Blowing Up Stuff! Visual Effects for Destruction
Louis Marcoux – Autodesk Canada

DG6940 This class is all about having fun with your 3D models in Autodesk® 3ds Max® 2012 and creating explosive visuals. You will see how different visual-effects tools in 3ds Max can be used to create convincing explosions and how objects can be destroyed. Learn various ways of making objects collapse and make decisions about how to finish effects shots so that they look believable.

Learning Objectives
At the end of this class, you will be able to:

- Understand how various materials break and react to explosions and destruction
- Break and fracture objects in the intent to explode and destroy them
- Use rigid bodies and cloth simulations to have physically based destruction animations
- Add particles, fire and smoke effects in an explosion and destruction scene

About the Speaker
Louis Marcoux is an Autodesk Technical Specialist for 3D Animation and Visual Effects since 2003. Prior to Autodesk, he was a real-time broadcast graphics specialist for 5 years, working with Discreet Logic and Vertigo X media. Louis received a Bachelor’s degree in Electrical Engineering from Polytechnique in Montreal and also holds a Bachelor of Communications degree from UQAM and a Bachelor of Fine Arts in Film Production from Concordia University.

Web site: http://area.autodesk.com/louis
Introduction
A few years ago, I presented a class at AU on visual effects for visualization. This class was rewarded with the “Best Speaker” award for that year. I was honored and surprised and it made me think about how I could push this idea further. A lot of the comments on the class were suggesting ideas and I felt that the attendees were eager to learn more on visual effects techniques. So, there came the idea for this class: blowing up things!

At first, I realized that this kind of class was not really useful for the pure design visualization task. Doing visual effects is time consuming and it is hard to justify the cost of such visuals in the overall aspect of an engineering project. But, on the other hand, it's fun! And if you think about it, visual impact is always what attracts the eye.

The Stardust Implosion in Las Vegas was a big visual event

So, I build the class by keeping in mind that the techniques covered are not essential to the design visualization workflow but that they will add weapons to your arsenal. They will give you food for thoughts for better visual impact on your projects. And, it's always a good practice to challenge ourselves to learn new things even if they are not essential to the day to day work.

So, let's get to it and blow some stuff up!

The 3 Steps of Destruction
In this class, we will see the whole process of taking an existing object and make it explode. To start, we'll look at how we can break objects in various ways. It is important to make the debris that will be deployed in all directions to create the explosion effect. So, step 1 will be a deep look at how we can break objects.

In step 2, will take a look at how we can animate the explosion with simulation tools. Some of the debris will keep their shape and some others will have to distort. We also see how to create initial forces that will explode the object and send the debris in all directions.

Finally, we'll see that it is important to add details to the explosion scene to make it effective. Smaller debris can be added to tie in the effect and also smoke and fire.
So, that’s how I divided this class in 3 steps:

1. Break
2. Simulate
3. Add details

**Part 1: Fracturing and Breaking Objects**

The first thing to do when it comes to destroying objects is to break them in pieces. Most of the objects will be built in 3ds Max scenes as whole objects. So, we need to figure out how they will break and we need to create the debris for the destruction animation. In a destruction context, a brick wall is not an object; it’s an assembly of bricks, breakable bricks and mortar.

In order to know how to break objects and assemblies of objects into debris, it is important to study the materials that make our objects and how they would react to the destruction force that will initiate the break up. A concrete structure that falls apart will not break in the same way if an explosion initiates the breaking. Same idea for a concrete structure vs a wood structure. They won’t break in the same way. So, it is important to do a bit a visual research on how the materials would break in real life; or, as wanted if you want it to be larger than life, like in Hollywood.

Another aspect of successful destructions is the remaining of objects or part of objects that are not breakable. Metal tubes can twist but they won’t necessarily break in pieces. So, it is important to leave the non-breakable objects as single objects. They’ll be simulated differently at the simulation phase.

The success of a good visual destruction is in the details. So, the breaking up in pieces and debris is the crucial phase to create those details. The more a destruction is chaotic and detailed, the more visually dramatic it will be.

Here are 5 techniques you can use to break objects in pieces.
Technique 1: Face Explosion Tools

*With Edit Mesh*

The Edit Mesh primitive and modifier have a tool to explode by faces. By setting the threshold angle to extreme or intermediate values, you can quickly create multiple objects from the faces or elements of an object.

Beside the Explode button, there is a value slider that represents the limit of the tolerance to combine 2 faces together in the explosion process. A value of 0.0 means that all faces will be broken separately and a value of 180.0 means that all faces that are continuous will be broken together. In between values allows some faces to be grouped together on the explosion. The tolerance value is an angle in degrees.

So, when an object is selected, a tolerance of 0.0 will break the object and create one new object per face.

If the tolerance is set to 180.0, all the non-continuous elements will be broken apart.
And values in between will keep the faces within the angle tolerance value together.

This is the simplest way of breaking up an object but it has limited flexibility.

For all objects broken into faces, it is important to give the new faces a thickness. For dynamic simulations later on, if the faces do not have thickness, they won’t simulate. Also, when created, the new objects will have their pivot point at the center of the original object. So, for simulation, it will also cause problems while try to solve the object transformation. So, in brief, when the object is broken in faces, you need to do the following:

1. Add a shell modifier for thickness
2. Center the pivot point of the new objects

Also, for speed optimization, you might benefit from collapsing the object to an edit mesh or edit poly rather than keeping all the modifiers in the scene.

*With Edit Poly*

With Editable Poly’s, there is no explode function. So, the process of separating elements and faces or polys has to be manual. But, on the other hand, it offers a better way to define how the object will break since it’s not triangle based. The best option to manage this is to understand the workflow and use a script to automate the process. First, let’s take a look at the manual workflow.

With Edit Poly’s, a box can be define by edges and polygons that are not necessarily triangles.
By using different tools to sculpt the edges of a model, you can refine how it will break. The main 2 tools that we can use to add and edit edges while maintaining the overall shape of the object are the Cut tool and the Face Constraint.

Cut allows you to add edges on the exiting faces. It is important to always connect the new edges to exiting edges in order to maintain a correct topology.

With the Face Constraint tool, you will be able to move vertices and edges while keeping the sub objects that is being moved on the adjacent faces. It helps create irregular edges on the model while maintaining the shape.

