

Burning Chair

This page provides a tutorial on creating a Burning Chair simulation with Chaos Phoenix in 3ds Max.

Overview

This is an Intermediate Level tutorial. Even though no previous knowledge of Phoenix is required to follow along, re-purposing the setup shown here to another shot may require a deeper understanding of the host platform's tools, and some modifications of the simulation settings.

Requires Phoenix 3.10.00 Official Release and V-Ray 3.60.04 Official Release for 3ds Max 2015 or newer. You can download official Phoenix and V-Ray from <https://download.chaos.com>. If you notice a major difference between the results shown here and the behavior of your setup, please reach us using the [Support Form](#).

The instructions on this page guide you through the process of creating a Burning Chair effect using Phoenix and 3ds Max.

The emission of fire is modulated by a VertexPaint modifier coupled with a Vertex Color texture used as a mask for the Phoenix Fire Source. Many Dynamics parameters related to Fire/Smoke simulations are explained in detail, with videos showing their effect on the simulation. The simulation is then written to disk, and the resulting cache files are used for the Phoenix Resimulation process to slow down the movement of the flames.

The Download button below provides you with an archive containing the start and end scenes, as well as a Phoenix Render Settings preset that you can use to quickly set up the volumetric shading options for a basic fire simulation.

[Download Project Files](#)

Units Setup

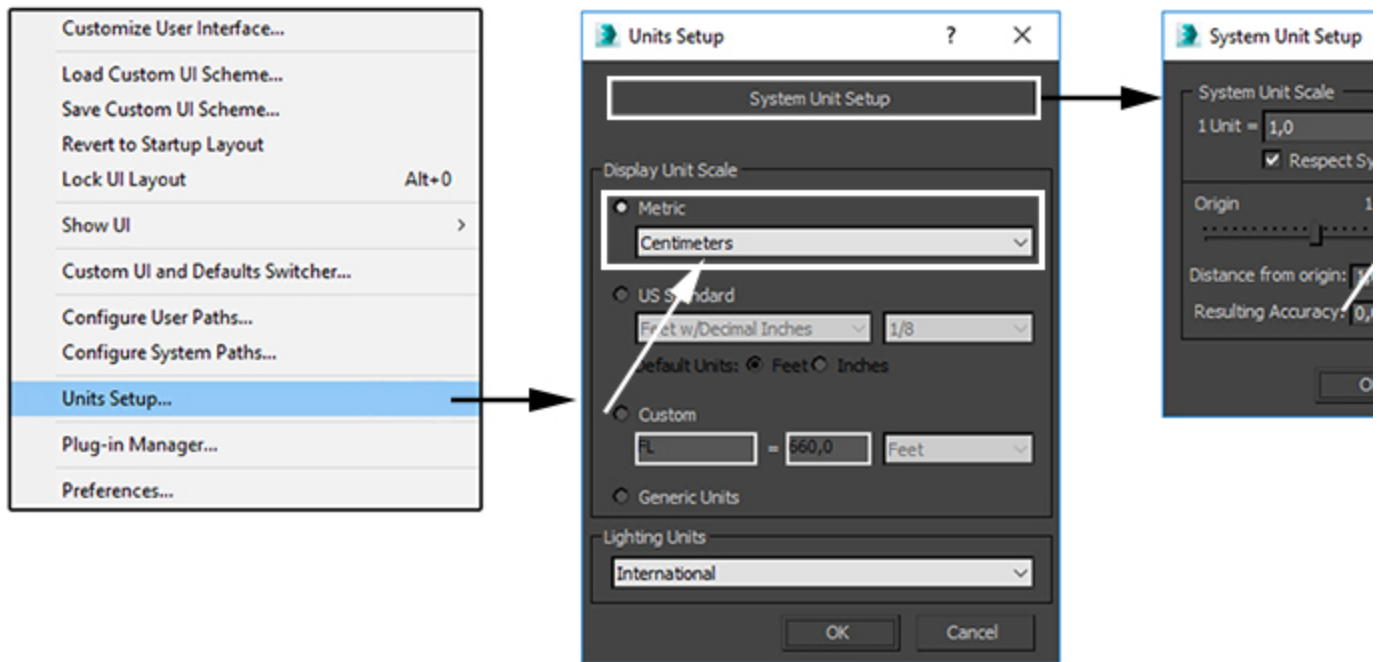
Scale is crucial for the behavior of any simulation. The real-world **size of the Simulator** in **units** is important for the simulation dynamics. Large-scale simulations appear to move more slowly, while mid-to-small scale simulations have lots of vigorous movement. When you create your Simulator, you must check the **Grid** rollout where the real-world extents of the Simulator are shown. If the size of the Simulator in the scene cannot be changed, you can cheat the solver into working as if the scale is larger or smaller by changing the **Scene Scale** option in the **Grid** rollout.

The Phoenix solver is not affected by how you choose to view the Display Unit Scale - it is just a matter of convenience.

A burning piece of furniture would be considered a small-scale simulation so we will view the units as Centimeters.

Go to **Customize Units Setup** and set Display Unit Scale to **Metric Centimeters**.

Also, set the **System Units** such that **1 Unit** equals **1 Centimeter**.

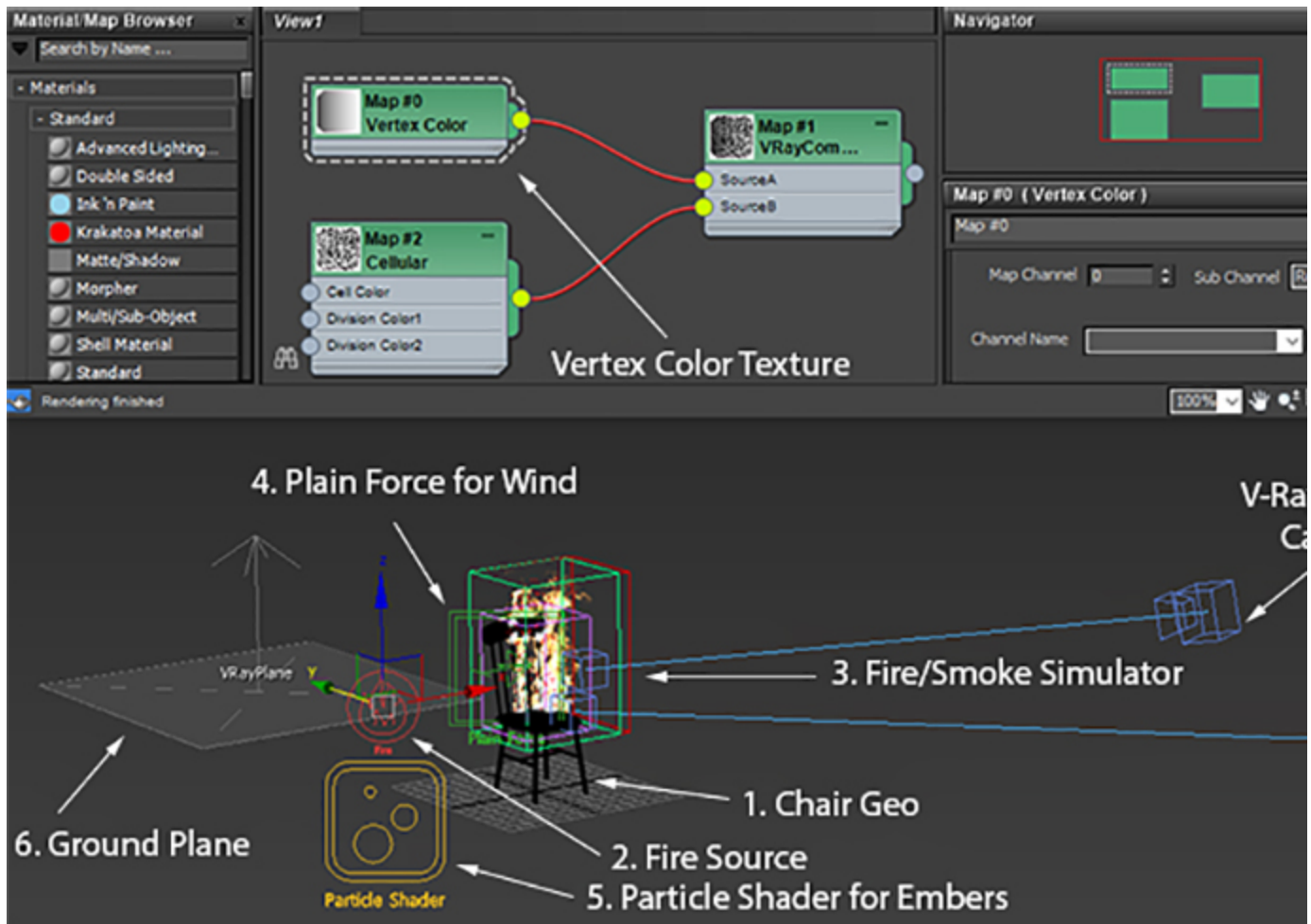


Scene Layout

The final scene consists of the following elements:

1. The chair geometry with a **VertexPaint modifier**.
2. A **Phoenix Fire/Smoke Source** with the chair geometry in its list. A **Vertex Color** texture is used as a **Mask** for the **Outgoing Velocity** parameter of the Source. The Vertex Color texture is reading the vertex data from the Vertex Paint modifier on the chair geometry. The **Source** is also emitting **Drag Particles** into the simulation. The Drag particles are used to represent embers.
3. A **Phoenix Fire/Smoke Simulator** for the fire.
4. A **Phoenix Plain Force** used to imitate wind blowing in the positive X direction.
5. A **Phoenix Particle Shader** to render the Drag Particles of the Source as points.
6. A **V-Ray Infinite Plane** for the ground.

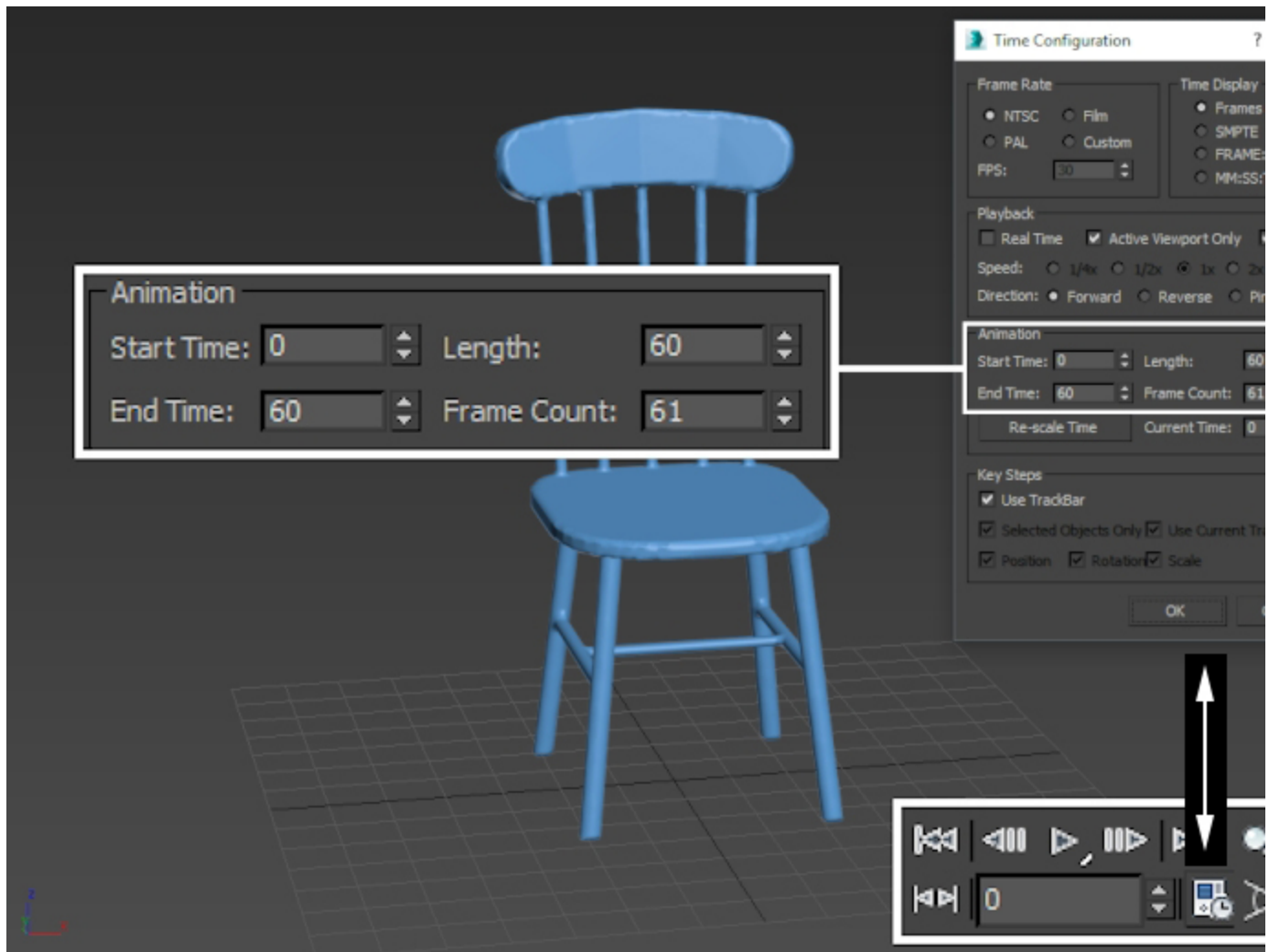
Note that there are no light sources present in the scene. The only source of illumination for the final image is provided by the fire.



Scene Setup

Set the **Time Configuration Animation Length** to 60 so that the Timeslider goes from 0 to 60.

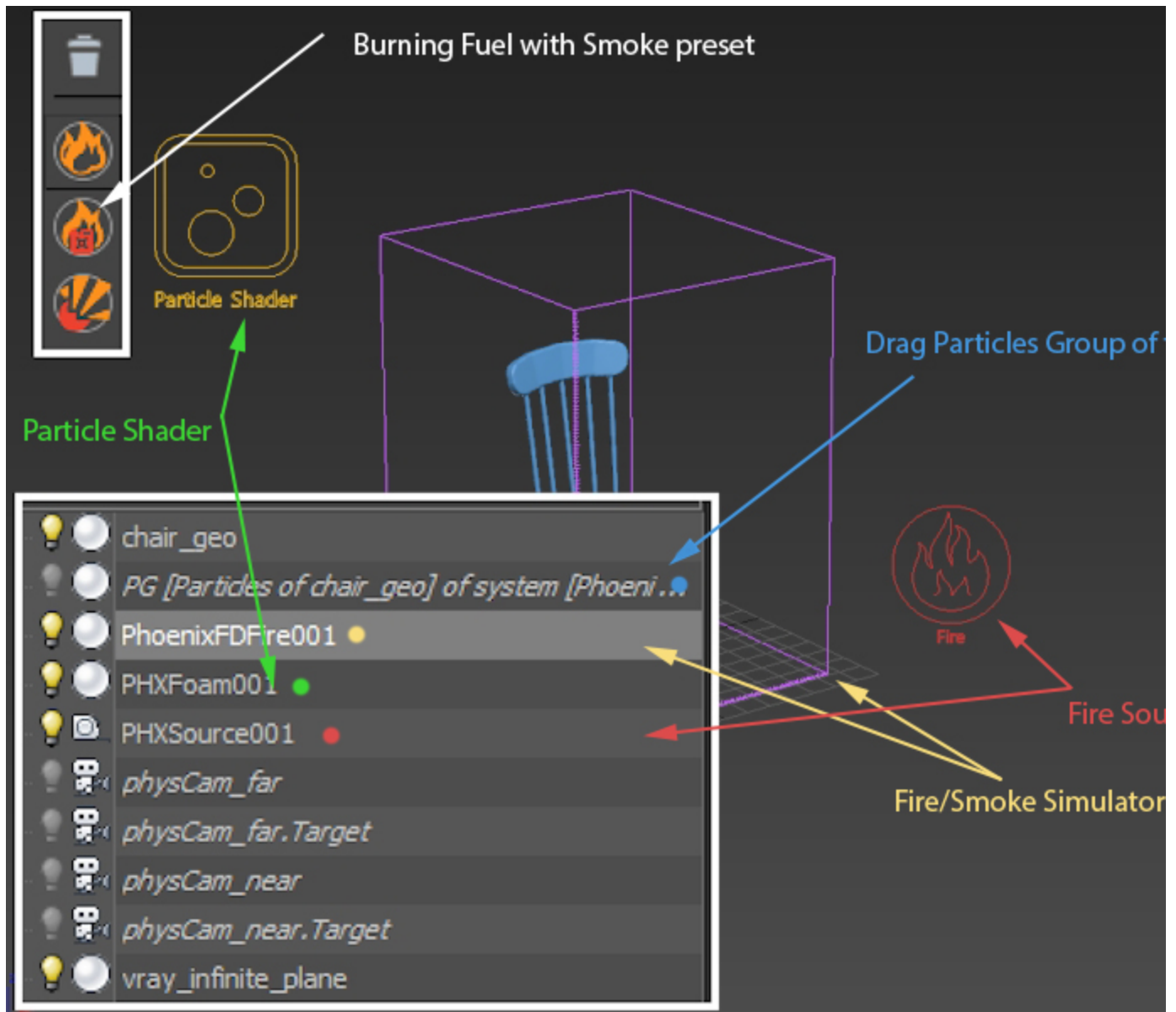
If you need any geometry animation that should interact with the simulation, prepare the animation at this step. After we're done with the basic simulation, we will rescale the timeline to slow down any animation present in the scene and we will use the Resimulation option to slow down the fire simulation as well.



