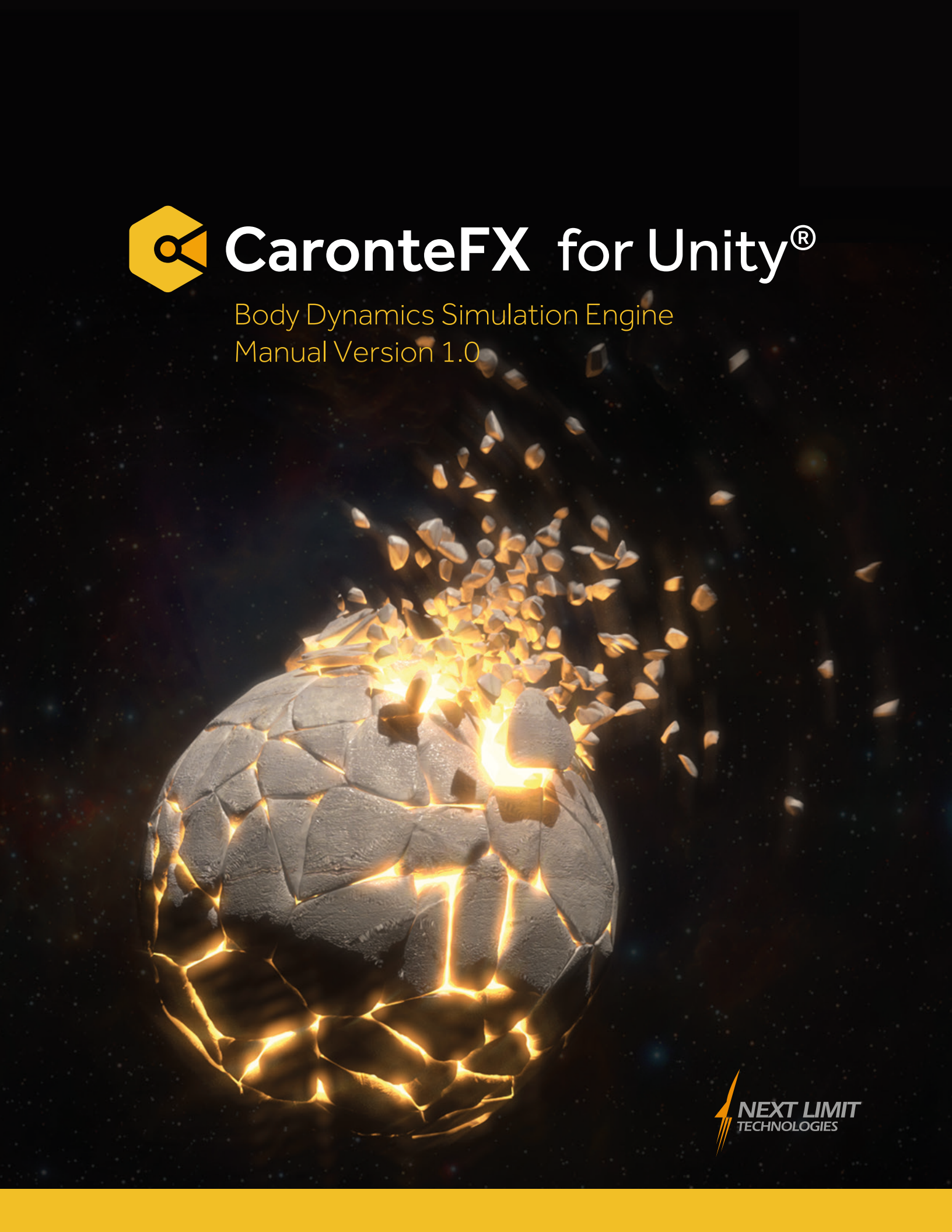
CaronteFX for Unity®

Body Dynamics Simulation Engine
Manual Version 1.0



NEXT LIMIT TECHNOLOGIES

Installing CaronteFX

System Requirements, Installation

What is CaronteFX?

CaronteFX for Unity® is an easy-to-use rigid and soft body dynamics engine. The package's tools are seamlessly integrated into Unity's® user interface.

System Requirements & Installation

System requirements depend on the type and complexity of your simulations. As a minimum system we recommend an Intel i5 (or comparable) processor and 4 GB RAM. For complex simulations, we recommend an Intel i7 (or better) processor with at least 8 GB RAM. Currently, CaronteFX for Unity® is only available for Windows operation systems.

The CaronteFX package can be stored at any location and directory. Each time you start a new Unity® project, the package has to be imported from

Assests > Import New Asset... > Custom package

Where to Find CaronteFX?

Once the CaronteFX package has been loaded it will appear as an asset in the "Project" window. To launch the CaronteFX user interface go to

Window > CaronteFX Editor

- A new window appears ("CaronteFX Ed")
- The editor can be positioned anywhere, e.g. below the "Inspector".

The first action is the creation of a Unity® GameObject:

- Click on "Create FX GameObject"
- The editor's layout will change and you can see a "CaronteFx_0" GameObject.
- All scene elements (rigid and soft bodies, joints, daemons, etc.) will be added here.

Updates

For information about new versions and updates on CaronteFX for Unity® and the manual please visit

http://www.carontefx.com

CaronteFX Basics

Workflows and Getting Started

The CaronteFX Scene Elements

Unity® uses GameObjects, and in fact all CaronteFX elements are GameObjects as well. This name convention can be misleading, because it is not always clear which kind of GameObject is actually meant: a native Unity® object or a CaronteFX node? Therefore it is important to separate things:

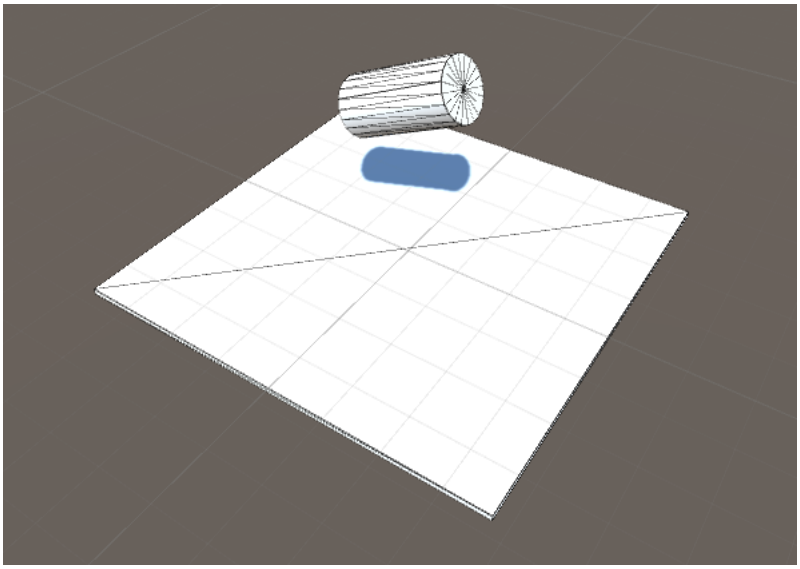
- "**Unity® GameObject**" describes all native objects from the "GameObjects" menu.
- "**CaronteFx GameObject**" describes a container where CaronteFX-specific nodes can be added to. This type is identified through its "CaronteFx_" prefix, followed by a number, e.g. "CaronteFx_0".
- "**CaronteFX SimulationObject**" stands for nodes from the CaronteFX window's tabs like "Bodies", "Joints", or "Daemons". These elements are always grouped under a "CaronteFx GameObject".

Your First CaronteFX Scene

CaronteFX does not create any basic primitives like spheres, cubes, or cylinders. It is also not a modelling or a UVW layout tool. But CaronteFX can use existing Unity® GameObjects and turn them into rigid and soft bodies, or disassemble them into fragments. It is also possible to stick fragments together to create convincing breaking effects, add explosions, and drive objects with motors.

Let's start with a very basic scene to see how CaronteFX works inside Unity®:

- Add a "Cube" object from **GameObject > 3D**.
- Scale the cube to create a ground plane.
- Add a "Cylinder" object, move it upwards, and rotate it.

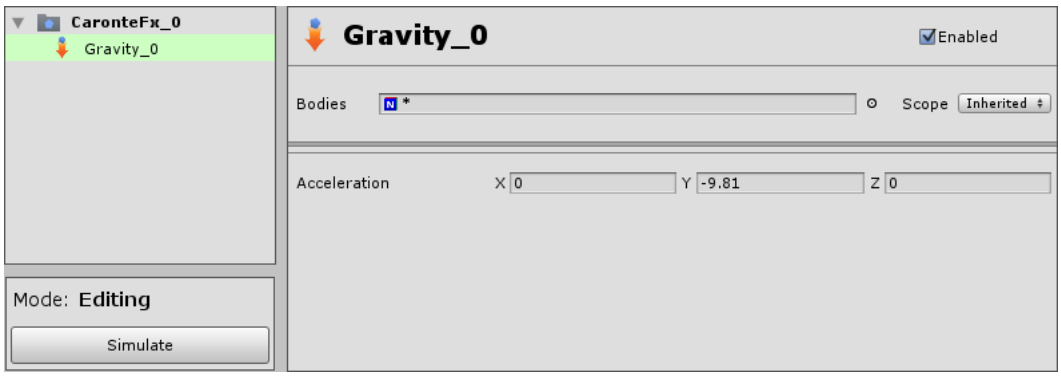


The setup of your first scene.

Now, go to the “CaronteFX Ed” window and press “Create FX GameObject”:

The newly created object contains a “Gravity_0” daemon – it has been added automatically. Gravity is a force pointing to the centre of a huge imaginary planet. This force will accelerate the cylinder and make it fall. The ground plane will act as an obstacle.

Click on “Gravity_0” and the big logo will be substituted through a set of parameters. Under “Acceleration” you see a default value:



- This value represents gravitational acceleration on Earth: 9.81 m/s^2.
- The negative sign indicates that all affected bodies will fall. Here along the Y axis.

Turning Unity® GameObjects into CaronteFx SimulationObjects

In the next step we have to make decisions. The cylinder should move, while the ground plane should remain motionless, but it has to interact with cylinder.

The Cylinder

Click on the “Bodies” tab and then on “Rigid”:

Under “CaronteFx_0” you see a new entry: “RigidBodies_0”. When selected, a new parameter set appears, but these settings are currently unimportant.

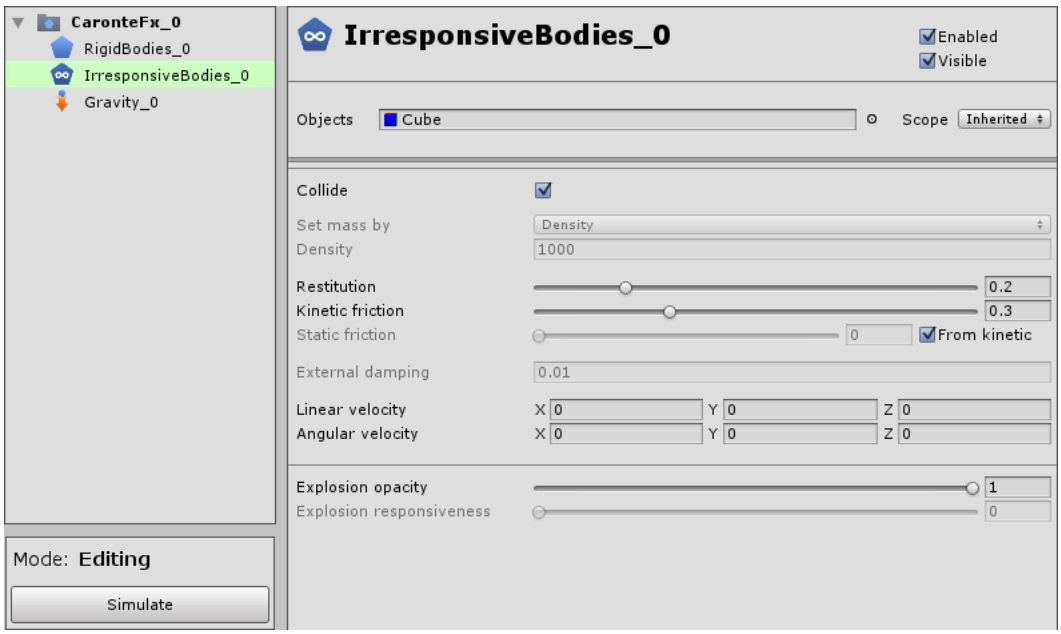
For now, the only interesting entry is the “Objects” slot :

- Drag the “Cylinder” object from the scene’s “Hierarchy” to the “Objects” field.
- Now this object has been turned into a rigid body and the parameters define its physical properties.

The Ground Plane (Cube)

In the next step, the ground plane is defined. Do you remember that this object should not move?

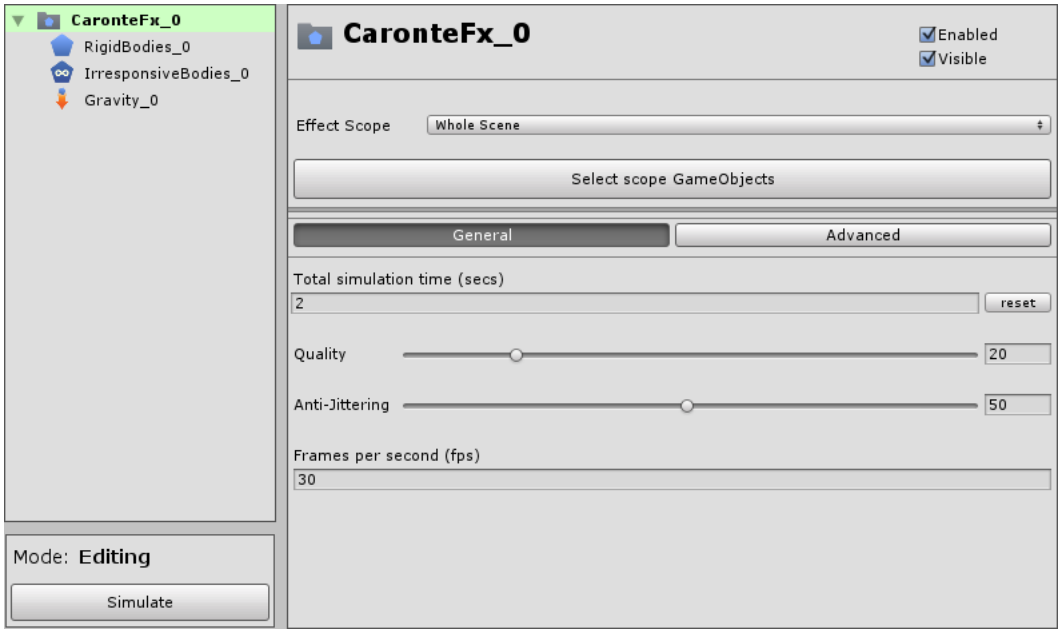
- Click on “Irresponsive”, select “IrresponsiveBodies_0”, and drag the “Cube” node to “Objects”.



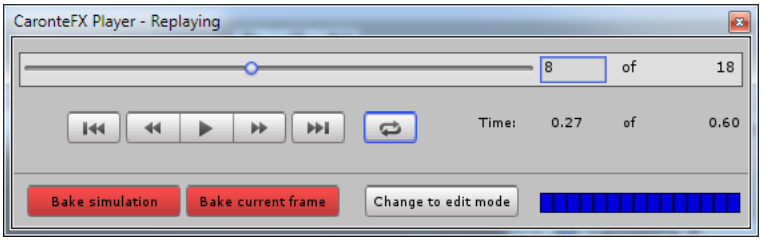
The Simulation

Now the scene has all ingredients for a CaronteFX dynamics simulation. For the final settings, click on “CaronteFx_0”. Another parameter set appears:

- The default “Total simulation time” is 10 seconds, but this is very long for the current scene. Change it to 2.0 seconds.
- With “Quality” it is possible to control the simulation’s accuracy. In a simple scene like that, a value of 20 is sufficient. This change will also speed up the simulation.



Finally, hit “Simulate”. A controller appears – the “CaronteFx Player” –, and you can also see the cylinder falling. After a short time the cylinder collides with the cube and comes to rest.

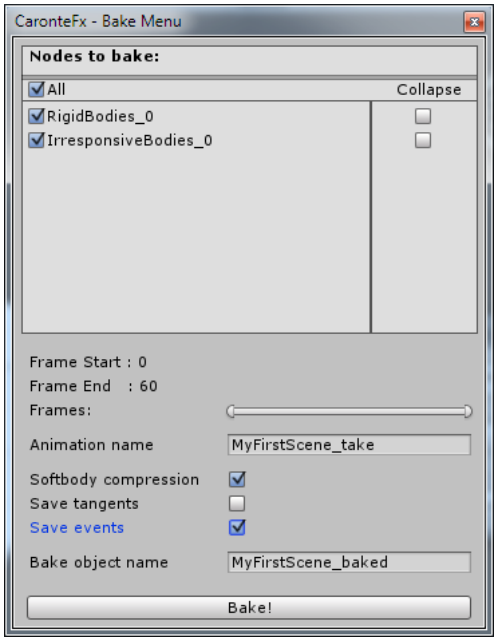


Baking and Creating Assets

In its current state, the simulation is only stored temporarily and will be lost when you close the player. If you want to keep the result permanently you have to bake it:

- Click on “Bake Simulation” (see outer left button in the image above).

- Enter new names and press “Bake!”.
- Specify the assets file location.



Reusing the Asset

Once the simulation has been baked and saved, it will be loaded to the current scene. You will also see that it contains exactly the same GameObjects as the original scene – and all properties are still editable. This means that you are able to create copies and different versions of the scene, bake them, and save everything as assets again.

Soft Bodies & Cloth

The presented workflow is exactly the same for soft bodies and cloth.

More Workflows

CaronteFX provides several possibilities for enhancing your workflow:

A very comfortable way to define Unity® GameObjects as rigid or soft bodies, etc. is to drag your nodes to the CaronteFx GameObject. When you drop the nodes a menu appears, asking you which type of SimulationObject you want to create.

Another method is to click on a SimulationObject's "Bodies" or "Objects" field. In both cases, a new window appears. Unity® GameObjects can be dragged from "Hierarchy" to this panel, where you can organize and arrange your nodes.

With the self-explaining entries of the window's "Edit" menu you are also able to work with Unity® GameObject selections. The "Add name Selector" entry is of special interest though: this feature is a convenient filter and accepts the * wildcard character.

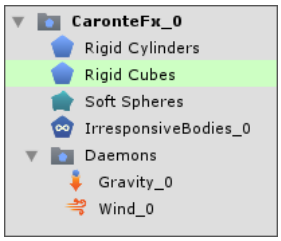
Unity® GameObjects can be renamed at any time and their names will be updated in the appropriate CaronteFx SimulationObjects automatically.

Building Groups and Clusters

So far, every CaronteFx SimulationObject ("Rigid" and "Irresponsive") contains a single node: a cylinder and a cube. But it is possible to control several nodes with just one SimulationObject.

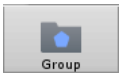
Create another scene or project with a new CaronteFx GameObject and an irresponsive ground plane. Then, add several primitives and spread them randomly. Press **Bodies > Rigid** (or **Soft/Cloth**) and drag the appropriate objects to the "Objects" slot. The scene is complete and you can click on "Simulate".

The main advantage with this method is that you have to make your parameter settings only once, but this might also lead to a uniform look. One method to avoid this is to create clusters of objects with equal properties, for example with equal mass, friction, or "Restitution" settings. To do this, multiple CaronteFx SimulationObjects are required.



For a better organisation, it is possible to rename the CaronteFx SimulationObjects. Right-click on an entry and choose “Rename”.

CaronteFx SimulationObjects can also be grouped with **General > Group**. Then, just drag and drop the objects to the new container.



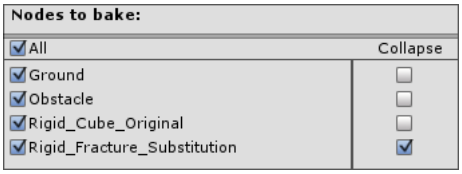
More About Baking and Collapsing

The “CaronteFX Player” (you see it when you hit simulate) provides a concenient possibility of baking a single frame and reuse it as an asset.

This comes in very handy, for example, if you want to spread debris or other objects over a plane:

- Shatter one or more fractured objects.
- Simulate to a frame you like and bake this specific frame as an asset.
- The assest’s bodies do not contain any motion information, like a 3D snapshot.

Another interesting option is “Collapse”. The appropriate checkbox appears in the “Bake simulation” window. Please note that the option is not available for “Bake current frame”.



When enabled the baked GameObjects are treated like one mesh, but the node-specific properties are kept. Without this function, each object is still treated as an individual mesh. The effect is that collpased objects require less resources in the runtime application – typically around 30%.

Changing Parameters

Simulation parameters influence each other and we therefore recommend changing one parameter at a time only to see how the objects react. You can, for example, start with the adjustment of mass or density, then “Restitution”, in the next step you define “Kinetic friction”, and finally “External damping”. This is an iterative process and with each simulation pass you refine the results until you get the desired output.

The question is which type of material you want to create:

- Materials like roughly cut wood or sandstone have high friction.
- Polished metal or glass surfaces are very even and have low friction.
- Rubber has a higher bounciness (“Restitution”) than plastic or metal.

An object’s shape also influences friction and air resistance. The latter is something you can feel when you hold out your hand through the open window of a fast driving car. In CaronteFX, this effect can be simulated with “External damping” and it will slow down a moving body effectively:

- Spherical or cylindrical objects, for example, can have low friction and “External damping” values due to their shape.
- Cubic objects have big contact areas and they are considered to have higher friction and “External damping”.
- Edgy objects can be simulated with higher “External damping” values, because such shapes create turbulence in real life.

Understanding Mass

Another fundamental property is mass and it is crucial to understand how mass works inside CaronteFX. There are two possibilities of determining mass under “Set mass by”:

- Mass (by single body)
- Density

First and foremost: mass is not weight by default. Only on Earth both terms describe the same in everyday language. The reason is that weight depends on where the object is located: on the Moon, Jupiter, or Mercure, for example. All these celestial bodies have different gravity values, and even on Earth gravity is not constant, but the standard value of 9.81 m/s^2 is a very good average. This is the default value of the “Gravity” daemon. In CaronteFX, weight can be calculated this way:

```
Weight = object’s mass * abs(gravity value)
```

Since there is no negative weight, the “Gravity” daemon’s default of -9.81 m/s^2 is converted to an absolute value without an algebraic sign.

Understanding Density

Then there is density, defined as

```
Density = mass / volume unit (e.g. kg per cubic metre)
```

Density is something you know from daily life. Different materials have different densities, e.g. wood and concrete. Density also becomes obvious when you mix fluids with different densities like water oil. A sphere made from steel will sink immediately when you throw it into water, while a beach ball floats on the surface.

As you can see above, density also depends on an object's volume. CaronteFX determines each scene element's volume automatically. But why is this so important? Imagine a situation where you have a group of objects with different shapes and dimensions. Now you apply a mass value of 50 kg with "Mass (by single body)". This value will be used for every object without considering its size. So, a sphere with a radius r = 1.0 m will have the same mass as a sphere with r = 10 cm.

This means that the spheres' densities have to be changed to get a realistic simulation. With the radius information from above it is easy to get the resulting values:

$Volume_{sphere} = 4/3 * Pi * r^3$
$Volume_{sphere1}$ (1.0 m) = 4.2 m^3
$Volume_{sphere2}$ (0.1 m) = 0.0042 m^3

$Density_{sphere1}$ = 50 kg / 4.2m^3 = 11.9 kg/m^3
$Density_{sphere2}$ = 50 kg / 0.0042 m^3 = 11,905 kg/m^3

You do not have to perform this calculation, because it is done by CaronteFX automatically. The example has just been added to show you what is happening.

Changing Mass Through Density

Working with a fixed mass value can lead to unwanted results. A typical situation is a fractured object. Here all pieces should have the same density, because they are normally made of the same material. But the fragments have different masses. To achieve this, use the "Density" option under "Set mass by". The internal calculation is now:

```
Mass = density * volume
```

For the two spheres you get

$Mass_{sphere1}$ = 1000 kg/m^3 * 4.2 m^3 = 4,200 kg
$Mass_{sphere2}$ = 1000 kg/m^3 * 0.0042 m^3 = 4.2 kg

Soft Body Parameters

The specific parameters of soft bodies appear a little more abstract, because we are not used to these properties from daily life. We do not have this natural feeling for deformations as we have it for mass, friction, or even density.

Start with a basic scene: a soft body capsule rests on an even irresponsive ground object, e.g. a cube. The capsule has a certain weight, determined by its mass and the “Gravity” daemon. If you have read the “Understanding Mass” chapter you know that a body’s weight is calculated as

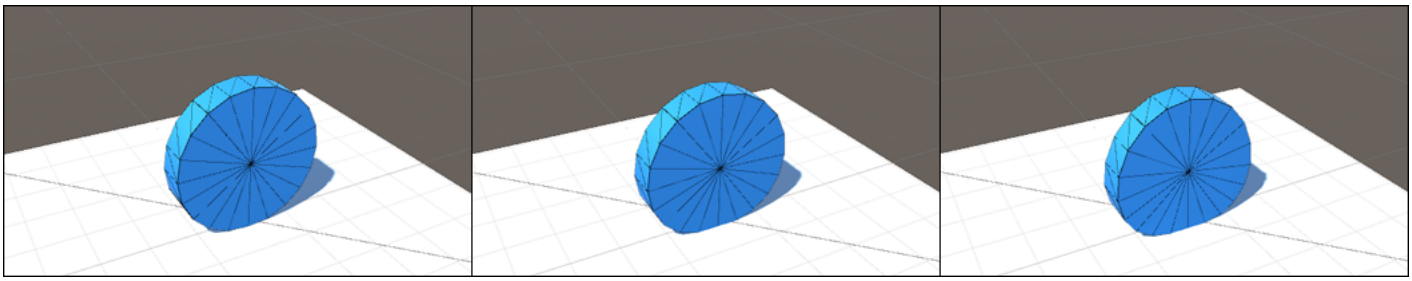
```
Weight = mass * abs(gravity value)
```

Stiffness

A body’s own weight causes a soft body to deform, because it will literally collapse under its own weight. This is the moment where stiffness properties come into play. The higher a body’s stiffness values, the more it behaves like a rigid body, and the less it suffers from deformation. In real world, a rigid body can be considered a soft body with extremely high stiffness.