Once the model is ready to be broken, the same idea as Edit Mesh applies: you need to decide if you break by elements or by polygons. This is defined by the sub object mode which the detach tool is invoked. Polygons can be detached or Elements can be detached. But all of them need to be detached manually because of the absence of an explode tool.
Detaching polygons is done in Polygon selection mode:

Detaching elements is done in Element selection mode:

With Edit Poly’s that are detached, it is also important to add a shell modifier for thickness and center the pivot point to the debris. If you did the process manually, it would be convenient to instance the shell modifier so that you can edit all of the thicknesses at once.

Doing all this can be quite time consuming even if it’s the best way to get the breaking results you desire. So, I created a script that allows a quick Edit Poly explosion that adds a shell modifier and finally collapse the stack and center the pivot point on all detached objects. The script is included with the files for the class. See conclusion for the details of where this can be found.
Running the script once will create a tool called “ExplodePolygonByFace” in the LM Tools category. You can add that tool in quad menus and be able to call it from the main viewport’s quad menus.

The workflow of the script goes like this.

1. Choose the tool you’ve just installed in the quad menus to open the dialog
2. Select all the objects you want to explode
   a. They don’t have to be Edit Polys, they will be converted in the process to Edit Poly’s

3. Define if you want to explode by elements or by polygons

4. Click “Explode”
   a. If you exploded by elements, the process is done and all elements are now new objects with centered pivot points.
   b. If you exploded by polygons, the thickness slider and the “collapse Stack & Center Pivot” button will be enabled
i. Go in wireframe mode to adjust the Shell value for the thickness of the new polygons

ii. Click “Collapse Stack & Center Pivot”. Warning: If you don’t click this button, the shell modifier will stay in the stack and the pivot point will not be centered.

5. After the Explode operation, you can change the selection and do another explosion.
**Technique 2: Breaking Volume**

For objects that need to show their volume, the core of the breaking needs to happen in Pro-Cutter, a Compound Object found within the Primitives tab.

![ProCutter](image)

ProCutter takes any 2D geometry, of geometry with faces on one side only, and use that geometry as a splice tool for the 3D geometry. Any 2D object can be used to splice a 3D volume. The result is a thick volume geometry spliced from the 2D object.

![3D geometry](image)

The UVW mapping of the cutter object gets applied to the faces that are cut on the new objects.

**Important note:** for 3ds Max 2012 users, the material of the cutter gets applied to the whole object which is not the desired behavior. We’ll see how to go around that issue at the end of this section.
Creating 2D cutters
There are different ways to create 2D cutters that will result in good fragmented objects. Here are a few techniques. For all those techniques, it is important that the cutter cuts through the whole section of the volume. Otherwise, the cut operation won’t work.

The simplest technique is to create a 2D plane and add some noise to it.

1. Create a 2D plane with enough subdivisions for the noise desired
2. Add a noise modifier and adjust the scale and the Z strength, keeping in mind that for many scenes, these values should be small.

Another good way for creating good cutters is to use 2D splines and the extrude them.

1. Draw 2D splines in the front/top/side viewports that will define the breaking pattern.
2. Select all the 2D splines and choose the Extrude modifier and in the modifier panel, increase the Amount of the extrusion to go across the object to cut.

Also, if the cut needs to have less regularity, it is possible to increase the number of segments in the extrude modifier and then use an edit poly modifier to adjust the vertices into a more refined cut pattern.

1. At the Line level, turn off “optimize” in the “Interpolation” rollout and change the number of steps to match the refinement needed.
2. At the Extrude level, increase the number of segments

3. Add Modifiers to add variations to the result. Below, the edit poly modifier is added and vertices have been move manually:
Sometimes, to be able to create good 2D cutters, it is necessary to work from a point of view or an angle that is not available in the orthogonal views. In this case, custom grids can be used and activated in order to create the 2D lines on it.

1. Create a custom grid, from the helpers menu, in the viewport

2. Orbit in the perspective view for the desired creation view. Make sure the grid is selected and press the Align to View button from the Align button. Align in Z
3. In Screen manipulation mode, align the grid so that it's in between the view and the object. The grid doesn't have to fill the object, it's considered infinite. But it needs to be in front of the object so that the extrusion is simpler to do.

4. Right click while the grid is selected and in the quad menus, choose “Activate Grid” in the top left corner.
5. Draw the 2D lines on the grid

To roll back to the home grid, select the custom grid again and right click to access the quad menus. Choose Activate Home Grid.
Another useful technique for realistic fragmentation of objects is the use of bitmap references for breaking patterns. By assigning the bitmap to a plane that can be moved like a grid, it is possible to choose a projection point of view for the breaking pattern and work from their using the Autogrid creation process.

1. Assign the bitmap to a plane using a standard material. If the image is black and white, it’s better to put it in the opacity map. This way, it’s possible to see through. It is also an option to use the visibility slider in the object properties if the bitmap doesn’t have clear opacity maps results.

2. Align the plane to the view using the same technique as for the custom grid
3. Enable the Autogrid option and start drawing lines on top of the plane and make sure that the first click of the line creation process is on top of the plane not beside.

4. Remove the plane to check the results, keeping in mind that the cuts will need to cross the whole object. We’ll see a bit later how to handle special cases of lines that don’t cross but would be needed in the effect.

A good practice for explosions is to create a bank of break patterns with breaking steps. They can be merged in scenes on demand and it can save a lot of time.
Using ProCutter
ProCutter needs 2 things in order to make a cut action. First, a set of cutters that completely cross the volume to cut and, second, a volume to cut that is water tight closed. For the cutters, a visual inspection is needed to see if the 2D geometry crosses the volume completely. For the cut volume, there are 2 tools that can be used to inspect and fix issues. The STL Check modifier inspects the model for irregularities, including opened edges. The Cap Holes modifier closes all open edges if possible.

1. add the STL Check modifier

2. Select the Open Edge option in the modifier and turn on Select Faces and “Check”.

3. Errors will be displayed as selected faces in the viewport
4. If there are errors, add the Cap Holes modifier below the STL Check

The resulting geometry should be correct and water tight closed.

If not, you’ll have to manually edit the model to make sure all open edges are closed and that there are no more errors. Otherwise, the cut will not work.

To cut with ProCutters, it is important to plan the steps of the cut. Since all cut objects must completely cross the volume to cut, there are occasions where you will need to cut in multiple phases. If all the cut objects completely cross the volume, then the cut can be done in one ProCutter operation.
Let take a close look at this breaking pattern:

Many of the 2D lines do not completely cross the breaking volume. So, to make that fragmentation work, we need to work in steps. The center line will create the first hole by making sure that the extrusion crosses the whole teapot in depth. Then, we will cut the outside part of the teapot with the lines going from the center to the exterior. Then, we’ll cut the pieces again with the remaining cutters.
Another important rule about ProCutter is that it cannot cut an object from a hole in it to the exterior.

