

C4 - An object destruction plug-in set for Lightwave 7.5b+

Created by Sebastian Goetsch & Nick Hatzichristos

C4:Destruct

Shatter Controls

Use the Shatter module to break up your object in solid pieces. It's a recursive, random operation primarily based on the "iterations" setting. Every iteration results in roughly twice the number of pieces. You can see a rough estimate of the resulting pieces as you adjust the setting. If "Iterations" is set to more than 8, multi-pass shattering is automatically enabled.

Controls :

Enable Shatter : Enables / Disables module

Shater in 3D / 2D : Object can be broken in 3D or 2D - a 2D shattering is a lot like using the Knife tool in one of the viewports, only it still produces solid pieces. 2D shattering is usually needed for "flat" objects such as windows, walls, etc.

2D plane : Used when shattering in 2D, selects the plane the breaking up will be performed on.

Iterations : Each iteration roughly doubles the number of resulting pieces. See the rough estimate of the resulting pieces right under this control.

Cuts Resolution : This controls the detail of the mesh of the cuts surfaces. The higher the setting, the more detail in the cuts meshes. Higher values will result in a slower operation.

Resolution Type : Can be set to Absolute ("Abs") or Relative ("Rel"). When set to Absolute, the detail of the cuts meshes will be the same in each iteration, resulting in roughly the same detail for even the smallest pieces. If set to Relative, the user-defined resolution setting will be used for the first iteration, and as pieces are cut into smaller pieces, their cuts resolution will be adjusted to each piece's size relatively to the original, uncut object's size.

Cuts Roughness : Controls how "rough", or "jittered" the cuts will be. Higher values work best with high enough "Cuts Resolution" values.

Cuts Surface : The surface name of the inner cuts polygons.

Material Controls

Enable Physical Properties : Enable this to have C4 : Destruct calculate mass centers for each fragment and (optionally, look below) mass information (in Kg) for each segment too. This information is used by C4 : Detonate in Layout for better simulation of forces / rotations / etc.

Apply Mass Value Vmap : This determines if the actual mass of each fragment will be calculated, along with the mass centers. No reason to enable this if you 're not planning on using the map, since the mass calculation does take extra time to complete.

named : Optionally enter a different name for the Mass Value Vmap. Default is "mass".

Material / Density : A drop-down list of materials, setting each one's real-world density value. Helps to more accurately simulate the chosen material, by enabling "Apply Mass Value Vmap" and choosing the right density from the list. You can also enter a density manually directly in the value field.

Output Controls

The Output Module takes a shattered object and saves each piece as a separate object file. Output shattered pieces if you intend to use plugins such as Impact or FX Break - these systems require each piece to be a self-contained object. You do not have to output pieces if you will be using C4's native system for animation, since it requires "reassembled" geometry (the result of the Shatter module).

Controls :

Enable Output : Enables / Disables module

Base name : Select a base name for the exported piece objects sequence

Merge points : Merge points individually on each piece prior to saving it out

Center Pivots : Move each piece's pivot point to the center of its volume. Recommended for easier selection when pieces are loaded in Layout

Triple Polygons : Triple polygons individually on each piece prior to saving it out.

Add air polygons : This adds "air polygons" on each piece. You can also enter a name for the air surface. For proper results, "Merge Points" must be enabled too. The module will automatically enable it if "Add air polygons" is selected.

Shrink : This will shrink each piece by a user-defined amount. This may be useful for dynamics systems such as Impact, where pieces should not touch each other in their initial positions.

- Abs(olute) will shrink all pieces equally by the specified amount, regardless of each one's size.
- Rel(ative) will shrink each piece by a percentage of its size.

Options

Shatter Process Monitoring : Best used with an Advanced or Expert Alert Level. With this setting on, the shatter module will inform you of the various phases it goes through while performing a shattering operation.

Enable Warnings : If enabled, C4 will warn you about things like losing your buffer and give you a last chance to cancel the operation.

Clean up on abort : With this setting enabled, aborting an operation will cause C4 to purge all geometry created by that time, and revert to the original object. If disabled, all geometry created by the time the operation was aborted will be kept.

Tips

Closed Volume

- The object to be shattered has to be a closed volume. Open surface meshes will give unpredictable results, and may even bring the operation to a halt with a code error.

Cuts Resolution

- Although more is obviously better in general, keep in mind that it's easy to get carried away and use a lot more than actually needed. If you're shattering an object into a lot of pieces (9 iterations - at least 512 pieces for example) you probably only need a pretty low cuts resolution. Even a setting of 1 can be enough, depending on the situation.
- Also, Cuts Roughness is directly dependent on the Cuts Resolution setting. Don't expect to notice any roughness if you've set a Cuts Resolution of 1-2, for example.