A car’s tyre is a good example:
When there is enough air inside, the tyre keeps its shape, because air and the tyre’s materials add stiffness to the body. Now air pressure is reduced and there is not enough resistance against the car’s weight – the tyre becomes more or less flat.

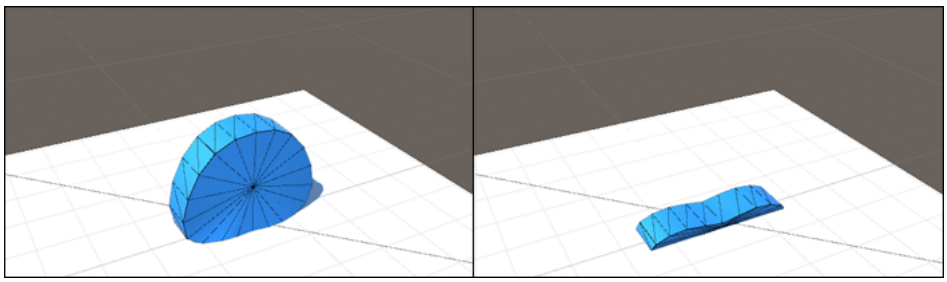
Start to decrease “Relative length stiffness” gradually, e.g. 1.0, 0.1, 0.01. When you simulate you will observe that the body will be more and more deformed with smaller values, but there is also a limit. The reason for this limit is “Relative volume stiffness”, because this parameter protects the body from volume changes. In this example the amount of deformation is moderate, because “Relative volume stiffness” still counteracts the body’s tendency to collapse. If you decrease volume stiffness as well the deformation will turn out even stronger.



Relative length stiffness: 1.0, 0.1 and 0.01 | Relative volume stiffness: 0.25

Damping

Now, shift the body vertically and let it drop onto the ground plane. You see that the body becomes even more flattened, due to the object's motion energy. With very low stiffness (length and volume) it might happen that the body becomes completely flat for a moment.



Stiffness is also the reason why the body tries to restore its shape and this causes a wobbling effect. To reduce this behaviour, increase "Internal damping", e.g. to 0.7.

Just play a little with different settings to get a feeling for how the parameters react and work together.

Creating a Fracture Scene with Joints

CaronteFX comes with three powerful, though easy-to-use, fracture tools. With these helpers it is possible to break any polygonal Unity® GameObject into pieces. This process is non-destructive and the original object will be kept.

Like in the scenes before, an irresponsive ground plane is needed, as well as a “Gravity” daemon, and at least one arbitrary Unity® GameObject. Then, add a “Uniform” object from the “Fractures & Tools” tab, and drag the object(s) you want to break into pieces to the “Objects” slot.

Breaking the Object

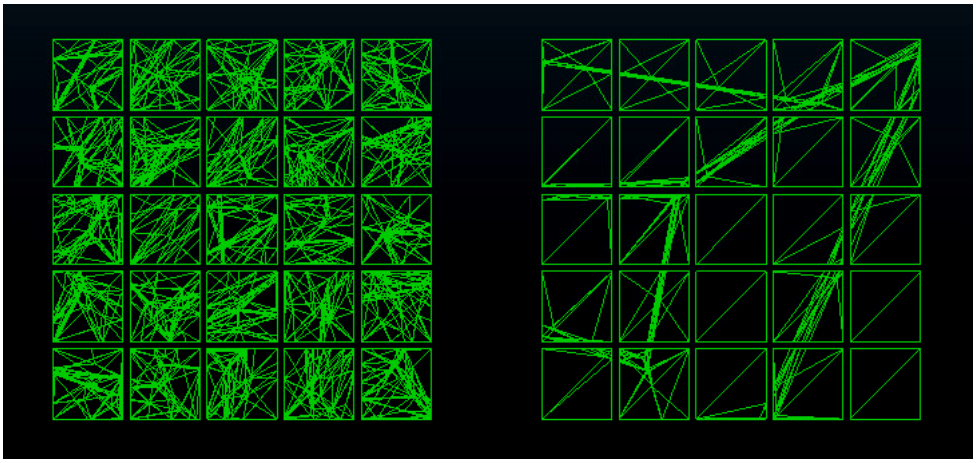
“Rough number of pieces” is certainly the most interesting parameter, because here you specify how many fragments you want to create. The actual number can be different from the input value in some cases, but most time you will get the exact number of pieces. The actual amount can be seen under “Stats”.

“Global pattern” is only relevant in conjunction with multiple objects. When active

- all nodes under “Objects” are treated as one coherent object
- you will see propagating cracks (see right image below)
- the total number of fragments corresponds with “Rough number of pieces” (see left image below).

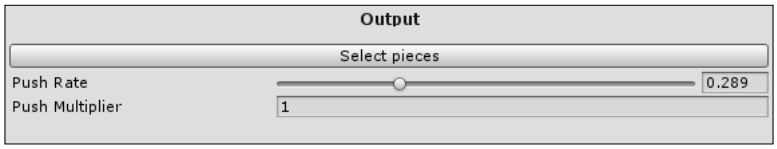
Otherwise every object is fractured individually, and the total amount of fragments is

```
Rough number of pieces * number of Unity® GameObjects
```

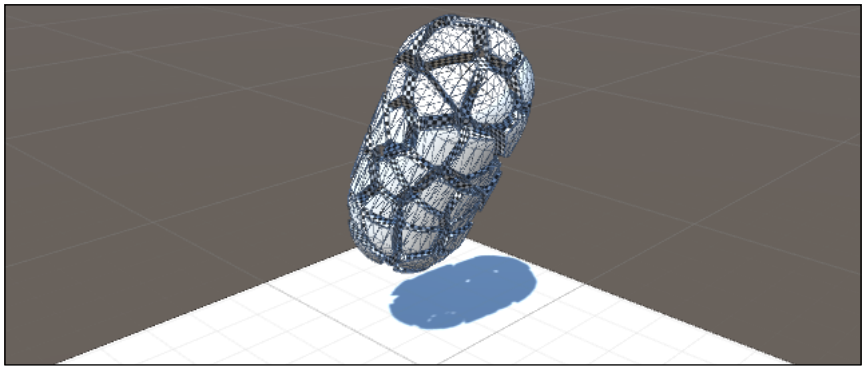


Once you have made your settings click on “Chop”. The individual pieces are grouped automatically and added to the “Hierarchy” window. If you do not like the result, just change the parameters and press “Chop” again. The fragments group will be updated automatically.

In order to see the pieces drag the “Push rate” slider.



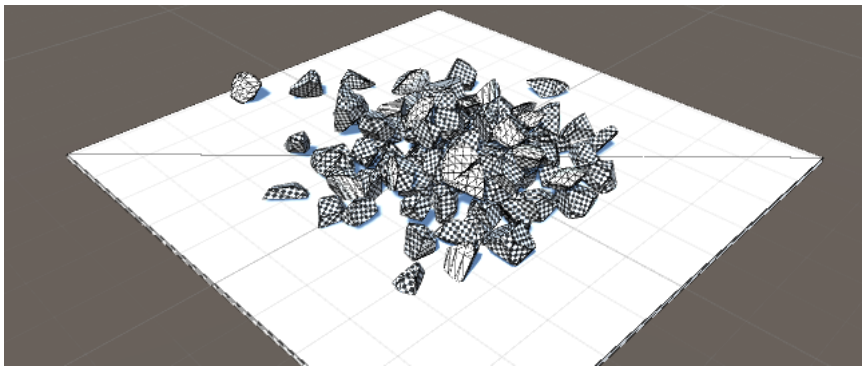
This feature gives you an exploded view of the entire structure.



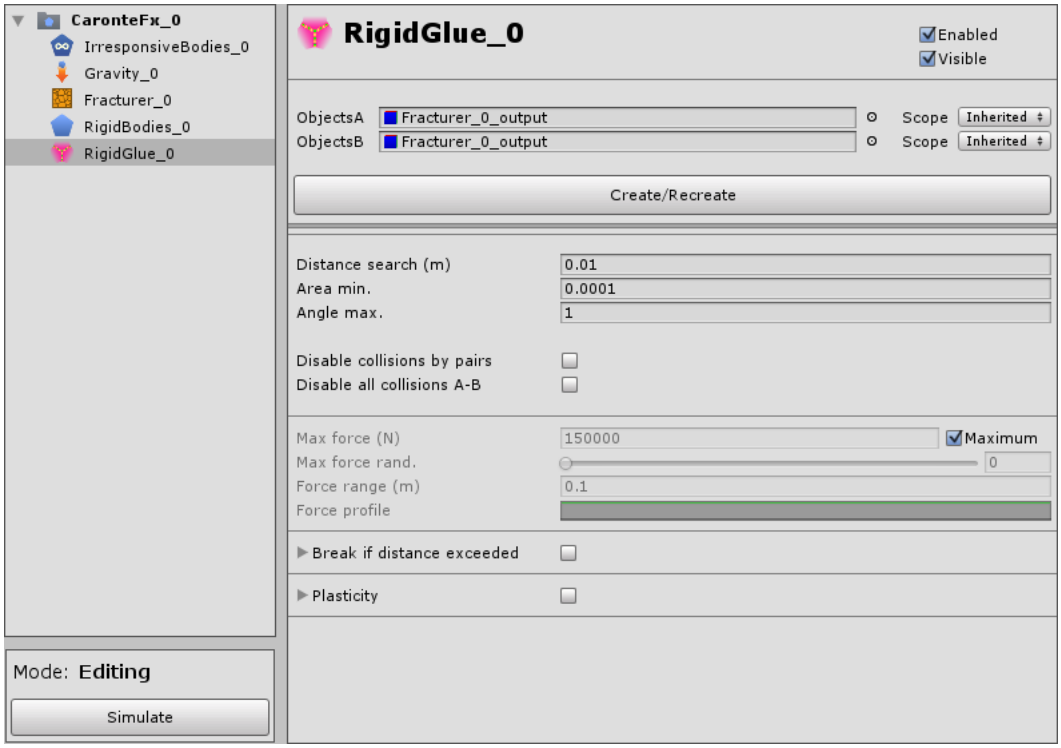
Connecting the Fragments

Now, the “Fracturer_0_output” group can be dragged from the “Hierarchy” window to a new “Rigid” SimulationObject. The rigid parameters can be adjusted according to the previous workflow descriptions. Mass is of special importance here and you should work with the “Density” option. This way each fragment will have a different mass according to its volume (→ see “Understanding Density”).

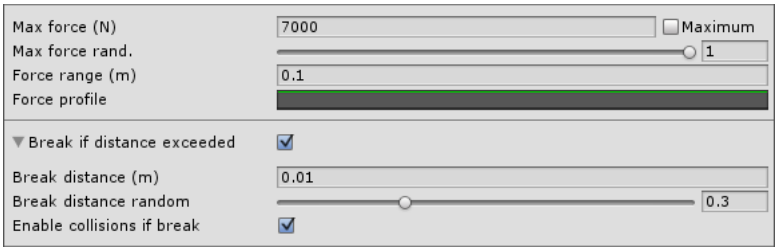
When you simulate you will observe that the fragments are separated as soon as the first piece hits the ground plane. This is not a very realistic result and the output looks more like a rubble mountain, not like a broken object.



Therefore a “Rigid Glue” SimulationObject is being added from the “Joints” tab. In this scene, all objects should be re-connected: drag the “Fracturer_0_output” group to “Objects A” and “Objects B”.

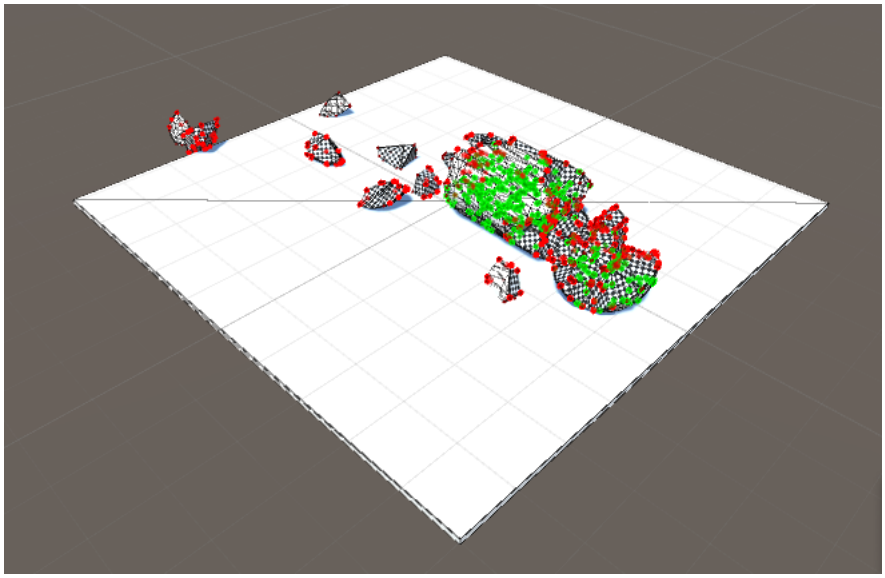


Start a simulation. What you most probably see is that the fragments stick together and do not break apart at impact. To change this, enable “Break if distance exceeded” and decrease “Max force (N)”, e.g. to 6000. It requires a few tests to find working forces, but you will get a feeling for this parameter soon.

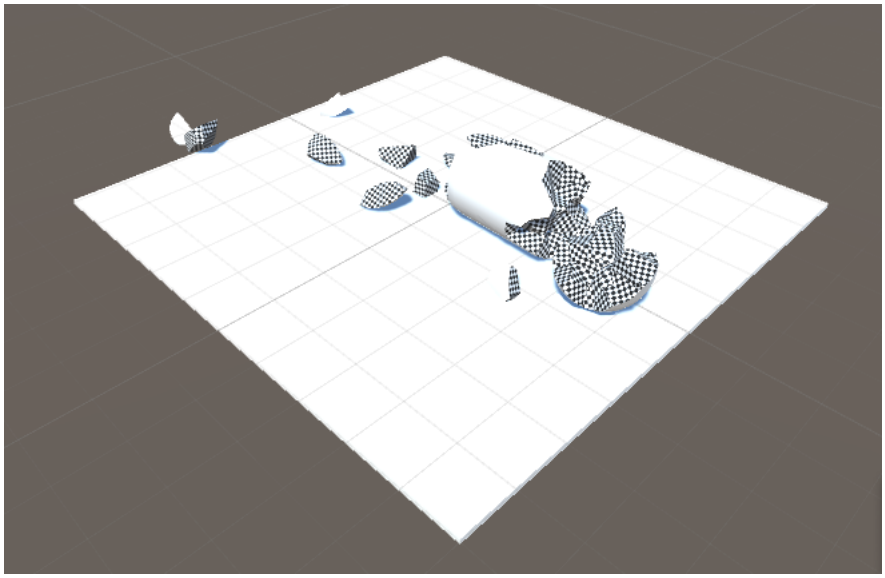


The effect of these settings is that the pieces start to separate when the given force limit is reached. Once there is a certain “Break distance (m)” between the fragments, the connection will break and the fragment is free to move. Intact connections are displayed as green boxes, broken links are red. Purple indicates intermediate states (images are available on the following page).

The result is a very realistic behaviour with clusters and a natural breaking process. With “Max force rand.” and “Break distance random” it is possible to further improve the object’s behaviour.



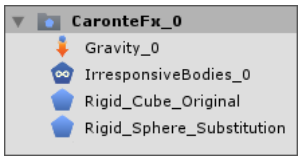
A simulated fractured capsule objects with visible joints. Red joints a broken, green links are still intact.



The same object with invisible joints. New faces are displayed with a checkerboard texture.

Substituter & Trigger Basics

In this first scene you will learn how to replace one GameObject through another. In this case, a rigid cube is substituted by a rigid sphere at a certain point in time. The setup is straight-forward and all you need is an irresponsive ground element, a gravity daemon, a rigid cube, and a rigid sphere.

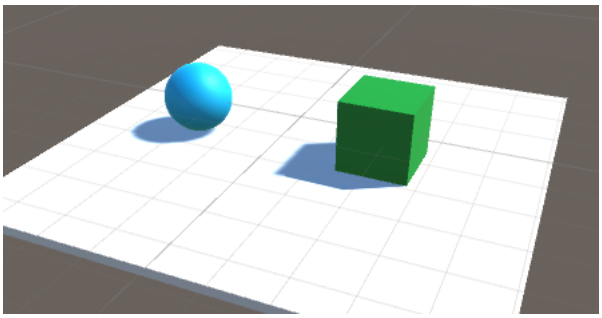


The positions of the cube and the sphere should be equal, and you will soon understand why. In this example, both bodies have the following coordinates:

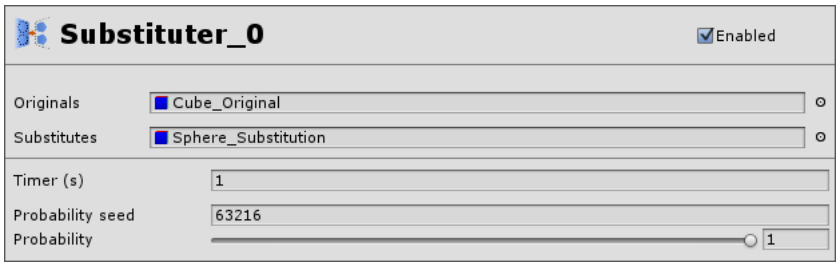
- Position.X = 0.0, Y = 2.5, Z = 2.0
- Scale is 1.5 in all directions

Open the “Rigid_Cube_Original” SimulationObject’s parameters and set “Initial linear velocity.Z” to -3.0. This will give the cube some extra push. The “Restitution” value of both rigid bodies is 0.7.

When you simulate you will see that both rigid bodies separate and go their own way. The sphere falls down immediately, the cube remains in the air a little longer due to its speed. Nice, but foreseeable.

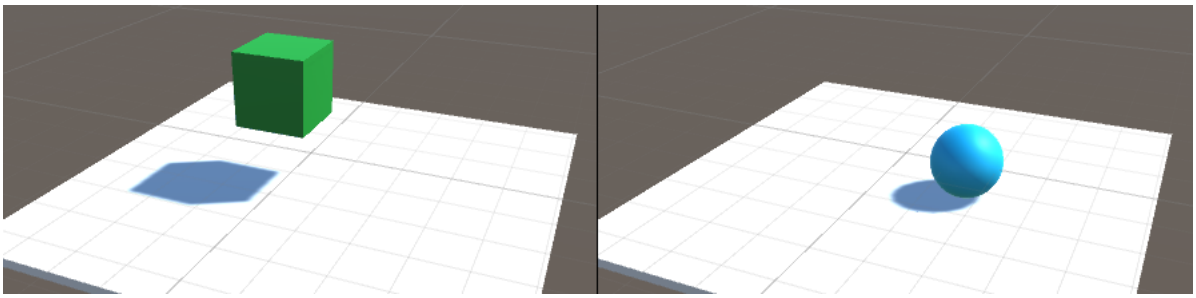


Now, add a “Substituter” from the “Actions” tab, and fill the parameters as shown below:



Simulation length should be limited to 3.5 seconds under CaronteFx_0 > Total simulation time (sec).

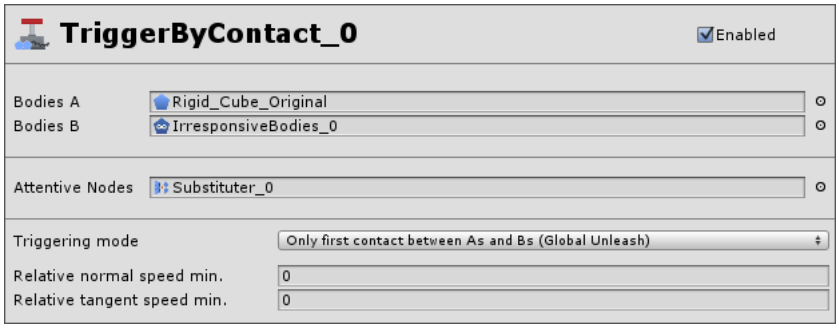
Hit simulate and see what happens:



At t = 1.5 seconds the cube becomes invisible and is replaced by the sphere. Before this event, the sphere has been parented to the cube. If both bodies have different positions you will see an offset.

Triggering the Substitution

It would be much more elegant if the substitution takes place when the cube hits the ground object, but this is difficult to adjust manually. Therefore, it is better to leave this job to a “Trigger By Contact” action. Here are the settings:

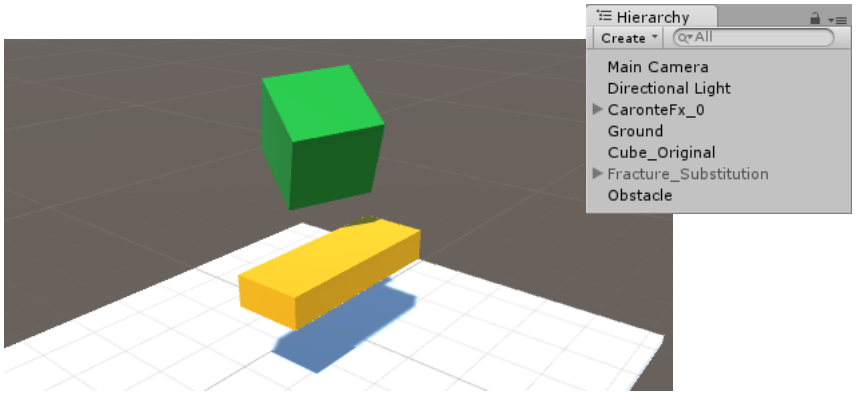


The “Substituter” node’s “Time (s)” parameter should be set to 0. Start another simulation and this time, the replacement happens exactly when the sphere gets in contact with the ground plane. Instead of “Trigger By Contact” it is also possible to use any other trigger node, of course.

Working with Fractured Objects

As you see, the basic workflow is easy to understand, but the current scene was nothing more than a proof of concept. One of the main applications with the “Substituter” is the control of fractured objects. The next scene is based on the current setup and there are just a few modifications:

- Remove the sphere (“Sphere_Substitution”) and its associated “Rigid” SimulationObject.
- Add a “Uniform” node from the “Fracture & Tools” tab.
- **Rigid_Cube_Original > Initial linear velocity.Z > 0.0**
- Break “Cube_Original” into some 25 pieces.
- Add another “Cube” Unity® GameObject and scale it. This cube will serve as an obstacle.

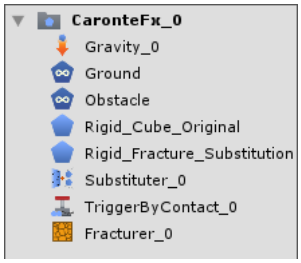


The idea is that “Cube_Original” should fall on the obstacle, but remain intact. When the cube touches the ground floor then the breaking process will be triggered. A situation like that is (almost) impossible to create with the “Substituter” and “Trigger By Contact” actions.

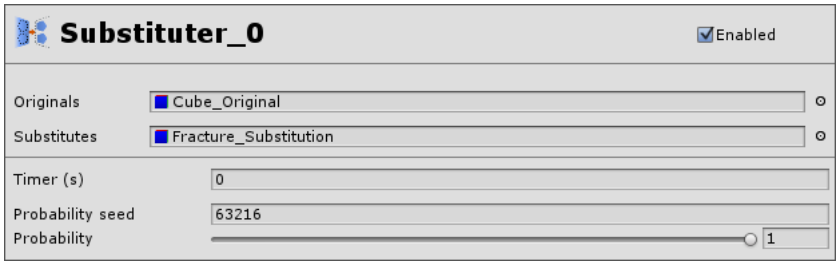
Before you start, a few more settings have to be made:

- Make the “Obstacle” object an irresponsive body (use a separate “Irresponsive” node).
- Make the fractured cube a separate “Rigid” body.

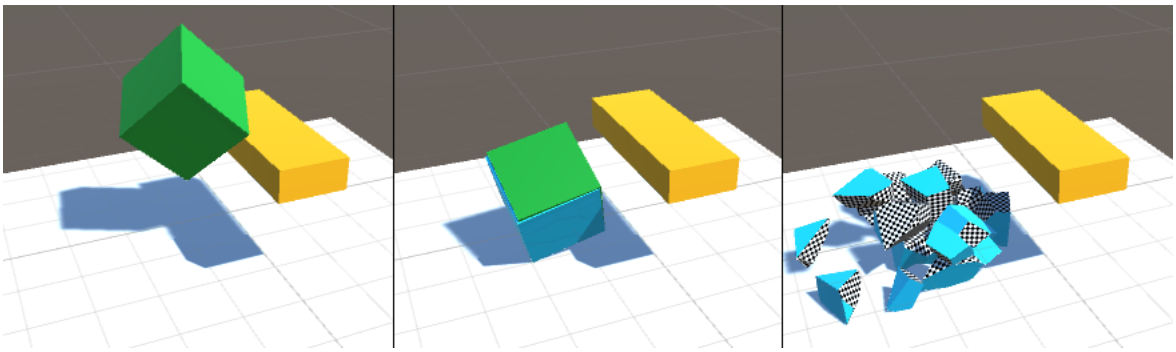
This is the complete SimulationObject tree:



While the “Trigger By Contact” node can remain untoched, the “Substituter” needs a change as well, as seen here:



When you simulate you will see that the behaviour is exactly as intended, and the cube breaks apart once it hits the ground.



Of course it is possible to create really complex setups with different trigger actions and times, various fractured objects, explosions, and even multiple substitutions.

Probability

Maybe you have also seen the “Probability” parameter? Imagine you have created 100 spheres and after 2 seconds you want to substitute some of these spheres through cubes. One idea is to create several groups, but this is a tedious task. A much more time-saving idea is the usage of “Probability”.
A value of

- 50 replaces roughly 50% of the spheres through cubes
- 81 replaces roughly 81% of the spheres.

If you do not like how the substitution is done simply change “Probability seed” and perform another simulation.