The object will need to be broken in two parts first then we will be able to continue. In order to do that, step 2 would be divided in 2. First, connect 2 lines together and then do the other cuts.
All the cut operations follow the same steps. When carefully planned, all the cuts will be successful. In order to use ProCutter, follow these steps:

1. Select a single cutter object in the scene

2. Choose the ProCutter Tool from the Compound Objects

3. Enable the following options in order to make the cut ready for the simulation (Auto Extract Mesh, Explode By Elements, Stock Outside Cutter, Stock Inside Cutter)
4. Pick all the other cutter objects you want to include in the current cut operation (steps as described above)

5. Pick all the objects to cut using the Pick Stock Objects tool

6. Delete or hide the cutters that are currently selected. Delete if you don’t need them anymore and hide if you still need them.
Repeat this procedure for all the cuts needed. In the case of our breaking patterns, here is what we get.

When done, it is important to center the pivot points of the fragments in order to get the objects ready for simulation later.

ProCutter Script
Martin Breidt, a script developer for 3ds Max shares a lot of useful tools for free on his web site:

http://scripts.breidt.net/

One of the script is called “mbFracture v0.1a”. This scripts automate the breaking of one object using the noise plane technique and ProCutter. It’s worth checking to save a bit of time fragmenting objects.
Scriptspot.com Script
Scriptspot.com is an infinite resource for tools in 3ds Max. If you do a search for Fracture Voronoi, you’ll find a free script that also breaks objects. It doesn’t use ProCutter as its core but it delivers good results fast.

The script lets you pick an object and define how many parts and iterations you want. Then, you can break the object and it maintains the volume.
**ProCutter andTextures in 3ds Max 2012**

ProCutter supports the ability to project the UV mapping and the material of the cutters on the cut surfaces of the object that is being cut. In 3ds Max 2012, there is an issue that prevents this from happening. If you have a setup like this:

And that you choose the settings discussed in the previous section:

You’ll end up with a result like this:
To avoid that, you can use the following settings instead:

This means that the new objects won’t be separate but all the pieces will be elements in the mesh. But, the mapping and textures get applied correctly. So, after the cut, you’ll have this result:

You can select the result and run the Explode Script that is discussed in the previous sections and explode by elements.
You’ll end up with correct elements, correct texture mapping and correct materials.

This is the quickest way I found to go around this issue.

By planning the texturing of the cuts, you can achieve great visual results that will be even more believable. So, keep this technique in mind!

**Technique 3: Breaking Hollow**
The ProCutter technique is great to achieve volume fragments. But, if the surface needs to be hollow, a few steps need to be taken into consideration. Basically, to create a hollow breaking with the same type of technique, you need to do the following:

1. Subtract the interior of the volume
2. Create an opened bridge between the exterior surface and the interior surface
3. Use technique 2 with ProCutter

The best way to do this is to create a reference copy of the original object and use the push modifier to reduce its size. Then, delete a face on the outside and corresponding face on the inside to bridge the open edges together.
Let's say you have an object like this:

By choosing the “clone” command in the quad menus or the edit menus or hit Ctrl-V, and choose “Reference” when the dialog opens:

You’ll end up with an exact copy of the model and a modifier stack that looks like this:
This means that the reference is maintained like an instance and if the original object is modified, this reference will receive the modifications except for what’s above the thick line. We can now add a “push” modifier above this thick line.

And use a negative value for the Push:

This creates the interior surface.
From there, you need to create a hole on the exterior surface. This can be done by adding an edit poly modifier on the exterior surface object and inset a face to delete its center:

Because of the referencing, the interior face is also deleted. This hole is necessary for the ProCutter operations to work.

When ready to create a single object, collapse both objects and attach them in edit poly.

Select the interior element.
Flip all the faces on it:

Then, select the open edges and use the bridge tool to connect them.

The object is now ready to be broken using the ProCutter technique described previously.
**Technique 4: Filling Volume with PFlow Particles**

PFlow is a powerful particle creation and manipulation tool inside of 3ds Max. Particles are very important in the creation of visual effects. For explosions, many aspects of PFlow can be looked at. We will show the key elements of how we can use the tools for the purpose of destruction. In essence, what we are trying to do is to take an existing object and replace it by its particle replica. Once it’s made of particles, it’s easy to create all sorts of disintegration effects.

To fill up the volume of an existing object with particles, you will need a setup that looks like this in PFlow. PFlow can be found in the Graph Editors menu.

Every operator has its function. And for an explosion and a volume filling, each operator needs to be setup in a specific way. Let’s take a look in details at what’s needed.
The render node needs to be present for all particles to be rendered.

![Render Node](image)

In the settings for this main operator, the Quantity Multiplier is key for optimizing the viewport performance and get a lot of particles at render time.

![Quantity Multiplier](image)

By reducing the amount of particles in the Viewport, you gain maximum performance while still be able to time your animations and see a preview of the effect. By increasing the number of particles at render time, you'll get a full effect with lots of details. It's always a good practice to adjust these settings as you are working in PFlow.

The birth operator is the one creating the particles in the scene.

![Birth Operator](image)

To fill the volume at a specific frame in order to replace the model by its particle replica, the emit and stop frames need to be identical. In this case, it's frame 0 but if the explosion happens later in time, the frames would need to be adjusted. The amount depends on the volume to fill. It's important to do render tests until the results are what you are looking for.

![Birth Settings](image)

To have a better understanding of the look of the particles, the display operator is set to geometry.
To place the particles in the object and fill its volume, you need to use the Position Object operator.

In that operator, you can pick the objects you want to emit from. This is where you pick the object for which you want to create a particle replica.

If you want the object to be filled with particles in its entire volume, it is important that the object is water tight closed like for ProCutter. If that's what you need for the explosion effect, you choose the Location setting: Volume.

If you want to create a particle setup that only fills the surface of the object to make it look hollow, you can choose the Location setting: Surface.

To match the material and the UV mapping of the original object with particle colors, you need 2 operators.
The Material Static operator assigns a material to the particles like any other materials in the scene. The best practice is to instance the same material as the object to this operator. With an instance, you are sure that if the material is modified on the object, the particles will get updated as well.

The Mapping Object operator grabs the UV mapping of an object to apply the material to the particles. So, you need to pick the mapping object and make sure you use the same mapping channel as the original object's material.

