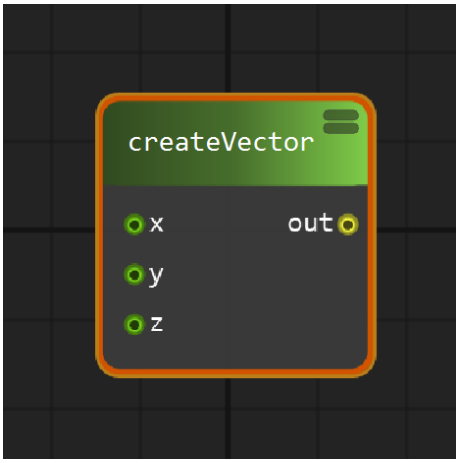
In the formulas we will be able to use “t” representing the time since the beginning of the movement.

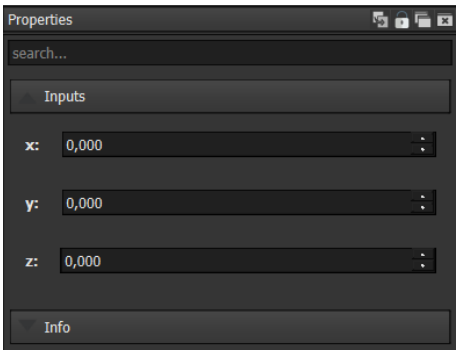
Parameters

Input (Input)	Output (Output)
- inExec: (optional) link with another block - Object: FreeCAD.Object - Loop: bool - Round trip: bool - Equation in x: string - Equation in y: string - Equation in z: string - Duration: float	- outExec : (optional) link with another block - Object: object - Final position: float

Vector

Create a vector

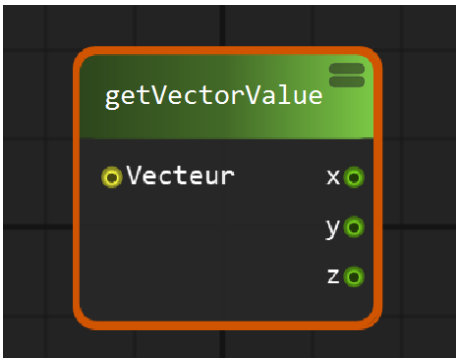


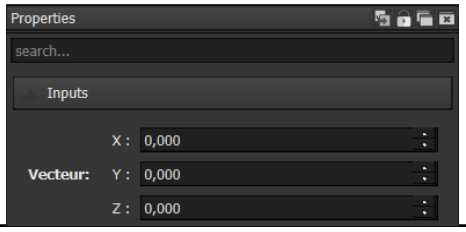


Description
The "CreateVector" node allows you to create a vector by entering coordinates (x,y,z). After execution, the node returns a created vector containing the filled values.

Parameters	
Input (Input)	**Output (Output)**
- component x: float - component y: float - component z: float	- out : vector of float

Retrieving the values of a vector

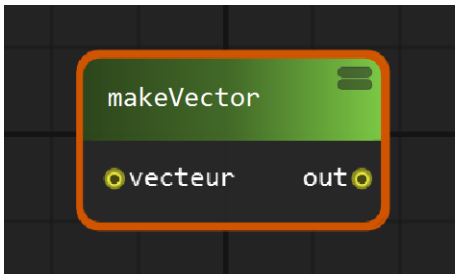


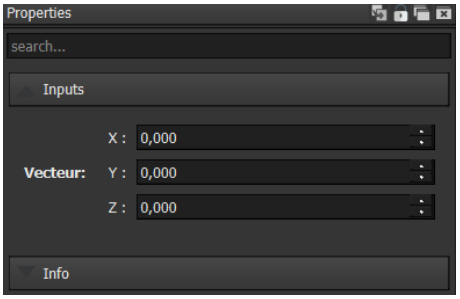


Description
The "GetVectorValue" used to retrieve the values of a vector by passing in parameter a vector. After execution the node returns the coordinate values of the vector.

Parameters	
Input (Input)	**Output (Output)**
- vector: float vector	- x component: float - y component: float - z component: float

Create a vector from a vector





Description	
The "MakeVector" used to create a vector from an existing vector pass as a parameter. After execution the node returns the newly created vector.	
Parameters	
Input (Input)	**Output (Output)**
- vector: float	- vector out: float vector