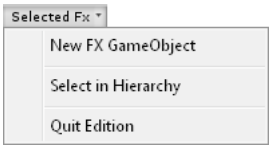
CaronteFX Scope

Filters, Groups, and Clusters

Multiple CaronteFx GameObjects

So far you have been working with a single CaronteFx GameObject: “CaronteFx_0”. All SimulationObjects are attached to this container. But, you also have the possibility of creating more objects and build groups of nodes. These groups can be simulated separately and they even have their own simulation settings – just click on “CaronteFx_0” to see the parameters.

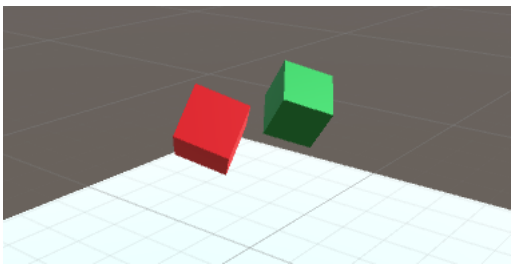
For a new CaronteFx GameObject click on the “Selected Fx” tab and choose “New FX GameObject”.



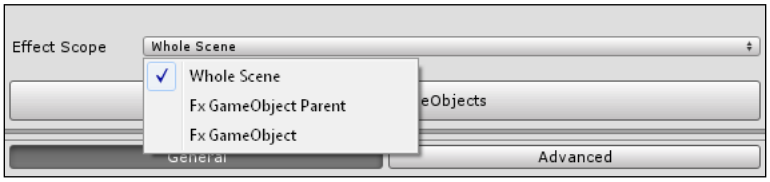
Maybe you have noticed that the CaronteFx GameObjects are also represented in the “Hierarchy” window? Here, they play an important role for a scene’s scope (see next chapter), because you are able to group a scene’s Unity® GameObjects under these containers in order to establish individual simulation clusters.

Understanding Scope

With scope it is possible to filter objects, build groups and clusters, and define how these objects will finally interact. The scope system allows you to include or exclude SimulationObjects selectively, based on your rules. Start with a basic scene, e.g. an irresponsive ground plane and two rigid or soft body cubes. One cube has a red material, the other node is green.



Your first stop with scope is the “CaronteFx_0” GameObject and its settings. There you find a menu named “Effect Scope”.



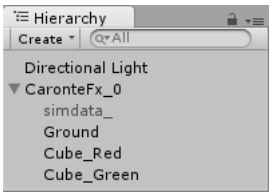
Effect Scope: Whole

The default setting is “Whole scene” and it tells you that all elements (with the same scope) are able to interact without restrictions – even from different CaronteFx GameObjects.

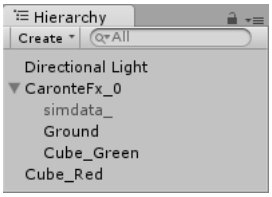
Effect Scope: Fx GameObject

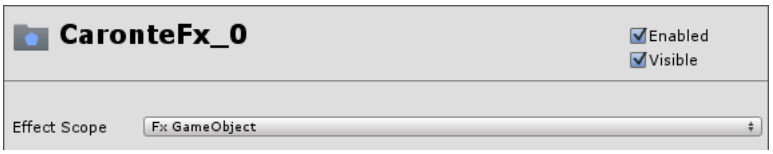
This option allows you to restrict the simulation’s scope to certain Unity® GameObjects:

- Go to the “Hierarchy” window and drag the three objects (“Ground”, “Cube red”, “Cube green”) to the “CaronteFx_0” entry.
- In the CaronteFx editor, change “Effect Scope” to “Fx GameObject”.
- Simulate.

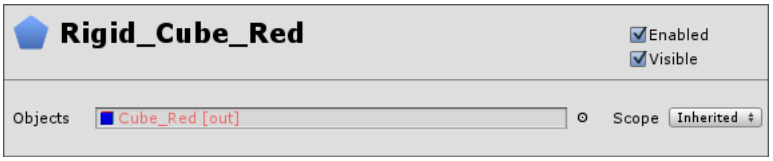


Again, all bodies are simulated and they are also able to interact. So far there is no difference to the “Whole scene” mode. Now return to “Hierarchy” and detach the red cube from “CaronteFx_0”. Start another simulation and you will see that only the green cube falls down and interacts with the ground. The red cube remains motionless, although it is a rigid body.



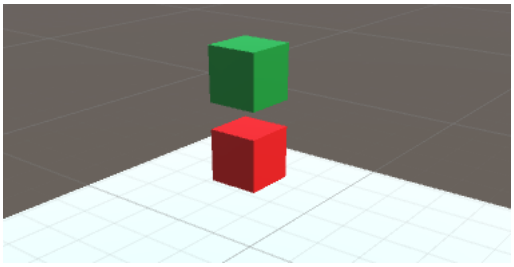


When you take a look at the “Objects” field of “Rigid_Cube_Red” you will notice a change, telling you that the red cube is out of scope now:

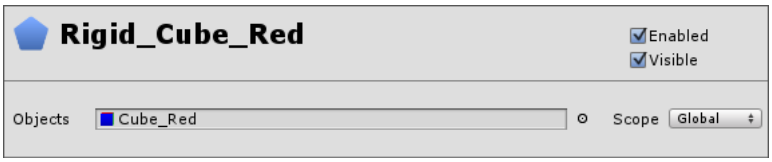


Global and Inherited

Change the red and green cubes' positions to make them interact as shown below:



With the current settings the green cube will go through the red object just as if it wasn't there. Go to "Rigid_Cube_Red" and set "Scope" to "Global". Simulate again. The red cube is not being affected by the "Gravity" daemon, but it is pushed by the green body.



Now guess what will happen when you set the "Gravity" daemon's local "Scope" to "Global"? If you think that the red cube will be attracted as well then you are right.

Effect Scope: Fx GameObject Parent

The mode of operation is similar to "Fx GameObject", but here you can set the scope for more than one CaronteFx GameObject (also see "Multiple CaronteFX GameObjects"). This it is possible to create separate groups of SimulationObjects, and treat them individually.

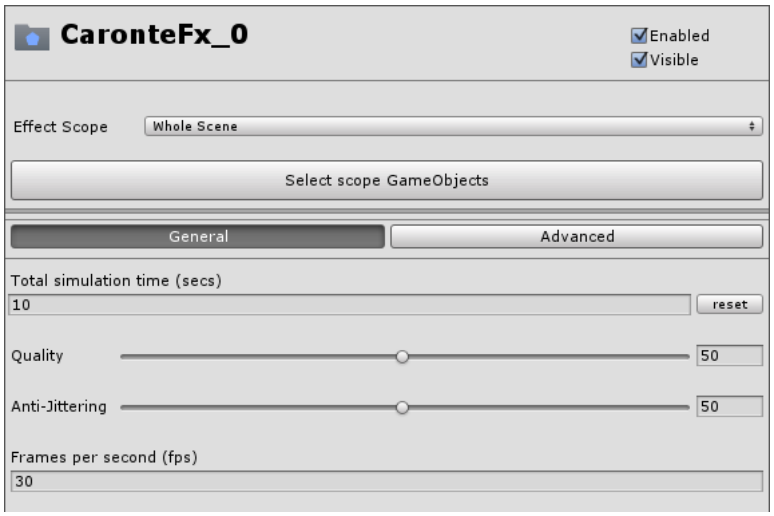
When you set a SimulationObject's local "Scope" to "Global", the object will be able to interact with SimulationObjects from different CaronteFx GameObjects. With "Inherited" the object's scope is restricted to its own CaronteFx tree.

Simulation Settings

Timesteps, FPS, Quality

CaronteFx GameObject and Simulation Parameters

A click on the “CaronteFx_0” tree reveals global simulation parameters and settings.



Effect Scope
A CaronteFx GameObject’s simulation nodes can be restricted to “Whole Scene”, “Fx GameObject Parent” or “Fx GameObject”.

For more information on how to work with scope please read the “Understanding Scope” chapter.

Select scope GameObjects
Choose the Unity® GameObjects from the “Hierarchy” window, and press this button to set their scope according to the option from “Effect Scope”. For more information on how to work with these options please read the “Understanding Scope” chapter.

General Parameters

Total simulation time (sec)
How long do you want the simulation to last? Enter an appropriate number of seconds here.

Reset
Apply CaronteFX’s default simulation time (10.0 s).

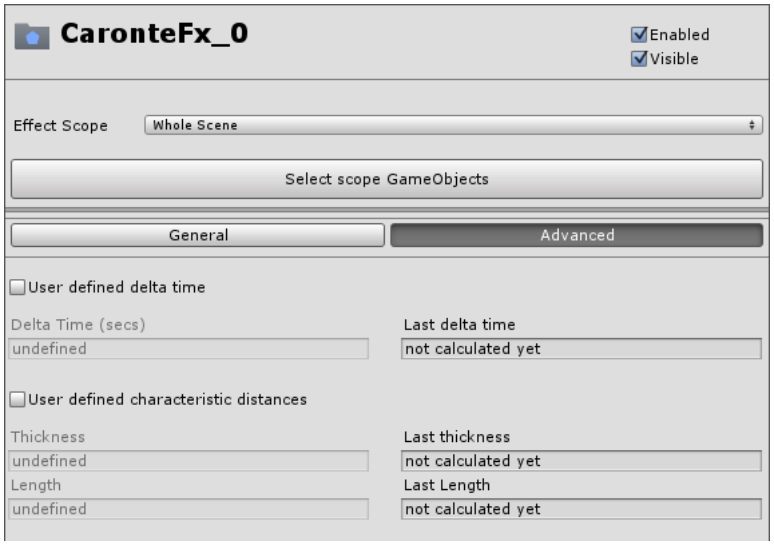
Quality
This parameter ranges from 1 to 100 and determines the simulation’s accuracy:

- With higher settings, simulation times increase.
- The default value is 50, but settings around 25 are sufficient in many cases.
- Change the simulation’s fps rate.

Anti-Jittering
With this parameter it is possible to gradually reduce trembling and jitter effects in a range from 0 to 100, where 100 completely removes the tremble.

Frames per second (fps)
Change the simulation's fps rate.

"Advanced" Parameters



User defined delta time
In simulations, time is not continuous, but subdivided into steps. The number of steps is responsible for the simulation's accuracy and the more steps there are, the better the final result. As a side-effect, simulation time increases.

CaronteFX determines the length of these steps, measured in seconds, automatically by default, and shorter values mean more steps. The associated parameter is the "Quality" slider. Alternatively you can also define the length of a time step manually by activating this option. Under "Delta time (secs)" you enter the desired value.

Delta time (secs)
Enter the length of the custom time step in seconds. Smaller steps/values will increase accuracy, but also simulation time. The value from "Last delta time" can be used as a reference.

Last delta time
This is a read-only field and shows the length of the simulation's last time step.

User defined characteristic distances
This feature maintains the behaviour of a scene's elements when you perform simulations in several parts.

Imagine a pre-fractured rigid body vase and soft body flowers falling on the floor:

- In the first part, only the vase dynamics is simulated, because the flowers will hardly contribute to the fragments' behaviour.
- Once you are satisfied with the behaviour of the vase's pieces, activate the flower objects and re-simulate.

What you might observe is a completely different behaviour, because RealFlow adjusted the scales automatically to react on the new situation:

- Start the simulation with all nodes (here: vase, flowers).
- After a short time you can stop the simulation.
- Check "User defined characteristic distances" and transfer the "Last used" values to their associated fields.
- Reset and simulate the parts individually – here: a) vase only and b) vase + flowers

If you decide to add completely new elements to an existing scene you have to repeat the four steps above and readjust CaronteFX.

Thickness
This parameter is related to the scene's objects thickness:

- Under "Last used" you can see the values the solver used during the last simulation.
- To make CaronteFX work with a fixed value, enter a new value.
- We recommend using the associated "Last thickness" value.

Length
This parameter describes the scene's characteristic object length:

- Under "Last used" you can see the values the solver used during the last simulation.
- To make CaronteFX work with a fixed value, enter a new value.
- We recommend using the associated "Last Length" value.

General Reference

Duplicate, Group

“General” Parameter Reference

There are two functional buttons without parameters:

Duplicate
Choose one or more nodes from the CaronteFx GameObject tree and click on this button to duplicate them.

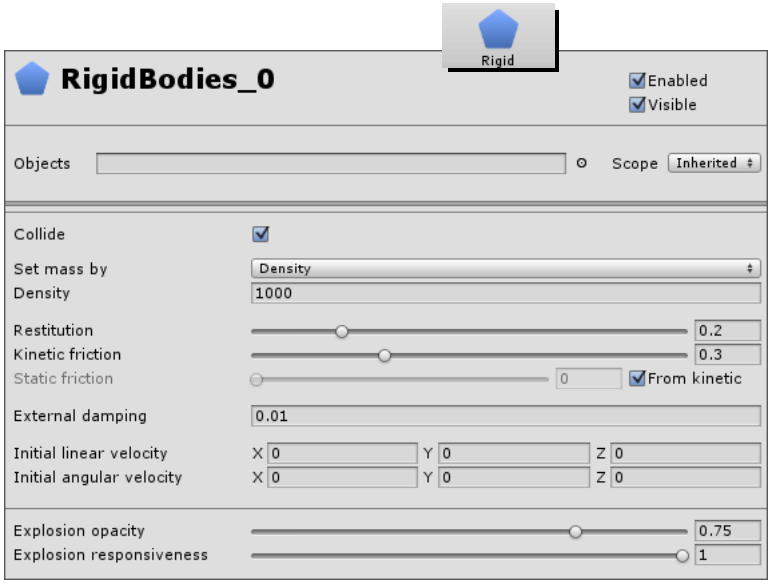
Group
Click on this button to group the selected nodes from the CaronteFx GameObject tree. It is also possible to create an empty group container and drag SimulationObjects to the bin.

Bodies Reference

Rigid, Irresponsive, Animated, Soft, Cloth

“Rigid” Parameter Reference

Rigid bodies are movable, not deformable, and they can interact with daemons and other body types (rigid, soft, irresponsive, animated).



Enabled
Enable or disable the selected body with this checkbox.

Visible
Show or hide the selected body in the viewport.

Objects
This selection specifies the scene’s rigid bodies. There are several ways to choose scene elements, for example:

- Drag and drop Unity® GameObjects from the “Hierarchy” window.
- Enter or copy/paste the name of a GameObject to the field directly

A left-click on the “Objects” field opens a node browser where you can see the already attached Game-Objects arranged in a list. Here it is also possible to manage your nodes:

- Remove an object by selecting and right-clicking on an entry. Then, choose “Remove”
- Rename an object by selecting and right-clicking on an entry. Then, choose “Rename”
- Drag and drop GameObjects from the “Hierarchy” window.
- Use the entries from the node bowser’s “Edit” menu.

Collide
When disabled the body will not be able to collide with other bodies anymore.

Scope
Only GameObjects within the defined scope will be processed. Other nodes are considered out of reach. There are two options – please read the “CaronteFX Scope” chapters for more information:

- “Global”. All GameObjects are able to interact without restrictions.
- “Inherited”. The scope will be used from the closest CaronteFX parent node of the selected object.

Set mass by
There are two options available – please read the “Understanding Mass” and “Understanding Density” chapters for more information:

- “Mass (per single body)”. Here you enter a specific mass in kg. If the body is composed of multiple rigid bodies the entered value will be applied to every body. If the bodies have different volumes, density will be varying.
- “Density”. This parameter is measured in kg/m^3. If the body is composed of multiple rigid bodies the entered value will be applied to every body. If the bodies have different volumes, mass will be varying.

Density | Mass per object
This parameter changes according to your selection under “Set mass by” set to “Density”. With “Density” it is, for example, possible to assign different masses to fragments. When “Mass per object” is active every body under “Objects” will receive exactly the same mass - regardless its volume/size. Please read the “Understanding Mass” and “Understanding Density” chapters for more information.

Restitution
When bodies collide they bounce off. The resulting effective bounciness is calculated from the “Restitution” values of all colliding bodies. This parameter ranges from 0 to 1.

Kinetic friction
In the real world, objects have more or less uneven surfaces and this circumstance causes a loss of motion energy. The resulting effective kinetic friction is calculated from the “Friction kinetic” values of all interacting bodies. This parameter ranges from 0 to 1.

Static friction
Imagine a cube resting on an inclined plane. When the bodies’ surfaces are very even the cube will start to slide quickly. With rough surfaces, it will take longer until the cube starts to move. In plain words, this property can be seen a measure for a body’s ability to start sliding. The resulting effective static friction is calculated from the “Friction static” values of all interacting bodies. This parameter ranges from 0 to 1. With “From kinetic” enabled, the value is taken from the “Kinetic friction” parameter.

External damping
This parameter decreases a rigid body's velocity (linear and angular), and very high settings can cause a body to stop completely. You can use any positive value.

Initial linear velocity
Here a start velocity is defined. The resulting velocity is calculated from the three axes; they also determine the object's motion direction. All positive and negative values, including 0, are allowed.

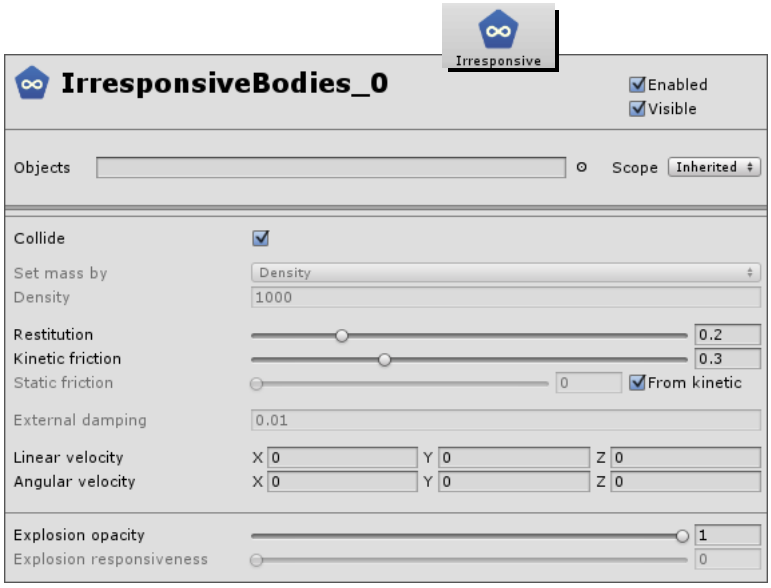
Initial angular velocity
The three values represent an object's rotation axes. Each value specifies how many degrees the object will cover per second. As a result the body will be spinning around its centre of gravity. All positive and negative values, including 0, are allowed.

Explosion opacity
Objects can attenuate or damp the effect of an explosion. Other bodies behind such a "blocker" are more or less affected by the explosion. This parameter ranges from 0 to 1, and specifies how strong this attenuation will be. With 1, the object will block the explosion completely.

Explosion responsiveness
Here you can adjust directly how strong a body will respond to an explosion. The parameter ranges from 0 to 1. With 1, the object will be fully affected by the explosion without any attenuation.

“Irresponsive” Parameter Reference

Irresponsive bodies are non-deformable and do not react on other dynamic elements. To other bodies they appear as objects with a huge mass. A typical example for an irresponsive body is a ground plane.



Enabled
Enable or disable the selected body with this checkbox.

Visible
Show or hide the selected body in the viewport.

Objects
This selection specifies the scene’s rigid bodies. There are several ways to choose scene elements, for example:

- Drag and drop Unity® GameObjects from the “Hierarchy” window.
- Enter or copy/paste the name of a GameObject to the field directly

A left-click on the “Objects” field opens a node browser where you can see the already attached Game-Objects arranged in a list. Here it is also possible to manage your nodes:

- Remove an object by selecting and right-clicking on an entry. Then, choose “Remove”
- Rename an object by selecting and right-clicking on an entry. Then, choose “Rename”
- Drag and drop GameObjects from the “Hierarchy” window.
- Use the entries from the node bowser’s “Edit” menu.

Collide
When disabled the body will not be able to collide with other bodies anymore.

Scope
Only GameObjects within the defined scope will be processed. Others are considered out of reach. There are two options - please read the “CaronteFX Scope” chapters for more information:

- “Global”. All GameObjects are able to interact without restrictions.
- “Inherited”. The scope will be used from the closest CaronteFX parent node of the selected object.

Set mass by
An irresponsive body’s mass is considered huge and therefore the parameter is locked permanently.

Density | Mass per object
An irresponsive body’s mass is considered huge and therefore the parameter is locked permanently.

Restitution
When bodies collide they bounce off. The resulting effective bounciness is calculated from the “Restitution” values of all colliding bodies. This parameter ranges from 0 to 1.

Kinetic friction
In the real world, objects have more or less uneven surfaces and this circumstance causes a loss of motion energy. The resulting effective kinetic friction is calculated from the “Friction kinetic” values of all interacting bodies. This parameter ranges from 0 to 1.

Static friction
Imagine a cube resting on an inclined plane. When the bodies’ surfaces are very even the cube will start to slide quickly. With rough surfaces, it will take longer until the cube starts to move. In plain words, this property can be seen a measure for a body’s ability to start sliding. The resulting effective static friction is calculated from the “Friction static” values of all interacting bodies. This parameter ranges from 0 to 1. With “From kinetic” enabled, the value is taken from the “Kinetic friction” parameter.

External damping
Irresponsive bodies cannot be moved dynamically, e..g through collisions. Therefore the parameter is locked permanently.

Initial linear velocity
Here a start velocity is defined. The resulting velocity is calculated from the three axes; they also determine the object’s motion direction. All positive and negative values, including 0, are allowed.

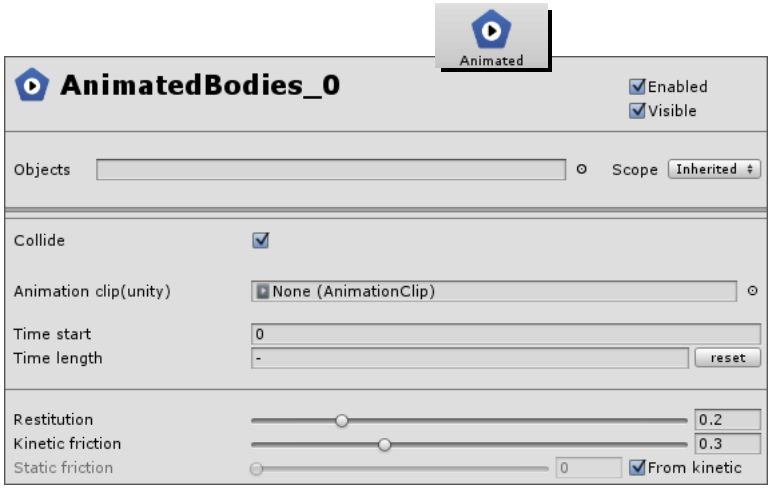
Initial angular velocity
The three values represent an object’s rotation axes. Each value specifies how many degrees the object will cover per second. As a result the body will be spinning around its centre of gravity. All positive and negative values, including 0, are allowed.

Explosion opacity
Objects can attenuate or damp the effect of an explosion. Other bodies behind such a “blocker” are more or less affected by the explosion. This parameter ranges from 0 to 1, and specifies how strong this attenuation will be. With 1, the object will block the explosion completely.

Explosion responsiveness
Here you can adjust directly how strong a body will respond to an explosion. The parameter ranges from 0 to 1. With 1, the object will be fully affected by the explosion without any attenuation.

“Animated” Parameter Reference

Choose an existing animation clip from inside Unity®. This can also be a previously baked CaronteFX simulation.



Enabled
Enable or disable the selected body with this checkbox.

Visible
Show or hide the selected body in the viewport.

Objects
This selection specifies the scene’s rigid bodies. There are several ways to choose scene elements, for example:

- Drag and drop Unity® GameObjects from the “Hierarchy” window.
- Enter or copy/paste the name of a GameObject to the field directly

A left-click on the “Objects” field opens a node browser where you can see the attached GameObjects arranged in a list. Here it is also possible to manage your nodes:

- Remove an object by selecting and right-clicking on an entry. Then, choose “Remove”
- Rename an object by selecting and right-clicking on an entry. Then, choose “Rename”
- Drag and drop GameObjects from the “Hierarchy” window.
- Use the entries from the node bowser’s “Edit” menu.

Collide
When disabled the body will not be able to collide with other bodies anymore.

Collide
When disabled the body will not be able to collide with other bodies anymore.

Scope
Only GameObjects within the defined scope will be processed. Others are considered out of reach. There are two options - please read the “CaronteFX Scope” chapters for more information:

- “Global”. All GameObjects are able to interact without restrictions.
- “Inherited”. The scope will be used from the closest CaronteFX parent node of the selected object.

Animation clip (unity)
Animation clips containe objects with predefined motions. Drag a clip to the empty field to use it in conjunction with CaronteFX.

Time start
The animation will start at the given simulation time measured in seconds.

Time length
This is the animation’s duration. After this period of time the body will remain motionless at it last position. “Time length” is given in seconds.

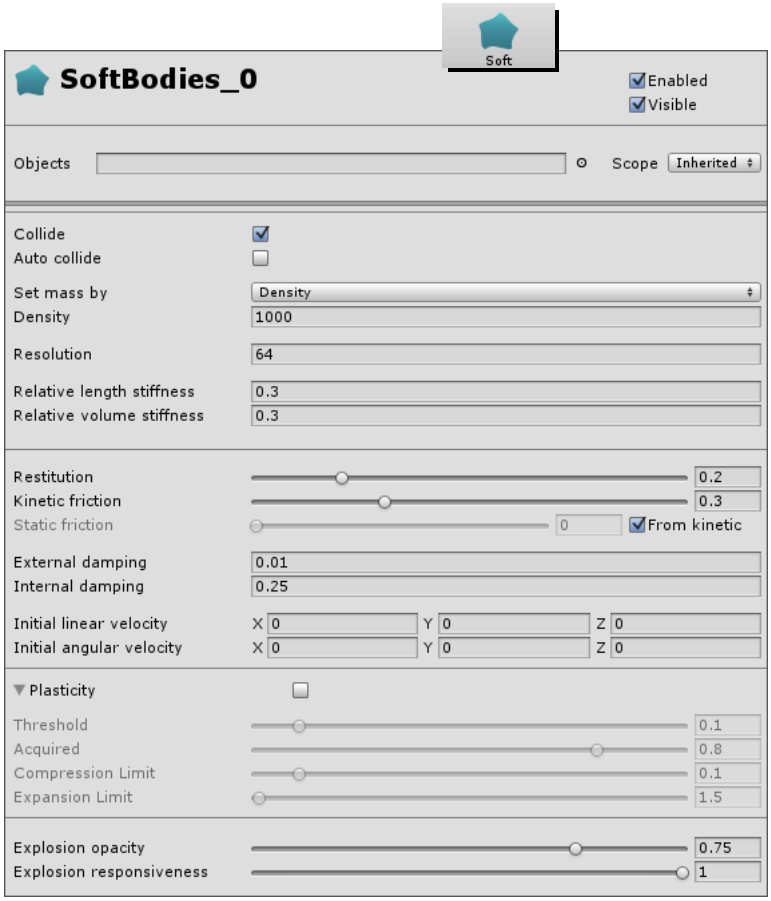
Restitution
When bodies collide they bounce off. The resulting effective bounciness is calculated from the “Restitution” values of all colliding bodies. This parameter ranges from 0 to 1.