The last thing we need to think about is the shape of the particles.
The shape can be made from a set of predefined shapes with the Shape operator.

![Shape operator](image)

Or, you can model your own debris and use the Shape Instance operator.

![Shape Instance operator](image)

To use multiple shape instances, you can model debris that will look like debris from the object exploding. This depends on what the object is made of.
Here, a rounded object has been modeled to mimic the shape of the side of a teapot. Then, the object was fragmented using ProCutter.

![Fragmented object](image)

When each fragment is an element in a single object, it is easy to create particles that randomly pick a shape in all the fragments. It adds a lot of variations and makes the explosion more believable. For multiple elements of a single object, the Shape Instance operator can be setup this way:

![Shape Instance operator](image)

You can adjust the scale depending on the size of the fragments and the variation you are after. Also, you can use the Rotation operator to introduce random rotation for all the debris. If the particle’s geometry doesn’t show in the viewport (it happens!), just roll back the display operator to dots or ticks.
When all these steps are done, you end up with a particle replica of your object made with a bunch of random fragments.

A good technique is to mix all the breaking methods. For example, if you break a glass, the impact spot can be made with particles to get very small debris in that area but wider fragments from ProCutter for the rest of the glass. The debris can even be made from the glass fragments as particle shapes.
Technique 5: Professional Plugins

If you intend to do a lot of object breaking, there are popular plugins for 3ds Max that can enhance this workflow. Here they are.

**Volume Breaker from Cebas**

[http://www.cebas.com](http://www.cebas.com)

From their web site: "*volumeBreaker is a volumetric geometry fracturing tool that will instantly create sub-geometry within any mesh - geometry that perfectly fits together and fills any given volume. With volumeBreaker, Cebas brings a Hollywood quality destruction tool to 3ds Max.*"

Price : 395$

**Rayfire from Rayfire Studios**


From their web site: "*RayFire Tool gives you the ability to fragment, destroy, demolish, wreck, break down, wreak havoc, blow up, burst, detonate, explode and do other similar things you have always dreamed of to do in Max...It offers interactive destruction and various fragmentation types.*"

Price : 325$
**Part 2: Simulation**

Now that all objects are made of debris, it is time to animate them to blow them up. This is done with simulation tools. Although it can be animated manually, it would take a long time to do so and the iterations and refinements would be even longer. So, it is easier to rely on simulation and it’s also a lot of fun.

To make an explosion or the destruction of a 3D model requires a bit of planning and, I would say, choreographing. Remember the big Hollywood explosions you have seen in your favorite movies, they happen in phases and steps to extend the dramatic effect of the initial explosion. A good explosion will be preceded by a good anticipation and will be followed by follow-through actions, like it is defined in the 12 principles of animations.

In this section of the class, we’ll take a look at 3 simulation tools in 3ds Max:

- Rigid bodies with MassFX
- Soft Bodies with the cloth modifier
- Particles animations

These tools will allow you to accomplish the most convincing destructions. By choreographing the use of those tools for your animations, you’ll be able to create very dramatic scenes. We’ll see that destructions may need some initial forces to have an explosive effect but not in every contexts. Sometimes, destruction will look better just by collapsing a structure on itself. Again, it’s all part of the planning of the destruction to make those decisions.

Since timing and space is very important in a destruction scene, it is important to think about finessing animations after the simulation process is over. Adjusting positions and timing of the resulting animation will make a simulation go from good to great.

To learn all these concepts, we’ll take a look at 5 simulation and refinement techniques.
**Technique 1: Rigid Body Simulations**

Most of all the debris that were created with previous techniques will be considered rigid bodies. Rigid bodies do no change shape during the simulation. Rigid body simulations happen in a tool call MassFX (added to 3ds Max in the 2012 version). It is accessible by right-clicking in an empty space in the 3ds Max toolbar. The list of toolbars then appears and you can enable the display of MassFX toolbar.

This toolbar offers of simplified workflow for simulations of rigid bodies, making it very fast to achieve good results.

In this class, we’ll focus mainly on the tools that are relevant to the types of animations we want to do. But it’s a very deep tool in itself.

The workflow with MassFX goes like this:

1. Select objects to be simulated
2. Define behavior in simulation
3. Define physical properties
4. Setup world
5. Simulate
6. Bake animations

And all these steps are presented in a linear fashion but it is always possible to iterate many times and go back a few steps to change settings and simulation parameters. Even after an animation is baked, it is still possible to go back and rework on the simulation. But, it a typical flow of events, it is how a simulation with MassFX for rigid bodies. Let’s take a look at these steps a bit closer.
Objects can be simulated with 3 behaviors:

**Dynamic:** objects are reacting to collision and forces and are animated by the simulation. The debris will typically be dynamic.

**Kinematic:** objects are animated in 3ds Max and do not react to collision and forces. But, they affect the simulation and create collisions with dynamic objects. Impacting objects will typically be kinematic. Objects can turn from kinematic to dynamic during a simulation. This is useful to change settings of a dynamic mesh over time.

**Static:** objects do not move and are not affected by forces and collisions. This is typical of environment objects surrounding the destruction.

Let’s take a case like this:

The central piece has been broken in debris and we would like it to collapse. The environment, in yellow, would stay intact and receive the debris. So, the debris would be assigned a dynamic behavior and the environment would be assigned a static behavior.
If an impact object coming at fast speed in on the central piece, if it’s animated in Max to impact the debris, then this object would be kinematic.

![Image of an impact object on debris](image)

Kinematic objects are essential to create initial forces in explosions. Without initial forces, debris would just fall on themselves without the explosive impact effect.

So, by selecting the objects and then using the MassFX toolbar, you can assign in one click the behavior to the selected objects.
When objects are assigned a behavior, they receive a MassFX modifier and a simulation cage (or collision mesh).

The simulation cage is based on the model’s Convex Hull. The Convex Hull is like a plastic wrapper around the object. It follows the surface of the object but doesn’t go into holes. It’s like a version of the model without holes. This simulation cage can be changed to get more accurate results or approximated to get faster results. For each modifier on each simulated object, the physical mesh can be change in the modifier panel.
The Convex Hull will give good results in most cases. But, if it happens that a geometry explodes instantly when simulation start and when there is no initial force or impact, it is most probably because 2 Convex Hulls are within each other. So, they get a repulsion force that makes it look like an explosion. In this case, it is better to switch the physical mesh type to “Original”.

For all the objects in the simulation, it will be important to define their physical properties. Those properties will influence how they interact with others during the simulation. They are defined in the modifier of the object, in the Physical Material rollout.