Phoenix Setup

Select the chair geometry and left-click on the **Burning Fuel with Smoke preset** from the **Phoenix Toolbar**.

This will generate an entire setup for you consisting of a Phoenix Fire/Smoke Simulator, a Phoenix Fire Source, and a Phoenix Particle Shader node used for rendering the Drag particles produced by the source.



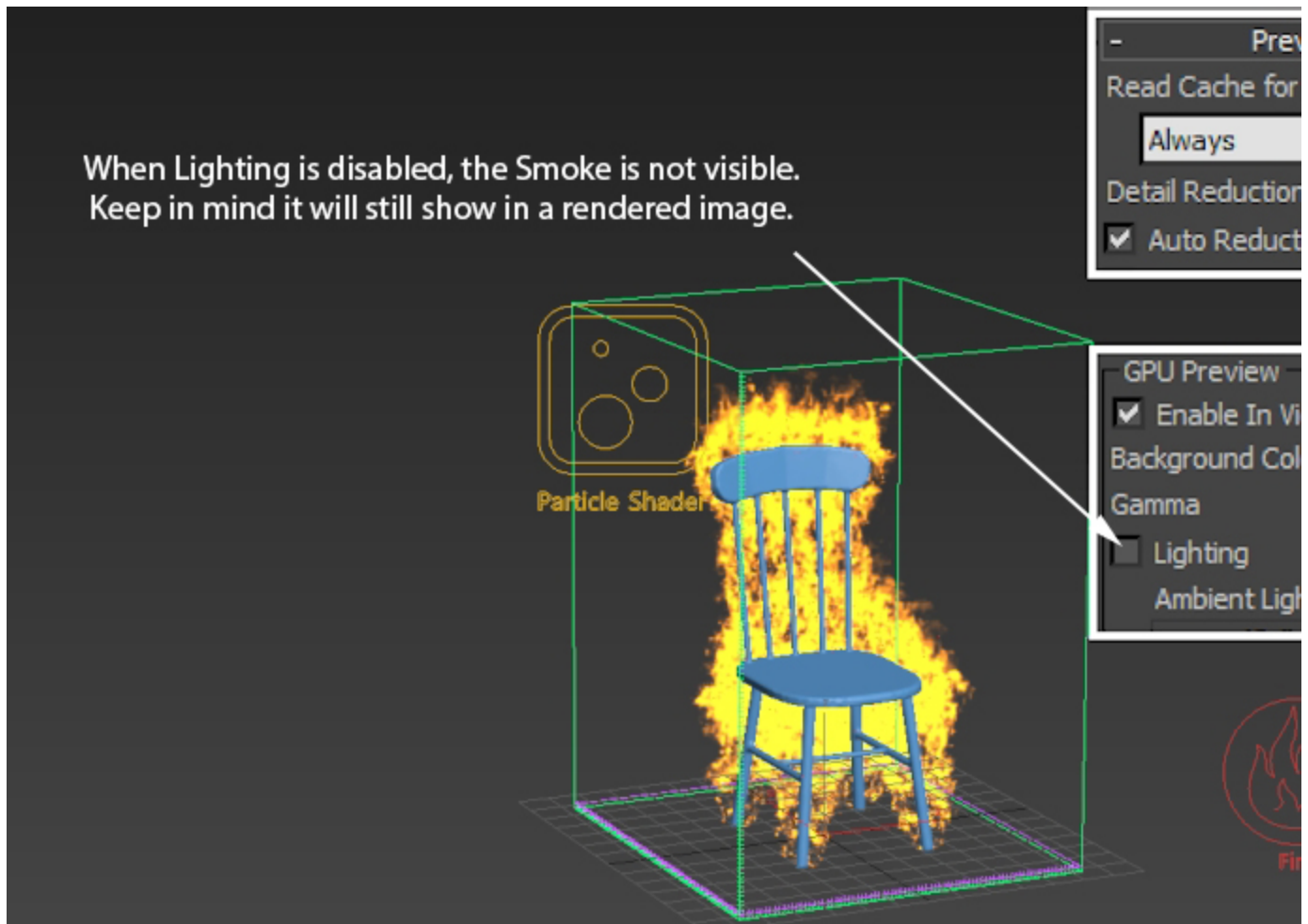
Phoenix provides a high-quality **GPU Preview** directly in the 3ds Max Viewport which should be enabled by default by the Burning Fuel preset.

You may want to disable it if you're working through a Remote Desktop connection or on a machine with integrated graphics.

To do so, go to the **Preview** rollout **GPU preview** section **Enable in Viewport**.

The **GPU Preview** section of the Preview rollout also contains **Lighting controls** which only affect the viewport preview and not the rendering.

You can also disable the Lighting altogether which will cause only the fire to be visible in the Viewport.



Here's how the simulation looks with the default Burning Fuel preset.

As you can see from the video to the right, there are plenty of issues to address:

1. **The emission is uniform** - the entire chair is burning instead of only certain sections of it. We will first take care of that by using a **VertexPaint modifier** to specify where the emission happens.
2. **The fire is too choppy** - we are going for a smoother/calmer look. The noisiness in the fire is caused by the low grid resolution, the Vorticity parameters in the Dynamics rollout of the Simulation, and to a lesser extent - the shading settings set up in the Rendering Volumetric Options Fire window. We resolve this by disabling the Vorticity parameters and tweaking the Fire rendering Color and Intensity curve of the Volumetric Shader.
3. **The fire is moving rather quickly** - this would be the desired behavior when simulating a burning flammable fluid but instead we would like to achieve a smoother flame. Think of a campfire - the flames produced by the Burning Fuel preset don't resemble the calm and elongated fire produced by burning wood. We resolve this by slowing down the cached sequence from the Input panel and resimulating over the slowed-down cache files with the Time-Bend Resimulation option enabled.

Varying the Emission with Vertex Color Texture

At the moment, the entire chair geometry is used for fire /smoke emission. This produces a homogeneous flame which looks unrealistic.

To break up the emission, we use a **VertexPaint** modifier to specify where the emission happens. We can then extract the painted information using a **Vertex Color texture** and apply that as a mask to the **Outgoing Velocity** parameter of the Phoenix Fire Source.

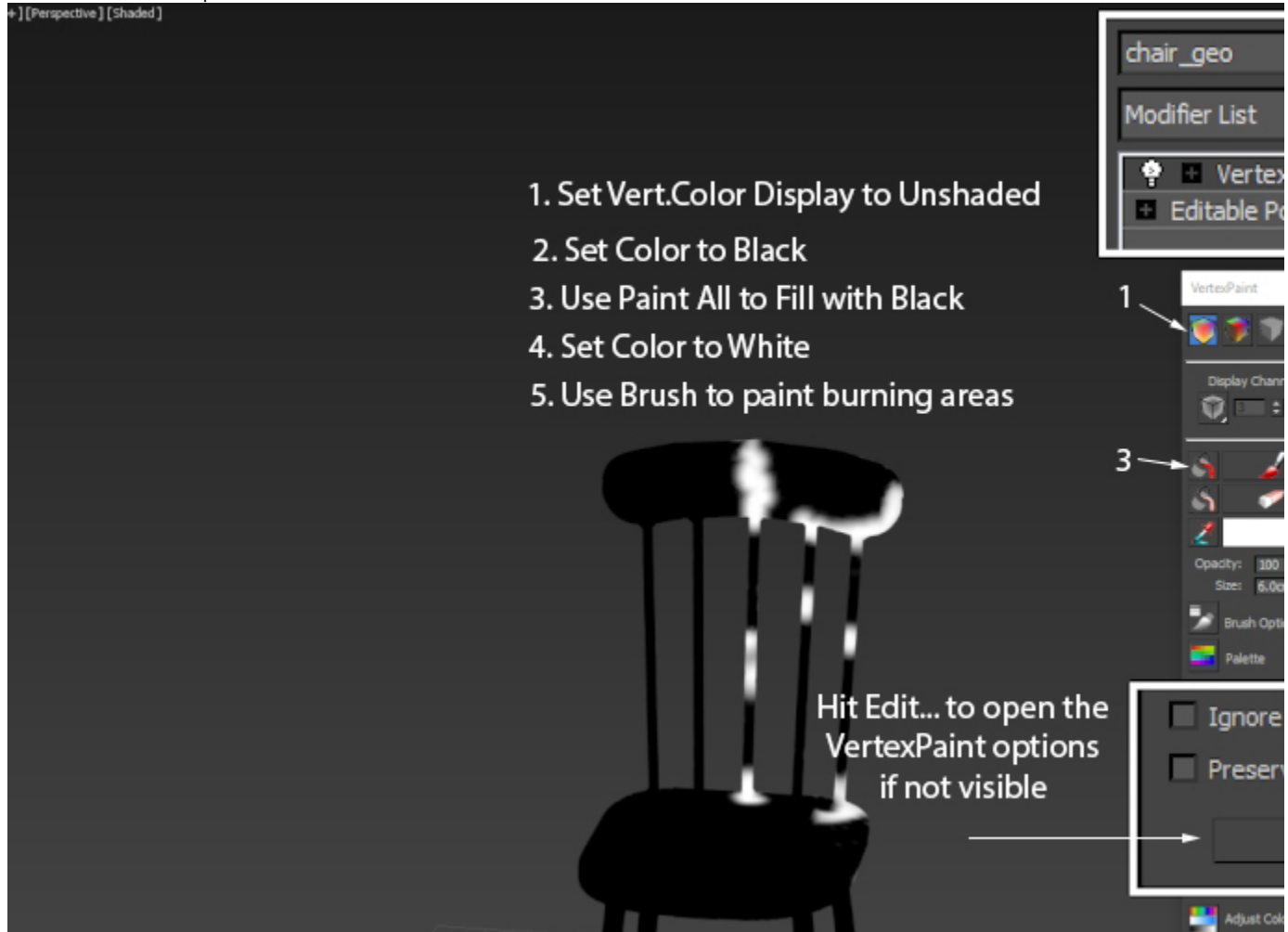
Select the chair geometry and apply a **VertexPaint** modifier. A **VertexPaint toolbar** should appear in the 3ds Max UI. If you accidentally close it, use the **Edit...** button under the VertexPaint modifier to bring it back.

Click the left-most icon at the top of the toolbar to set the **Vertex Color display** to **Unshaded** - this makes it easier to see what you're doing.

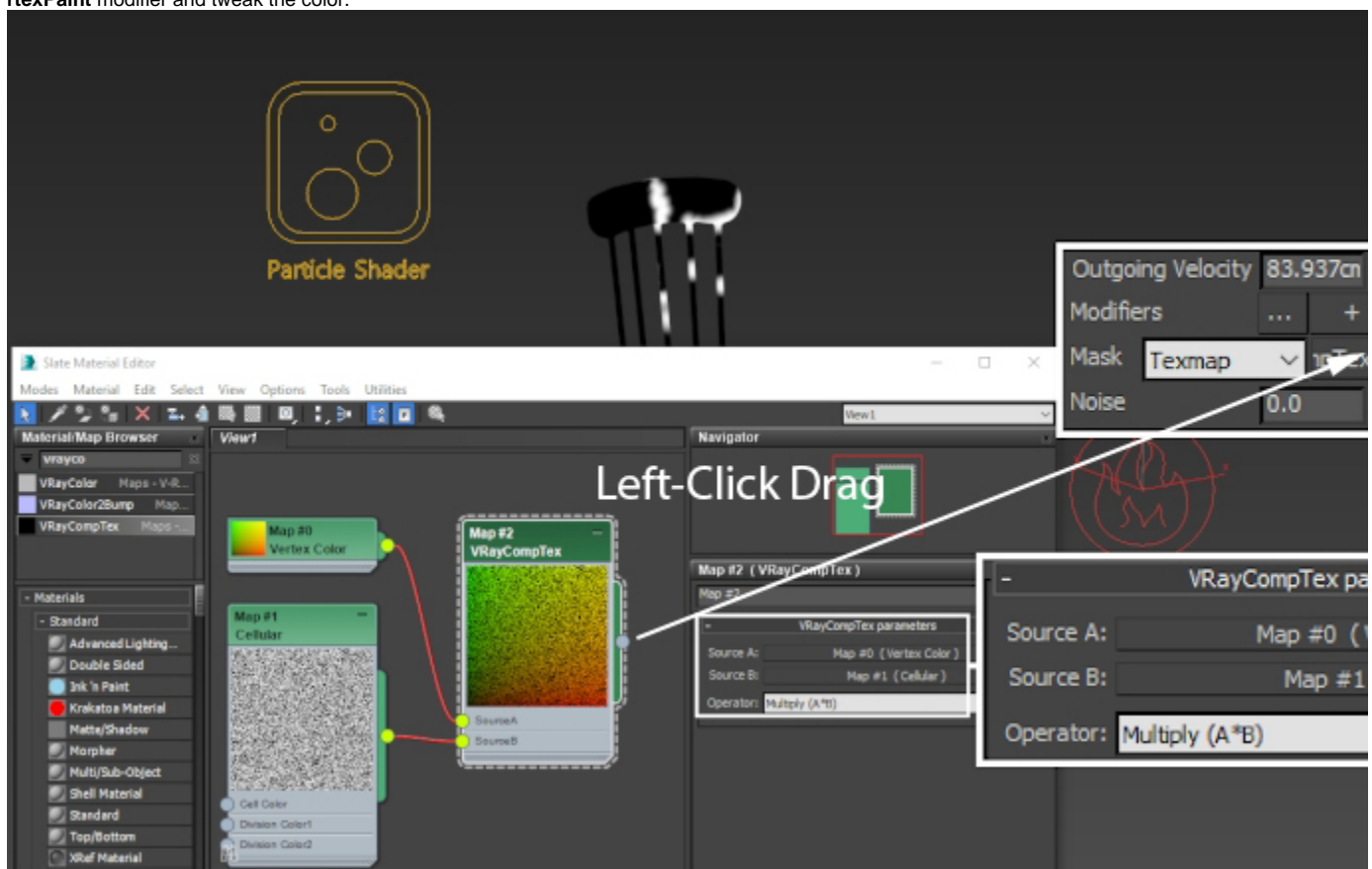
Then, set the Color to **Black** (0, 0, 0) and use the **Paint Bucket** icon to **make the entire chair black**.

Activate the **Brush tool**, set the Color to **White** (255, 255, 255) and paint the areas that you want to be burning.

When finished, click the 3-rd icon (left to right) at the top to disable the vertex color preview.



The painted Vertex Color on the chair geometry will heavily affect the simulation result. You can always go back to the **VertexPaint** modifier and tweak the color.