Kinetic friction
In the real world, objects have more or less uneven surfaces and this circumstance causes a loss of motion energy. The resulting effective kinetic friction is calculated from the “Friction kinetic” values of all interacting bodies. This parameter ranges from 0 to 1.

Static friction
Imagine a cube resting on an inclined plane. When the bodies’ surfaces are very even the cube will start to slide quickly. With rough surfaces, it will take longer until the cube starts to move. In plain words, this property can be seen a measure for a body’s ability to start sliding. The resulting effective static friction is calculated from the “Friction static” values of all interacting bodies. This parameter ranges from 0 to 1. With “From kinetic” enabled, the value is taken from the “Kinetic friction” parameter.

“Soft” Parameter Reference

Soft bodies are movable, deformable, and they can interact with daemons and other body types (rigid, soft, irresponsive, animated). You can decide whether the deformation is temporary or permanent.



Enabled
Enable or disable the selected body with this checkbox.

Visible
Show or hide the selected body in the viewport.

Objects
This selection specifies the scene’s rigid bodies. There are several ways to choose scene elements, for example:

- Drag and drop Unity® GameObjects from the “Hierarchy” window.
- Enter or copy/paste the name of a GameObject to the field directly.

A left-click on the “Objects” field opens a node browser where you can see the attached GameObjects arranged in a list. Here it is also possible to manage your nodes:

- Remove an object by selecting and right-clicking on an entry. Then, choose “Remove”
- Rename an object by selecting and right-clicking on an entry. Then, choose “Rename”
- Drag and drop GameObjects from the “Hierarchy” window.
- Use the entries from the node bowser’s “Edit” menu.

Collide
When disabled the body will not be able to collide with other bodies anymore.

Auto collide
When active, CaronteFX will calculate collisions between the object’s vertices and polygons (also known as self-collision). With complex geometry, simulation time will increase noticeably.

Scope
Only GameObjects within the defined scope will be processed. Others are considered out of reach. There are two options - please read the “CaronteFX Scope” chapters for more information:

- “Global”. All GameObjects are able to interact without restrictions.
- “Inherited”. The scope will be used from the closest CaronteFX parent node of the selected object.

Set mass by
There are two options available (please also see “Understanding Mass” and “Understanding Density”)

- “Mass (per single body)”. Here you enter a specific mass in kg. If the body is composed of multiple rigid bodies the entered value will be applied to every body. If the bodies have different volumes, density will be varying.
- “Density”. This parameter is measured in kg/m^3. If the body is composed of multiple rigid bodies the entered value will be applied to every body. If the bodies have different volumes, mass will be varying.

Density | Mass per object
This parameter changes according to your selection under “Set mass by” set to “Density”. With “Density” it is, for example, possible to assign different masses to fragments. When “Mass per object” is active every body under “Objects” will receive exactly the same mass - regardless its volume/size. Please read the “Understanding Mass” and “Understanding Density” chapters for more information.

Resolution
A soft body is subdivided into flexible elements and the number of elements defines the simulation’s accuracy. With higher settings you are able to achieve a higher level of detail, but at the cost of longer simulation times.

The object’s maximum “Resolution” depends on its number of triangles and for higher accuracy more (= smaller) triangles are needed.

Relative length stiffness
The higher the value, the more rigid the object will finally appear, because it will be more resistant against changes along its longitudinal axes (width, length, height). This parameter accepts any positive value. With 1, for example, a soft body will nearly maintain its shape when resting on a horizontal plane. In this case, the body is able to resist its own weight without being deformed.

Relative volume stiffness
The higher the value, the more rigid the object will finally appear, because it will be more resistant against changes of its original volume. This parameter accepts any positive value. With 1, for example, a soft body will nearly maintain its shape when resting on a horizontal plane. In this case, the body is able to resist its own weight without being deformed.

Restitution
When bodies collide they bounce off. The resulting effective bounciness is calculated from the “Restitution” values of all colliding bodies. This parameter ranges from 0 to 1.

Kinetic friction
In the real world. objects have more or less uneven surfaces and this circumstance causes a loss of motion energy. The resulting effective kinetic friction is calculated from the “Friction kinetic” values of all interacting bodies. This parameter ranges from 0 to 1.

Static friction
Imagine a cube resting on an inclined plane. When the bodies’ surfaces are very even the cube will start to slide quickly. With rough surfaces, it will take longer until the cube starts to move. In plain words, this property can be seen a measure for a body’s ability to start sliding. The resulting effective static friction is calculated from the “Friction static” values of all interacting bodies. This parameter ranges from 0 to 1. With “From kinetic” enabled, the value is taken from the “Kinetic friction” parameter.

External damping
This parameter decreases a rigid body’s velocity (linear and angular), and very high settings can cause a body to stop completely. You can use any positive value.

Initial linear velocity
Here a start velocity is defined. The resulting velocity is calculated from the three axes; they also determine the object’s motion direction. All positive and negative values, including 0, are allowed.

Initial angular velocity
The three values represent an object’s rotation axes. Each value specifies how many degrees the object will cover per second. As a result the body will be spinning around its centre of gravity. All positive and negative values, including 0, are allowed.

Plasticity
When enabled, parts of the object will remain deformed. The final amount of permanent deformation depends on the associated parameters.

Threshold
As long as this limit is not reached the deformation will still be able to recover. If the deformation is stronger it will remain permanently. The parameter ranges between 0 and 1. A value of 0.5 means that the change of volume in any direction has to be at least 50% of the object's original size to produce permanent deformation.

An example:
Let's assume there is a cube with 1 m x 1 m x 1 m and a "Threshold" of 0.5. The deformation in one or more directions has to be least 0.5 m or to get a permanent deformation, e.g. 1 m x 0.5 m x 1.0 m.

Acquired
This parameter represents the percentage of deformation that will remain permanent. With 0.0, the object will recover completely, with 1.0 the entire deformation will be kept.

Compression limit
With this parameter it is possible to prevent a soft body from becoming very flat. The parameter ranges between 0 and 1. With 0.5, for example, deformations will be permanent when the body's compression in any direction is 50% of its original size. Deformations beyond this limit will be able to recover.

Expansion limit
The mode of operation is very similar to "Compression limit", but here the soft body is prevented from becoming too thick. The parameter accepts any positive value. With 2.0, for example, deformations will be permanent when the body's expansion in any direction is 50% of its original size. Deformations beyond this limit will be able to recover.

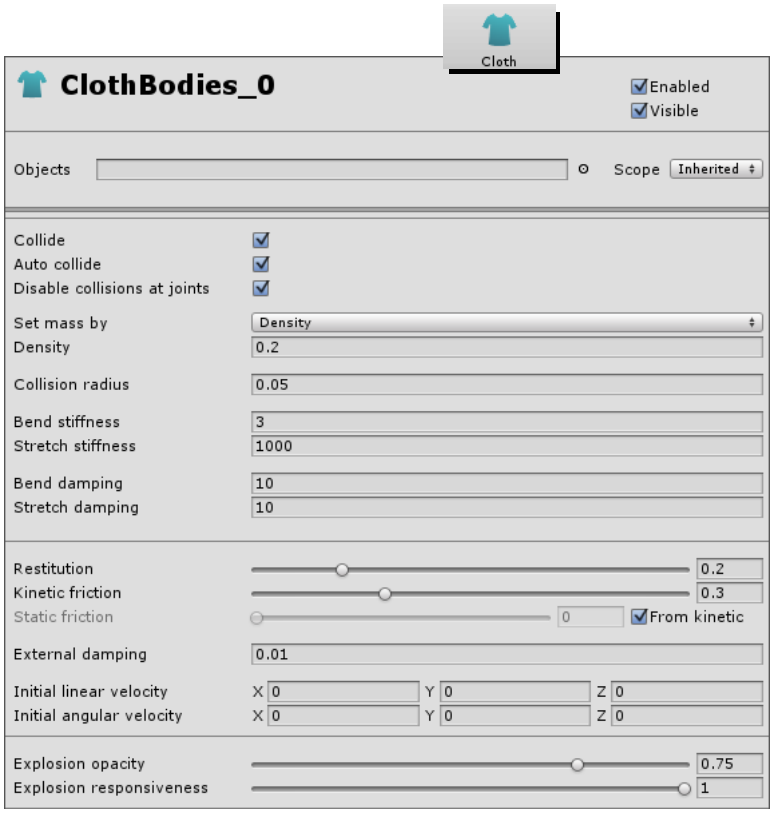
Internal damping
A soft body always has a certain amount of internal motion ("wobbling"). With "Internal damping" and "Restitution" it is possible to control this effect. "Internal damping" accepts any positive value. With high settings, the body will stop wobbling quickly, and it will also experience smaller bounces.

Explosion opacity
Objects can attenuate or damp the effect of an explosion. Other bodies behind such a "blocker" are more or less affected by the explosion. This parameter ranges from 0 to 1, and specifies how strong this attenuation will be. With 1, the object will block the explosion completely.

Explosion responsiveness
Here you can adjust directly how strong a body will respond to an explosion. The parameter ranges from 0 to 1. With 1, the object will be fully affected by the explosion without any attenuation.

“Cloth” Parameter Reference

Cloth bodies are movable, deformable, and they can interact with daemons and other body types (rigid, soft, cloth, irresponsive, animated). You can decide whether the deformation is temporary or permanent.



Enabled
Enable or disable the selected body with this checkbox.

Visible
Show or hide the selected body in the viewport.

Objects
This selection specifies the scene’s rigid bodies. There are several ways to choose scene elements, for example:

- Drag and drop Unity® GameObjects from the “Hierarchy” window.
- Enter or copy/paste the name of a GameObject to the field directly.

A left-click on the “Objects” field opens a node browser where you can see the attached GameObjects arranged in a list. Here it is also possible to manage your nodes:

- Remove an object by selecting and right-clicking on an entry. Then, choose “Remove”
- Rename an object by selecting and right-clicking on an entry. Then, choose “Rename”
- Drag and drop GameObjects from the “Hierarchy” window.
- Use the entries from the node bowser’s “Edit” menu.

Collide
When disabled the body will not be able to collide with other bodies anymore.

Auto collide
When active, CaronteFX will calculate collisions between the object’s vertices and polygons (also known as self-collision). With complex geometry, simulation time will increase noticeably.

Disable collision at joints
This option is only relevant in conjunction with a joiint SimulationObject, e.g. “Rigid Glue”. When checked collision between the cloth object and its joined body will not be calculated near the joints’ location.

Scope
Only GameObjects within the defined scope will be processed. Others are considered out of reach. There are two options - please read the “CaronteFX Scope” chapters for more information:

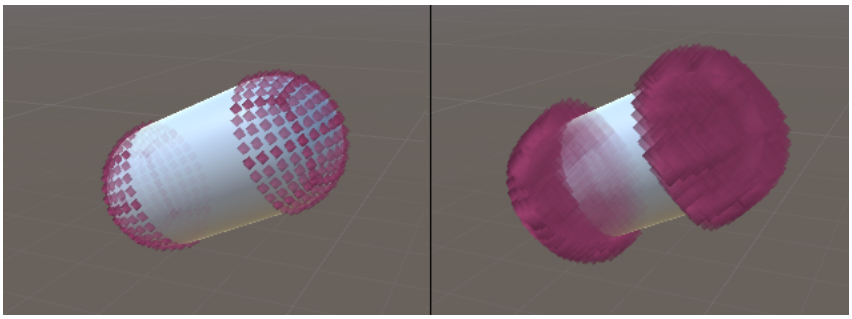
- “Global”. All GameObjects are able to interact without restrictions.
- “Inherited”. The scope will be used from the closest CaronteFX parent node of the selected object.

Set mass by
There are two options available (please also see “Understanding Mass” and “Understanding Density”)

- “Mass (per single body)”. Here you enter a specific mass in kg. If the body is composed of multiple rigid bodies the entered value will be applied to every body. If the bodies have different volumes, density will be varying.
- “Density”. This parameter is measured in kg/m^3. If the body is composed of multiple rigid bodies the entered value will be applied to every body. If the bodies have different volumes, mass will be varying.

Density | Mass per object
This parameter changes according to your selection under “Set mass by” set to “Density”. With “Density” it is, for example, possible to assign different masses to fragments. When “Mass per object” is active every body under “Objects” will receive exactly the same mass - regardless its volume/size. Please read the “Understanding Mass” and “Understanding Density” chapters for more information.

Collision radius
This parameter is used to improve (self-) collision. With higher values you will observe less intersections between collision objects, but also an increasing distance between cloth and collision object. “Collision radius” is displayed in the viewport as purple cubes of variable size at the cloth’s vertices. The value is given in metres. Here are two examples with “Collision radius” values of 0.05 and 0.2:



Bend stiffness
Here you define the cloth body’s resistance against bending forces. With higher values, cloth objects become more and more rigid. The cloth will also try to restitute its original shape and this results in a wobbling effect. To suppress this effect, increase “Bend damping”.

Stretch stiffness
When cloth interacts with other objects you will observe that the cloth will be stretched. Here you define the cloth body’s resistance against stretching forces. With higher settings, cloth objects appear more rigid. With very lower values (e.g. 50), the cloth object might become completely flat. Please avoid values smaller than 10. With higher values, cloth objects become more and more rigid. The cloth will also try to restitute its original shape and this results in a wobbling effect. To suppress this effect, increase “Stretch damping”.

Bend damping
If you observe a wobbling effect or extreme contraction with small “Bend stiffness” you should increase this parameter.

Stretch damping
If you observe a wobbling effect or extreme contraction with small “Stretch stiffness” you should increase this parameter.

Restitution
When bodies collide they bounce off. The resulting effective bounciness is calculated from the “Restitution” values of all colliding bodies. This parameter ranges from 0 to 1.

Kinetic friction
In the real world, object have more or less uneven surfaces and this circumstance causes a loss of motion energy. The resulting effective kinetic friction is calculated from the “Friction kinetic” values of all interacting bodies. This parameter ranges from 0 to 1.

Static friction
Imagine a cube resting on an inclined plane. When the bodies’ surfaces are very even the cube will start to slide quickly. With rough surfaces, it will take longer until the cube starts to move. In plain words, this property can be seen a measure for a body’s ability to start sliding. The resulting effective static friction is calculated from the “Friction static” values of all interacting bodies. This parameter ranges from 0 to 1. With “From kinetic” enabled, the value is taken from the “Kinetic friction” parameter.

External damping
This parameter decreases a rigid body’s velocity (linear and angular), and very high settings can cause a body to stop completely. You can use any positive value.

Initial linear velocity
Here a start velocity is defined. The resulting velocity is calculated from the three axes; they also determine the object’s motion direction. All positive and negative values, including 0, are allowed.

Initial angular velocity
The three values represent an object’s rotation axes. Each value specifies how many degrees the object will cover per second. As a result the body will be spinning around its centre of gravity. All positive and negative values, including 0, are allowed.

Explosion opacity
Objects can attenuate or damp the effect of an explosion. Other bodies behind such a “blocker” are more or less affected by the explosion. This parameter ranges from 0 to 1, and specifies how strong this attenuation will be. With 1, the object will block the explosion completely.

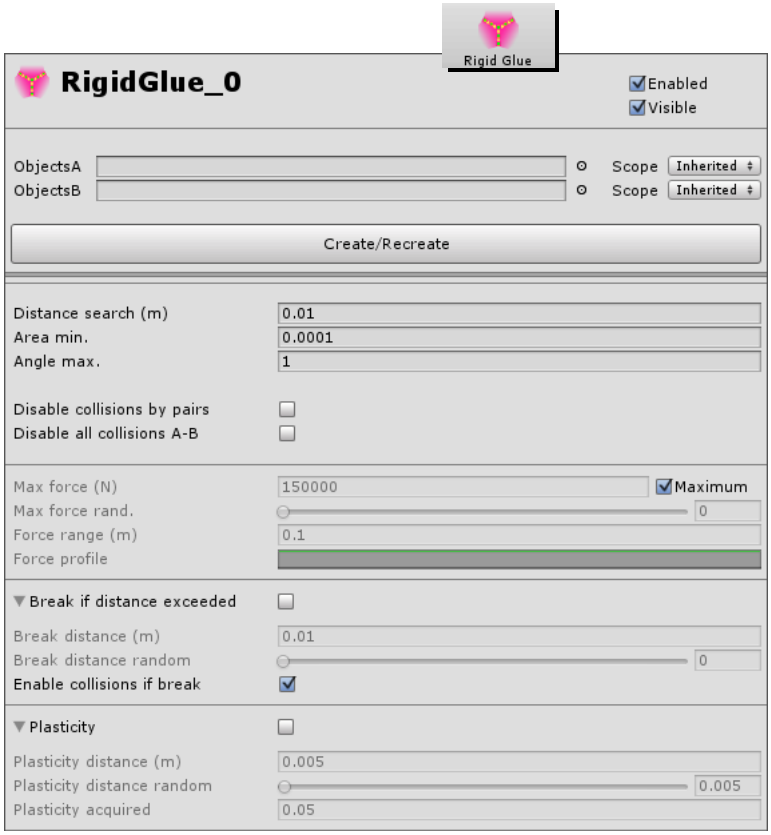
Explosion responsiveness
Here you can adjust directly how strong a body will respond to an explosion. The parameter ranges from 0 to 1. With 1, the object will be fully affected by the explosion without any attenuation.

Joints Reference

Glue, Area, Vertices, Leaves, Locators

“Rigid Glue” Parameter Reference

With this type it is possible to quickly connect rigid bodies with varying orientation. “Rigid Glue” is perfectly suited for joining prefractured pieces.



Enabled
Enable or disable the selected body with this checkbox.

Visible
Show or hide the selected body in the viewport.

ObjectsA
Here, the first group of objects is specified you want to connect to a second group, found under “ObjectsB”.

ObjectsB
This is the second group of objects you have to choose to create the links. If you attach exactly the same nodes to “ObjectsA” and “ObjectsB” connections between all selected items will be created. A good example is a brick wall.

Create/Recreate
Establish or recreate the connections according to your settings.

Scope
Only GameObjects within the defined scope will be processed. Others are considered out of reach. There are two options - please read the “CaronteFX Scope” chapters for more information:

- “Global”. All GameObjects are able to interact without restrictions.
- “Inherited”. The scope will be used from the closest CaronteFX parent node of the selected object.

Distance search (m)
With this parameter you can define the distance at which CaronteFX should look for objects to connect. By increasing this value you can look for object pairs which are further away.

Area min.
To establish joints, the contact area between the faces of two objects must be greater than this value. It is safe to leave the default value in most cases. The value is measured in metres.

Angle max.
Joints will only be created when the angles between nodes’ faces are smaller than this value. The value is given in degrees.

Disable collisions by pairs
When enabled, CaronteFX stops processing collisions between all body pairs connected by joints. This results in shorter simulation times.

Disable all collisions A-B
The difference to “Disable collisions by pairs” is that nodes from “ObjectsA | B” do not necessarily have to be joined, for example when the distances between them are too big. If you disable collisions between pieces when you already know that they will not interact you will see an increase in simulation speed. An example is a pre-fractured object where you do not need collision between the fragments until it begins to break apart.

Max force (N)
Define the force which is needed to separate the objects. When the acting force is smaller than “Max force (N)” the joint remains intact and the objects are pulled back. It requires some testing and experience to find working settings.

Max force rand.
The actual value will be between 0 and the given positive value. The total force is calculated this way:
`Total force = Max force (N) + Max force rand.`

Force range (m)
The “bandwidth” in which the force acts on the joints, given in metres.

Force profile
You can draw a curve to determine the force at different distance points. This way it is possible to define a decreasing range of forces with growing distance, force peaks, etc.

Break if distance exceeded
With the associated “Break distance (m)” you can define a distance at which a joint will break.

Break distance (m)
If the given distance between two joints is exceeded, the connection is broken. The unit is metres.

Break distance random
The actual value will be between 0 and the given positive value. This value is given in metres and the total distance is calculated this way: `Total distance = Break distance + Break distance random`

Enable collisions if break
When active, objects with broken joints will not interpenetrate. The way, how the collision is handled also depends on “Disable collision by pairs” and “Disable all collisions A-B”:

- If “Disable collision by pairs” is active only collisions between a single “ObjectsA” body and its connected “ObjectsB” node will be simulated.
- If “Disable all collisions A-B” is active a single “ObjectsA” body is able to collide with all nodes from “ObjectsB”, and vice versa.
- If none of these nodes is active the result is the same as with “Disable all collisions A-B”.

Plasticity
With “Plasticity” you can make fragments stick together as if they were connected with a kind of underlying tissue or grid, while other parts can still leave the assembly.

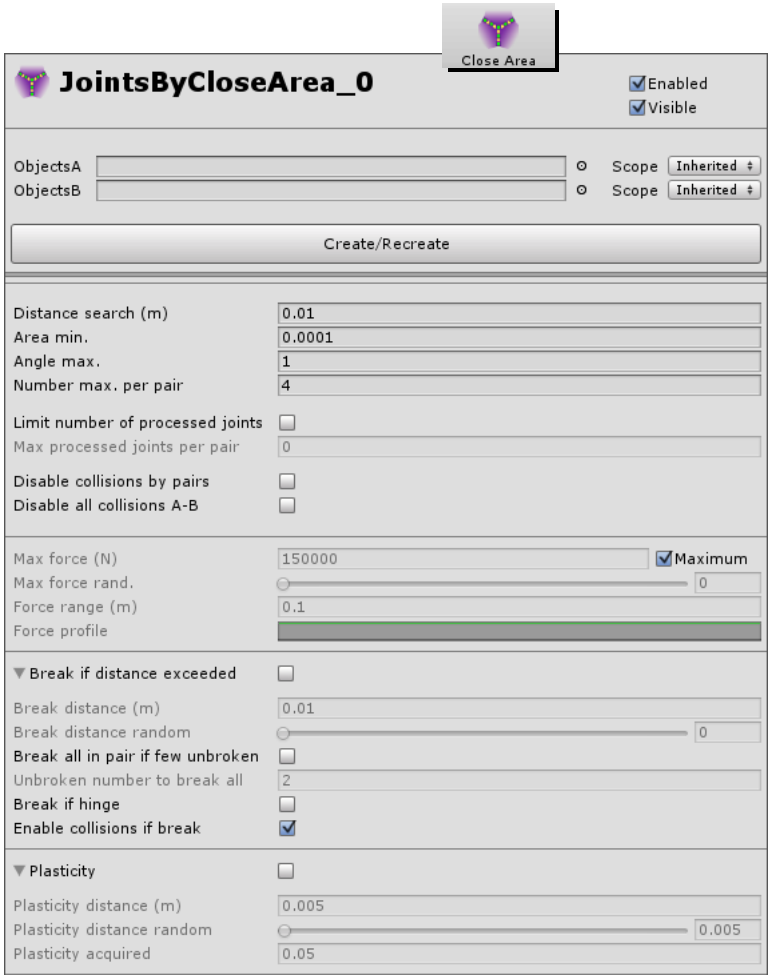
Plasticity distance (m)
When the distance between two joints exceeds this value the connection between joints becomes irreversible – it behaves like an overstretched rubber band. This value is given in metres.

Plasticity distance random
The actual value will be between 0 and the given positive value. This value is given in metres and the total distance is calculated this way:
`Total distance = @ Plasticity distance + @ Plasticity distance random`

Plasticity acquired
The parameter’s range goes from 0.0 to 1.0. A setting of 0.5 will keep approximately 50% of a joint’s deformation as permanent. The other half is able to relax and turn back to its initial state.

“Close Area” Parameter Reference

The tool looks for the bodies’ polygons within a given area and connects them. Rigid, soft, cloth, and irresponsive bodies are supported.



Enabled
Enable or disable the selected body with this checkbox.

Visible
Show or hide the selected body in the viewport.

ObjectsA
Here, the first group of objects is specified you want to connect to a second group, found under “ObjectsB”.

ObjectsB
This is the second group of objects you have to choose to create the links. If you attach exactly the same nodes to “ObjectsA” and “ObjectsB” connections between all selected items will be created. A good example is a brick wall.

Create/Recreate
Establish or recreate the connections according to your settings.

Scope
Only GameObjects within the defined scope will be processed. Others are considered out of reach. There are two options - please read the “CaronteFX Scope” chapters for more information:

- “Global”. All GameObjects are able to interact without restrictions.
- “Inherited”. The scope will be used from the closest CaronteFX parent node of the selected object.

Distance search (m)
With this parameter you can define the distance at which CaronteFX should look for objects to connect. By increasing this value you can look for object pairs which are further away.

Area min.
To establish joints, the contact area between the faces of two objects must be greater than this value. It is safe to leave the default value in most cases. The value is measured in metres.

Angle max.
Joints are created when the angles between the nodes’ faces are smaller than this value (in degrees).

Number max. per pair
With this parameter you can limit the number of connections. CaronteFX selects the most significant joints. Higher settings create more connections.

Limit number of processed joints
This option restricts the number of joints that are actually processed during simulation. The number of joints does not change, but CaronteFX will choose the ones which are best suited.

Max processed joints per pair
By default, all joints are processed at the same time and this can be a time-consuming task. With this parameter it is possible to “group” the joints to accelerate the simulation. An example: a bone linked to its surrounding flesh. The bone’s vertices are used to create the joints, let’s say 1,000 and “Max processed joints per pair” is set to 50. During calculation all of them will be processed, but only 50 at once.