The density and the mass are directly linked. Since all objects fall down at the same speed with gravity, this settings really defines the momentum in relation with other objects. So an object with higher density will be more difficult to move or affect by objects with lower density. Big fragments will get a bigger mass and small fragments will have a smaller mass. If the fragments all come from the same object and the same material, their density should be the same.
The friction defines if objects will resist sliding on each other. For objects collapsing and exploding, this value is best if set low. If the friction is too high, it happens that object won’t start moving because they are stuck onto each other. Values of 0.0 make sure that sliding happens immediately and objects start moving when you want them to move.

The bounciness will make object bounce. Nothing surprising there… For big massive structures like concrete, the value should be around 0.1. Otherwise, the objects will look light when the land on the ground. This value really defines how objects feel to weight when they touch the ground.

Another aspect of the modifier that can be of interest is Initial Motion rollout.

In this rollout, you can assign an initial speed to the object and an initial spin. For explosions, that can be useful because it sets things in motion without the requirement of an impact object. It’s a great way to initiate a structure collapsing. It can be long to setup the initial velocity and initial spin for multiple fragments.
But, someone wrote a nice script to automate the setup of an explosion by adjusting these settings for multiple objects at once. It’s a good free add-on to your arsenal if you intent to blow up a lot of objects in your work.

http://loran-cg.blogspot.com/

The next step in the workflow is to setup the world. World properties are accessible when the MassFX tools dialog is opened from the toolbar.
They are under the World tab of the dialog:

The settings that we are mainly interested in are in the Rigid Bodies group. The substeps represent the quality of the simulation. More substeps gives a more refined simulation but takes longer to calculate. The solver iterations will be discussed later and they are tied to constraints. The Collision Overlap is the most crucial setting to worry about when we deal with fragments of objects.

The collision overlap defines the tolerance for objects to interpenetrate. Since we have built debris that are very close to each other, when the collision overlap is too low, the system might explode on the first simulation frame. Objects that are considered to be interpenetrated are immediately repulsed.

To create a quick explosion effect, you can reduce the collision overlap. This will make the object’s debris to push away from each other, resulting in an explosion effect. If you want a structure to collapse on itself, collision overlap will have to be permissive enough to get the objects to fall on each other from the start rather than being repulsed and projected. This is why this setting is very important.
Here are two examples of the same simulation with high and low collision tolerance. After one frame of simulation, small tolerance results in an explosion like simulation. High tolerance results in a collapsing like simulation.

When all this is setup, the scene is ready to be simulated. This is an iterative process where you can simulate and adjust settings that are not giving the desirable result. It is never right on the first attempt so it is normal to try multiple times before getting it right.

The simulation happens in the MassFX toolbar. If you have a setup with kinematic objects, it is important to simulate with animations. If not, you can simulate without animation which will give a faster playback.

Once the results are to your satisfaction, you can bake this simulation. This means that all objects will receive keyframes on their transformation channels. For animation tweaking and playback from the time slider, it is a mandatory step.
Baking is done in the tools tab of the MassFX tools dialog.

![MassFX Tools](image)

You can bake selected objects or all the objects in the simulation. Also, after baking, you can turn objects into kinematic objects and do more simulations using these objects. You can also turn kinematic objects into dynamic ones at a specific frame.

So this is the workflow for simulating rigid bodies, with or without an initial impact. Initial impact objects can help drive the direction of explosions. It depends on your desired result to see if you need them or not.

![Simulation with and without impact](image)
**Technique 2: Soft Bodies Simulations**

Soft bodies are 3D models that will deform as they are simulated by 3ds Max. This is the case of twisted metal, plastic or any other element in an explosion that will change its shape because of the pressure or because heavier objects around it force it to deform.

Soft body’s simulations in 3ds Max 2012 can be done with the cloth modifier. By adjusting the settings of the cloth to make them very resistant, we can make the cloth objects behave almost like rigid bodies. This means that intermediate values can give us great results to simulate the deformation of metal or plastic in a destruction sequence.

A rigid body simulation will be stored in the transformation channels of the objects being simulated. This is very different than doing a cloth simulation. For cloth, all transformations are stored at the vertex level, into the mesh. It is the mesh that is being animated, not the position or rotation of the object.

This is a fundamental concept that needs to be understood because to match rigid body’s simulations with soft bodies, you need to take that into consideration. Basically, you can run a rigid body simulation and take the result in a soft body simulation. But, to take a soft body simulation and bring it into a rigid body simulation is practically impossible because none of the soft bodies have inertia or spin. They are just immobile deforming meshes even if they look like they are moving and spinning.
All objects that need to be part of the soft body simulation need to have the same cloth modifier instanced on all of them. This is done easily from within the cloth modifier itself once it’s applied to one of the object.

In the cloth modifier, there are 3 types of objects.

- **Cloth**: to be simulated in the cloth modifier, objects must be of type cloth. All deformable objects are of cloth type.
- **Collision**: for objects to be taken into account in the simulation, they need to be made collision objects. All collision objects will not be influenced by the cloth simulation but they will affect all cloth objects that they enter in collision with. Animated objects in 3ds Max can be collision objects.
- **Inactive**: objects not being taken into account in the simulation. Of no use in our case.

The object types are defined in the cloth properties.

The cloth modifier can become very complex if you explore all aspects of it. But we’ll focus on the parameters that will let us create quick results for destruction objects. First, let’s take a look at the properties that we need to adjust in order to take a cloth and make it behave like a soft body and a rigid body.
The properties that we need to take into consideration are Bend, Curve, Stretch, Compress, Shear and Plasticity. They are expressed in a level of resistance. This means that the higher the value, the higher the object will be resisting to the change involved. Let's take a closer look at those settings and the result that they will give us.

- **Bend**: resistance level to bending. Values between 0 and 300 will result in variations of soft bodies and a value of 2000 will result in a rigid-like body.
- **Curve**: another resistance level to bending but based on angles between faces. Values between 0.0 and 1.0 will result in variations of soft bodies. Value of 1.0 will result in rigid-like body.
- **Stretch**: resistance level to stretching. Values between 0 and 300 will result in variations of soft bodies and a value of 2000 will result in rigid-like body.
- **Compress**: the opposite of stretch resistance, it’s the resistance level to compress. Again, values between 0 and 300 will result in variations of soft bodies and a value of 2000 will result in rigid-like body.
- **Shear**: resistance to shearing. Values between 0 and 300 will result in variations of soft bodies and a value of 2000 will result in a rigid-like body.
- **Plasticity**: ability of an object to maintain its current shape. Higher values result in rigid-like bodies and lower values tend to allow more shape changes. Values between 0 and 100 will result in soft bodies variations and value of 100 will result in rigid-like body.