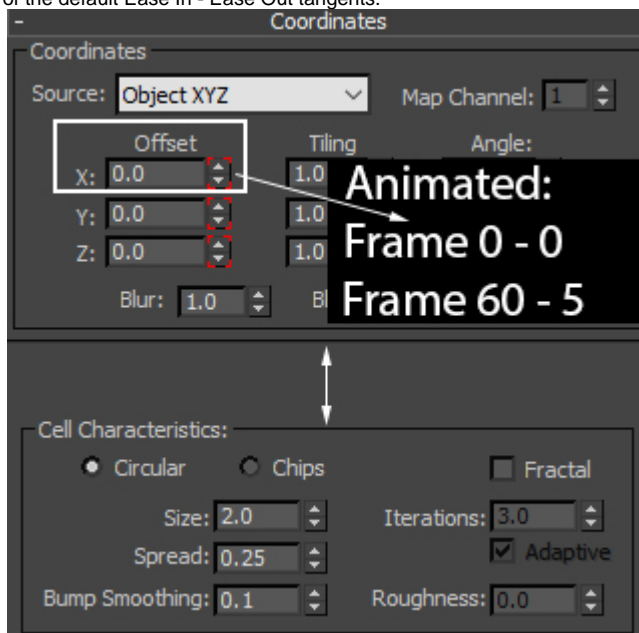
Set the **Cellular type** to **Circular**, with a **Size** of **2.0** and a **Spread** of **0.25**. Those values are an artistic choice and in no way set in stone - feel free to change them if your setup requires it.

Animate the **Offset X** parameter to go from a value of **0** to **5** over the range of frames 0 to 60. This will cause the sampling coordinates for the Cellular noise to shift over time, adding one more layer of randomness to the emission.

Open the **Curve Editor** and set the tangents for the animation to **Linear**.

If you can't find the Cellular texture coordinates in the list, make sure to **disable the Show Only filters under the View Filters** options box. You should then be able to find it under SME Map # Properties Coordinates.

This is done so the animation of the Cellular noise doesn't speed up in the beginning and slow down at the end, which is the effect of the default Ease In - Ease Out tangents.



Here's how the simulation looks with the **VRay Comp** texture used as a **Mask** for the **Outgoing Velocity**.

We still need to tweak the simulation parameters and make some speed optimizations along the way.

Select the **Phoenix Simulator** and open the **Grid** rollout.

Set the **Scene Scale** to **1.5** - this will slightly slow down the simulation.

Set the **Cell Size** to **0.65** and change the X/Y/Z dimensions of the grid such that the painted areas of the chair are included in the bounds of the Simulator box. This will save some simulation time as those empty voxels will not be calculated in vain.

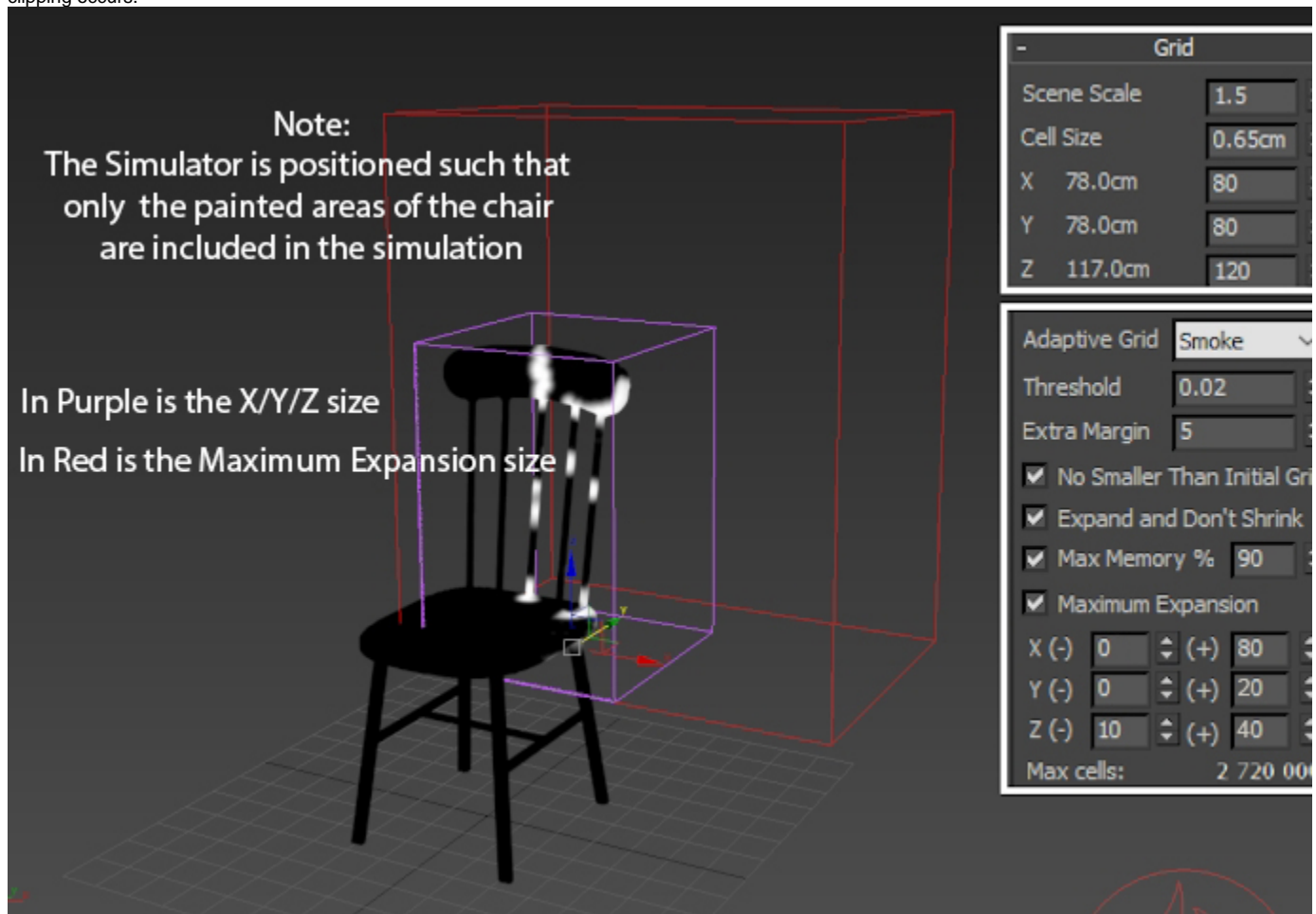
Set the **Adaptive Grid** option to **Smoke**, and the **Threshold** to **0.02**.

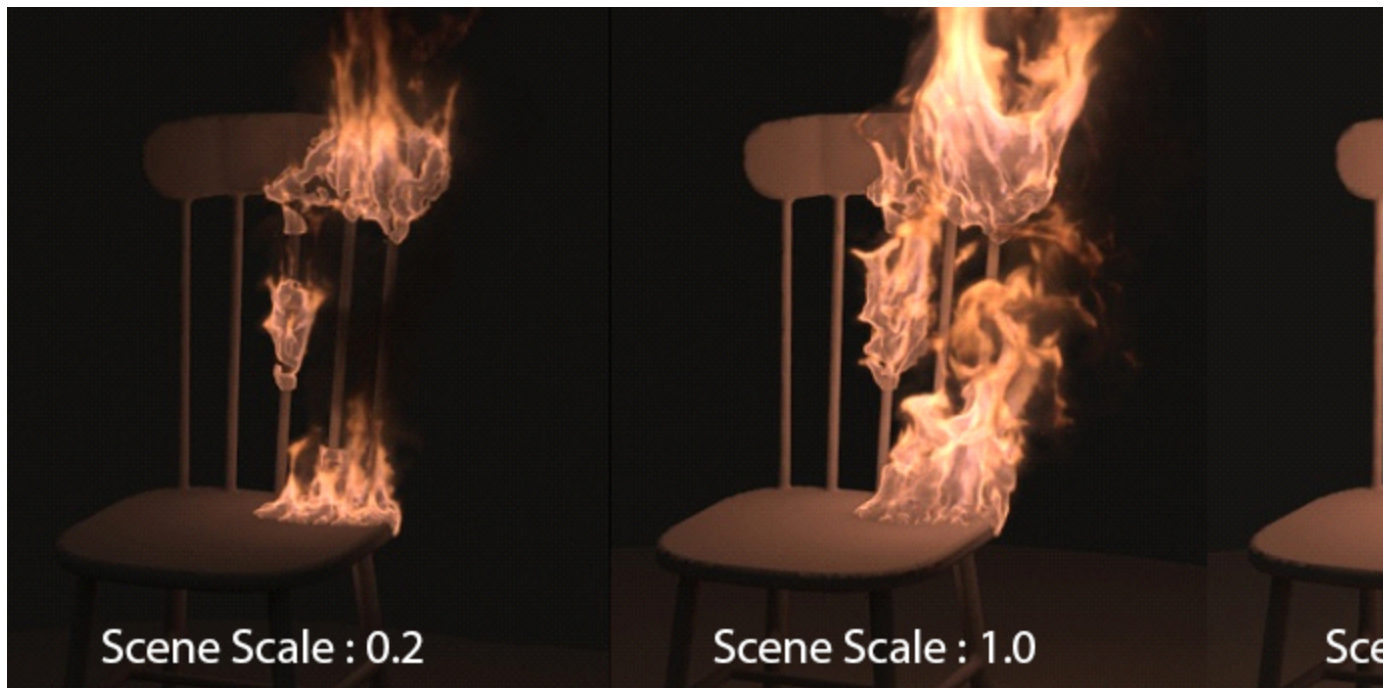
The Adaptive Grid option allows the container to grow in size once the content of the specified channel near the walls of the simulator reaches the Threshold value. In this case, the container will increase in size when there is smoke near the walls.

Enable the **Expand and Don't Shrink** option - because fire simulations tend to be turbulent, it is possible that a certain portion of the container will oscillate between being completely empty and filled with smoke/temperature. This could cause the Adaptive Grid option to constantly resize the bounds of the simulation, producing undesirable results (such as sharply cutting off the fire).

Enable **Maximum Expansion** and set the X/Y/Z dimensions according to your needs. In this tutorial, a Phoenix Plain Force is used to add wind blowing from the left so extra space is provided to the right where the fire is expected to go.

Set the **Extra Margin** to **5** - this option allows the Adaptation to keep a number of voxels close to the walls as a buffer zone and expand the grid earlier than usual. This can be especially useful when simulating fast-moving objects or explosions as it allows the simulator to expand before any clipping occurs.





Select the **Phoenix Fire Source** created by the Burning Fuel preset and make sure that the Chair geometry is included in the **Emitter Nodes** list at the top.

By default, the **Emit Mode** should be set to **Surface Force** - this will produce flames only on the surface of the emission geometry.

Set the **Outgoing Velocity** to **400cm**. The Outgoing Velocity parameter controls the strength of the emission. If you'd like the flames to be calm and short, reduce this value accordingly.

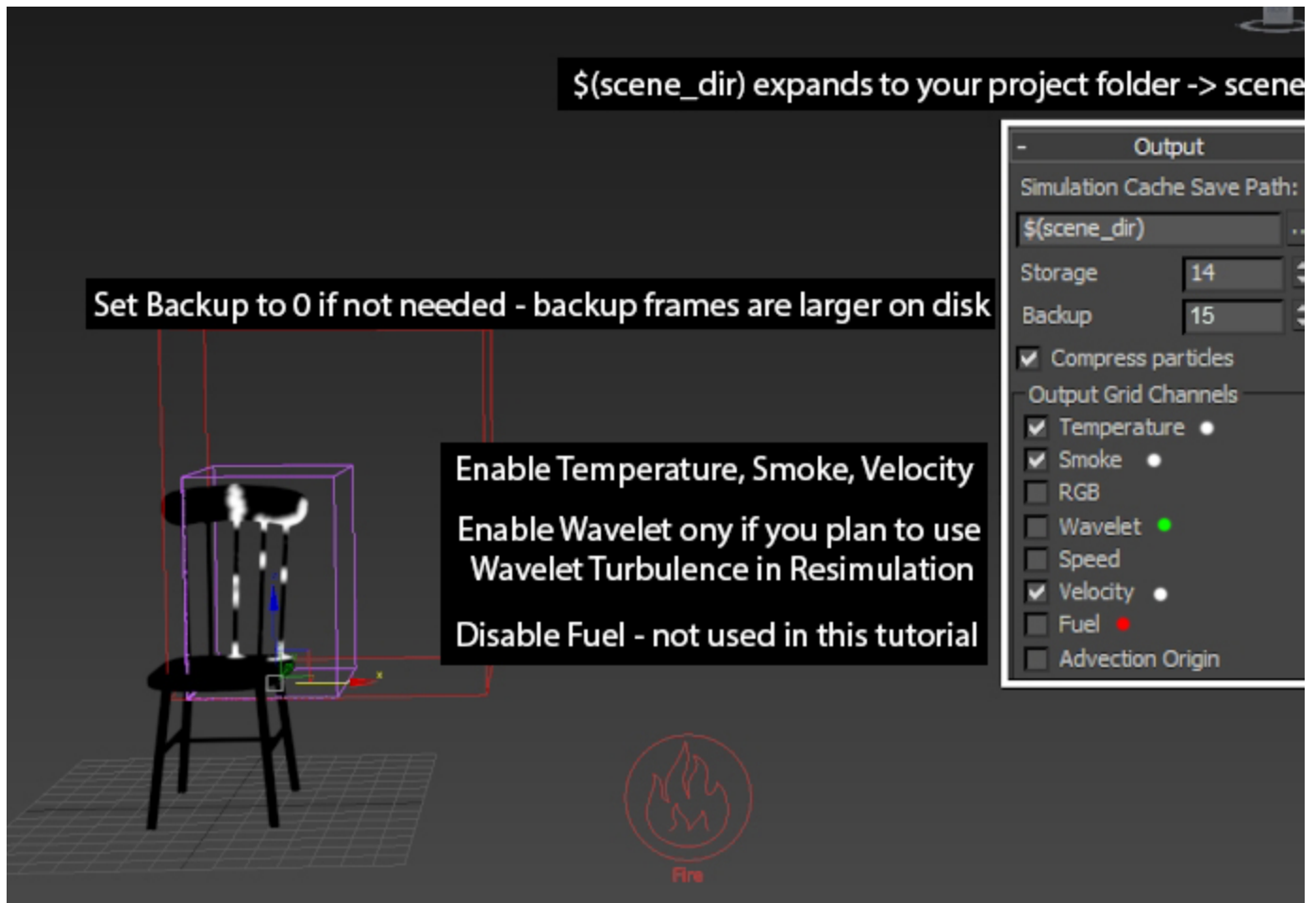
Disable **Fuel**, and enable **Smoke** and **Temperature** emission. The Burning Fuel preset we selected earlier sets up a simulation that uses the fuel channel to produce temperature and smoke - this behavior is controlled from the Fuel rollout of the Phoenix Simulator.

However, instead of relying on this, a much simpler and easier to control way to produce fire is to only emit Smoke and Temperature, and to tweak the Shading/Rendering options as necessary. Fuel and burning are better suited for explosive effects.

Set the **Particles** value to **2.5** - this parameter controls the number of particles (in thousands) born each second. We emit Drag particles to represent the embers produced by the process of burning.