Disable collisions by pairs
When enabled, CaronteFX stops processing collisions between all body pairs connected by joints. This results in shorter simulation times.

Disable all collisions A-B
The difference to “Disable collisions by pairs” is that nodes from “ObjectsA | B” do not necessarily have to be joined, for example when the distances between them are too big. If you disable collisions between pieces when you already know that they will not interact you will see an increase in simulation speed. An example is a pre-fractured object where you do not need collision between the fragments until it begins to break apart.

Max force (N/m2)
Define the force which is needed to separate the objects. When the acting force is smaller than “Max force (N/m2)” the joint remains intact and the objects are pulled back. With “Maximum” enabled, touching objects cannot become separated, no matter what happens.

Max force rand.
The actual value will be between 0 and the given positive value. The total force is calculated this way:
`Total force = Max force (N) + Max force rand.`

Force range (m)
The “bandwidth” in which the force acts on the joints, given in metres.

Force profile
You can draw a curve to determine the force at different distance points. This way it is possible to define a decreasing range of forces with growing distance, force peaks, etc.

Break if distance exceeded
With the associated “Break distance (m)” you can define a distance at which a joint will break.

Break distance (m)
If the given distance between two joints is exceeded, the connection is broken. The unit is metres.

Break distance random
The actual value will be between 0 and the given positive value. This value is given in metres and the total distance is calculated this way: `Total distance = Break distance + Break distance random`

Break all in pair if few unbroken
It is advised to enable this option if you can see constraint effects between nodes which should be completely separated. An example: Some bricks of a wall stick together, because not all joints are broken. Those hinged bricks can be separated with this option. The broken joints limit is defined under “Unbroken number to break all”.

Unbroken number to break all
If the number of broken joints is reached by a particular pair of linked bodies, all remaining connections between them will be broken.

Break if hinge
There are cases where objects are loosely connected and the joints behave like hinges. Enable this feature to avoid such a behaviour.

Enable collisions if break
When active, objects with broken joints will not interpenetrate. The way, how the collision is handled also depends on “Disable collision by pairs” and “Disable all collisions A-B”:

- If “Disable collision by pairs” is active only collisions between a single “ObjectsA” body and its connected “Objects B” node will be simulated.
- If “Disable all collisions A-B” is active a single “ObjectsA” body is able to collide with all nodes from “ObjectsB”, and vice versa.
- If none of these nodes is active the result is the same as with “Disable all collisions A-B”.

Plasticity
With “Plasticity” you can make fragments stick together as if they were connected with a kind of underlying tissue or grid, while other parts can still leave the assembly.

Plasticity distance (m)
When the distance between two joints exceeds this value the connection between joints becomes irreversible – it behaves like an overstretched rubber band. This value is given in metres.

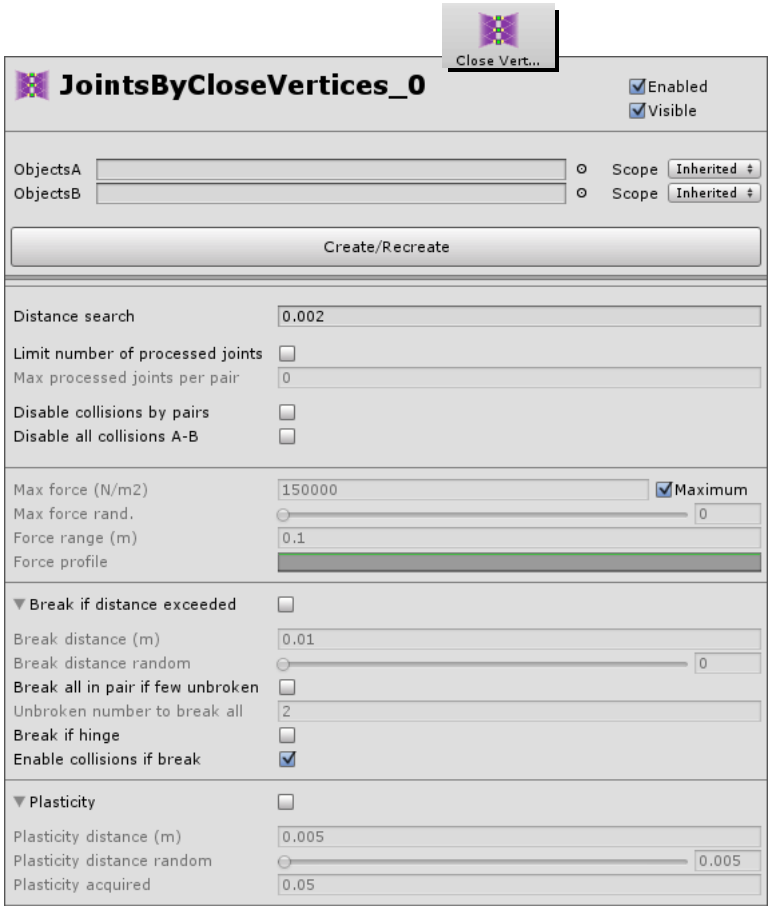
Plasticity distance random
The actual value will be between 0 and the given positive value. This value is given in metres and the total distance is calculated this way:

```
Total distance = @ Plasticity distance + @ Plasticity distance random
```

Plasticity acquired
The parameter’s range goes from 0.0 to 1.0. A setting of 0.5 will keep approximately 50% of a joint’s deformation as permanent. The other half is able to relax and turn back to its initial state.

“Close Vertices” Parameter Reference

The tool looks for the bodies’ vetrtices within a given area and connects them. Rigid, soft, cloth, and irresponsive bodies are supported.



Enabled
Enable or disable the selected body with this checkbox.

Visible
Show or hide the selected body in the viewport.

ObjectsA
Here, the first group of objects is specified you want to connect to a second group, found under “ObjectsB”.

ObjectsB
This is the second group of objects you have to choose to create the links. If you attach exactly the same nodes to “ObjectsA” and “ObjectsB” connections between all selected items will be created. A good example is a brick wall.

Create/Recreate
Establish or recreate the connections according to your settings.

Scope
Only GameObjects within the defined scope will be processed. Others are considered out of reach. There are two options - please read the “CaronteFX Scope” chapters for more information:

- “Global”. All GameObjects are able to interact without restrictions.
- “Inherited”. The scope will be used from the closest CaronteFX parent node of the selected object.

Distance search
With this parameter you can define the distance at which CaronteFX should look for objects to connect. By increasing this value you can look for object pairs which are further away.

Limit number of processed joints
This option restricts the number of joints that are actually processed during simulation. The number of joints does not change, but CaronteFX will choose the ones which are best suited.

Max processed joints per pair
By default, all joints are processed at the same time and this can be a time-consuming task. With this parameter it is possible to “group” the joints to accelerate the simulation. An example: a bone linked to its surrounding flesh. The bone’s vertices are used to create the joints, let’s say 1,000 and “Max processed joints per pair” is set to 50. During calculation all of them will be processed, but only 50 at once.

Disable collisions by pairs
When enabled, CaronteFX stops processing collisions between all body pairs connected by joints. This results in shorter simulation times.

Disable all collisions A-B
The difference to “Disable collisions by pairs” is that nodes from “ObjectsA | B” do not necessarily have to be joined, for example when the distances between them are too big. If you disable collisions between pieces when you already know that they will not interact you will see an increase in simulation speed. An example is a pre-fractured object where you do not need collision between the fragments until it begins to break apart.

Max force (N/m2)
Define the force which is needed to separate the objects. When the acting force is smaller than “Max force (N/m2)” the joint remains intact and the objects are pulled back. With “Maximum” enabled, touching objects cannot become separated, no matter what happens.

Max force rand.
The actual value will be between 0 and the given positive value. The total force is calculated this way:
`Total force = Max force (N) + Max force rand.`

Force range (m)
The “bandwidth” in which the force acts on the joints, given in metres.

Force profile
You can draw a curve to determine the force at different distance points. This way it is possible to define a decreasing range of forces with growing distance, force peaks, etc.

Break if distance exceeded
With the associated “Break distance (m)” you can define a distance at which a joint will break.

Break distance (m)
If the given distance between two joints is exceeded, the connection is broken. The unit is metres.

Break distance random
The actual value will be between 0 and the given positive value. This value is given in metres and the total distance is calculated this way: `Total distance = Break distance + Break distance random`

Break all in pair if few unbroken
It is advised to enable this option if you can see constraint effects between nodes which should be completely separated. An example: Some bricks of a wall stick together, because not all joints are broken. Those hinged bricks can be separated with this option. The broken joints limit is defined under “Unbroken number to break all”.

Unbroken number to break all
If the number of broken joints is reached by a particular pair of linked bodies, all remaining connections between them will be broken.

Break if hinge
There are cases where objects are loosely connected and the joints behave like hinges. Enable this feature to avoid such a behaviour.

Enable collisions if break
When active, objects with broken joints will not interpenetrate. The way, how the collision is handled also depends on “Disable collision by pairs” and “Disable all collisions A-B”:

- If “Disable collision by pairs” is active only collisions between a single “ObjectsA” body and its connected “ObjectsB” node will be simulated.
- If “Disable all collisions A-B” is active a single “ObjectsA” body is able to collide with all nodes from “Objects B”, and vice versa.
- If none of these nodes is active the result is the same as with “Disable all collisions A-B”.

Plasticity
With “Plasticity” you can make fragments stick together as if they were connected with a kind of underlying tissue or grid, while other parts can still leave the assembly.

Plasticity distance (m)
When the distance between two joints exceeds this value the connection between joints becomes irreversible – it behaves like an overstretched rubber band. This value is given in metres.

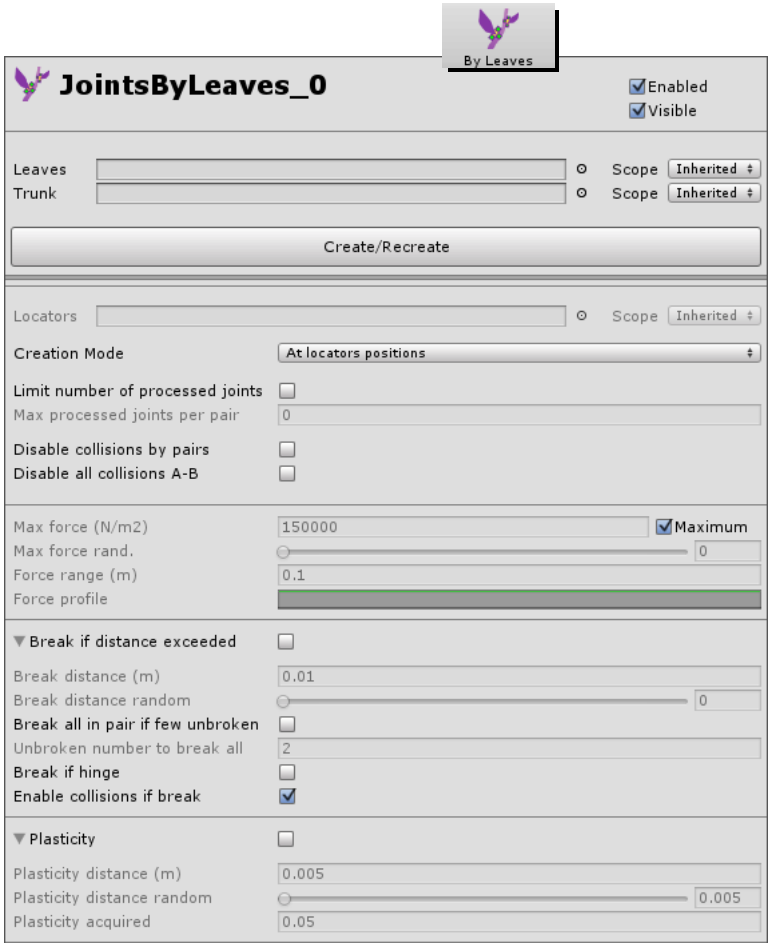
Plasticity distance random
The actual value will be between 0 and the given positive value. This value is given in metres and the total distance is calculated this way:

```
Total distance = @ Plasticity distance + @ Plasticity distance random
```

@ Plasticity acquired
The parameter’s range goes from 0.0 to 1.0. A setting of 0.5 will keep approximately 50% of a joint’s deformation as permanent. The other half is able to relax and turn back to its initial state.

“By Leaves” Parameter Reference

This mode is suitable for plants and similar setups. The leaves have to be very close to the tree’s branches.



Enabled
Enable or disable the selected body with this checkbox.

Visible
Show or hide the selected body in the viewport.

Leaves
The leaf objects have to be added here. Leaves have to be very close to the tree’s branches.

Trunks
The trunk objects have to be added here.

Create/Recreate
Establish or recreate the connections according to your settings.

Scope
Only GameObjects within the defined scope will be processed. Others are considered out of reach. There are two options - please read the “CaronteFX Scope” chapters for more information:

- “Global”. All GameObjects are able to interact without restrictions.
- “Inherited”. The scope will be used from the closest CaronteFX parent node of the selected object.

Locators
Locators determine, where the joints will be created. Under “Creation Mode” you find three methods to connect the objects.

Creation Mode
These options determine how the objects are linked together:

- “At locators positions”. Here you create objects and place them between the objects meant to be linked. The joints will be created at the pivot points of the helper nodes. All locator nodes have to be added to “Locators”.
- “At locators vertices”. With this mode, the locator’s vertex positions are used to connect the two nearest objects from “Leaves” and “Trunks”. Objects will be linked, even if they are far away from the joint.
- “At locators bbox centres”. This mode allows you to use any objects shape as a locator. The geometrical centre of a bounding box around the selected locator(s) is used to create the joints.

Limit number of processed joints
This option restricts the number of joints that are actually processed during simulation. The number of joints does not change, but CaronteFX will choose the ones which are best suited.

Max processed joints per pair
By default, all joints are processed at the same time and this can be a time-consuming task. With this parameter it is possible to “group” the joints to accelerate the simulation. An example: a bone linked to its surrounding flesh. The bone’s vertices are used to create the joints, let’s say 1,000 and “Max processed joints per pair” is set to 50. During calculation all of them will be processed, but only 50 at once.

Disable collisions by pairs
When enabled, CaronteFX stops processing collisions between all body pairs connected by joints. This results in shorter simulation times.

Disable all collisions A-B
The difference to “Disable collisions by pairs” is that nodes from “Objects A | B” do not necessarily have to be joined, for example when the distances between them are too big. If you disable collisions between pieces when you already know that they will not interact you will see an increase in simulation speed. An example is a pre-fractured object where you do not need collision between the fragments until it begins to break apart.

Max force (N/m2)
Define the force which is needed to separate the objects. When the acting force is smaller than “Max force (N/m2)” the joint remains intact and the objects are pulled back. With “Maximum” enabled, touching objects cannot become separated, no matter what happens.

Max force rand.
The actual value will be between 0 and the given positive value. The total force is calculated this way:
`Total force = Max force (N) + Max force rand.`

Force range (m)
The “bandwidth” in which the force acts on the joints, given in metres.

Force profile
You can draw a curve to determine the force at different distance points. This way it is possible to define a decreasing range of forces with growing distance, force peaks, etc.

Break if distance exceeded
With the associated “Break distance (m)” you can define a distance at which a joint will break.

Break distance (m)
If the given distance between two joints is exceeded, the connection is broken. The unit is metres.

Break distance random
The actual value will be between 0 and the given positive value. This value is given in metres and the total distance is calculated this way: `Total distance = Break distance + Break distance random`

Break all in pair if few unbroken
It is advised to enable this option if you can see constraint effects between nodes which should be completely separated. An example: Some bricks of a wall stick together, because not all joints are broken. Those hinged bricks can be separated with this option. The broken joints limit is defined under “Unbroken number to break all”.

Unbroken number to break all
If the number of broken joints is reached by a particular pair of linked bodies, all remaining connections between them will be broken.

Break if hinge
There are cases where objects are loosely connected and the joints behave like hinges. Enable this feature to avoid such a behaviour.

Enable collisions if break
When active, objects with broken joints will not interpenetrate. The way, how the collision is handled also depends on “Disable collision by pairs” and “Disable all collisions A-B”:

- If “Disable collision by pairs” is active only collisions between a single “Objects A” body and its connected “Objects B” node will be simulated.
- If “Disable all collisions A-B” is active a single “Objects A” body is able to collide with all nodes from “Objects B”, and vice versa.
- If none of these nodes is active the result is the same as with “Disable all collisions A-B”.

Plasticity
With “Plasticity” you can make fragments stick together as if they were connected with a kind of underlying tissue or grid, while other parts can still leave the assembly.

Plasticity distance (m)
When the distance between two joints exceeds this value the connection between joints becomes irreversible – it behaves like an overstretched rubber band. This value is given in metres.

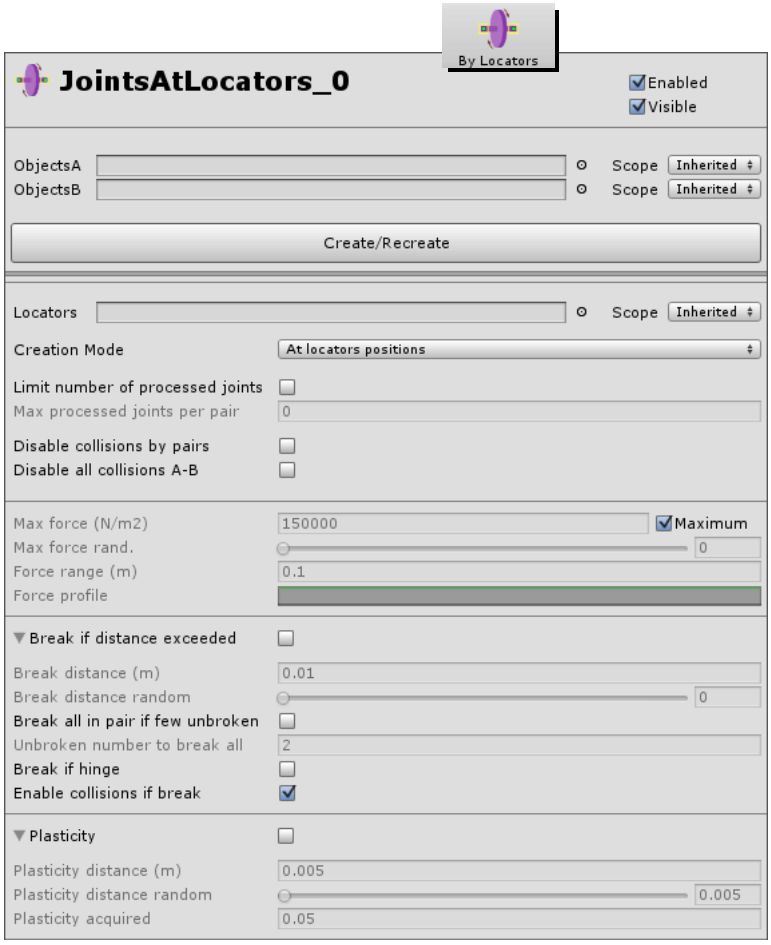
Plasticity distance random
The actual value will be between 0 and the given positive value. This value is given in metres and the total distance is calculated this way:

```
Total distance = @ Plasticity distance + @ Plasticity distance random
```

Plasticity acquired
The parameter’s range goes from 0.0 to 1.0. A setting of 0.5 will keep approximately 50% of a joint’s deformation as permanent. The other half is able to relax and turn back to its initial state.

“At Locators” Parameter Reference

Here you can connect bodies (rigid, soft, cloth, irresponsive) via helper Unity® GameObjects, e.g. cubes or spheres. This tool is perfectly suited if you want to create hinges.



Enabled
Enable or disable the selected body with this checkbox.

Visible
Show or hide the selected body in the viewport.

ObjectsA
Here, the first group of objects is specified you want to connect to a second group, found under “Objects B”.

ObjectsB
This is the second group of objects you have to choose to create the links. If you attach exactly the same nodes to “Objects A” and “Objects B” connections between all selected items will be created. A good example is a brick wall.

Create/Recreate
Establish or recreate the connections according to your settings.

Scope
Only GameObjects within the defined scope will be processed. Others are considered out of reach. There are two options - please read the “CaronteFX Scope” chapters for more information:

- “Global”. All GameObjects are able to interact without restrictions.
- “Inherited”. The scope will be used from the closest CaronteFX parent node of the selected object.

Locators
Locators determine, where the joints will be created. Under “Creation Mode” you find three methods to connect the objects.

Creation Mode
These options determine how the objects are linked together:

- “At locators positions”. Here you create objects and place them between the objects meant to be linked. The joints will be created at the pivot points of the helper nodes. All locator nodes have to be added to “Locators”.
- “At locators vertices”. With this mode, the locator’s vertex positions are used to connect the two nearest objects from “Leaves” and “Trunks”. Objects will be linked, even if they are far away from the joint.
- “At locators bbox centres”. This mode allows you to use any objects shape as a locator. The geometrical centre of a bounding box around the selected locator(s) is used to create the joints.

Limit number of processed joints
This option restricts the number of joints that are actually processed during simulation. The number of joints does not change, but CaronteFX will choose the ones which are best suited.

Max processed joints per pair
By default, all joints are processed at the same time and this can be a time-consuming task. With this parameter it is possible to “group” the joints to accelerate the simulation. An example: a bone linked to its surrounding flesh. The bone’s vertices are used to create the joints, let’s say 1,000 and “Max processed joints per pair” is set to 50. During calculation all of them will be processed, but only 50 at once.

Disable collisions by pairs
When enabled, CaronteFX stops processing collisions between all body pairs connected by joints. This results in shorter simulation times.

Disable all collisions A-B
The difference to “Disable collisions by pairs” is that nodes from “Objects A | B” do not necessarily have to be joined, for example when the distances between them are too big. If you disable collisions between pieces when you already know that they will not interact you will see an increase in simulation speed. An example is a pre-fractured object where you do not need collision between the fragments until it begins to break apart.

Max force (N/m2)
Define the force which is needed to separate the objects. When the acting force is smaller than “Max force (N/m2)” the joint remains intact and the objects are pulled back. With “Maximum” enabled, touching objects cannot become separated, no matter what happens.

Max force rand.
The actual value will be between 0 and the given positive value. The total force is calculated this way:
`Total force = Max force (N) + Max force rand.`

Force range (m)
The “bandwidth” in which the force acts on the joints, given in metres.

Force profile
You can draw a curve to determine the force at different distance points. This way it is possible to define a decreasing range of forces with growing distance, force peaks, etc.

Break if distance exceeded
With the associated “Break distance (m)” you can define a distance at which a joint will break.

Break distance (m)
If the given distance between two joints is exceeded, the connection is broken. The unit is metres.

Break distance random
The actual value will be between 0 and the given positive value. This value is given in metres and the total distance is calculated this way: `Total distance = Break distance + Break distance random`

Break all in pair if few unbroken
It is advised to enable this option if you can see constraint effects between nodes which should be completely separated. An example: Some bricks of a wall stick together, because not all joints are broken. Those hinged bricks can be separated with this option. The broken joints limit is defined under “Unbroken number to break all”.

Unbroken number to break all
If the number of broken joints is reached by a particular pair of linked bodies, all remaining connections between them will be broken.

Break if hinge
There are cases where objects are loosely connected and the joints behave like hinges. Enable this feature to avoid such a behaviour.

Plasticity
With “Plasticity” you can make fragments stick together as if they were connected with a kind of underlying tissue or grid, while other parts can still leave the assembly.

Plasticity distance (m)
When the distance between two joints exceeds this value the connection between joints becomes irreversible – it behaves like an overstretched rubber band. This value is given in metres.

Plasticity distance random
The actual value will be between 0 and the given positive value. This value is given in metres and the total distance is calculated this way:

```
Total distance = @ Plasticity distance + @ Plasticity distance random
```

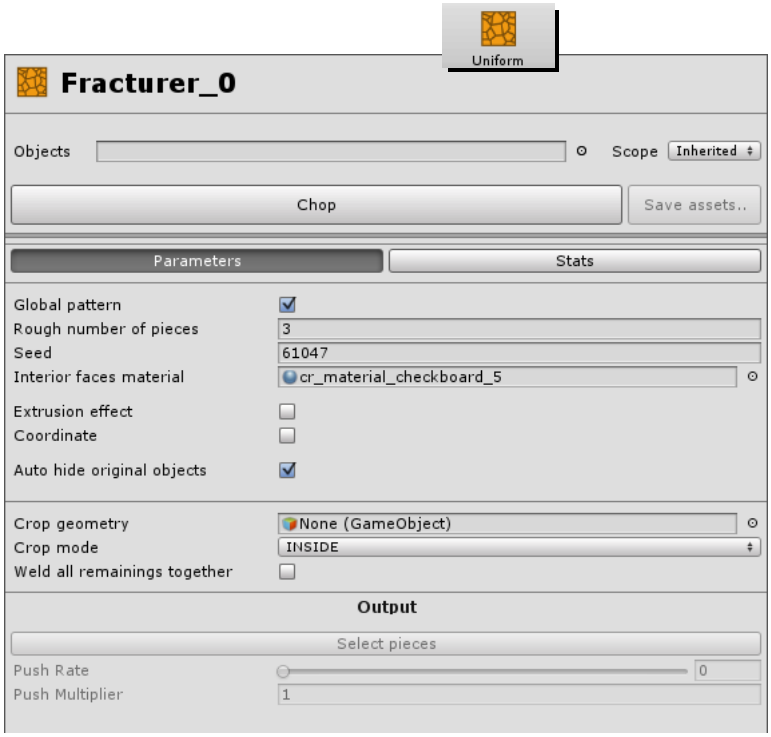
Plasticity acquired
The parameter’s range goes from 0.0 to 1.0. A setting of 0.5 will keep approximately 50% of a joint’s deformation as permanent. The other half is able to relax and turn back to its initial state.

Fractures Reference

Uniform, Geometry, Radial

“Uniform” Parameter Reference

This tool creates a random distribution of fragments.