In order to avoid wasting time, to go from a very soft body to a very rigid body, you can proportionally adjust all those values at once. It will give an overall believable effect. If you need to adjust subtleties, you can but for explosions and destructions, that might not be necessary. Examples:

- **Very soft body**, 10% of maximum soft body’s range: Bend 50.0, Curve 0.1, Stretch 50.0, Compress 50.0, Shear 20.0, Plasticity 10.0
- **Half soft body**, 50% of maximum soft body’s range: Bend 150.0, Curve 0.5, Stretch 150.0, Compress 150.0, Shear 150.0, Plasticity 50.0
- **Close to rigid body**, 100% of maximum soft body’s range: Bend 300.0, Curve 1.0, Stretch 300.0, Compress 300.0, Shear 300.0, Plasticity 100.0
- **Rigid-like body**, excessive resistance values:
The density will also have an impact since it represents the weight of a cloth object. So, if you have multiple objects simulated together and that they should be of different weight, you need to adjust their relative density. Higher density will also impose more pressure on the soft body to change its shape so higher values of resistance will be needed to prevent shape changes.

An important item of note is that the cloth simulation will require a ground to be added. Unlike MassFX and rigid body simulations, there is no default ground. So you need to create a collision object in the simulation for the ground.

For a good simulation, a few parameters need to be adjusted in the Simulation Rollout.

Self-Collision needs to be enabled and the number of iterations to solve the self-collision needs to be at least 2. The subsamples and steps can be adjusted for a finer simulation depending on the results you get the first time. Increasing the subsample will take longer but will have better collision handling, especially at high speed. If the step parameter is lower, smaller calculation steps will be taken resulting in higher quality simulations.
To simulate the animation, you can directly use the simulate button in the simulation group.

![Simulation Group](image)

The local simulation is useful for character cloth workflows but for destruction and explosions, it is rare that we need to do local simulations. If you are unsatisfied with simulations, you can erase them or truncate them from current frame.

To create variations of solidity in the simulation, you can simulate up to a certain frame and then truncate the end of the simulation. Then, you can change the cloth settings and simulate again. This will simulate with the new settings starting from the existing simulation. This is a useful technique to create objects that will strongly deform on impact and then stay rigid for the rest of the simulation. For example, metal objects that are deformed by the explosion but then stay rigid when they hit the ground.

Here are 2 workflows that we can use for creating soft bodies simulations in the context of destruction.

**soft bodies in a rigid body simulation**

This is the case of a structure that would break as rigid bodies but pieces of metal would be in the structure, being deformed by the solid objects breaking.
First, you simulate only the rigid bodies collapsing. Then, you add the soft bodies and you make all the rigid bodies become collision objects in the cloth simulation.

Since soft bodies can be tricky to simulate because of the complex nature of those simulations, it is wise to work in isolated parts and work on a few pieces at a time.

**Soft bodies turned into rigid bodies**
This is the case of an explosion or strong impact that will deform objects. But, the deformed objects are resistant enough to keep their shape after the impact. A good example would be a car explosion where the metal pieces keep their twisted shape as they hit the ground. The explosion is strong enough to deform the pieces but when away from the impact, they tend to keep their shape.

In this setup, the simulation is started where all the deformable objects have low resistance value. After the simulation is done, you can scrub the timeline where all the pieces have left the explosion area and are flying in the air.
From there, you can truncate the animation to remove the simulation from there. All the settings can then be adjusted so that all bodies continue to act as rigid-like bodies. Simulate again to get the rest of the simulation done where objects deform a lot less.

To create explosions like this car, you can animate the radius of a sphere going from very small to very large, encompassing all deformable objects, in a short amount of time.

This sphere can be made a collision object for the start of the animation and then turned off when all objects have left the explosion area. It can be hidden for rendering when the simulation is done.

**Technique 3: Constraining Fragments**

When fragments are simulated, they are all independent from each other and nothing makes them hold together. Sometimes, it is useful to have certain debris be attached to each other before they break in parts. A good example of that would be a piece of concrete that falls as one piece and then breaks on the ground. For that type of behavior, we have a set of tools to attach and constrain fragments together in the simulation.
For rigid bodies’ simulation, MassFX offers a set of constraints that can make objects be connected to others with variable degrees of freedom. It is also possible to simulate the non-fragmented object first and then simulate the fragments by themselves from the points of impact. For soft bodies, the cloth modifier allows to attach a set of vertices to a surface. We’ll see these 3 approaches in this section.

In the MassFX tools, there are 6 constraints that can be applied to link 2 objects together and define their degrees of freedom. But, fundamentally, each constraint is based on 3 levels of degrees of freedom: translation, swing and twist. And each of those degrees can be locked, limited or completely free.

In the MassFX toolbar, adding a constraint is easy. Pick the 2 rigid body objects that you want to constrain and pick the type of constrain you need. If the objects are not defined as rigid body yet, you’ll get a warning to define them as rigid bodies. Here is an overview of the constraints:

- **Rigid**: no freedom of movement
- **Slide**: freedom of translation only
- **Hinge**: freedom of swing in one axis only
- **Twist**: freedom of twist only
- **Universal**: freedom of swing in 2 axis
- **Ball & socket**: full freedom of movement
For each of the constraints, you can adjust how it works within limits or full freedom in the modifier stack. This is true for translation, swing and twist.

![Translation Limits](image)

Also, you can define how hard it is to get away from the original position, swing and twist of the object by adjusting the spring values.

![Spring](image)

You can make all those constraints breakable. This automates the breaking of fragments when the force that tries to break them reaches a certain value. It is a work of trials and errors to find the right breakable values for the effect you are after.

![Connection](image)
If you have multiple objects to constrain together, it can become a long task because all constraints can be applied to only 2 objects. So, you need to pair up all adjacent fragments in order to constrain them all together. If you have hundreds of fragments, it can become impractical. In those cases, let’s take a look at a technique that can save some time.

For debris that need to act as one before they break apart, you can start planning at the fragmentation phase. You can keep the original geometry before it is fragmented and use a clone to fragment it. In the end, you have the original and the fragments.

You can link all the fragments as child of the original object.

Simulating the original object as a rigid body will give the movement for the whole structure and the child objects will follow the same movement. Once simulated, you can bake the animation on the main object and remove the MassFX modifier on it (or just collapse to an editable poly from the quad menus when the object is selected).