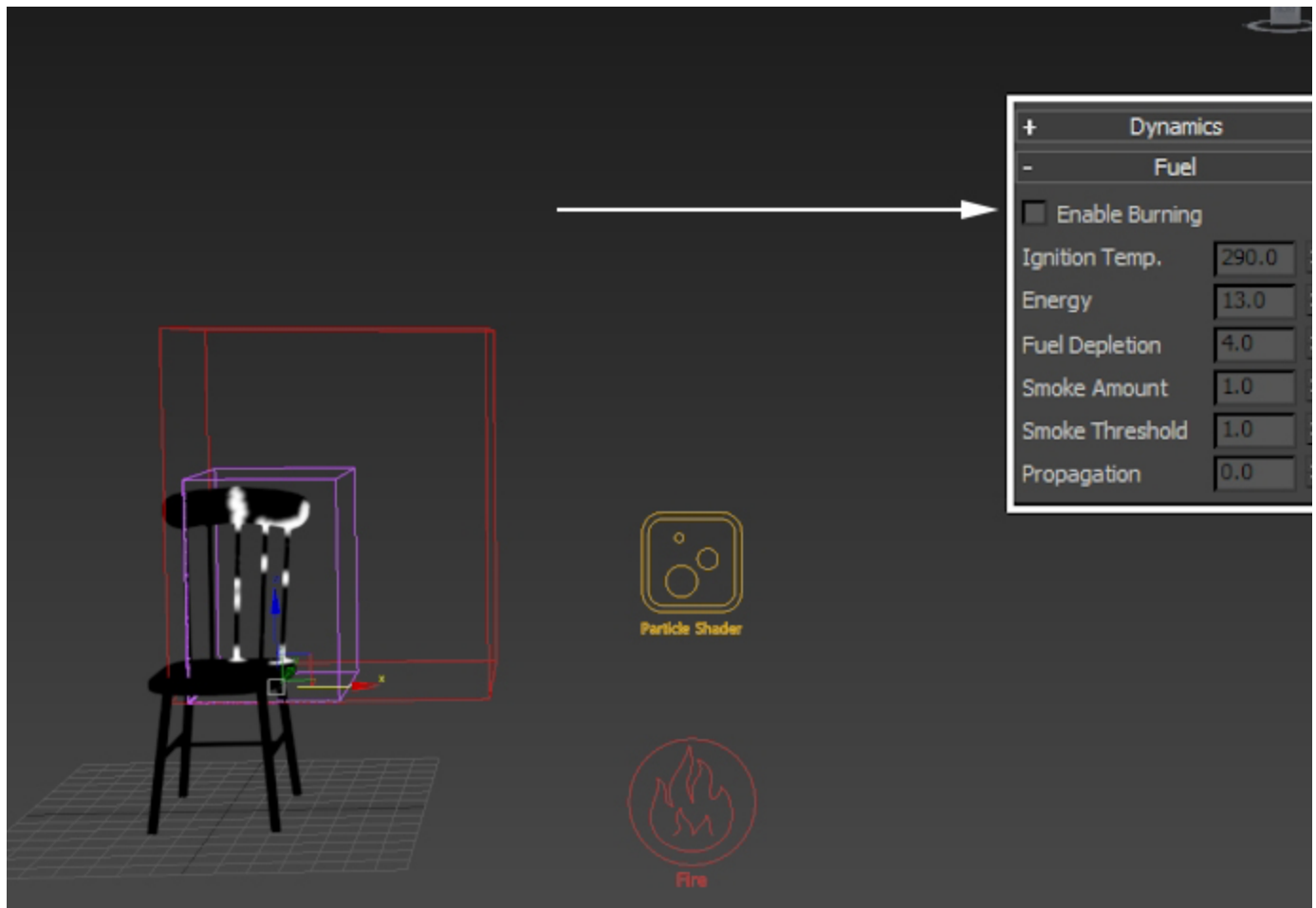
Drag particles are simply Phoenix's version of Particle Advection. As the simulation progresses, the Drag particles are carried (dragged) along by the Velocity channel together with the smoke.

Make sure to enable **Export Drag Particle IDs** - Phoenix needs the IDs for the Resimulation process. If you want to render with motion blur, you also need to enable output of Drag Particle Velocity from the Source.



Open the **Fuel** rollout and deselect **Enable Burning**.

This option was enabled by the Burning Fuel with Smoke preset we used as a starting point. Since we disabled the emission of fuel, the burning process will not be utilized and keeping this enabled will only consume resources in vain.



Here's how the simulation looks with the higher **Outgoing Velocity** rate and reduced **Cell Size**.

Note that we also tweaked the **Scene Scale** parameter which affects all the dynamics of the Phoenix Simulator.

The fire is still too turbulent and noisy for our purpose so we take care of that next.

Phoenix Simulator Properties

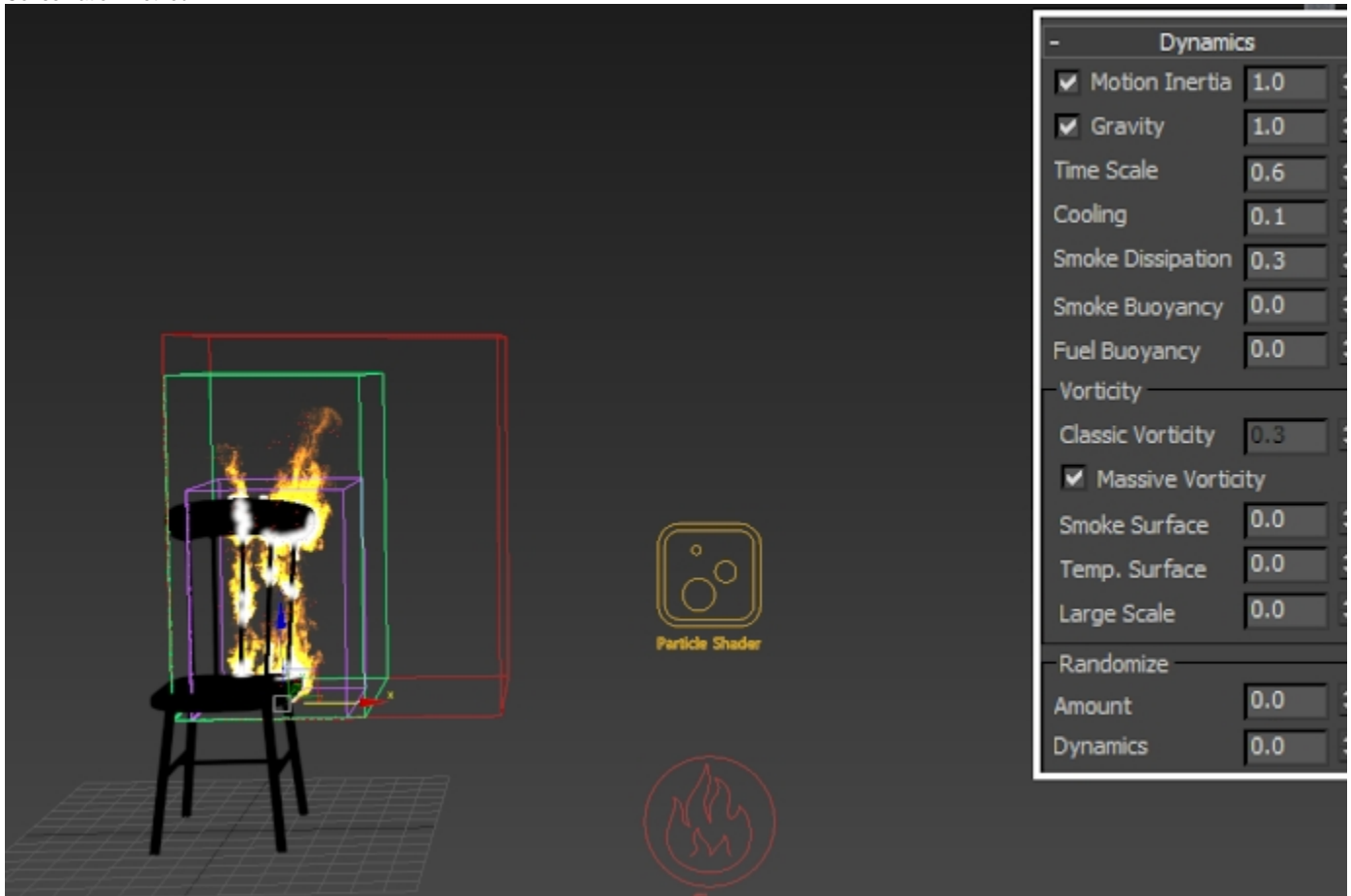
Select the **Simulator** and open the **Dynamics** rollout.

Set the **Time Scale** to **0.6**. The Time Scale option works as a multiplier for the velocity in the simulation. The lower this value is, the lower the effect of the forces and dynamics parameters such as buoyancy, vorticity, etc. will be.

Reduce the **Cooling** to **0.1**. This parameter will heavily influence the height of the flames in the simulation. Reduce it for taller flames, increase it for shorter ones.

Set the **Smoke Dissipation** to **0.3**. The effect of this change will only be visible in the smoke. You may omit this step in case you plan to only render the fire.

Set all the **Vorticity** and **Randomize** parameters to **0** to disable them entirely. For this simulation, we rely on the detail produced by the emission textures and the **Buffered Conservation** method.



Here's the simulation with **Time Scale** set to **0.6**.

As you can see from the video to the right, the fire is now slightly less turbulent. This is going in the right direction for us.

Here's how things changed after reducing the **Cooling** to **0.1**.

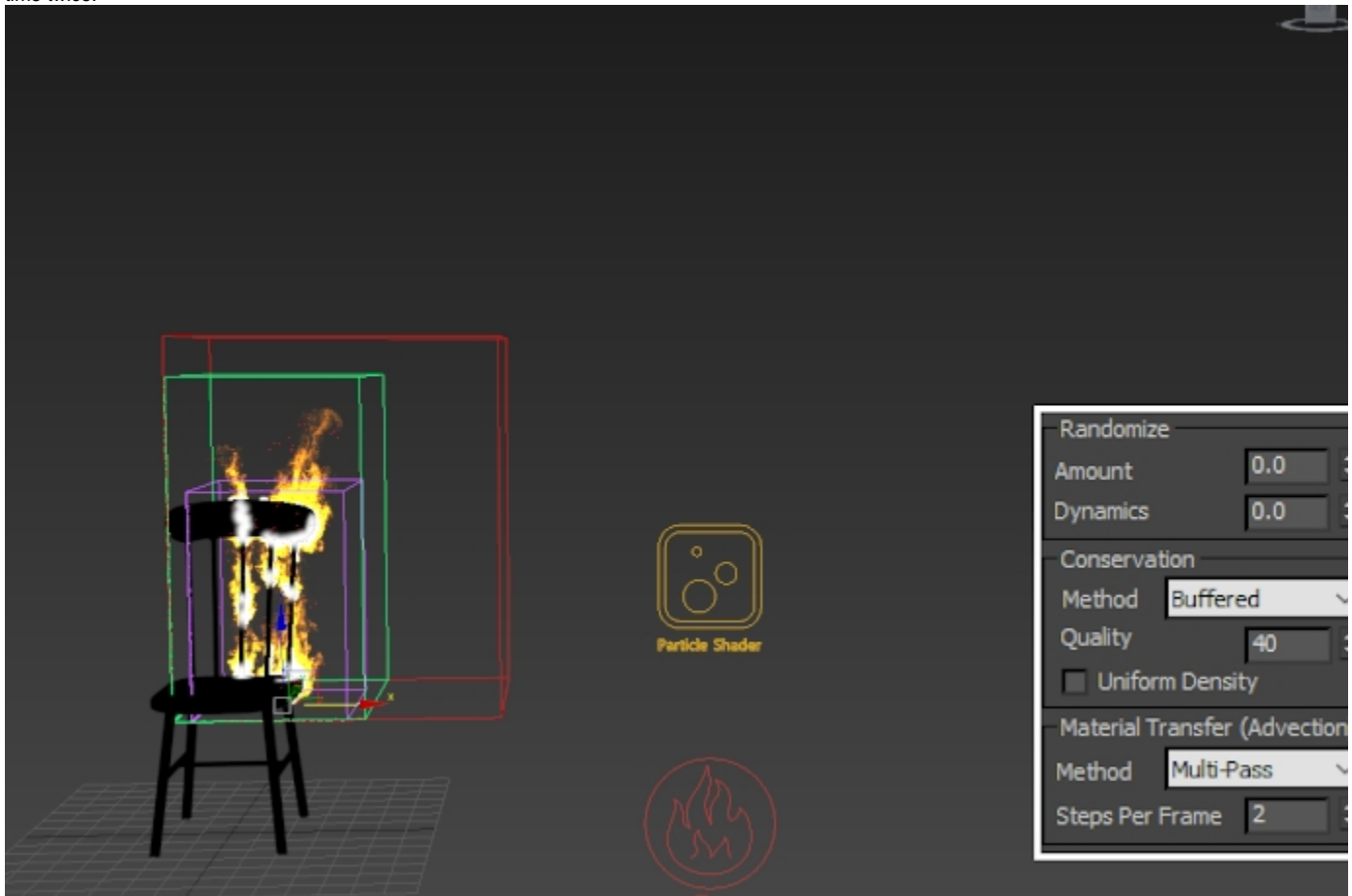
The flames are now longer - as they should be with this type of burning.

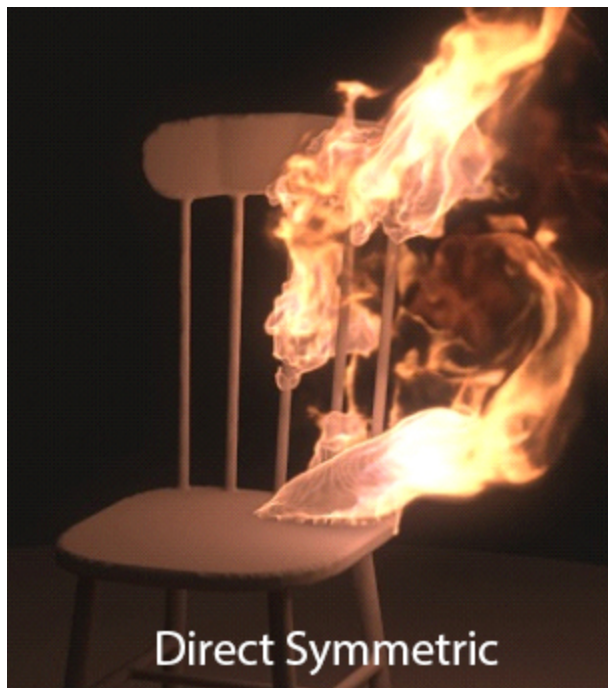
Turning off the **Vorticity** removed the noisiness in the simulation.

In the next step, we add some detail back into the sim by setting the **Conservation Method** to **Buffered**.

Still under the **Dynamics** rollout, in the **Conservation** section, set the **Method** to **Buffered**, with a **Quality** of **40**, and disable **Uniform Density**.

You may tweak the **Steps Per Frame** parameter if your setup requires it. For this tutorial, the **Steps Per Frame** are set to **2**. A higher **Steps Per Frame** value will produce a smoother simulation. This is usually required if the emitter object is moving quickly and producing stepping in the emission, or the outgoing velocity is so high that the fluid becomes noisy. Try to keep this parameter to a value as low as possible because it increases your simulation time linearly, i.e. increasing it twice will increase the simulation time twice.