Objects
This selection specifies the scene’s rigid bodies. There are several ways to choose scene elements, for example:

- Drag and drop Unity® GameObjects from the “Hierarchy” window.
- Enter or copy/paste the name of a GameObject to the field directly

A left-click on the “Objects” field opens a node browser where you can see the attached GameObjects arranged in a list. Here it is also possible to manage your nodes:

- Remove an object by selecting and right-clicking on an entry. Then, choose “Remove”
- Rename an object by selecting and right-clicking on an entry. Then, choose “Rename”
- Drag and drop GameObjects from the “Hierarchy” window.
- Use the entries from the node bowser’s “Edit” menu.

Scope
Only GameObjects within the defined scope will be processed. Others are considered out of reach. There are two options - please read the “CaronteFX Scope” chapters for more information:

- “Global”. All GameObjects are able to interact without restrictions.
- “Inherited”. The scope will be used from the closest CaronteFX parent node of the selected object.

Chop
Press this button to create fragments. You can enter new values and press this button again to update the result.

Global pattern
This option is only relevant in conjunction with multiple objects. When active

- all nodes under “Objects” are treated as one coherent object
- you will see propagating cracks
- the total number of fragments corresponds with “Rough number of pieces”.

Otherwise every object is fractured individually, and the total amount of fragments is
`Rough number of pieces * number of Unity® GameObjects`

Rough number of pieces
Here you can specify how many fragments you want to create. In some cases the resulting number of fragments slightly differs from the input value. You can enter any positive integer greater than 0.

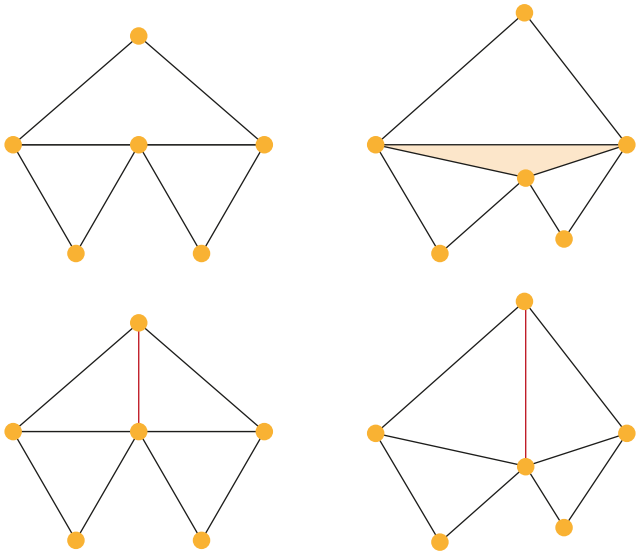
Seed
If you want to achieve a different distribution of fragments change this value – you can enter any positive integer.

Interior faces material
The specified material will applied to the newly created faces of the fractures.

Extrusion effect
With enabled the result is an effect that is similar to the look of a jigsaw puzzle:

- The pieces are cut out like cookies and are not arranged in layers or stacks.
- “Extrusion effect” is useful if you want to punch a hole into a wall or create the effect of a thin breaking pane of glass.

Coordinate
The best way to explain this feature is an illustration:



- Above you see three triangles and their vertices.
- The problem here is that not all of the triangles are connected through common vertices.
- When such a structure is bended, e.g. in a soft body simulation, you most probably see unwanted holes.

To avoid this, CaronteFX restructures the mesh by adding new triangles with shared vertices. Now, the body can be deformed without holes. Of course, the new structure will only be applied to the fragments, not the original object(s).

Auto hide original objects
The fracture tools are non-destructive and the original objects will be kept. When this option is enabled, the original object is hidden.

Crop geometry
Cropping is used to create custom clusters of fragments or increase the number of fragments in user-defined areas with the help of a reference object.

Crop mode
Here you specify how the cropping process will be applied based on the reference object from “Crop geometry”: strictly inside or outside the steering object, or including nodes which are inside the steering object’s bounding box.

Weld all remainings together
When active the remaining, non-cropped fragments will be reconnected to a single piece.

Select pieces
This button makes the freshly created output the current object selection.

Push Rate
By dragging the slider you can separate the pieces. The distance ranges between 0 (touching) and 1 (maximum separation). If you want to create bigger gaps between the objects please use “Push Multiplier”.

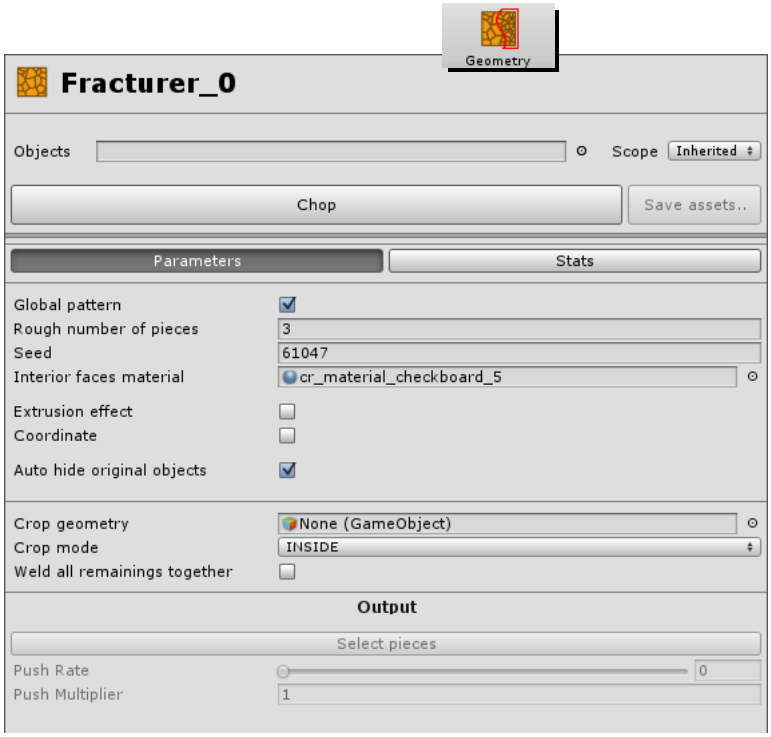
Push Multiplier
Create bigger gaps between the fragments with this slider (please see “Push Rate”).

“Uniform” Stats

This section is for information purposes only. You see the original number of objects, vertices, and faces. Below is a statistics with the results.

“Steering Geometry” Parameter Reference

With this tool a reference object is used to control zones of higher or lower fragment concentration, e.g. for defining impact areas.



Objects
This selection specifies the scene’s rigid bodies. There are several ways to choose scene elements, for example:

- Drag and drop Unity® GameObjects from the “Hierarchy” window.
- Enter or copy/paste the name of a GameObject to the field directly

A left-click on the “Objects” field opens a node browser where you can see the attached GameObjects arranged in a list. Here it is also possible to manage your nodes:

- Remove an object by selecting and right-clicking on an entry. Then, choose “Remove”
- Rename an object by selecting and right-clicking on an entry. Then, choose “Rename”
- Drag and drop GameObjects from the “Hierarchy” window.
- Use the entries from the node bowser’s “Edit” menu.

Scope
Only GameObjects within the defined scope will be processed. Others are considered out of reach. There are two options - please read the “CaronteFX Scope” chapters for more information:

- “Global”. All GameObjects are able to interact without restrictions.
- “Inherited”. The scope will be used from the closest CaronteFX parent node of the selected object.

Chop
Press this button to create fragments. You can enter new values and press this button again to update the result.

Global pattern
This option is only relevant in conjunction with multiple objects. When active

- all nodes under “Objects” are treated as one coherent object.
- you will see propagating cracks
- the total number of fragments corresponds with “Rough number of pieces”.

Otherwise every object is fractured individually, and the total amount of fragments is
`Rough number of pieces * number of Unity® GameObjects`

Rough number of pieces
Here you can specify how many fragments you want to create. In some cases the resulting number of fragments slightly differs from the input value. You can enter any positive integer greater than 0.

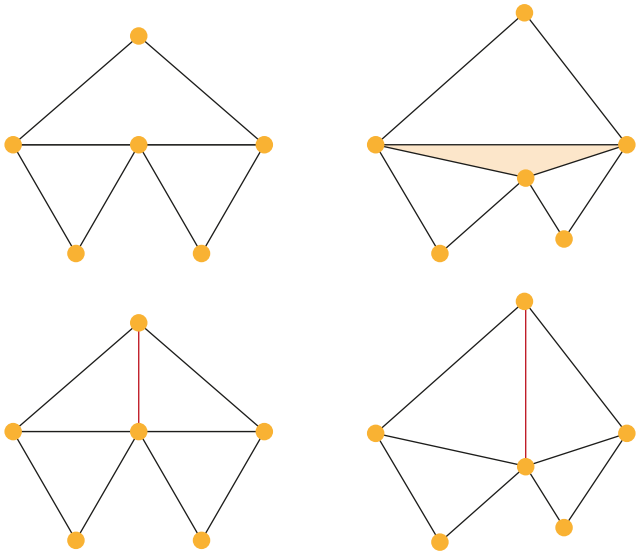
Seed
If you want to achieve a different distribution of fragments change this value – you can enter any positive integer.

Interior faces material
The specified material will applied to the newly created faces of the fractures.

Extrusion effect
With enabled the result is an effect that is similar to the look of a jigsaw puzzle:

- The pieces are cut out like cookies and are not arranged in layers or stacks.
- “Extrusion effect” is useful if you want to punch a hole into a wall or create the effect of a thin breaking pane of glass.

Coordinate
The best way to explain this feature is an illustration:



- Above you see three triangles and their vertices.
- Not all of the triangles are connected through common vertices.
- When such a structure is bended, e.g. in a soft body simulation, you most probably see unwanted holes.

To avoid this, CaronteFX restructures the mesh by adding new triangles with shared vertices. Now, the body can be deformed without holes. Of course, the new structure will only be applied to the fragments, not the original object(s).

Auto hide original objects
The fracture tools are non-destructive and the original objects will be kept. When this option is enabled, the original object is hidden.

Steering geometry
Select an object by clicking on the empty field (“None (GameObject)“). The chosen object is used to indicate the zone with the highest concentration of pieces.

Steering resolution
The steering geometry is rasterized for the fragmentation process and this parameter defines the quality of that process. Higher values will represent the shape of the steering geometry much better.

Highest concentration
According to the selected option the highest concentration of fragments is created “Inside”, “Outside” , “Along the boundary” or “Out of the boundary” of the steering object.

Lowest/Highest concentration rate
This value determines the ratio of the number of fragments in areas with higher and lower fragment concentration. With smaller values the transition between these areas becomes sharper. An example:

- You want to create 100 pieces inside the zone with the highest concentration per fragment in the low concentration area.
- The rate is calculated as 1 / 100 = 0.01
- With 0.001, the ratio is 1: 1000 objects.

Transition length
This parameter affects the transition between zones of higher and lower fragment density:

- With small values, there is a more or less sudden change between the low- and high-density zones.
- Higher values create smoother distribution.
- The value is given in meters and with a value of 3, the transition between the zones will be gradually changed over 3 metres.

Crop geometry
Cropping is used to create custom clusters of fragments or increase the number of fragments in user-defined areas with the help of a reference object.

Crop mode
Here you specify how the cropping process will be applied based on the reference object from “Crop geometry”: strictly inside or outside the steering object, or including nodes which are inside the steering object’s bounding box.

Weld all remainings together
When active the remaining, non-cropped fragments will be reconnected to a single piece.

Select pieces
This button makes the freshly created output the current object selection.

Push Rate
By dragging the slider you can separate the pieces. The distance ranges between 0 (touching) and 1 (maximum separation). If you want to create bigger gaps between the objects please use “Push Multiplier”.

Push Multiplier
Create bigger gaps between the fragments with this slider (please see “Push Rate”).

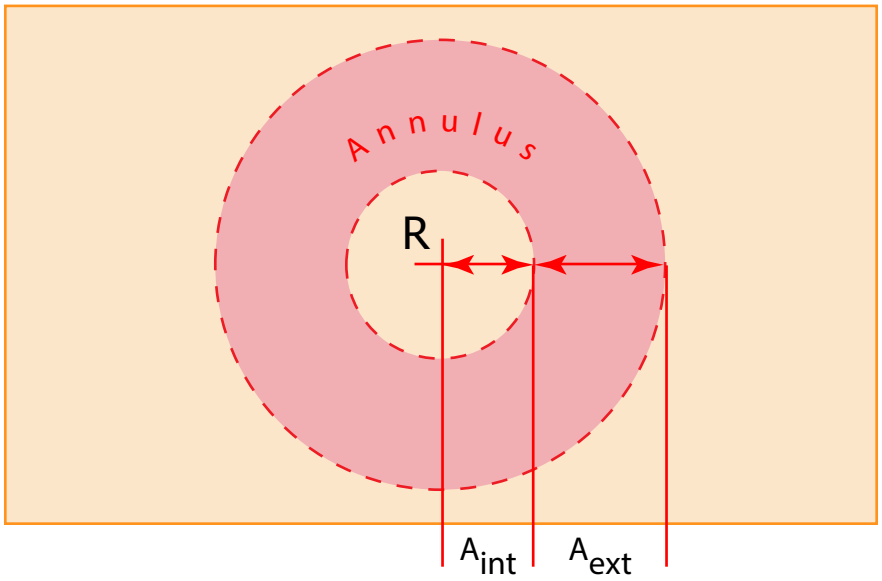
“Steering Geometry” Stats

This section is for information purposes only. You see the original number of objects, vertices, and faces. Below is a statistics with the results.

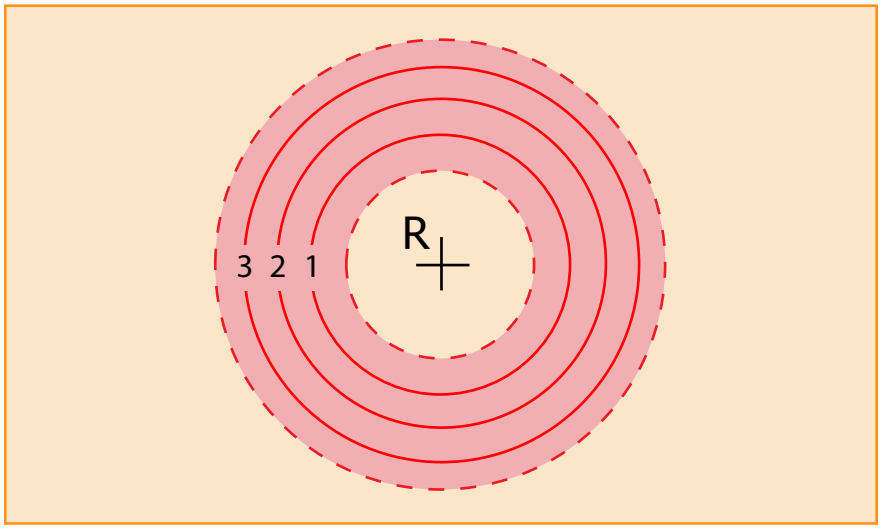
“Radial” Basic Descriptions

With this tool it is possible to create radial patterns, suited to the creation of breaking glass or stucco.

Annulus (red area)
This is the area in which the rings are created, defined through “Annulus internal radius” (**A_{int}**) and “Annulus external radius” (**A_{ext}**).

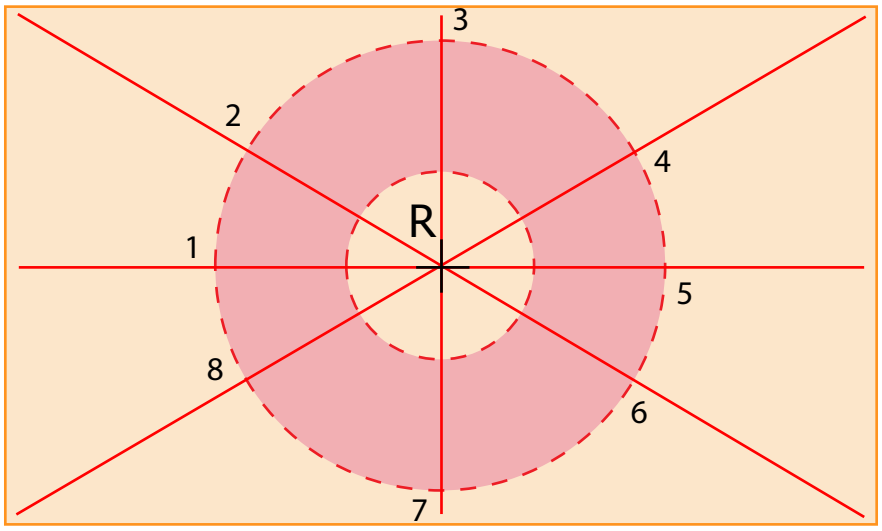


Rings (red circles)
The radial patterns consists of concentric rings around a centre (**R** = “Reference system” object). The numbers represent the “Rings number inside annulus” parameter.



1 - 3 = Rings number inside annulus

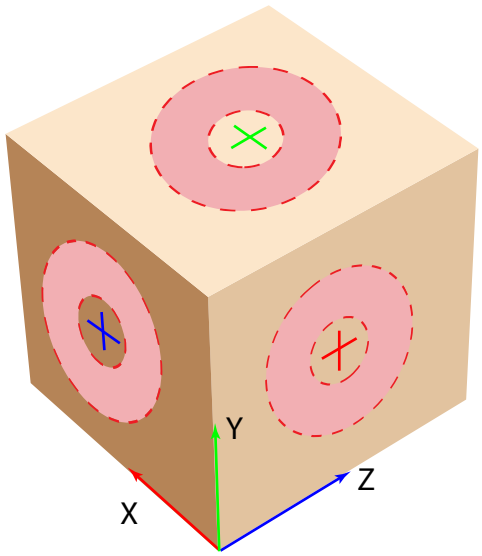
Rays (red lines)
A star-shaped structure that origins in the centre of the radial pattern (**R** = “Reference system” object) and cuts the rings. The numbers represent the “Rays number” parameter.



1 - 8 = Rays number

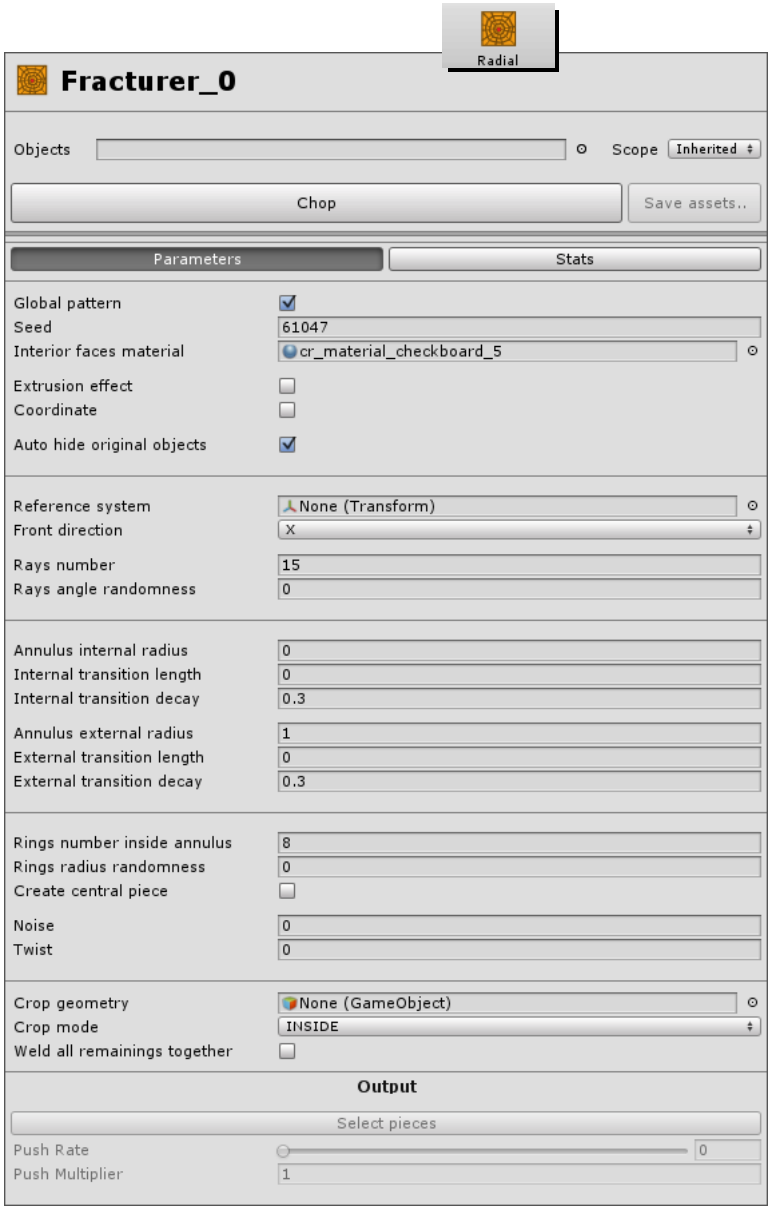
Reference System
The fragmentation pattern is created at the position of a helper object (“Reference system”) and projected on a 2D plane along the source’s X, Y, or Z axis:

- The pattern’s orientation is specified with the “Front direction” options: X, Y, and Z.
- When the source object is rotated the fragmentation pattern will be rotated as well.



“Radial” Parameter Reference

With this tool a reference object is used to control zones of higher or lower fragment concentration, e.g. for defining impact areas.



Objects

This selection specifies the scene’s rigid bodies. There are several ways to choose scene elements, for example:

- Drag and drop Unity® GameObjects from the “Hierarchy” window.
- Enter or copy/paste the name of a GameObject to the field directly.

A left-click on the “Objects” field opens a node browser where you can see the attached GameObjects arranged in a list. Here it is also possible to manage your nodes:

- Remove an object by selecting and right-clicking on an entry. Then, choose “Remove”
- Rename an object by selecting and right-clicking on an entry. Then, choose “Rename”
- Drag and drop GameObjects from the “Hierarchy” window.
- Use the entries from the node bowser’s “Edit” menu.

Scope
Only GameObjects within the defined scope will be processed. Others are considered out of reach. There are two options - please read the “CaronteFX Scope” chapters for more information:

- “Global”. All GameObjects are able to interact without restrictions.
- “Inherited”. The scope will be used from the closest CaronteFX parent node of the selected object.

Chop
Press this button to create fragments. You can enter new values and press this button again to update the result.

Global pattern
This option is only relevant in conjunction with multiple objects. When active

- all nodes under “Objects” are treated as one coherent object.
- you will see propagating cracks (see right image below)
- the total number of fragments corresponds with “Rough number of pieces” (see left image below).

Otherwise every object is fractured individually, and the total amount of fragments is
`Rough number of pieces * number of Unity® GameObjects`

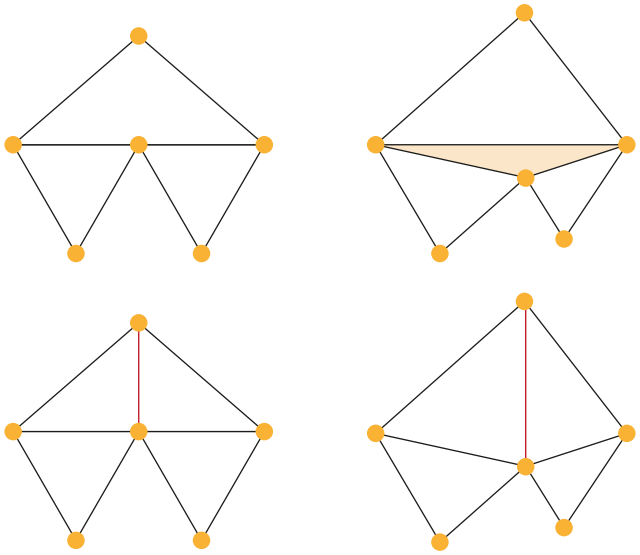
Seed
If you want to achieve a different distribution of fragments change this value – you can enter any positive integer.

Interior faces material
The specified material will applied to the newly created faces of the fractures.

Extrusion effect
With enabled the result is an effect that is similar to the look of a jigsaw puzzle:

- The pieces are cut out like cookies and are not arranged in layers or stacks.
- “Extrusion effect” is useful if you want to punch a hole into a wall or create the effect of a thin breaking pane of glass.

Coordinate
The best way to explain this feature is an illustration:



- Above you see three triangles and their vertices.
- The problem here is that not all of the triangles are connected through common vertices.
- When such a structure is bended, e.g. in a soft body simulation, you most probably see unwanted holes.

To avoid this, CaronteFX restructures the mesh by adding new triangles with shared vertices. Now, the body can be deformed without holes. Of course, the new structure will only be applied to the fragments, not the original object(s).

Auto hide original objects
The fracture tools are non-destructive and the original objects will be kept. When this option is enabled, the original object is hidden.

Reference System
The fragmentation pattern is created at the position of a helper object (“Source”) and projected on a 2D plane along the source’s X, Y, or Z axis:

- The pattern’s orientation is specified with the “Front direction” options: X, Y, and Z.
- When the source object is rotated the fragmentation pattern will be rotated as well.

Front direction
Here you specify the orientation of the radial pattern based on the rotation of the node from “Reference System”. This way it is possible to rotate the web pattern on the surface of the fragmented object.

Rays number
Here you can determine how many rays you want to create.

Ray angle randomness
By default, all rays are created with the same angle between them. If you want to displace the rays enter a number between 0.0 and 1.0 (highest randomness).

Annulus internal radius
This is the inner distance of the annulus from the radial pattern's centre – the centre is specified through the object under "Reference system". The unit is metres.

Internal transition length
Here you determine how far from the "Annulus internal radius" additional rings are created. The unit is metres.

Internal transition decay
Here you determine how fast the rings outside the annulus (measured from the inner boundary of the annulus) are separated. The value ranges between 0 (no decay) and 1 (maximum decay).

Annulus external radius
This is the outer distance of the annulus from the radial pattern's centre – the centre is specified through the object under "Source".

External transition length
Here you determine how far from the "Annulus external radius" additional rings are created. The unit is metres.

External transition decay
Here you determine how fast the rings outside the annulus (measured from the outer boundary of the annulus) are separated. The value ranges between 0 (no decay) and 1 (maximum decay).

Rings number inside annulus
Enter, how many rings you want to create inside the annulus – the area between "Annulus internal radius" and Annulus external radius".