From there, you can make the fragments as kinematic rigid bodies but only until the frame where the impact happens and they should break apart.
Simulating again will create the animation for the debris.

You can then animate the transparency of the material of the debris and the main object so that the main object is visible until impact and invisible after and do the opposite for the debris. This will result in an animation where the object acts as a whole and then breaks in debris on impact.

Of course, this can be iterated on multiple levels to create big structures destructions in steps. This can take a bit of time but it's a great way to choreograph a dramatic destruction.
For soft bodies’ constraints, the cloth modifier allows a subset of vertices to behave in a specific way in the simulation. So, it is possible to attach vertices to an external rigid body surface or to have vertices keep their initial rigid body animation. When doing the cloth simulation, the vertices that do not have specific behavior assigned to them are simulated.

To assign a set of vertices to follow a surface, you need to create a group of vertices by going in the Group sub object of the cloth modifier.

From there, you can select vertices and hit the “create group” button in the modifier stack. It's good to use soft selection after the group is created to have a smooth transition from the simulated vertices to the vertices that need to follow the surface.
When vertices are selected in a group, you can assign a behavior to them from the whole set of behaviors that are available.

The only 2 we will take a look at are Preserve and Surface.

The Preserve behavior keeps the original animation on the vertices. It’s a good way to take an object and simulate it as a rigid body. Then, use the cloth modifier to animate only a sub set of vertices that will behave as soft body in the animation.
The Surface behavior attaches the vertices of the group to an object surface. The object must be a collision object in order to be able to be used as a surface to follow. They maintain their position relation to that surface from the beginning of the animation. It's a great way to attach soft bodies to objects in the simulation and have them follow other objects.

It's by using a combination of all these techniques that you can have fragments and objects act together in a destruction sequence.

**Technique 4: PFlow Particles Simulations**

Particles can be set in motion by adding operators in the Particle Flow Graph. By defining at which frame the explosion needs to happen, you can turn static particles into particles in motion. The operators that are needed are Speed, Spin and Force. Speed and Spin allows you to define specific movement and Force lets you use different forces that affect speed and motion.
To get an explosion going from particles that have been created using the first section’s method, you can add an Age Test operator that defines when the particles should be set in motion.

Absolute age defines a specific frame. In this case, frame 10 with no variation. That means that all particles are set in motion at frame 10.

A new event can be created with a speed & spin operator, that will maintain the original particles properties but not their motion.

To create a direction for the explosion, you can use the PFlow icon in the viewport and orient it the way you want the impact to project the particle debris.
In the Speed operator, you can make the direction completely random for an explosion without a direction or along the icon arrow direction. Adding some variation angle will make the explosion a bit more random.

![Speed Operator](image1)

The speed and the variation factor depend on the scale of the scene. It’s a trial and error that will give good results. The spin operator can make the debris spin in random 3D directions and again, it’s a trial and error process to get the right motion from the spin speed and variation.

![Spin Operator](image2)

Combining these tools together gives you a great impact effect with particles.

![Impact Effect](image3)
Once the explosion happened, it is good to add forces that will give more realism to the movement. You can add forces by adding a Force operator per force that you want to have in the effect.

The main 3 forces that are useful in explosions are gravity, drag and wind. These forces are added in the viewport from the Space Warp tab.

The gravity will make objects fall. The drag will slow them down to simulate air resistance and the wind will get particles to react to wind forces. The wind is not that useful for debris effects but if you add smoke or fine dust as particles (as we'll see later), wind can enhance the believability. But when debris should not be affected by wind, there is no need to bring it in the equation and add calculation time.

For the Force operator, the Influence is by default very high. It should be brought down to values closer to 100% to get better effects. Again, it's a trial and error work to get the right balance between the forces to get the effect that you want.
An important item of note when doing rigid body simulations and that particles are emitting from objects in movement, in the Position Object helper, the “lock on emitter” option should be enabled.

When enabled, the particles will move with the emitter as it is animated in the scene. This allows an animation where main objects are animated as rigid bodies and some of the rigid bodies then explode in particles to give more details to the explosion sequence.
If the particles should enter in collision with objects or the ground and not go through them, you can add a Collision operator to the event. By default, the Collision operator will make particle bounce from the deflectors that are listed in the operator.

The UDeflectors allow you to define objects in the scene to be used as deflectors and the Deflector is a planar deflector that can be used for walls and ground.
The last thing to mention about particle simulations is that there is a point in time where you need to make the original object go from visible to invisible and the particles to go from invisible to visible. This should be done with materials rather than using the visibility track. By using a material, it is easy to use the material as instances in the scene and use only one material for the animation of many objects. It’s much more productive. So, you make a copy of the object’s material and you assign the copy to the particles in the Material operator. The original object’s material goes from opaque to transparent and the particles material goes from transparent to opaque.

**Technique 5: Refining Animations**

When doing rigid body simulations, you get baked animations on objects. When comes the time to adjust those animations, it can become difficult to add keyframes and offsets from the original animation. If you want to have an object fall into a different position after a simulation, or re-time some aspects of the animation, you’ll need to follow the techniques mentioned in this section. This is often needed to make some final adjustments to the shot for good staging. We’ll see how to adjust transformations with animation layers and we’ll see how to re-time using the animation ranges.
Animation layers allow to build layers of keyframe channels to produce the final animation. It is a great way to add offsets or adjust animations when a lot of keyframes are present on the base animation channel for objects. For example, after baking an object simulation, you may want this object to jump higher or fall further.

Animation layers are not enabled by default on objects. They need to selectively be added to the objects on which you want to make refinements to the animation. The workflow is simple:

1. Select the object you want to add layers to
2. Open the Animation Layers tool bar by right clicking in an empty area of the 3ds Max icon menu
3. Enable animation layers for the current selection

4. Add an animation layer for the selection

5. Use the default controller type to get animation curves that are keyable

6. Animate using the standard keying tool while on the new animation layer

With that workflow, you’ll be able to adjust, exaggerate and refine the animations that you will get from baked simulation animations.

For adjusting the timing of an animation in 3ds Max, you can use the animation selection ranges. They can be enabled in the animation time slider area.
When the selection range option is enabled, you can select multiple keyframes for multiple objects and adjust them all together to scale the time of those keyframes.

To speed up a portion of an animation, you can grab the end handles and place them closer together.

To slow down, do the opposite and place the handles further away from each other.