Semi-Random Shattering

- The shattering algorithm is random, the exact results can not be predicted. However, if you have an irregular shape you want to shatter, one way to semi-control the operation is to manually cut it in a couple of pieces first (perhaps according to its form, thickness, etc.), put the pieces in different layers and run C4 on them individually.

Multiple Meshes / Single Layer

If you have an object that is made up of several individual meshes (for example 4 balls in the same layer - that's still considered one object) it's probably a good idea to separate the meshes before shattering. The same is true (and even more so) if there is intersecting geometry in the same layer.

Known Issues

Physical Properties application is too slow. Currently looking into ways to speed it up. Might be improved in later versions.

If you get the message "...add small offset to the data and retry", chances are you have some of your object's polygons lying entirely and perfectly on a world axis plane (i.e. XY, YZ or XZ). This is an extremely rare case in real world situations, but I found it can happen when you're trying the plugin out for the first time, probably quickly creating boxes and stuff... :)

if you find a bug in C4 : Destruct, report it, and it will be added to the list of "known issues", and of course be on its way to being squashed !

C4:Detonate

Auto Update : If enabled the trajectories and collisions are recalculated whenever the frame slider moves from a time =initial time.(This is especially useful if you hit the play button and want to change parameters while LW plays the scene)

Update Fragment Motion : the manual version of Auto Update

Update Geometry : Press that button after you have modified anything in Modeler and after selecting a Mass Map or Mass Distribution Map

Explosion Controls

Center : Select an object that represent the center of the explosion. By default the exploding object itself is the center. If the selected center is a Null, you can add the C4_ShowRadius custom handler to that Null to see the explosion radius in Layout

Initial Time/As Frame : Defines when the explosion starts. If 'As Frame' is enabled the value is a frame number, otherwise seconds.

Radius : All Fragments inside the radius are "hit" by the explosion and will fly on their own trajectory. Fragments outside the radius are transformed like the base object. A Fragment can be hit only once.

Invert Radius : If enabled fragments outside the explosion radius are hit and those inside are not.

Power : Defines the initial velocity of fragments hit by the explosion. A fragment with mass = 2.5 which is hit with power = 10.0 has an initial velocity of $10.0 / 2.5 = 4.0$ m/s Max.

Duration : Collisions are only detected in the interval [Initial Time, Initial Time + Max. Duration]. This value is always interpreted as seconds!

Fragments Controls

Vertical / Horizontal Attenuation : Scale all fragments initial velocity vectors. If you think for example the fragments are flying to high at the beginning, simply reduce the y-component of all fragments velocity by entering 50% in the Vertical Attenuation field. Negative attenuation will increase the velocity.

Mass Distribution Map : Select a weight map that modifies the fragments centers of mass. By default the center of mass is the center of the axis aligned bounding box (AABB) OR if you enable physical properties application in C4 : Explode in Modeler there will be a custom 3D-VMAP named "centercoords" that contains one entry per fragment describing the centers position. Either way there's a default center of mass for each fragment. You can modify this default center using the weight map: If you want to shift the center towards a particular point, give that point a map value >0%. A map value of 100% means this point is the new center of mass. If several points have map values the center will be interpolated. Points that have a map value of 0.0 or no value at all are not considered in the calculation.

Mass Map : Select a weight map that describes fragment masses (resistance). If you enable mass map calculation in C4 : Explode in Modeler, physically accurate information will be stored in a user-named Vmap ("mass" by default), so all you have to do is select it here. If you want to manually define the mass for a particular fragment, select one of it's points and apply the mass value. Values are absolute and are in kilograms (Kg). Make sure that none of the other points in this fragment have a map value applied, setting their values to 0 is not sufficient. If no map is selected the default mass for all fragments is 1.0 A fragment with mass 0.0 is not affected by the explosion !

Min./Max. Angular Velocity (AV): At the beginning of the explosion each fragment receives a rotation about a random axis with a random velocity between Min. and Max. AV. The values are interpreted as °/s.

Restrict Trajectory / XYZ : If enabled the initial direction for all fragments is the vector XYZ. The length of XYZ is ignored, so you dont have to worry about normalization.

Min. / Max. Trajectory Distortion (TD) : Distort the initial velocity of all fragments by a random angle between Min. and Max. TD

Collision Controls

Ground Collision / Level : Enable collision with the plane Y = Ground Level. Make sure that the fragments are located above ground level when they are hit by the explosion. Otherwise collision detection might fail.

Collision Object : see hints

Vertical / Horizontal Elasticity : Define how strong a fragment bounces from the ground or collision object.