Rings radius randomness
By default, all rings inside the annulus have exactly the same distance from each other; in transition zones, the rings obey the decay parameters – with a value of 0 (= no decay), the rings will be equispaced as well. To avoid this, enter a degree of randomness between 0.0 and 1.0 (maximum randomness).

Create central piece
When enabled a fragment is created directly at the centre of the radial pattern.

Noise
Add a small amount of random displacement to create a more random look. Values range between 0.0 and 1.0 (maximum noise).

Twist
You can also displace the radial pattern to get a look similar to a spiral or the arrangement of sunflower seeds. Valid values are between -1.0 (maximum twist counterclockwise) and 1.0 (maximum twist clockwise). With 0, no twist will be applied.

Crop geometry
Cropping is used to create custom clusters of fragments or increase the number of fragments in user-defined areas with the help of a reference object.

Crop mode
Here you specify how the cropping process will be applied based on the reference object from “Crop geometry”: strictly inside or outside the steering object, or including nodes which are inside the steering object’s bounding box.

Weld all remainings together
When active the remaining, non-cropped fragments will be reconnected to a single piece.

Select pieces
This button makes the freshly created output the current object selection.

Push Rate
By dragging the slider you can separate the pieces. The distance ranges between 0 (touching) and 1 (maximum separation). If you want to create bigger gaps between the objects please use “Push Multiplier”.

Push Multiplier
Create bigger gaps between the fragments with this slider (please see “Push Rate”).

“Radial” Stats

This section is for information purposes only. You see the original number of objects, vertices, and faces. Below is a statistics with the results.

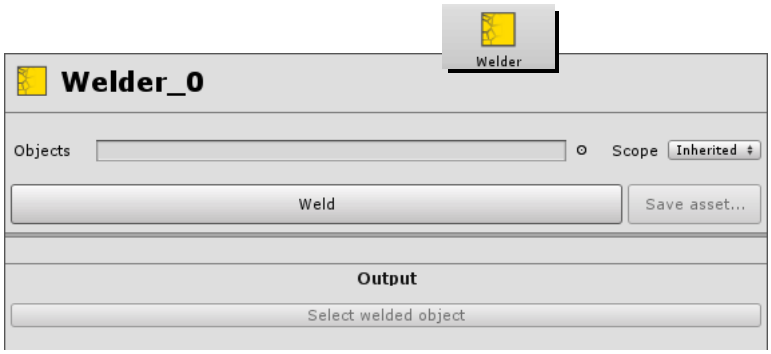
Tools Reference

Welder, Selector, Tesselator, Helper Mesh, etc.

“Welder” Parameter Reference

With this tool it is possible to create clusters of different objects with a single click. This process is non-destructive and the original objects will be kept.

An example:
Let’s say you have fractured a body into pieces and there are some fragments you want to keep together in any case. Simply add the appropriate nodes to the welder’s “Object” field, and click on “Weld”. Please note that the nodes you want to connect do not necessarily have to touch or overlap.



Objects
This selection specifies the scene’s rigid bodies. There are several ways to choose scene elements, for example:

- Drag and drop Unity® GameObjects from the “Hierarchy” window.
- Enter or copy/paste the name of a GameObject to the field directly

A left-click on the “Objects” field opens a node browser where you can see the attached GameObjects arranged in a list. Here it is also possible to manage your nodes:

- Remove an object by selecting and right-clicking on an entry. Then, choose “Remove”
- Rename an object by selecting and right-clicking on an entry. Then, choose “Rename”
- Drag and drop GameObjects from the “Hierarchy” window.
- Use the entries from the node bowser’s “Edit” menu.

Scope
Only GameObjects within the defined scope will be processed. Others are considered out of reach. There are two options - please read the “CaronteFX Scope” chapters for more information:

- “Global”. All GameObjects are able to interact without restrictions.
- “Inherited”. The scope will be used from the closest CaronteFX parent node of the selected object.

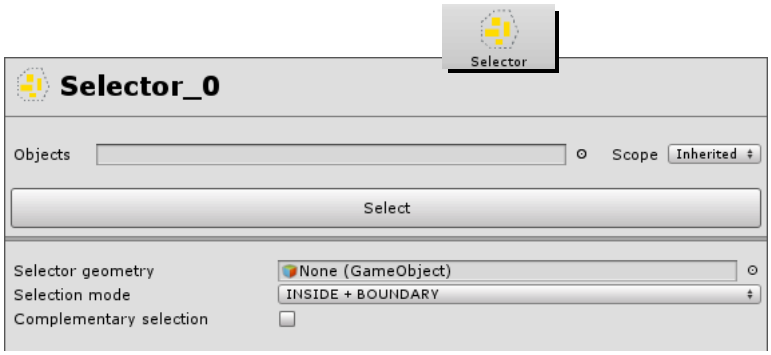
Weld
Click on this button to trigger the welding process.

Save asset...
The freshly created object can be saved as an asset with this button.

Select welded object
This button makes the freshly created output the current object selection.

“Selector” Parameter Reference

This tool is used to select fragments with the help of a reference object and weld them together.



Objects
Imagine a fractured cube. Now you want to select and weld some of the inner pieces to create a core that should not break into fragments. To do this, just add a small sphere, for example, to select the area where the core should be created.

Here, under objects, all the fragments of the cube are added. CaronteFX will find the pieces inside the sphere (or better: the reference object under “Selector geometry”) automatically and weld them.

There are several ways to choose scene elements, for example:

- Drag and drop Unity® GameObjects from the “Hierarchy” window.
- Enter or copy/paste the name of a GameObject to the field directly.

A left-click on the “Objects” field opens a node browser where you can see all GameObjects arranged in a list. Here it is also possible to manage your nodes:

- Remove an object by selecting and right-clicking on an entry. Then, choose “Remove”
- Rename an object by selecting and right-clicking on an entry. Then, choose “Rename”
- Drag and drop GameObjects from the “Hierarchy” window.
- Use the entries from the node bowser’s “Edit” menu.

Scope
Only GameObjects within the defined scope will be processed. Others are considered out of reach. There are two options - please read the “CaronteFX Scope” chapters for more information:

- “Global”. All GameObjects are able to interact without restrictions.
- “Inherited”. The scope will be used from the closest CaronteFX parent node of the selected object.

Select
When you press this button CaronteFX will start the selection and welding process. It is possible to update your results after changes with another click on this button.

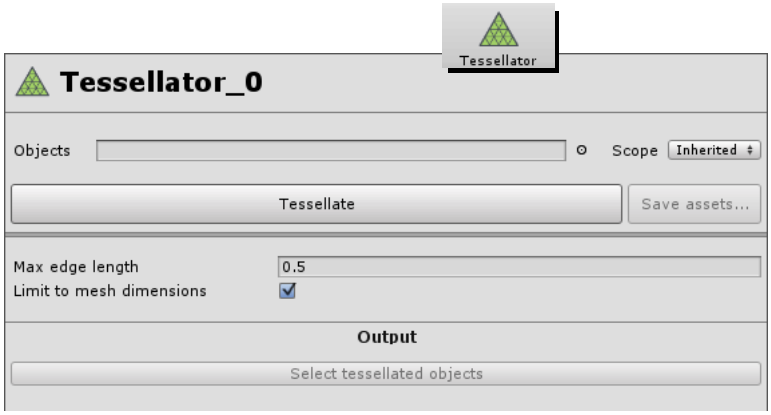
Selector geometry
This is the Unity® GameObject you want to use as reference or locator to select and weld the fragments from “Objects”.

Selection mode
- “Inside”. You can select only the pieces wich are exactly inside the selector geometry.
- “Inside + Boundary” You can select pieces wich are inside the selector geometry and are overlapping with the reference object’s surface.

Complementary selection
Invert you current selection of objects.

“Tesselator” Parameter Reference

Especially soft bodies profit from high-resolution meshes, because this body type is deformed at the triangles’ vertices. Simulation results become better and better with a increasing number of triangles. This tool allows you to subdivide an object.



Objects
This selection specifies the scene’s rigid bodies. There are several ways to choose scene elements, for example:

- Drag and drop Unity® GameObjects from the “Hierarchy” window.
- Enter or copy/paste the name of a GameObject to the field directly.

A left-click on the “Objects” field opens a node browser where you can see all GameObjects arranged in a list. Here it is also possible to manage your nodes:

- Remove an object by selecting and right-clicking on an entry. Then, choose “Remove”
- Rename an object by selecting and right-clicking on an entry. Then, choose “Rename”
- Drag and drop GameObjects from the “Hierarchy” window.
- Use the entries from the node bowser’s “Edit” menu.

Scope
Only GameObjects within the defined scope will be processed. Others are considered out of reach. There are two options - please read the “CaronteFX Scope” chapters for more information:

- “Global”. All GameObjects are able to interact without restrictions.
- “Inherited”. The scope will be used from the closest CaronteFX parent node of the selected object.

Tesselate
Press this button to subdivide the object(s). You can enter new values and press this button again to update the result.

Save asset...
The freshly created object can be saved as an asset with this button.

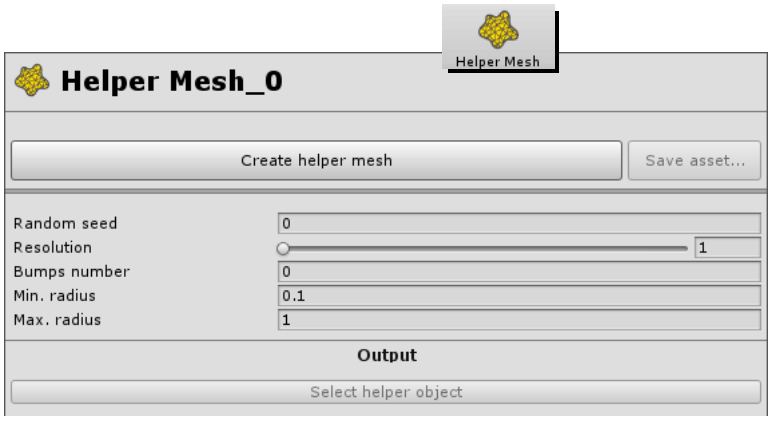
Max edge length
This value is measured in metres and accepts any positive value. The smaller the value, the more triangles you will get.

Limit to mesh dimensions
With this option it is possible to avoid excessive tesselation, because the tool's effect will be limited to the bounding box of the object you want to subdivide.

Select tessellated object
The objects which have been modified with this tool can be selected with a click on this button.

“Helper Mesh” Parameter Reference

Helper meshes can be used as cropping objects. e.g. for CaronteFX’s fracture tools, collision objects, or as steering geometry for the “Geometry” fracture tool.



Create helper mesh
When you create a mesh from the default parameters you will get a sphere. To change the mesh’s shape, modify the available parameters, and press the button again.

Save asset...
The freshly created object can be saved as an asset with this button.

Random seed
When you change “Random seed” you will get different shapes and variations from the same parameters. The parameter’s effect is only visible with a “Bumps number” greater than 0.

Resolution
The parameter ranges from 1 to 3 (= maximum resolution) and is used to control the mesh’s number of triangles.

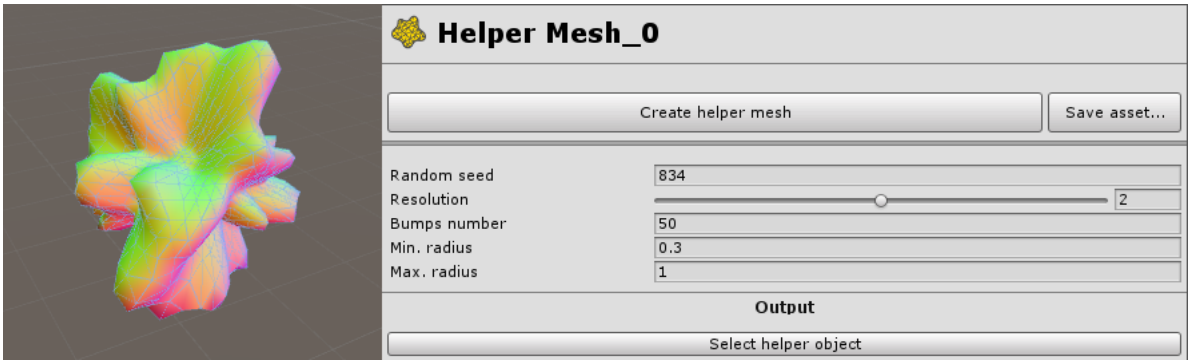
Bumps number
A helper mesh consists of seamlessly blended spheres of different size. The complexity of the mesh is controlled with number. If you want to change the size of the spheres change “Min. | Max. radius”.

Min. radius
A helper mesh consists of seamlessly blended spheres of different size. The size of the smallest sphere is controlled with this parameter (see next page for example images).

Max. radius
A helper mesh consists of seamlessly blended spheres of different size. The size of the biggest sphere is controlled with this parameter (see next page example images).

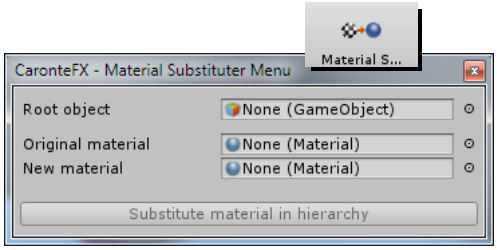

Helper Mesh_0
Create helper mesh
Save asset...
Random seed
53
Resolution
2
Bumps number
1
Min. radius
0.1
Max. radius
1
Output
Select helper object


Helper Mesh_0
Create helper mesh
Save asset...
Random seed
6544
Resolution
2
Bumps number
122
Min. radius
0.8
Max. radius
1
Output
Select helper object


Helper Mesh_0
Create helper mesh
Save asset...
Random seed
834
Resolution
2
Bumps number
50
Min. radius
0.3
Max. radius
1
Output
Select helper object

“Material Substituter” Parameter Reference

This is tool allows you to replace a material in an entire hierarchy of Unity® GameObjects. You select the root object, and the tool will replace the original material (can be empty) through a new material (cannot be empty) inside the specified hierarchy.



Root object
Here, the highest ranking object is being added. A typical example is the root object with fragments. The materials of the nodes under the root object will be substituted.

Original material
If you want to replace a specific material drag it to this slot, but the field can remain empty.

New material
Drag the material you want to apply to the nodes under the root object to this field. Please bear in mind that this slot cannot be empty.

Substitute material in hierarchy
Click on this button to start the replacement process.

Servos Reference

Linear & Angular Servos, Linear & Angular Motors

Motors & Servos Introduction

With the help of servos and motors you can accelerate and move objects in a physically correct manner.

A car is a good example, because everybody is familiar with its driving behaviour and has a basic idea of the concepts behind the technology. In a car, a motor is (normally) powered by fuel. The burning of the fuel creates energy that can be transferred to a drive system. This system will finally accelerate the car and set the vehicle in motion. The reason why the car can actually move is friction. On an even and polished surface, the car's wheels would simply slip. You have certainly already seen this on icy or wet streets.

Servos and motors can mimic this behaviour and set any object in motion with physically correct acceleration and even braking forces. Servos and motors can also react to changes of direction and interact with the ground texture. e.g. a stony, uneven ground.

Motors & Servos Types

A system with servos and motors always consists of at least two elements: two bodies linked via a servo/motor node. This is already the entire setup and the objects can be either irresponsive or active, have different shapes or size or are absolutely identical. In the following descriptions we will only discuss a two-body system, because it is much easier to understand, but everything said here is also valid for three or more objects.

In this system we have body A with active rigid body dynamics properties and body B in irresponsive mode. Object B is located in the scene's origin. Normally, all targets refer to object B's local system, but in this special case we can use the world coordinate system for convenience.

When we talk about a moving body we have to look at the performed kind of motion. Does it move along a certain path or rotate around a hub? The first case is also called linear motion and this is something you experience when you ride a bike, drive a car or go by train. In the second case, the motion is a rotation, e.g. the blades of a fan. Although all these motions appear differently, there have two things in common:

To get a body in motion, a force is needed and this force depends on the body's mass and the amount of friction ("Static friction" and "Kinetic friction") the object experiences

The second similarity is that a moving body always has a certain velocity. The object's speed is the result of acceleration. You can observe acceleration processes in daily life, for example when a train leaves a station after a halt. At the beginning, the motion is very slow, but becomes faster and faster until the body has reached its final (or target) velocity. From this moment, the object does not have to be accelerated anymore. With rotations it is the same. When you switch on a fan, it also takes some time until the blades have reached their maximum speed.

Until now, our object performs a motion at a certain velocity, but that is only one part of the story, because the body should stop when it has reached a certain point in space or lower its speed at least. The process is called braking or deceleration. The strength of the braking process again depends on a force that must be high enough to counteract the body's motion and slow it down.

With these everyday examples, we have covered the entire servos and motors system:

- **Motor** Velocity Linear. When you use this node, the object tries to reach a certain target velocity.
- **Motor** Velocity Angular. In this case, the target is the angular velocity, or more commonly, the amount of degrees a body covers within a certain time-span.
- **Servo** Position Linear. The object tries to achieve a certain position. The resulting velocity depends on the adjusted forces, powers, mass and friction values, as well as other parameters.
- **Servo** Position Angular. Here, it is slightly different, because a mounted object cannot reach a given position, so in this case we are talking about angular positions, e.g. orientations.

Target vs. Target

One thing that has to be clarified is the "target" term, because, so far, it has been used in different ways. We have talked about target positions and target velocities, but what does that actually mean for a simulation?

The target position is the point in space the objects want to reach. A body's target velocity, on the other hand, is the speed the bodies want to acquire. All target position and velocity settings are made under the MultiServo's "Target" panel and there you have to specify three values for X, Y and Z. The following table shows you the different meanings of the "Target" values:

- **Motor** Velocity Linear. The target velocity the object should achieve in metres per second.
- **Motor** Velocity Angular. The target rotation speed the object should achieve in degrees per second.
- **Servo** Position Linear. The target distance the object should cover in metres.
- **Servo** Position Angular. The targeted amount of degrees the object should cover.

Please remember that these explanations are simplified and describe a two-body system with two objects: A and B. All targets refer to the object B's local system instead of the world system. In the case that object B is located in the scene's origin, the world system can be used. To get a better idea of these relations, please take a look at following examples:

In the first scenario we have a car (rigid body) in front of a house (irresponsive body). The car should move away from the house in X direction with a velocity of 8 m/s. Here, the car is object A and the house is object B – the reference node. The "Target" values of the used "Servo Velocity Linear" are [8.0, 0.0, 0.0].

The second scene contains a vase with dry flowers (irresponsive body) and a candle (rigid body). Both objects have a certain distance to each other. Now you want to avoid the candle's flame igniting the flowers. For this purpose, the candle should be moved away from the vase by 0.2 m in X direction.

The vase acts as the reference object here and is tagged as object B, while the candle is object A. To perform the desired motion, a "Servo Position Linear" is used with a "Target" of [0.2, 0.0, 0.0].

As you can see from this example, it is important which node acts as object A or B, because each action will be performed relative to object B.

Physical Interaction of Bodies Linked by a MultiServo

Another interesting thing with the MultiServos system is how the coupled bodies move. Since the objects are physically related, they react on each other's behaviour. As a consequence, Servos and motors allow you to simulate realistic phenomena like recoiling effects. For example, if both elements are free to move and you have entered a target distance of 20 m in X direction then each object will move 10 m – but only if they have exactly the same properties (mass, object friction, air friction etc.).

If they are different, CaronteFX will automatically calculate which distance each object will cover. If one object is fixed, the movable node will cover the entire 20 m. With velocities it is similar and both nodes "share" the target velocity accordingly.

Forces, Torques, Power

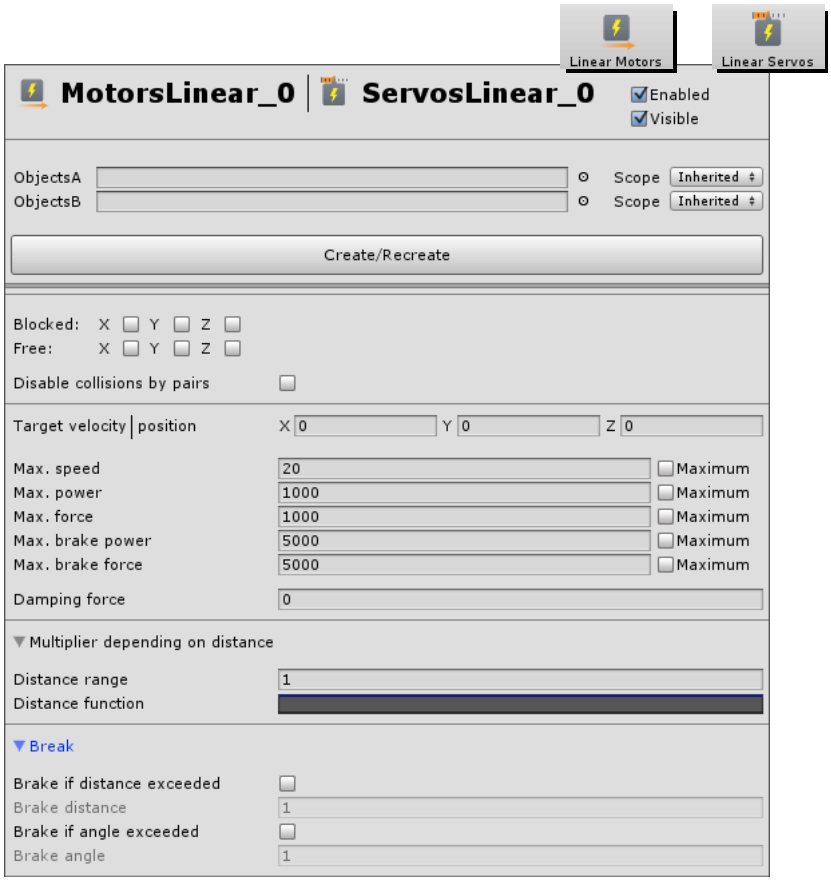
In order to accelerate the bodies, a force is required and if the motion is circular, we refer to torques. A torque is what you apply when you open a pickle jar or the cap of a bottle, and is measured in Newton metres [Nm]. The unit of forces is Newton [N].

Another parameter you can see is "Power" and it is given in Watts [W]. Power plays an important role as well. Power is responsible for how fast a current state can be changed in order to achieve the required target. The following example helps you to understand the role power plays with servos and motors: Imagine a motor with a very high force, but low power. In this case, the force creates a linear displacement, but this change of position is carried out slowly. When you use the same force as before and increase power to a very high value, the displacement is carried out immediately.

When you simulate, CaronteFx starts up in order to produce the change you want with the conditions you set. However, once the goal is achieved, it is necessary to stop and not go beyond it. This is why our motor also needs values for force and power to break. For example, imagine you want to move an object a certain distance. You start your motor and begin to push the object away, increasing the covered distance. Once you have covered exactly the distance you wanted, you have to brake, to not go too far from your final goal. If you do not have enough force or power to break immediately, the object will miss its target position and tries to go back to it again.

“Motors Linear” & “Servos Linear” Parameter Reference

Here you can choose the objects that will be used to establish a setup and how they will be linked. The order in which nodes will be grouped in “ObjectsA” or “ObjectsB” is important and makes a difference, because the connection between the nodes is based on relative positions.



ObjectsA

Here you can specify the first node group. When you click on the parameter’s input field, a node browser will be opened so you can select the desired objects.

ObjectsB

The mode of operation is exactly the same as with “ObjectsA”. “ObjectsB” are different from “ObjectsA” in that their local coordinate system is used to interpret the “Target” value(s) of the servo. The motor will be created between each “ObjectsA” and its closest “ObjectsB”.

Create/Recreate

Press this button to create the motor/servo or to update it after a change of parameters.

Blocked X | Y | Z
These parameters depend directly on “Free [X|Y|Z]” and can influence each other. When “Blocked [X|Y|Z]” is set to “Yes”, the motor/servo will act with infinite force and power on the appropriate axis. With “No”, forces and power will be limited to your force settings. When an axis is blocked, it cannot act freely anymore.

A blocked axis tells the motor/servo to act in any case. Blocking defines in which direction a motion must be performed. Let’s say you want to mimic the behaviour of a shock absorber. In such a case, a car’s wheels should not be displaced horizontally in relation to the car, but you want to allow vertical displacement. This can be done by creating a “Linear Servo” node between the wheels and the car body, which has the horizontal axes blocked and the vertical axis neither blocked nor free.

With this setup the wheels can move horizontally with car, preventing them to be detached from it, but they can also move vertically according to the specified maximum forces and powers of the absorber. Now the wheels will be able to perform an up-and-down movement when the cars (and its components) move over bumpy ground.

Free X | Y | Z
All of these parameters have exactly the same functionality: if set to “Yes”, the node’s axis can act completely freely. Another case is when an object’s axes are neither free nor blocked. Here, the motor’s/servo’s influence will be limited to the adjusted forces and powers.

A rising balloon is a nice example for free objects. In this case you want the balloon to perform a motion along a vertical axis, but the motion in horizontal directions should not be influenced by the motor’s/servo’s forces – only by wind or collisions with other objects. To achieve this behaviour, a “Linear Servo” node will be used with a target in the vertical axis (Y or Z – this depends on your scene setup). To allow horizontal movement, the appropriate axes (“Free X” and “Free Z”/”Free Y”) must be set to “Yes”.

Disable collision by pairs
By default (unchecked), collisions between the connected body pairs are taken into consideration.

Target velocity (Motors) | Target position (Servos)
A target consists of three coordinates for each spatial direction: X, Y and Z represent the target velocity the object should achieve given in metres per second (motors) or metres (servos).

Max speed
This is the maximum relative speed between the linked objects that can be reached in order to approach the target position. This value is measured in metres per second. Please ensure that your “Max speed” value is not lower than the target speed, or else the target will never be achieved.

Max power
A motor/servo needs a certain amount of power to accelerate an object. You can define a maximum value here, given in Watts.

Max force
This parameter can be used to limit the forces between linked nodes when they are accelerated and measured in Newtons.

Max brake power
Here you can define the maximum brake power that can be applied by the motor/servo in order to slow down the object. As a power, this value is given in Watts.

Max brake force
This is the counterpart to “Max force” and used to limit the maximum brake force that can act on an object. Like any other force, this one is also measured in Newtons.