Alternatively, you can look at the 3ds Max Motion mixer that has a time warp feature. It’s a great visual tool to mix and adjust animations. The workflow is a bit longer than using animation selection ranges but it’s very powerful. To adjust the time with the motion mixer, you need to follow these steps.

1. Select all objects for which the retiming is needed
2. Choose “save animation” in the animation menu and give it a name. It will save an XAF file which stores all the animation for the current selection.
3. Open the Motion Mixer from the Graph Editor’s menu

4. Add Max Objects and keep the current selection active

5. Right-click on the track and load a new clip and choose the one you just saved
6. Enable Time-Warps mode

7. Right-click on the clip and add a time warp

8. Add time boundaries markers on the clip by clicking on it

9. Use the top half of the markers to scale the time and do the time warps

The motion mixer shows a dotted line that helps visualize how the timing is affected on the clip.
With the motion mixer, you can have multiple tracks affecting different objects in the scene. You can use these multiple tracks to time different sections of explosions or destruction shots.

![Motion Mixer Screenshot](image)

Part 3: Adding Details
All the simulations will give great animations. But, the success is in the details. Adding details is a crucial part of making an effect work. It will tie in the shot and make it go from good to great. In this section, we’ll look at 5 different techniques to add details to a scene. These techniques can be used for destruction scenes but you can expand their usage to other goals.

Technique 1: using PFlow to add small objects
When a structure explodes or gets destroyed, there are many small objects that were used to build this structure but that are not necessarily visible when it is intact. When the object breaks, these small objects can be projected out and their presence in an explosion scene adds to the realism. So, in a car, for example, you can have small screws, radio buttons, wires, handles that are projected out of the car when it explodes. Instead of simulating everything as rigid bodies, you can create an independent particle system that will throw all these objects out in a random fashion. It’s a very efficient way to add details in a scene.
Using PFlow, the technique is very similar to the one we have looked at before for exploding particles. The idea is to replace the object emitter by a simple emitter right at the center of an explosion.

This is done by using the default Position Icon operator instead of the Position Object operator. Then, the volume of the icon can be set at the origin of all the particles.
The shape instances can be made of all the small objects you can think of that are relevant to the subject being destroyed.

You can time the speed of the particles with the speed of the explosion simulation.
Technique 2: break particles debris on collision

We have seen before how particles can be set to bounce on impact with a collider. But, it is also possible to make the impact break the object into smaller objects, faking the fracture effect on collision.

To achieve that effect, you need to replace the Collision operator by a Collision Spawn operator.

This operator deletes the original particle on collision and breaks it into pieces. By adding variation in size and speed, you get an effective breaking up effect.
By connecting an event to the output of the collision spawn, you can redefine the shape of the new fragments and model debris as you would like them. Or, reuse the same shape instance operator to randomize the shape of the debris.

In this particular event (above), a Rotation operator was added so that the new debris are randomly rotated when they appear and not perfectly aligned. And, a simple collision operator has been added to have the new debris bounce. But, this operator could also have been a Collision Spawn in order to break the new debris in smaller debris, cascading the effect multiple times.
Technique 3: adding trails to particles

Objects that are emerging from and explosion can deteriorate as they move, leaving a trail of dust, small debris or smoke. Particles can add trails in a very quick way, using the Spawn operator.

By setting up this operator to spawn by travel distance, the particles leave a trail behind.

By adjusting the speed variation and the divergence, you can get different types of trails spreading. Like any other event in PFlow, the look and motion of those new particles can be tweaked with operators in a new event. So, forces like wind can be added to create the effect of dust or hashes.
This technique can transform a fragment into something that looks hot and burning or a piece of concrete into a fragment that emits dust as it flies away from its original position.

**Technique 4: adding small debris to breaking faces of objects**

When a concrete structure breaks, at the fracture line, there are some smaller debris that detach from the bigger blocks. To achieve this small debris effect, we can use a technique that we have studied before which is to emit particles from objects in the scene. But, this time, we'll refine the emitting volume to narrow it down to selected faces on the objects. This way, you can select the fracture faces of 2 fragments that are breaking up and add particles only on the fracture surfaces.
To get particles to emit only from fracture faces, you need to go on each fragment model that you want to emit from and select the fracture faces in face mode.

Also, in the Position Object operator, you need to specify that particles should be emitted from selected faces.
The look and motion of particles can be defined with the operators that we have seen in previous sections. Here is an example using small debris as particle shapes with a gravity force.

![Image of debris with gravity force](image)

This technique can be used to create dust puffs by using smaller particles.

**Technique 5: Using FumeFX for fire, smoke and dust**

FumeFX is a powerful plugin to 3ds Max. It is not included with 3ds Max and needs to be purchased separately at www.afterworks.com. But because it is so commonly used in film visual effects for explosion, fire and smoke that I thought it would be appropriate to show a bit of it. It's fun and the results are impressive.

![Image of fire and smoke](image)

FumeFX is a fluid simulation tool focused on fuel, fire and smoke. It is very advanced and the goal of this overview is just to give an introduction to the concepts. You can expand your knowledge on FumeFX with Allan Mackay’s Advance Visual Effects DVD’s. What’s good about the tool is that it takes into consideration all objects you want when doing the fluid simulation making it extremely believable.
The workflow in FumeFX starts with a simulation cage that defines the limits of the simulation. All fluids must be contained in that cage and the cage is also in charge of defining the precision of the simulation.

The fluids need to be emitted by something in the scene. That can be a dummy object, a real object or particles. Once that is defined, FumeFX knows where to create fuel, fire and smoke. The emitters are defined as helpers in the scene and that’s through those helpers that you can define particle systems & objects emitters.
Like for rigid bodies and cloth simulations, there is a need in FumeFX to include all objects that will affect the simulation into that simulation. A lot of settings can be changed in the simulation to get various types of effects.

When the simulation happened, the look and feel of the result can be adjusted for rendering and lighting. Again, many settings are available to achieve all sorts of effects. But make sure lights are included in the illumination tab for the fluids to be lit by scene lights.
Once the look is defined, you can render using Mental Ray or Scanline and the fluids will be part of your scene. Or, render them separately and combine them later in compositing.

Conclusion

All of the techniques that were covered in this class are like steps in a dance. Every step is important but it’s the overall choreography that will make the final effect. So, by combining all the techniques learned, you’ll be able to create amazing destruction sequences. I hope you have learned from this presentation and if you want to continue learning, I will make the material for the class available here.

http://www.louismarcoux.com/AU2011_DG6940
user name:: AU2011_Atendees
password: WeWereThere

Have a good AU 2011!

lm