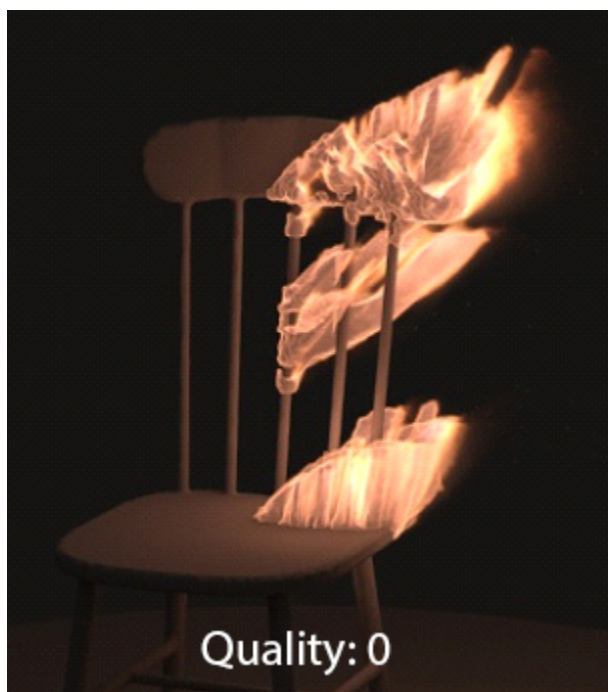




Direct Symmetric



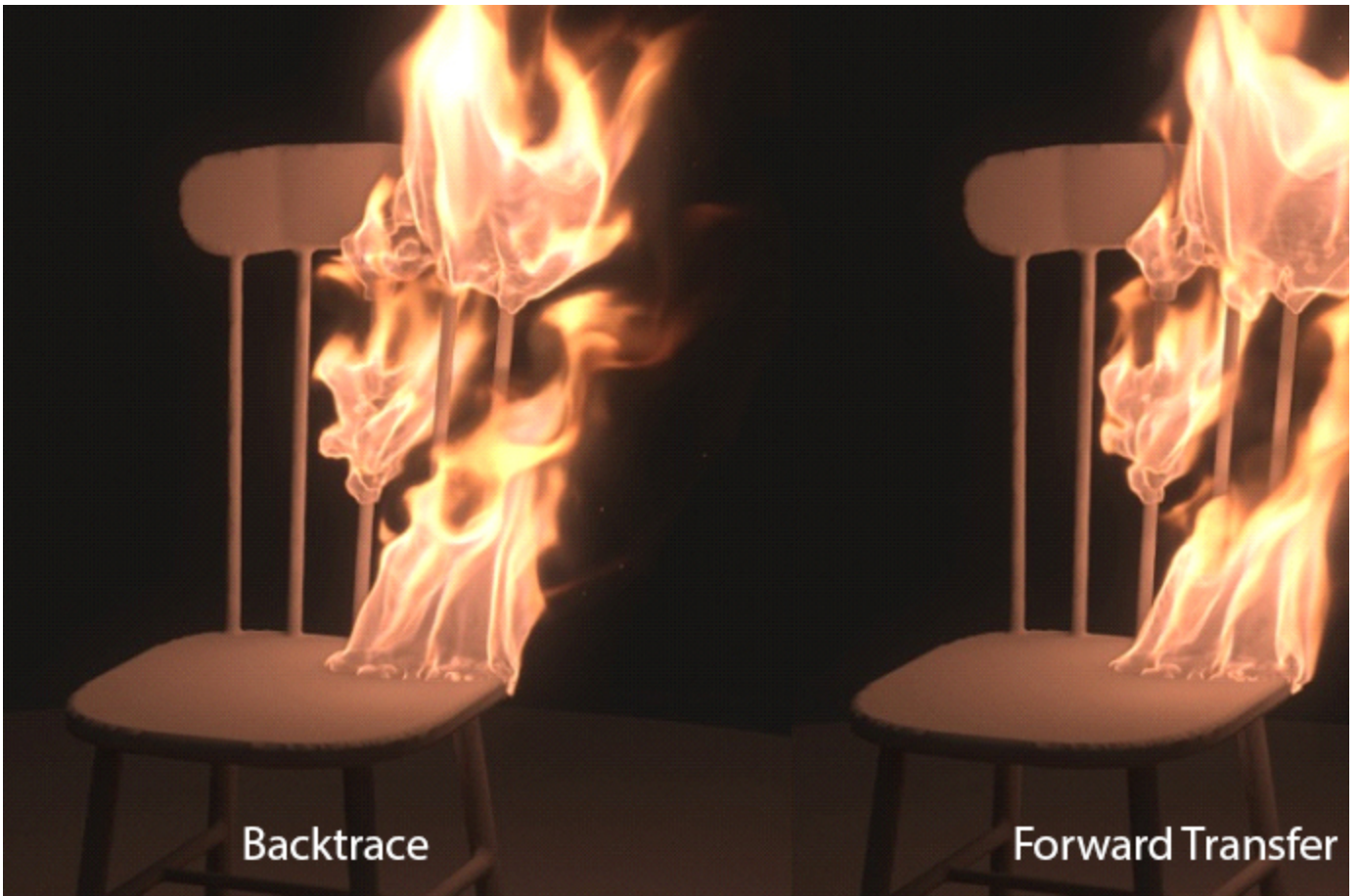
Direct Smooth



Quality: 0

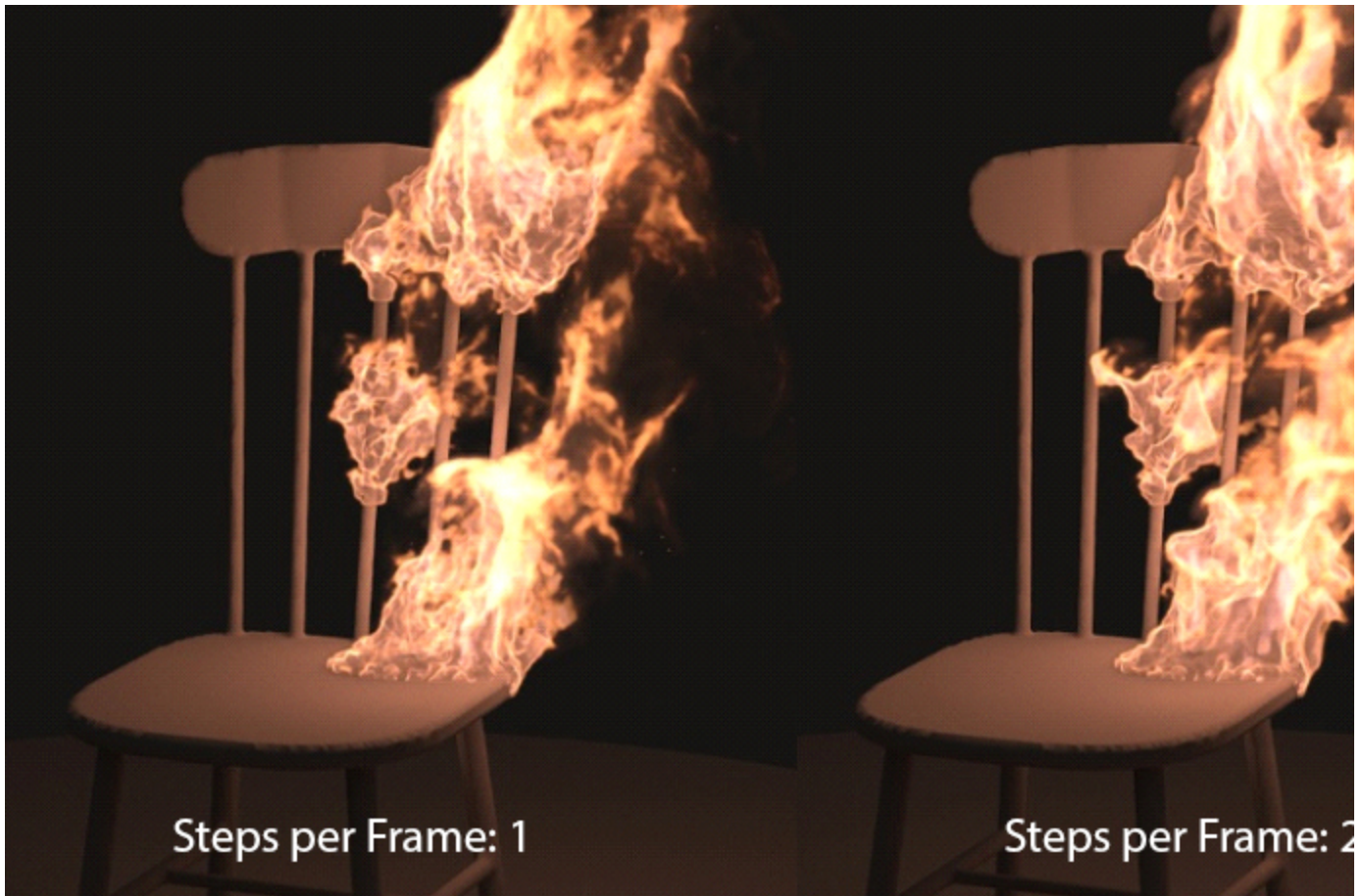


Quality: 10



Backtrace

Forward Transfer



Here's the results with the **Buffered Conservation Method**, **Quality** of **40** and **Uniform Density** disabled.

To add additional realism to the simulation, we use a **Phoenix Plain Force** to simulate the effect of wind.

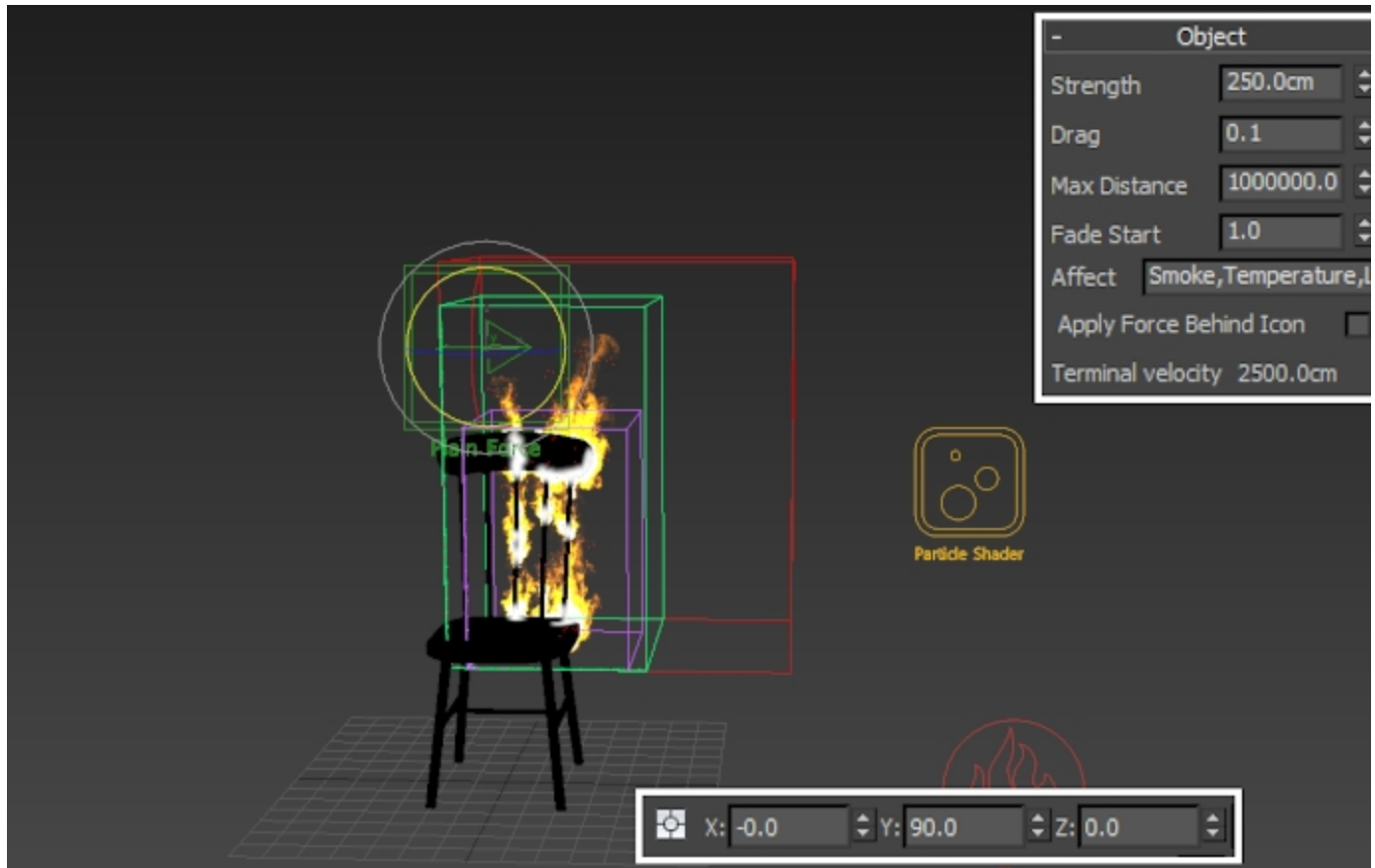
Go to the **Helpers** tab **Phoenix** and add a **Phoenix Plain Force**. The Plain Force is a simple directional force.

Rotate it so it points in the **positive X direction** and set the **Strength** to **250**.

To see how a force will affect the simulation, add it to the **Forces** list under the **Preview** rollout of the **Phoenix Simulator**. Then, **enable Forces**.

This will generate a collection of arrows showing you the direction of the force. You can control the density of the arrows with the **Detail Reduction** parameter at the top of the Preview rollout.

Note: if GPU preview is enabled, the Forces preview will not show up in the viewport.



After adding the **Plain Force**, the fire is blown to the right instead of going straight up. Some extra detail in the flames is also revealed.

You could experiment with other forces as well. A **Phoenix Turbulence** force with a **low Fractal Depth** (so that the noise is not grainy) could give you a good result as well.

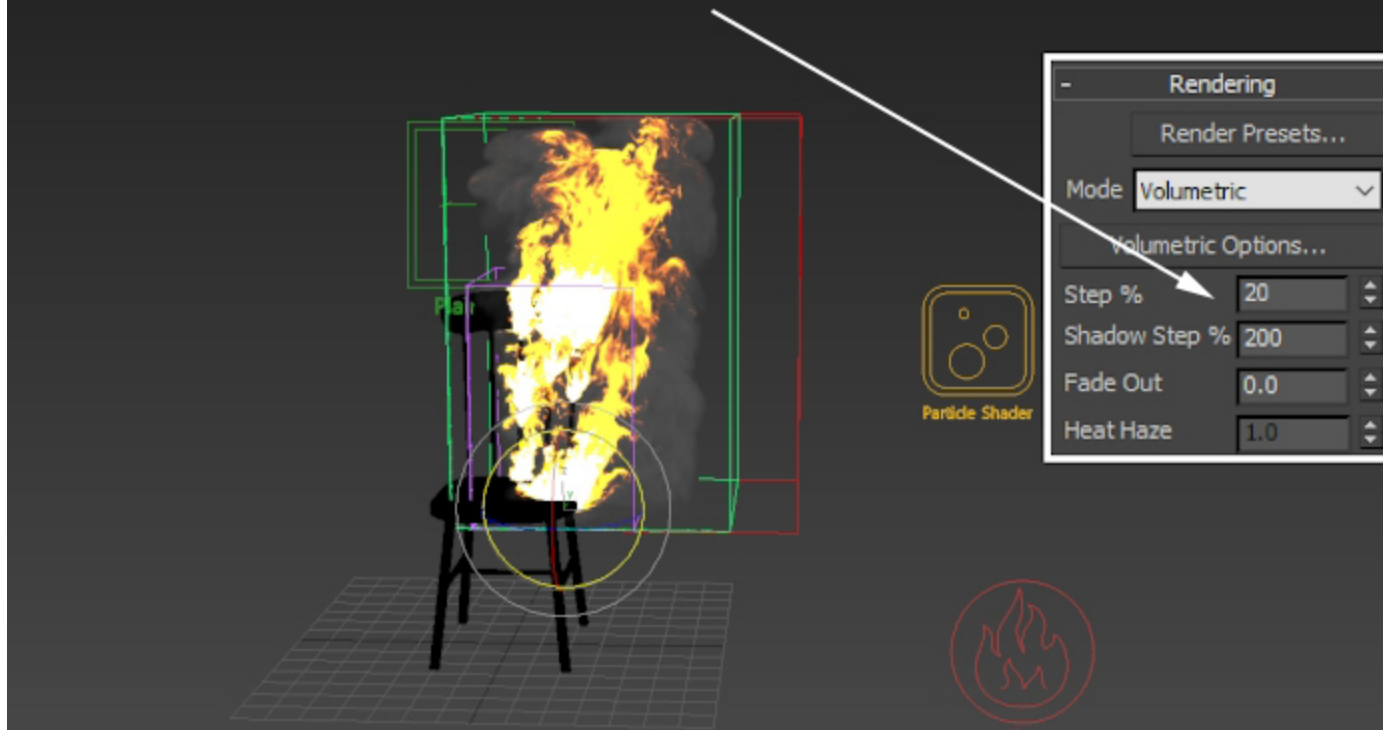
Phoenix Render Settings

Open the **Rendering** rollout. The Rendering rollout contains all the shading and rendering parameters necessary for a Fire/Smoke simulation.

You can gain a decent rendering speed-up by increasing the **Step%** parameter in the Rendering rollout during the look development stage.

The **Step%** controls the ray marching step of the camera rays as a percentage of the cell size. As the renderer traces rays through the simulator, this value controls how often to get information from the grid. If this value is more than 100, some cells will start getting skipped and artifacts may appear but rendering will be faster. For the final render, reduce this value back to 20 as the toolbar preset set it.