Damping force
In reality, forces can be weakened because of many unspecific influences. With this parameter it is possible to mimic this effect and create more realistic and natural motion. The unit for this parameter is Newton seconds per meter.

Multiplier depending on distance
When checked it is possible to the maximum force and power depending on the distance between the objects.

Distance range
The “bandwidth” in which force and power act on the objects, given in metres.

Distance function
You can draw a curve to determine the force/pwoer at different distance points. This way it is possible to define a decreasing range of forces and powers with growing distance, force peaks, etc.

Break if distance is exceeded
The default setting is “No” . In this mode the motor/servo rig remains intact and working. When the threshold under “Break distance” is reached or exceeded, the motor/serov connection will be released and stop working.

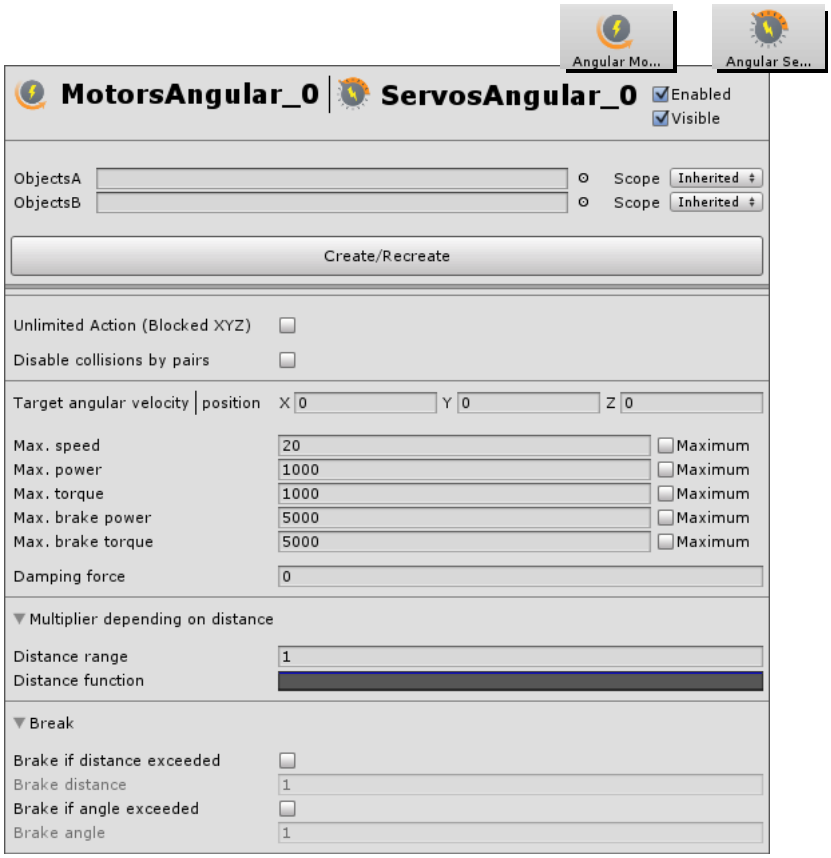
Break distance
Here you define the distance at which a motor/servo breaks and becomes inactive. “Break distance” is measured in meters.

Break if angle exceeded
The mode of operation is exactly the same as with “Break if distance exceeded”, but here the decisive criteria is the angle between the target and the reference object.

Break angle
Define an angle to break the motor/servo connection. This value is given in degrees.

“Motors Angular” & “Servos Angular” Parameter Reference

Here you can choose the objects that will be used to establish a setup and how they will be linked. The order in which nodes will be grouped in “ObjectsA” or “ObjectsB” is important and makes a difference, because the connection between the nodes is based on relative positions.



ObjectsA
Here you can specify the first node group. When you click on the parameter’s input field, a node browser will be opened so you can select the desired objects.

ObjectsB
The mode of operation is exactly the same as with “ObjectsA”. “ObjectsB” are different from “ObjectsA” in that their local coordinate system is used to interpret the “Target” value(s) of the servo. The motor will be created between each “ObjectsA” and its closest “ObjectsB”.

Create/Recreate
Press this button to create the motor/servo or to update it after a change of parameters.

Unlimited Action (Blocked XYZ)
When “Unlimited Action (Blocked XYZ)” is set to “Yes”, the motor/servo will act with infinite force and power on the appropriate axis. With “No”, forces and power will be limited to your force settings. This tells the motor/servo to act in any case. Blocking means that a motion must be performed.

Disable collision by pairs
By default (unchecked), collisions between the connected body pairs are taken into consideration.

Target angular velocity (Motors) | Target angular position (Servos)
A target consists of three coordinates for each spatial direction: X, Y and Z represent the target velocity the object should achieve given in degrees per second (metres) or degrees (servos).

Max speed
This is the maximum relative speed between the linked objects that can be reached in order to approach the target position. This value is measured in metres per second. Please ensure that your “Max speed” value is not lower than the target speed, or else the target will never be achieved.

Max power
A motor/servo needs a certain amount of power to accelerate an object. You can define a maximum value here, given in Watts.

Max force
This parameter can be used to limit the forces between linked nodes when they are accelerated and measured in Newtons.

Max brake power
Here you can define the maximum brake power that can be applied by the motor/servo in order to slow down the object. As a power, this value is given in Watts.

Max brake force
This is the counterpart to “Max force” and used to limit the maximum brake force that can act on an object. Like any other force, this one is also measured in Newtons.

Damping force
In reality, forces can be weakened because of many unspecific influences. With this parameter it is possible to mimic this effect and create more realistic and natural motion. The unit for this parameter is Newton seconds per meter.

Multiplier depending on distance
When checked it is possible to the maximum force and power depending on the distance between the objects.

Distance range
The “bandwidth” in which force and power act on the objects, given in metres.

Distance function
You can draw a curve to determine the force/pwoer at different distance points. This way it is possible to define a decreasing range of forces and powers with growing distance, force peaks, etc.

Break if distance is exceeded
The default setting is “No” . In this mode the motor/servo rig remains intact and working. When the threshold under “Break distance” is reached or exceeded, the motor/serov connection will be released and stop working.

Break distance
Here you define the distance at which a motor/servo breaks and becomes inactive. “Break distance” is measured in meters.

Break if angle exceeded
The mode of operation is exactly the same as with “Break if distance exceeded”, but here the decisive criteria is the angle between the target and the reference object.

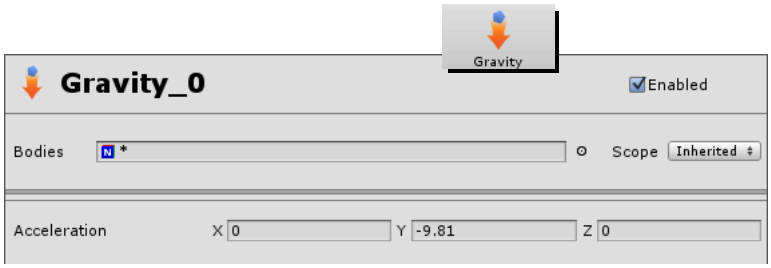
Break angle
Define an angle to break the motor/servo connection. This value is given in degrees.

Daemons Reference

Gravity, Explosion, Wind

“Gravity” Parameter Reference

In plain words, gravity makes things fall. Gravity is location-dependent and bodies experience different gravity on Earth, the Moon, or Mars. With CaronteFX it is possible to specify any direction.



Enabled
Enable or disable the daemon’s influence with this checkbox.

Bodies
The asterisk (*) means that all bodies in the scene will be affected by this daemon. If you want to restrict the daemon to certain objects click on the field and make your selection in the appearing window.

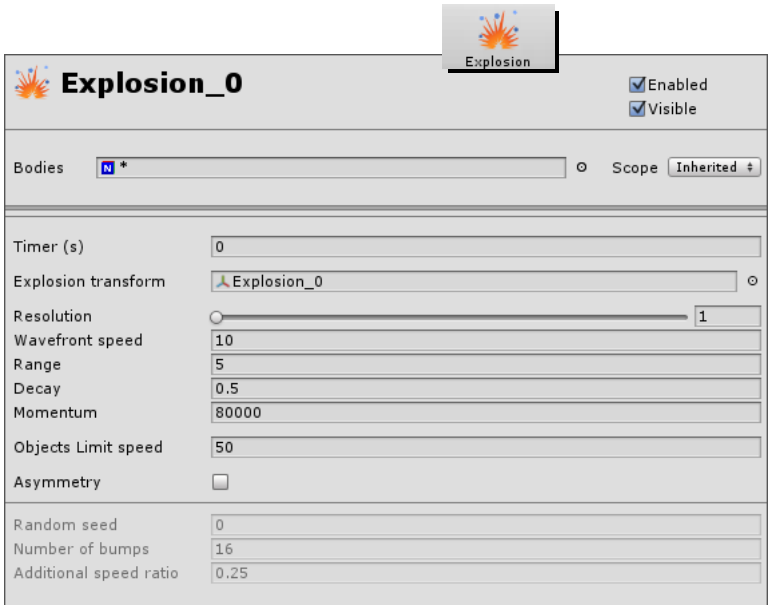
Scope
Only GameObjects within the defined scope will be processed. Others are considered out of reach. There are two options - please read the “CaronteFX Scope” chapters for more information:

- “Global”. All GameObjects are able to interact without restrictions.
- “Inherited”. The scope will be used from the closest CaronteFX parent node of the selected object.

Acceleration
In 3D space, forces always consist of three coordinates – they define the force’s direction and strength. The default value of -9.81 m/s^2 indicates that gravity is oriented downwards along the scene’s vertical axis.

“Explosion” Parameter Reference

Add an explosion to your scene, destroy objects, and cause some chaos.



Enabled
Enable or disable the daemon’s influence with this checkbox.

Visible
Show or hide the daemon’s viewport gizmo

Bodies
The asterisk (*) means that all bodies in the scene will be affected by this daemon. If you want to restrict the daemon to certain objects click on the field and make your selection in the appearing window.

Scope
Only GameObjects within the defined scope will be processed. Others are considered out of reach. There are two options - please read the “CaronteFX Scope” chapters for more information:

- “Global”. All GameObjects are able to interact without restrictions.
- “Inherited”. The scope will be used from the closest CaronteFX parent node of the selected object.

Timer (s)
Define a time in seconds when the explosion should happen.

Explosion transform
The explosion’s centre is set to the reference object’s midpoint. Please note that the reference object’s animation/dynamic motion is not considered.

Resolution
Accepted values range from 0 to 3 and define the explosion's "quality".

Wavefront speed
This parameter can be considered the explosion's "exposure". With smaller values, the objects will be exposed to the explosion for a longer time and this results in a stronger motion. With high values, the explosion's wavefront is very fast and the effect vanishes quickly. All positive values are accepted.

Range
Here, the explosion's effect range given in metres. The viewport gizmo changes according to your settings, indicating the range. Objects beyond this point will not be affected.

Decay
The parameter is used to "focus" the explosion and you can enter values between 0 and 1. With 1.0, for example, you see a very strong effect in a narrow area around the centre of the explosion. You will also notice a quick loss of energy with increasing distance from this centre. With smaller values, the explosion's range or scope is wider. "Decay" affects the viewport gizmo.

Momentum
This parameter can be seen as the explosion's strength. Higher values will create more destruction, and affected objects will be moved stronger.

Objects Limit speed
Affected objects can become very fast. To avoid this it is possible to set a speed limit here measured in metres per second.

Asymmetry
When disabled the explosion acts with exactly the same strength in all directions. This might lead to a uniform look. To get more natural results, activate the checkbox and control the explosion's behaviour with the unlocked parameters.

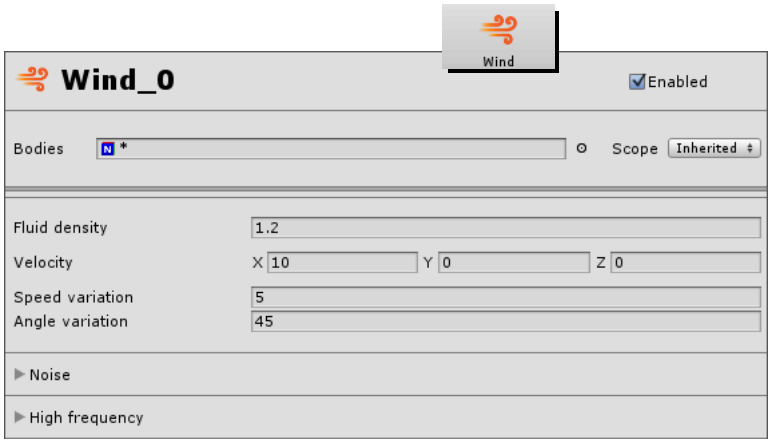
Random seed
Seed values are used to get different results from equal starting conditions.

Number of bumps
Bumps can be seen as explosion zones with varying strength. Especially with lots of objects, the impact of these bumps becomes better visible, because the objects/fragments are accelerated differently.

Additional speed ratio
An explosion's wavefront is not uniformly distributed, but there are zones with higher velocity. The speed of these "bumps" is controlled with this parameter: `Wavefront speed * Additional speed ratio`

“Wind” Parameter Reference

Wind is an important natural phenomenon and CaronteFX’s daemon provides a very realistic wind model with many features.



Enabled
Enable or disable the daemon’s influence with this checkbox.

Bodies
The asterisk (*) means that all bodies in the scene will be affected by this daemon. If you want to restrict the daemon to certain objects click on the field and make your selection in the appearing window.

Scope
Only GameObjects within the defined scope will be processed. Others are considered out of reach. There are two options - please read the “CaronteFX Scope” chapters for more information:

- “Global”. All GameObjects are able to interact without restrictions.
- “Inherited”. The scope will be used from the closest CaronteFX parent node of the selected object.

Fluid density
The default value represents the density of air. With 0, wind will not have any affect at all, because this value represents a perfect vacuum. Higher settings can be used to mimic rooms filled with smoke or fog, lower settings, e.g. 0.7 0 .9 are suited for simulating high altitudes.

Velocity
In 3D space, forces always consist of three coordinates – they define the force’s direction and overall strength. Each value is given in metres per second.

Speed variation
In nature, wind does not blow with constant velocity and there is always a certain amount of randomness. Variation is given in metres per second.

Angle variation
Wind direction is not constant and there is always a certain amount of randomness. Variation is given in degrees.

Temporal period
CaronteFX's noise functions create a pattern that is repeated over time. With 0 the noise pattern remains constant during simulation. With higher settings, the noise pattern will change faster, but it will also be repeated in shorter cycles.

Spatial period
Here the overall size or frequency of the noise in space is controlled. Be aware that high values lead to more noise.

Amplitude rate
This function adds more detail to the wind by adding a substructure, a structure that be imagined like small bumps. The size, or better: height, of these bumps is `Amplitude rate * Velocity`. The number of bumps, on the other hand, is controlled with "Speed up".

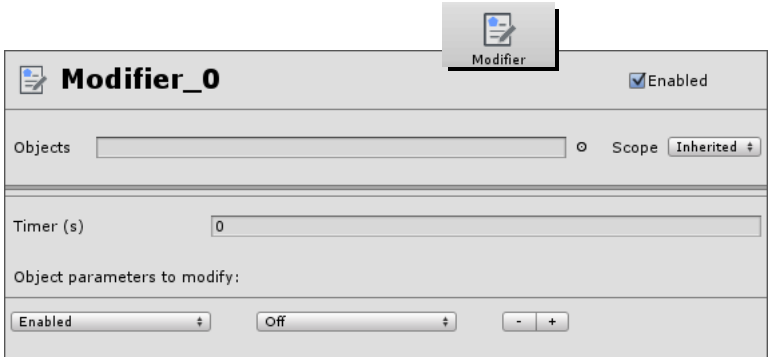
Speed up
Here you control the amount of detail you want to add to the wind daemon. Please also see "Amplitude rate" for more information.

Actions Reference

Contact Emitter, Trigger by Time, Explosion, etc.

“Modifier” Parameter Reference

This action works like a switch. You choose a property and define a time when it will be turned on, off, or when it will be flipped.



Enabled
Enable or disable the tool’s influence with this checkbox.

Objects
Click on the field, choose an entry from “Selectable Nodes”, and shift them with “<<” to the left column. When you close the window the nodes appear under “Attentive Nodes”. To remove a node, click on the empty field again, select one or more entries from the left column and hit “Del”.

Scope
Only GameObjects within the defined scope will be processed. Others are considered out of reach. There are two options - please read the “CaronteFX Scope” chapters for more information:

- “Global”. All GameObjects are able to interact without restrictions.
- “Inherited”. The scope will be used from the closest CaronteFX parent node of the selected object.

Timer (s)
Define a time in seconds when the action/modifaction should be executed.

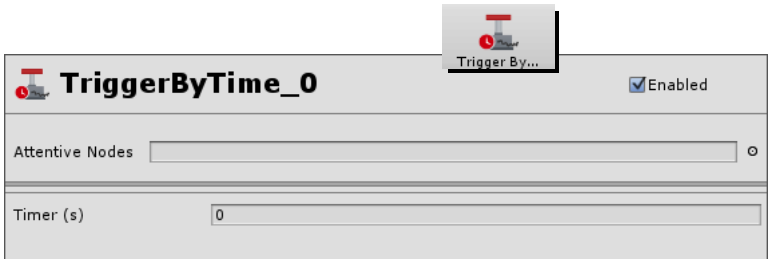
Object parameters to modify
The first menu (default = “Enabled”) contains several objects properties and you can choose one option to be switched or modified. Once you have selected a property, the second field will be changed accordingly:

- “Enabled” set to “Off” will disable the node(s).
- “Velocity linear”, for example, allows you to define a certain velocity in metres per second.
- “Force multiplier” will amplify the affected daemons.
- “Plasticity” is used to make a soft body permanently deformed on demand.

With the “+” and “-” buttons it is possible to add more switches.

“Trigger By Time” Parameter Reference

This action works like a timer. Once the adjusted simulation is reached the attached nodes will be activated.



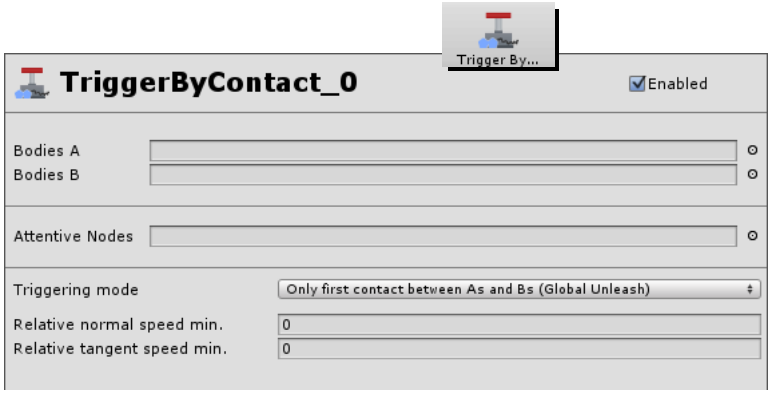
Enabled
Enable or disable the tool’s influence with this checkbox.

Attentive Nodes
The empty field accepts daemons only and contains the nodes you want to activate. Click on the field, choose an entry from “Selectable Nodes”, and shift them with “<<” to the left column. When you close the window the nodes appear under “Attentive Nodes”. To remove a node, click on the empty field again, select one or more entries from the left column and hit “Del”.

Timer (s)
Here, the simulation time is adjusted when the nodes will be activated. Please bear in mind that the “Explosion” daemon has its own timer and is not affected by this action.

“Trigger By Contact” Parameter Reference

An action can be activated when objects collide.



Enabled
Enable or disable the tool’s influence with this checkbox.

Bodies A | Bodies B
For a contact-driven action two objects are required. When the nodes from “Bodies A” collide with the nodes from “Bodies B” the action will be executed.

The empty fields accept bodies only: “Rigid”, “Soft”, etc. Click on the field, choose an entry from “Selectable Nodes”, and shift them with “<<” to the left column. When you close the window the nodes appear under “Bodies A | Bodies B”. To remove a node, click on the empty field again, select one or more entries from the left column and hit “Del”.

Attentive
The empty field accepts daemons only and contains the nodes you want to activate. The selection of nodes works in exactly the same way as described under “Bodies A | Bodies B”.

Triggering mode
There are two options available:

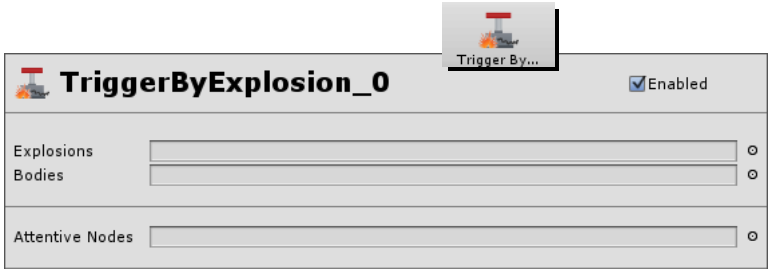
- “Only first contact between As and Bs (Global Unleash)”
- “Each first contact of each pair (Partial Unleash)”

Relative normal speed min.
If the relative normal speed between “Bodies A” and “Bodies B” is greater than the given value the event will be triggered.

Relative tangent speed min.
If the relative tangetial velocity between “Bodies A” and “Bodies B” is greater than the given value the event will be triggered.

“Trigger By Explosion” Parameter Reference

Actions can also be activated through explosions.



Enabled
Enable or disable the tool’s influence with this checkbox.

Explosions
The empty field accepts explosions daemons only. Click on the field, choose an entry from “Selectable Nodes”, and shift them with “<<” to the left column. When you close the window the nodes appear under “Attentive Nodes”. To remove a node, click on the empty field again, select one or more entries from the left column and hit “Del”.

Bodies
When the explosion’s wavefront reaches the bodies, specified here, the event will be triggered. The selection of nodes works in exactly the same way as described under “Explosions”.

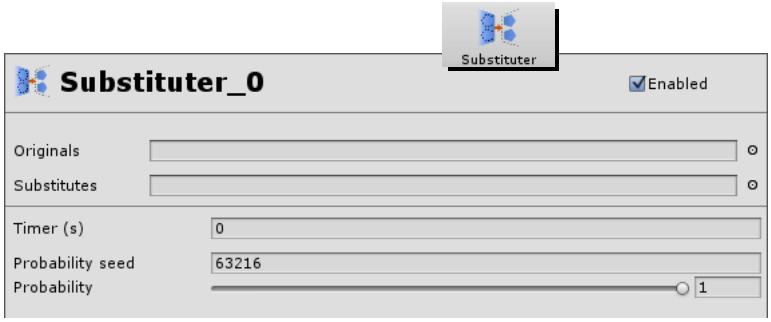
Attentive Nodes
The empty field accepts daemons only and contains the nodes you want to activate. The selection of nodes works in exactly the same way as described under “Explosions”.

“Substituter” Parameter Reference

The substituter will automatically handle the change of visibility, and enable the states of the originals and the substitutes at the moment of the substitution. The substitutes inherit position, velocity and shape (in the case of soft bodies and cloth objects) of the originals.

This tool is closely connected to the “Trigger By Contact” node, because this last one can modify the substitution behavior: depending on the triggering mode of the connected “Trigger By Contact” node it is possible to perform make complete substitutions (substitute all bodies in the “Originals” field – Global Unleash) or partial substitutions (only substitute the bodies triggering – Partial Unleash).

Please read the “Substituter & Trigger Basics” chapter for an introduction and an example.



Enabled
Enable or disable the tool’s influence with this checkbox.

Originals
Click on the field, choose an entry from “Selectable Nodes”, and shift them with “<<” to the left column. When you close the window the nodes appear under “Original”. To remove a node, click on the empty field again, select one or more entries from the left column and hit “Del”.

Substitutes
Here you add the nodes you want to use as replacements. Click on the field, choose an entry from “Selectable Nodes”, and shift them with “<<” to the left column. When you close the window the nodes appear under “Substitutes”. To remove a node, click on the empty field again, select one or more entries from the left column and hit “Del”.

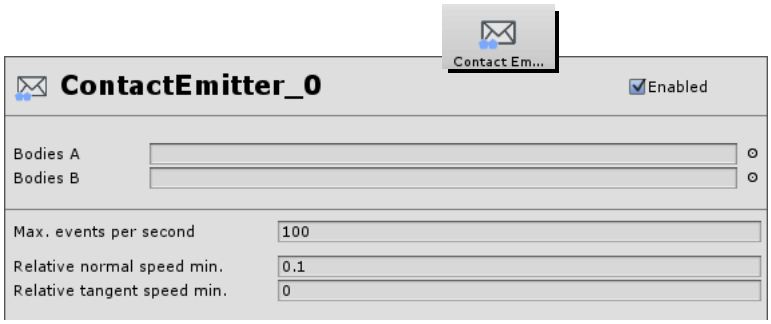
Time (s)
Here, the simulation time is adjusted when the nodes will be substituted.

Probability seed
Seed values help to get different results from equal starting conditions.

Probability
This parameter randomizes the substitution process in scenes with multiple object. With 0, none of the originals will be replaced, with 0.5 approximately 50% will be substituted, and with 1.0 all of the original nodes will replaced (100%).

“Contact Emitter” Parameter Reference

Contact events contain information about position, velocities and momentum of the bodies involved. These events can be included to the baked files when the option is selected in the bake menu. The contact events are useful to synchronize the baked animations with sounds or particle effects.



Bodies A | Bodies B
For a contact-induced action two objects are required. When the nodes from “Bodies A” collide with the nodes from “Bodies B” the action will be executed.

The empty fields accept bodies only: “Rigid”, “Soft”, etc. Click on the field, choose an entry from “Selectable Nodes”, and shift them with “<<” to the left column. When you close the window the nodes appear under “Bodies A | Bodies B”. To remove a node, click on the empty field again, select one or more entries from the left column and hit “Del”.

Max. events per second
The number of events per second can be limited here.

Relative normal speed min.
If the relative normal speed between “Bodies A” and “Bodies B” is greater than the given value the event will be triggered.

Relative tangent speed min.
If the relative tangetial velocity between “Bodies A” and “Bodies B” is greater than the given value the event will be triggered.

CaronteFX for Unity®

http://www.nextlimit.com