Angular Vel. Attenuation : Define how much the angular velocity is reduced after a collision.

Min. / Max. Bounce Velocity (BV) : If a fragments velocity after a collision is <Min. BV or >Max. BV the object will not bounce.

Sliding Friction : Define how much a sliding fragment is slowed down by friction. (This is an experimental feature and might not always work as expected)

Other Controls

Gravity XYZ : Enter the gravity vector here

Time Warp : Controls the speed of the whole explosion.

1.0 : everything runs in normal Layout time

2.0 : everything runs twice as fast

0.0 : freeze

2.0 : double speed backwards

etc.

You can blend between different values using the envelope button.

Time Warp is also available as stand-alone ItemMotionHandler:
<http://www.informatik.hu-berlin.de/~goetsch/TW/>

Motion Blur Warp : Controls the time for motion blur passes.

0% : motion blur time is warped like the standard frame times

100% : motion blur is warped back to original time

If you want to freeze the animation but keep the motion blur of the normal motion, enter 100% here.

Randseed : All random values are derived from the seed.

Adopt Object Motion : If enabled a fragment's velocity resulting from object rotation/translation (at the time it is hit by the explosion) is added to the initial velocity

Center Evaluation : Decrease the interval if your center object moves very fast and not all fragments in it's trajectory are hit. Increase iterations if two fragments start moving at the same time though they are obviously hit by the explosion at different times. (This will not happen as long as you stay in the usual spatial and temporal dimensions)

Hints

An object can have only one instance of C4 applied ! If you think you need several explosion centers, move the one center very fast to "simulate" the others.

Like in SG_Explosion2, a fragment consists of all points that are connected via polygons. (The fragment recognition algorithm is a lot faster now).

Collision objects : must consist of triangles only - should consist of as less faces and points as possible - polys should be bigger than the fragments they are colliding with - should not be animated, the object geometry is evaluated (scale, rotation, translation, no bones, no morph) at the initial time and any changes afterwards will have no effect on the collision detection.

This is not a rigid body dynamics simulation ! The more complex your collision object is, the more likely the collision detection will fail. A fragment's rotation has no influence on it's bounce direction.

Fragments without mass map have a default mass of 1.0. If you want a fragment to be unaffected by the explosion, use a mass map value of 0.0

If you want to see the explosion radius in Layout, apply the C4_Showradius custom handler to the null object that defines the center. If you can't see the radius you probably have not selected the null as center object before.

The number of explosions in a scene is limited to 2000.

Several explosions can have the same center object, but you will only see one radius in Layout (if the C4_ShowRadius CustomObjectHandler is enabled)

The initial trajectories are computed in the following order :

1. Compute vector V from the center object to the fragment's center of mass OR IF trajectory is restricted THEN use the vector specified in X,Y,Z
2. Rotate V about a (perpendicular) random vector with "Min. Trajectory Distortion" <= random rotation angle <= "Max. Trajectory Distortion"
3. Scale V according to "Power"
4. Divide V by fragment mass
5. IF "Adopt Object Motion" is enabled THEN approximate the fragments velocity (resulting from object rotation/translation) when it's hit by the explosion and add it to V
6. Multiply V.x by (1 - "Horizontal Attenuation")
Multiply V.y by (1 - "Vertical Attenuation")
Multiply V.z by (1 - "Horizontal Attenuation")

If you "Enable Physical Properties" in C4 : Destruct in Modeler, a custom (3D) vmap named "centercoords" is automatically applied. It contains one entry per fragment describing the center of mass in object (local) coordinates. If you move the object in Modeler after this calculation, these vmap values are not changed, so the coordinates won't be correct *. In Layout the plugin automatically looks for this map. If it exists, the points of a fragment are scanned until a point with a vmap entry is found. The mass map is evaluated the same way. The first point with a map value is used for the mass of the fragment. Other points with map values in the fragment are ignored. Since you can't predict which point will be the first in the scanning process, you should take care that only one point per fragment has a map value applied.

* this will be corrected in the next version update - the "physical properties application module will work as a completely separate module, making it possible to re-calculate mass & mass centers without having to break up again

Known Issues

Save your scene before changing the 'Center Evaluation Interval'. If LW crashes using the new interval, reload the scene and try a slightly different value.

If ground collision is enabled but the fragments don't bounce, there are 3 possible reasons:

- 'Duration' is too short
- The fragment's initial position is below the ground level
- It's a bug :)

If a collision object is selected but fragments don't bounce at all or bounce though they don't collide with the object, there are also several possible reasons :

- It's a bug
- The collision object is not tripled
- The collision object is animated
- 'Duration' is too short
- The collision object's triangles are too small