You may increase the Step % during the LookDev stage for faster rendering.
Set it back to 20 or lower for the final render.



Here's how a rendered image looks before any tweaks of the Volumetric Shader.

As you can see, the fire is too bright, and detail in the flames can hardly be noticed.

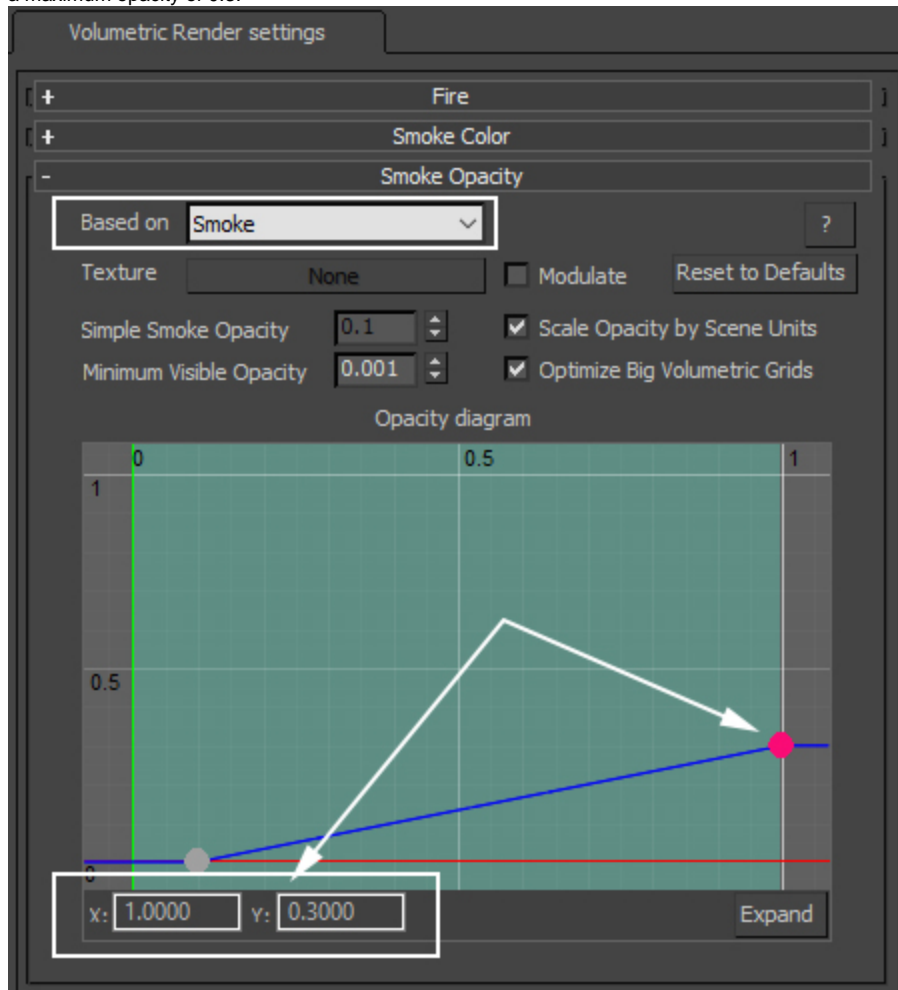


Click the **Volumetric Options** button to access the Volumetric Shading settings.

Open the **Smoke Opacity** rollout and set **Based on** to **Smoke**.

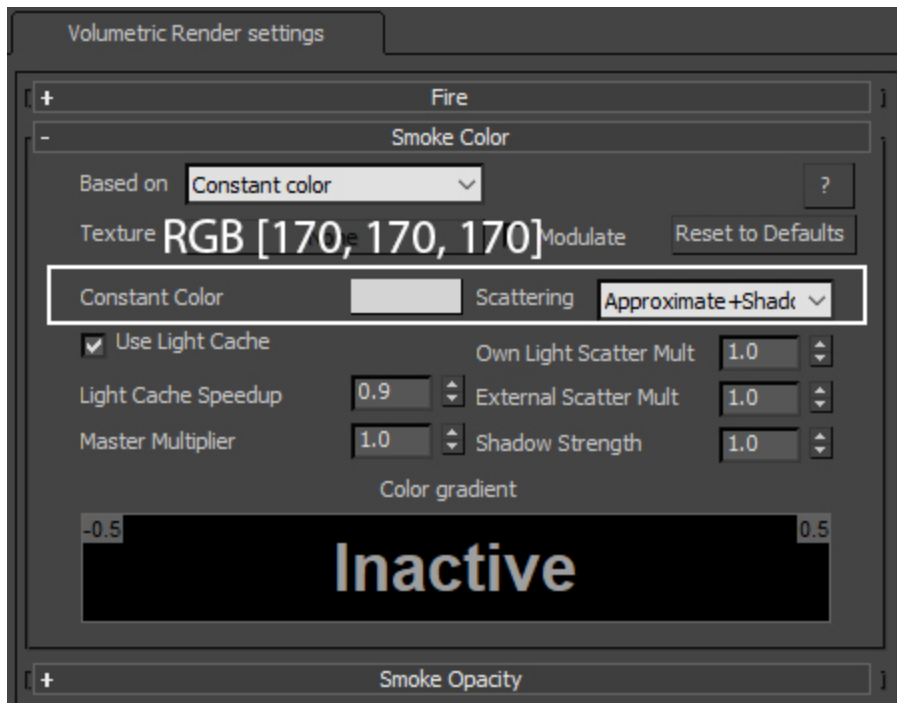
Set the second curve point in the **Opacity Diagram** to **X: 1.0, Y: 0.3**.

This will tell the Volumetric Shader to render the smoke such that the densest areas of the Smoke Channel will render with a maximum opacity of 0.3.



Open the **Smoke Color** rollout and set the **Constant Color** parameter to **RGB (170, 170, 170)** to make the smoke brighter.

Change the **Scattering** parameter to **Approximate+Shadows**. Scattering controls how the light rays are scattered inside the volume. The Ray-traced option tends to be much slower to render but physically-accurate while the Approximation options use internal Phoenix algorithms to speed up the rendering process. The difference between the Approximate and Approximate+Shadows is that the latter also affects the strength of shadows cast by the volume over the scene geometry.



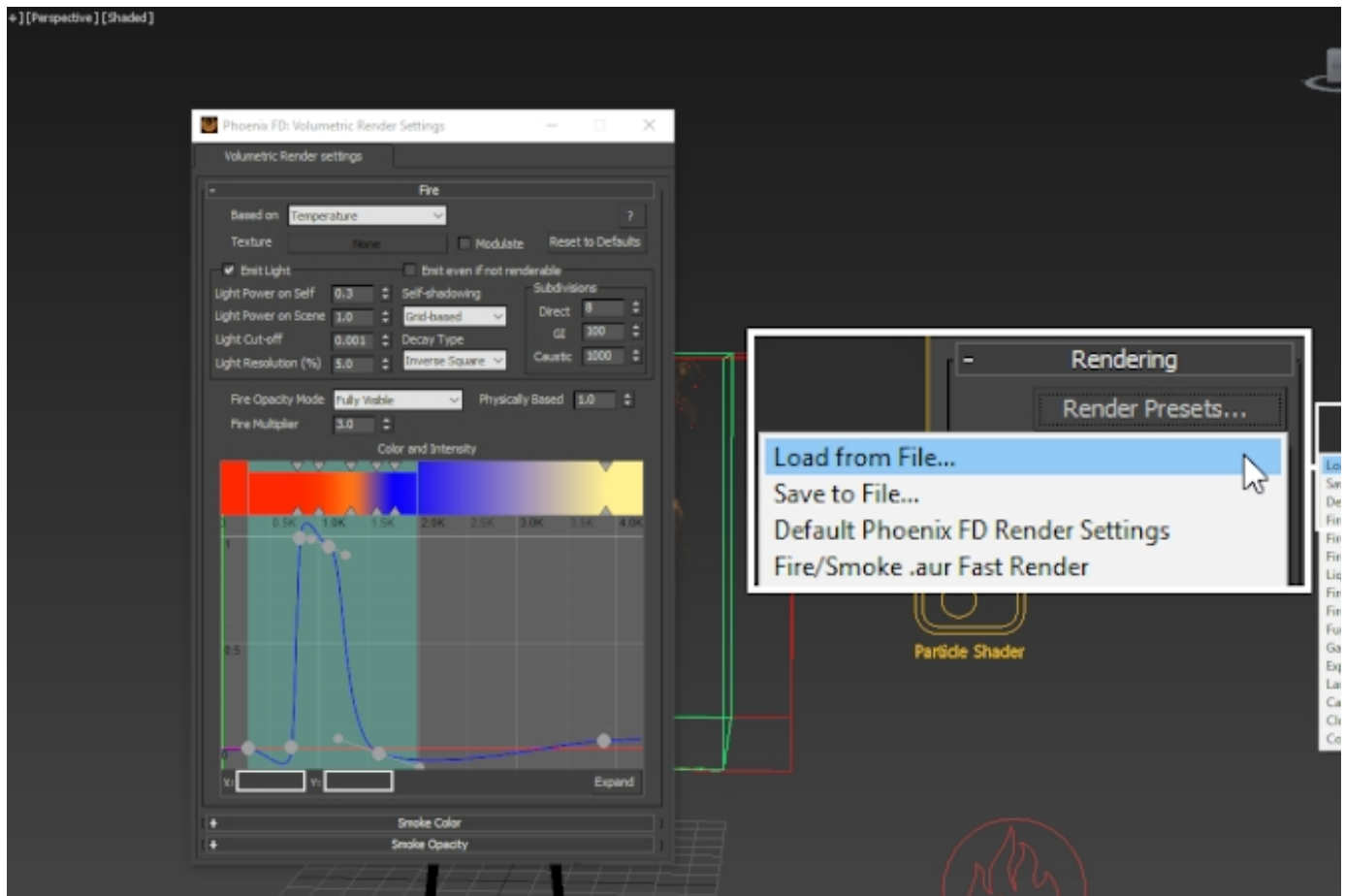
Open the **Fire** rollout. This section controls the emissive (fire) color of the Volumetric Shader and the light emitted by the simulator. Unlike the Smoke Color which needs an external light to become visible, the fire will be visible immediately.

For this tutorial, the **Fire Opacity Mode** is set to **Fully Visible**, with a **Fire Multiplier** of **15**. Reduce the **Light Power on Self** to **0.3**.

The process of setting up the **Intensity and Color Ramp** will usually have the greatest impact on the shading of your fire simulation.

As you can see from the screenshot, there are a dozen or so manually placed colors on the ramp, and the curve has been edited too. Since it's not feasible to describe this in a tutorial, we provide you with a **Render Preset** file that you can load in your scene to get the exact same settings.

To **load** the provided **preset**, go to the **Rendering Render Presets button** and select the *Phoenix_Burning_Chair_Flames.tpr* preset file that came with the project files for this tutorial.



Here's how a rendered image looks with the Volumetric Shader tweaks.



For a more realistic look to the fire, we'll add a **V-Ray Distance texture** so that the flames near the chair disappear.

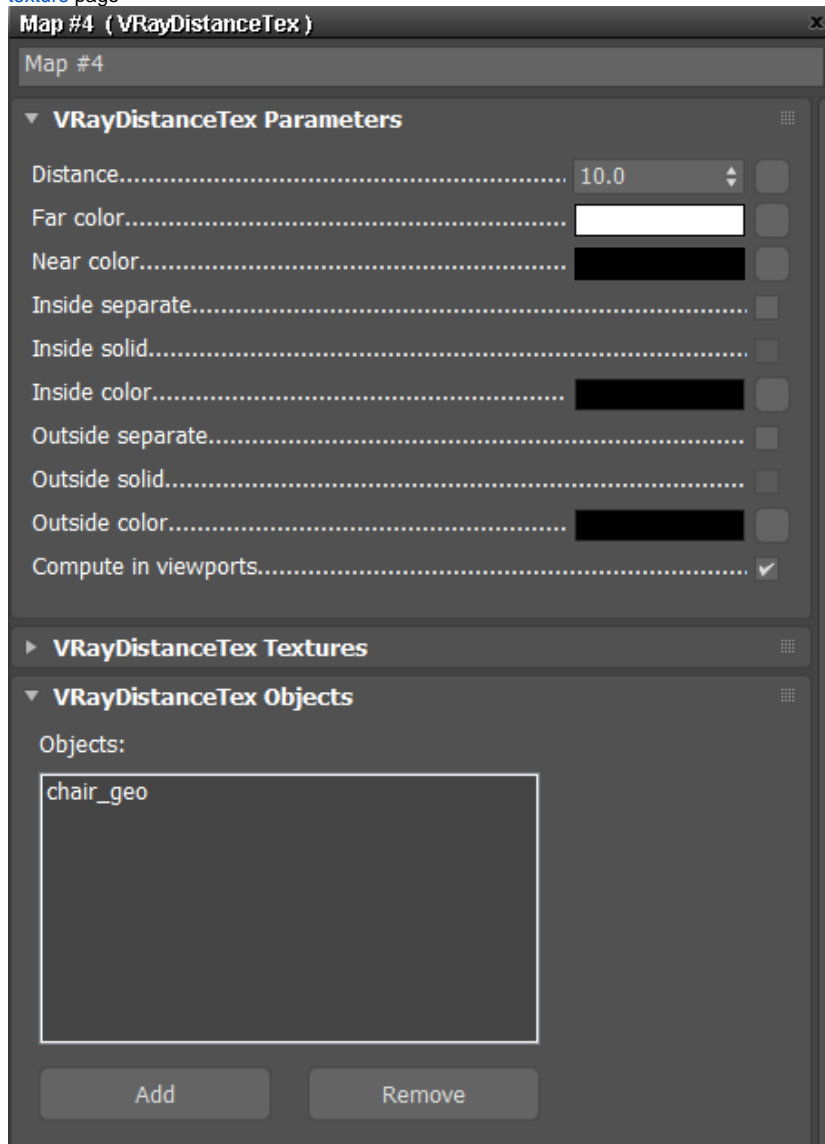
Open the **Material Editor** and create a **V-Ray Distance** texture. This texture is a V-Ray specific procedural texture that returns a different color based on a point's distance to an object(s) specified in a selection list.

Keep the **default settings** as they are.

Then, add the *chair_geo* to the Objects list.

In our case, the **Near Color** is set to **Black (0, 0, 0)** and the **Distance** is set to **10**, which means that the parts of the fire that are within a radius of 10 will not be rendered, and all parts that are at a distance greater than 10 will be rendered.

For more information, please check the [V-Ray Distance texture](#) page



Next, open the **Volumetric Render Settings window Fire rollout** and add the **V-Ray Distance** texture in the **Texture** slot.

Then, enable **Modulate**. Enabling this option multiplies the Based on channel by the V-Ray Distance texture placed in the Texture slot.



Volumetric Render Settings

▼ Fire

Based on Temperature ?

Texture Map #4 (V-RayDistanceTex) ☒ Modulate Reset to Defaults

☒ Create Fire Lights ☐ Create Lights Even If Not Renderable

Light Power on Self 0.3

Light Power on Scene 1.0

Light Cut-off 0.001

Light Resolution % 5.0

Self-shadowing Grid-based

Decay Type Inver...quare

Subdivisions

Direct 8

GI 100

Caustic 1000

Fire Opacity Mode Fully Visible

Physically Based 1.0

Don't Use Opacity Texture

No Map

Fire Multiplier 15.0

Here's how a rendered image looks with the V-Ray Distance texture.

The fire still looks a bit noisy - the Resimulation will take care of that.



Retiming the Phoenix Simulation

Suppose you want to slow the simulation down twice.

To achieve that, you need to take care of two things:

1. You need to slow down the **motion of the objects** in your scene.

2. You need to slow down the **Phoenix simulation** by reducing the **Play Speed** from the **Input rollout** of the **Simulator**.

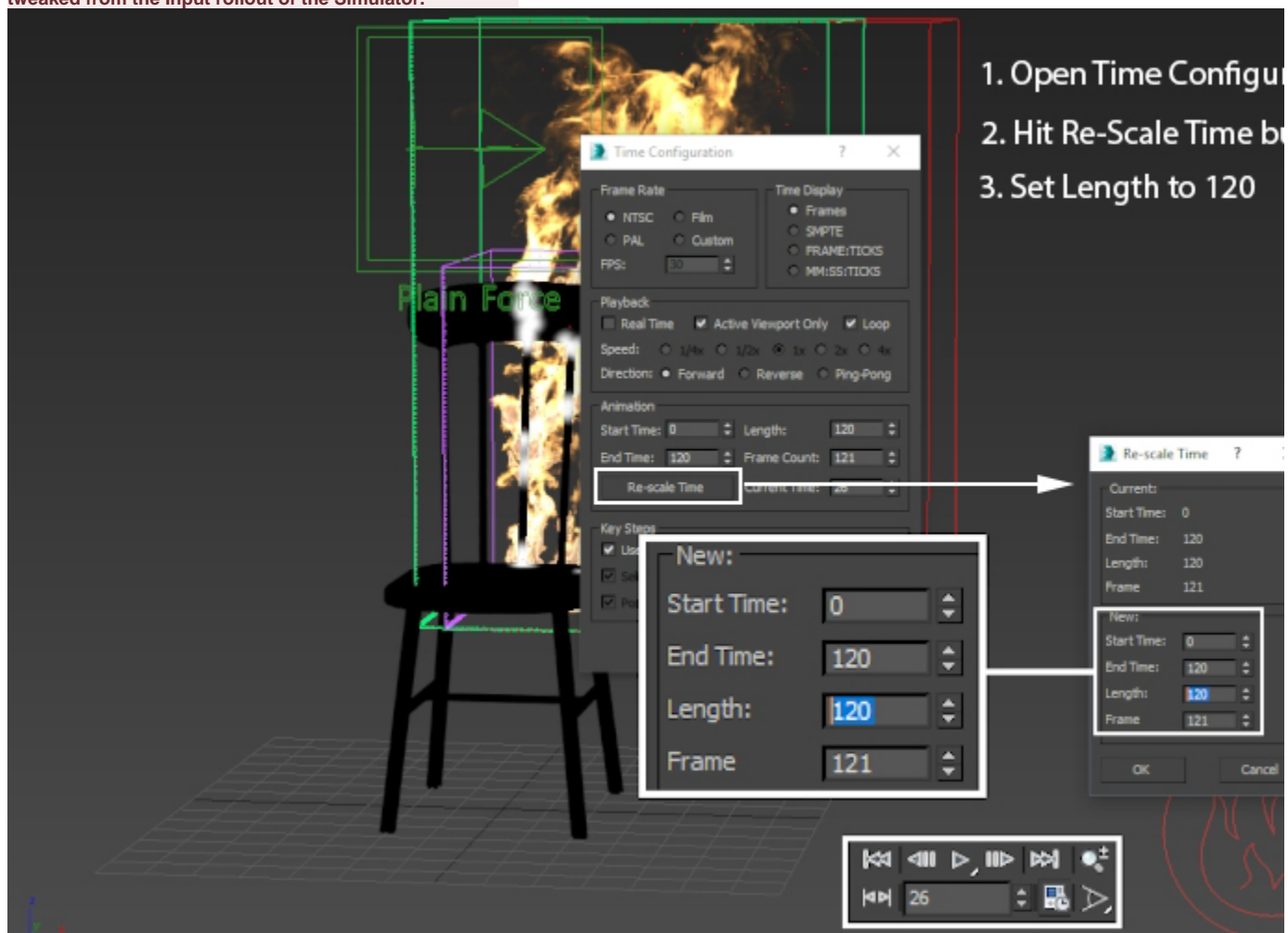
To slow down the animation of your scene objects, open the **Time Configuration** window, click the **Re-scale Time** button and set the **Length** to **120** (assuming you are working with the provided scene file which has a total animation length of 60).

The total length of the animation of will now be doubled.

It's very important to stress that this will not affect the Phoenix Simulation.

If you have 60 frames simulated, Re-Scale the timeline to 120 frames and hit the Play button, the simulation will play from frame 0 to frame 60, just like before the Re-Scaling.

The play speed of a Phoenix simulation can only be tweaked from the Input rollout of the Simulator.



Now that the animation speed of the scene objects is reduced in half, we need to do the same for the **Phoenix simulation**.

Open the **Input** rollout and set the **Play Speed** to **0.5**.

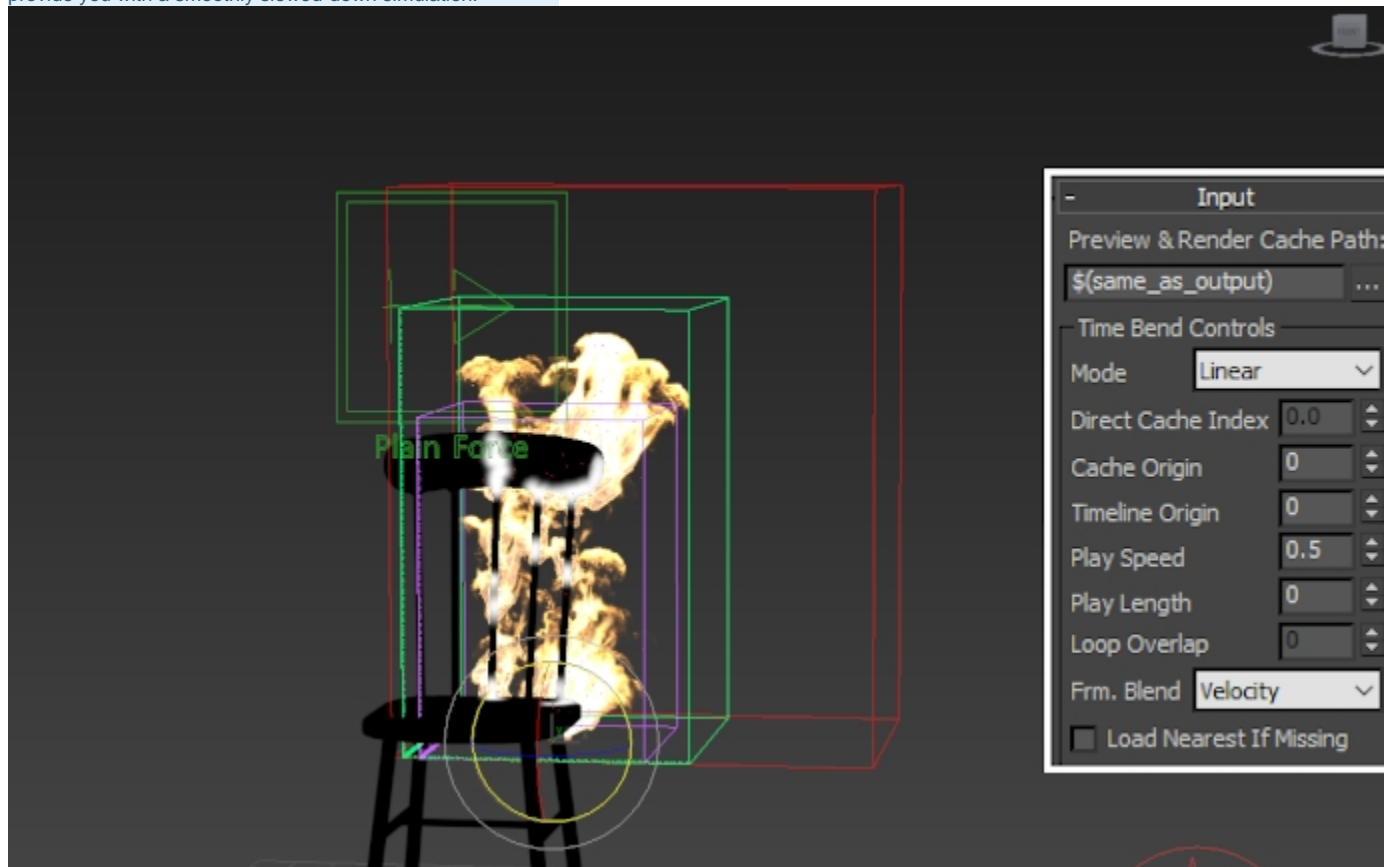
Set the **Frame Blending** method to **Velocity**.

The **Play Speed** parameter works as a multiplier for the Playback speed. When set to 1, it reads one cache file for each corresponding timeline frame. If the value is not precisely 1, Phoenix generates an intermediate frame by blending the two adjacent cached frames.

This **Blending** process works really well with FLIP (water) simulations as FLIP particles are fixed points in space, and consequently, the in-between particle positions can be easily determined from the previous and next frames' cache files.

For Fire/Smoke simulations, however, more advanced techniques are required to achieve a slow-down effect.

Reducing the **Play Speed** is good enough for most Smoke simulations but attempting to blend Fire simulations tends to produce flickering in the fire intensity. The way around this issue is to slow down the simulation from the Input Play Speed and run a **Resimulation**. Since the Resimulation process reads the cache files as specified in the Input panel, this will recalculate the dynamics at the reduced speed and provide you with a smoothly slowed-down simulation.



Open the **Resimulation** rollout and enable **Grid Resimulation** and **Particle Resimulation**.

Note that when you enable Resimulation, Phoenix will try to read the cache files for preview and rendering from the Resimulation Output Path instead of the 'regular' Output. Don't be alarmed if the viewport goes blank - you can always go back to the original cache files by simply disabling the resimulation.

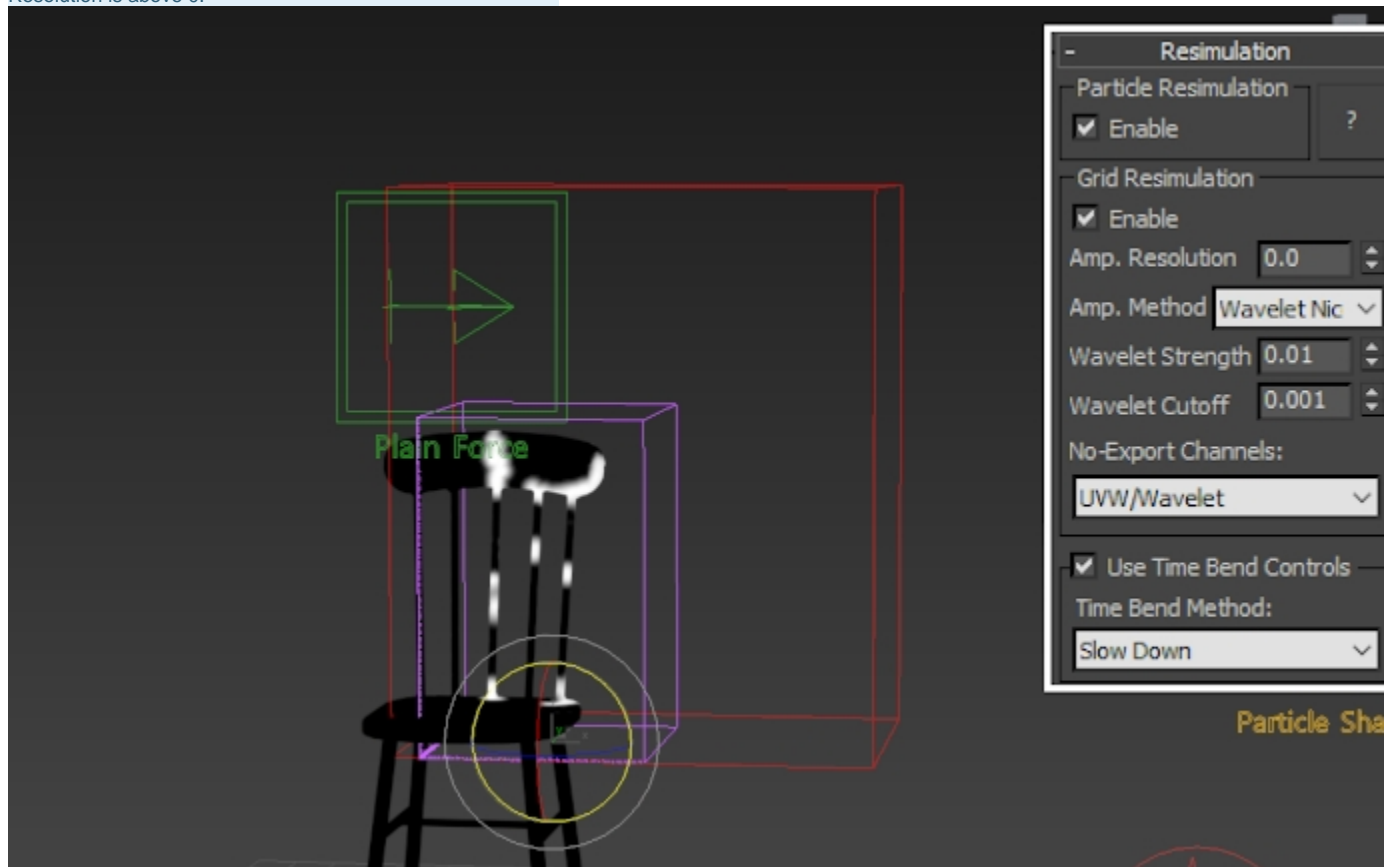
The **Grid Resimulation** takes care of the **Temperature /Smoke/Velocity Grid Channels** while the **Particle Resimulation** will affect the **Drag particles** emitted by the fire source.

Set the **Amp. Resolution** to **0**. This parameter is used to increase the resolution of the grid for the Resimulation process only (just like the Cell Size in the Grid rollout controls the simulator's resolution). Because we simply want to recalculate the cached sequence, we leave the Amplify Resolution at 0.

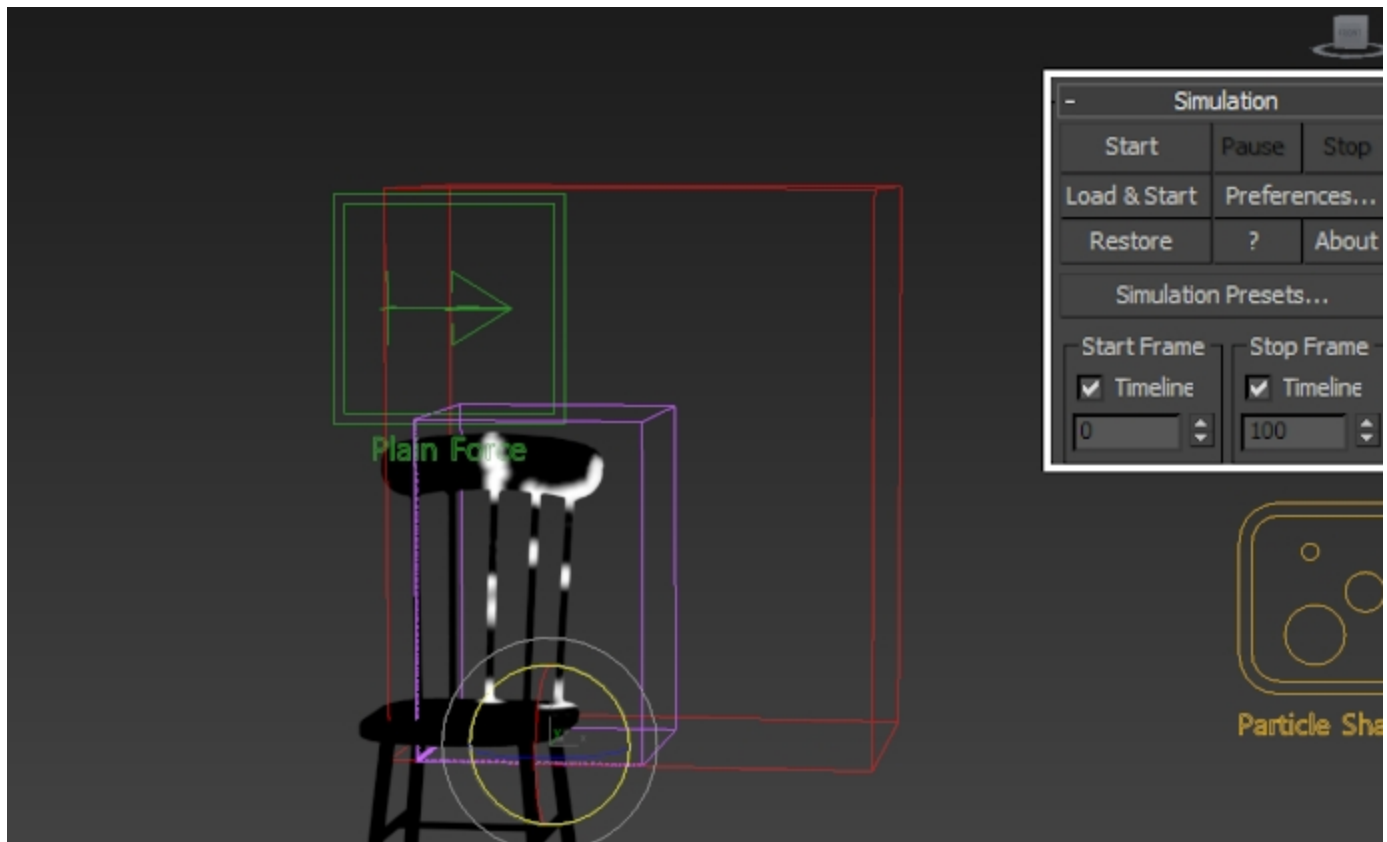
Enable **Use Time Bend Controls** and set the **Method** to **Slow Down**. This option works great when trying to produce a slower version of your base simulation, as is the case with our fire.

Set **Wavelet Strength** to **0.01**. Wavelet Turbulence is used to generate wispy detail in a simulation but we're not looking to add additional details so the Strength is reduced to its lowest possible value.

If you want to use **Wavelet Turbulence**, you need to **Enable Wavelet Grid channel export** from the **Output** rollout of the simulator. Note that it will take effect only if Amplify Resolution is above 0.



Hit **Start** to begin the Resimulation.



You can use the **Amp. Resolution** option to add extra detail to the simulation during the Resimulation process.

The Amplify Resolution setting controls the number of voxels that will be used for the grid during Resimulation.

Here's the simulation, resimulated with **Amp. Resolution** of 1.

Camera and Lighting

As mentioned in the beginning, the only illumination source for this scene is the fire itself.

The scene file called *Phoenix_Burning_Chair_START_3dsMax2015.max* already has the cameras set up for you.

However, for reference, here are the *physCam_far* camera properties:

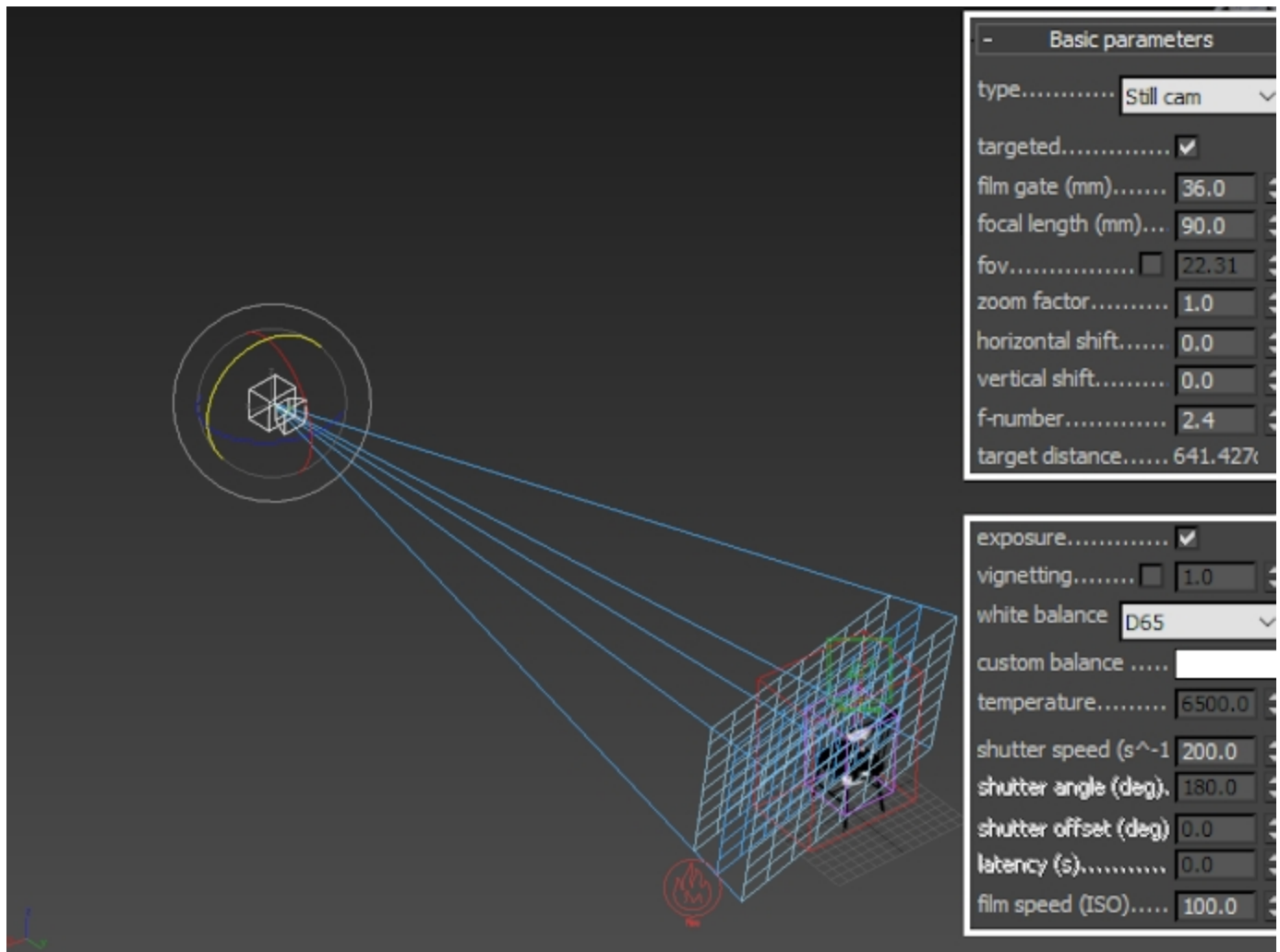
A **V-Ray Physical camera** is used for the rendering with the following settings:

F-Number: 2.4;

Shutter Speed: 200;

ISO: 100;

Focal Length: 90.



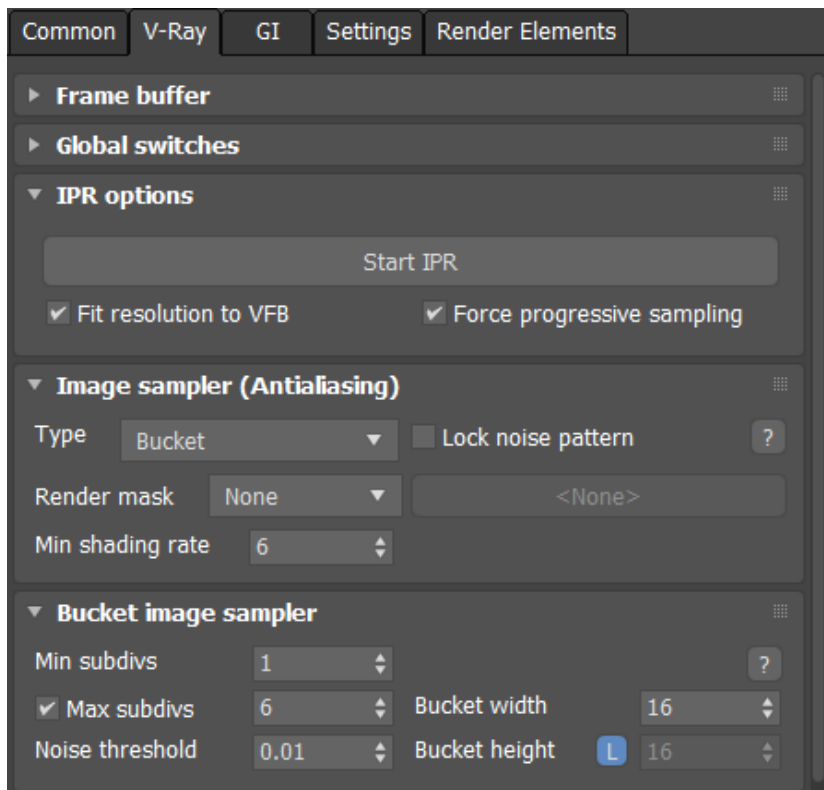
Render Settings

In the Render Setup tab:

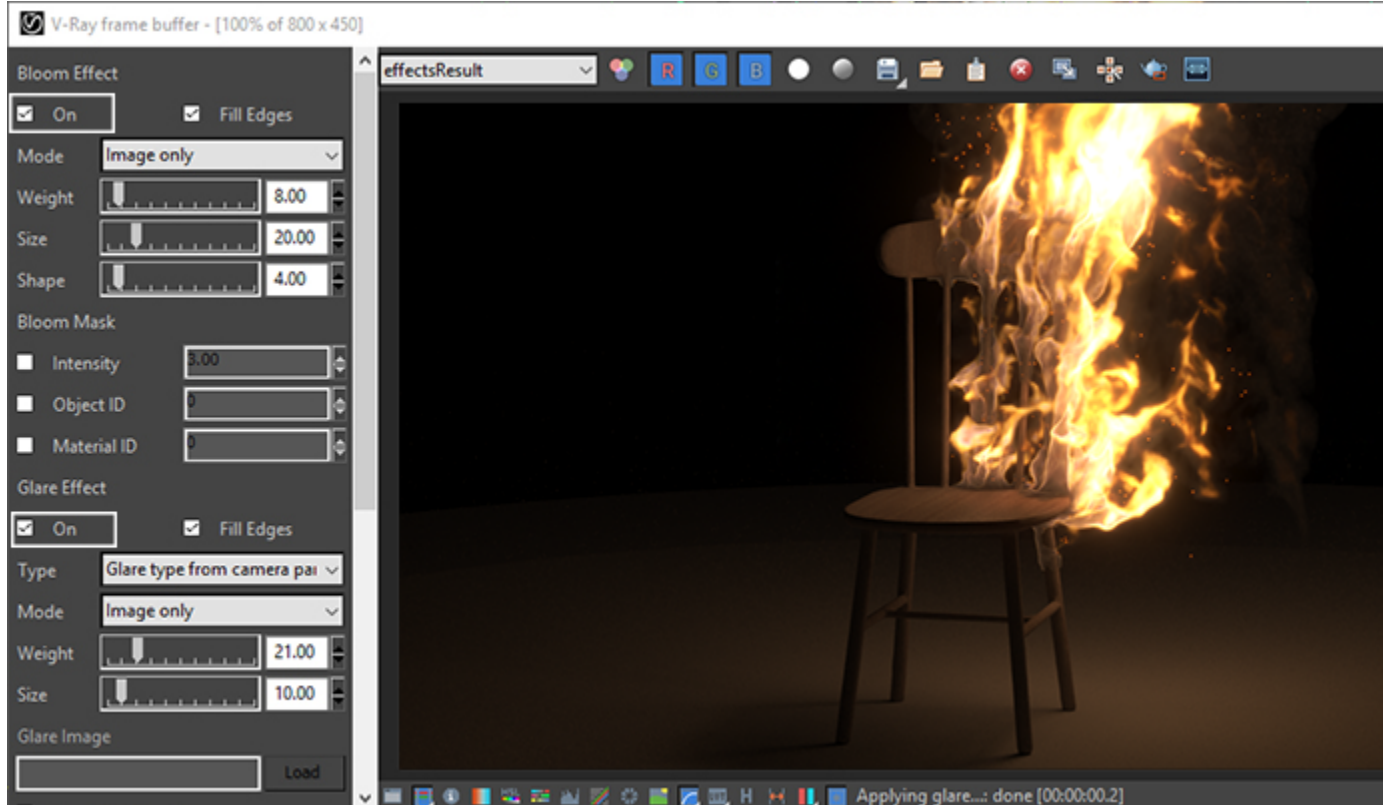
The **Image sampler Type** is set to **Bucket**.

The **Max subdivs** are set to **6**.

The **Bucket width** is set to **16**.



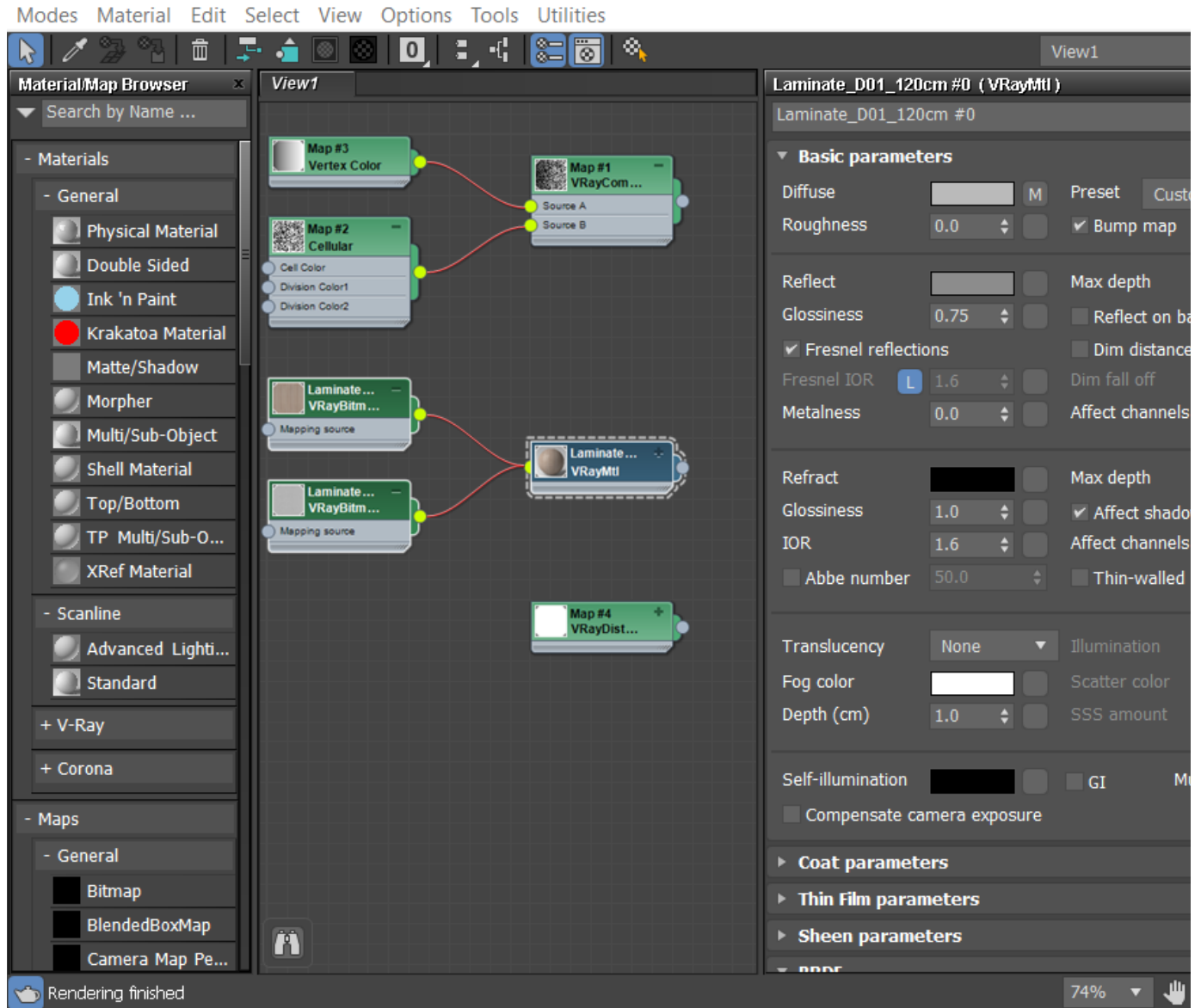
For a nicer looking image, you can enable **Bloom and Glare effects** from the **Lens Effects panel** in the **V-Ray Frame Buffer**.



For the final render, we use a wood material called *Laminate_D01_120cm* for the chair geometry. The material can be found in the **V-Ray Material Library**.

You can also experiment with other materials.

3 Slate Material Editor



And here is the final result.