

Basic Liquids QuickStart

This page provides a step-by-step guide for creating a simple liquid simulation using Chaos Phoenix for 3ds Max.

Overview

This is an Entry Level tutorial which requires no previous knowledge of Phoenix. A basic understanding of 3ds Max would be helpful but is not a prerequisite for being able to follow along.

This tutorial is created using **Phoenix 4.30 Official Release** and **V-Ray 5, Hotfix 2 Official Release** for 3ds Max 2018. 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 setting up a basic liquid simulation in Phoenix for 3ds Max. By the end of the tutorial, you will be able to create your own liquid simulations, understand how the simulation interacts with other objects and will know the basics of editing some of the main settings of the Phoenix Liquid Simulator.

This tutorial is a companion to go along with the QuickStart video posted on our YouTube channel.

The **Download** button below provides you with an archive containing the start and end scenes.

[Download Project Files](#)

The video is created using Phoenix 3.0, but the text version of the tutorial is updated and uses Phoenix 4.30 Official. In any case of doubt you may refer to the text.

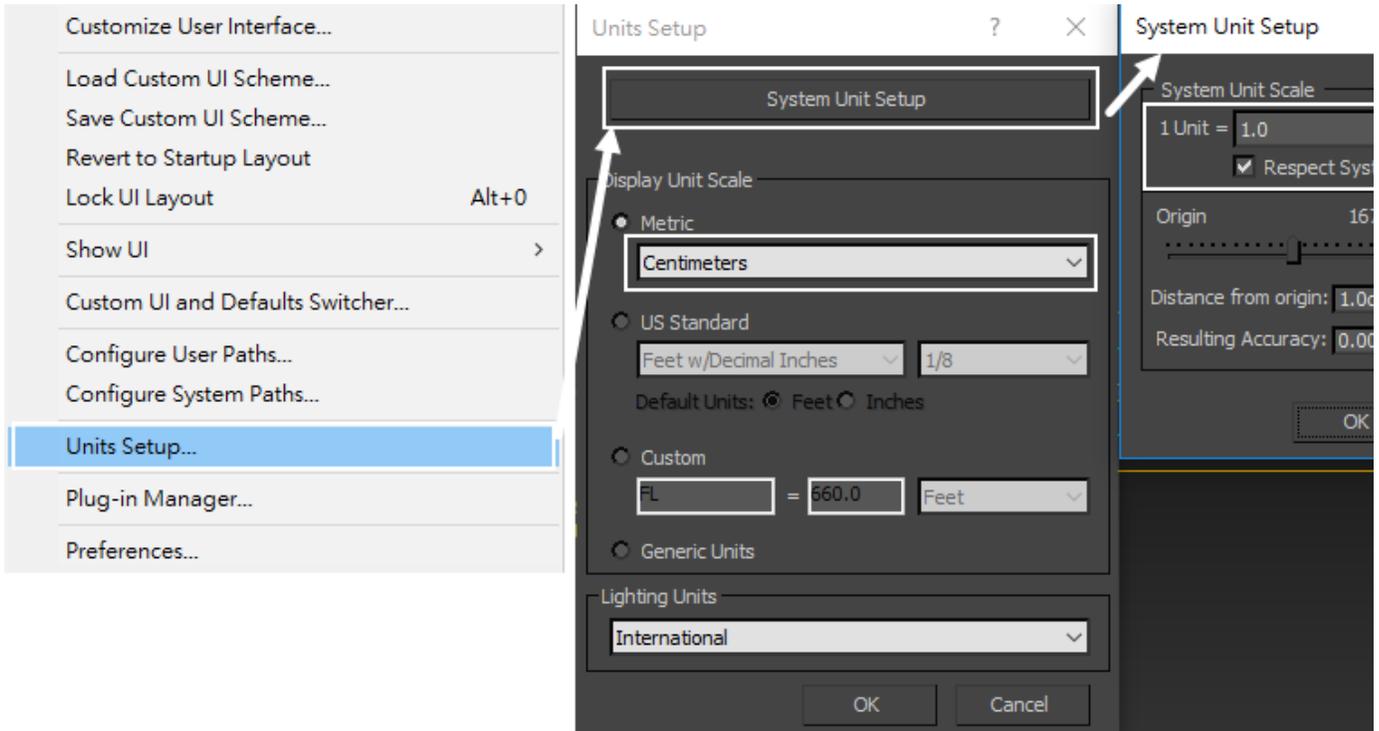
System Units Setup

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.

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

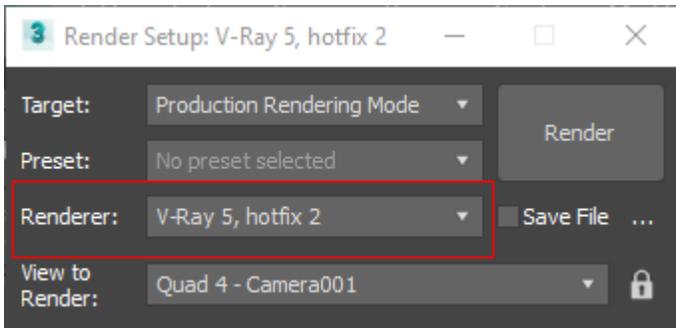
Also, set the **System Units** such that **1 Unit** equals **1 Centimeter**.



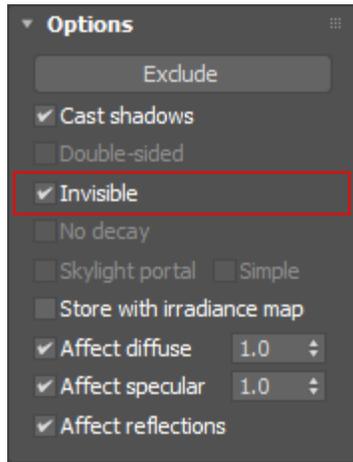
The units do not have to be set to centimeters to get the simulation to work. What's important is that the emitter (in this case, the sphere) is set to the proper real-world scale so that the simulation is correct for that situation. This is how Phoenix is able to know if it's simulating water in a glass or a boat on ocean waves.

Lightning and Rendering

To prepare your scene for rendering a simulation make sure V-Ray is set as your Renderer in the **Render Setup** window.



Add a **Dome Light** to your scene and set it to be **Invisible**.



Scene Setup

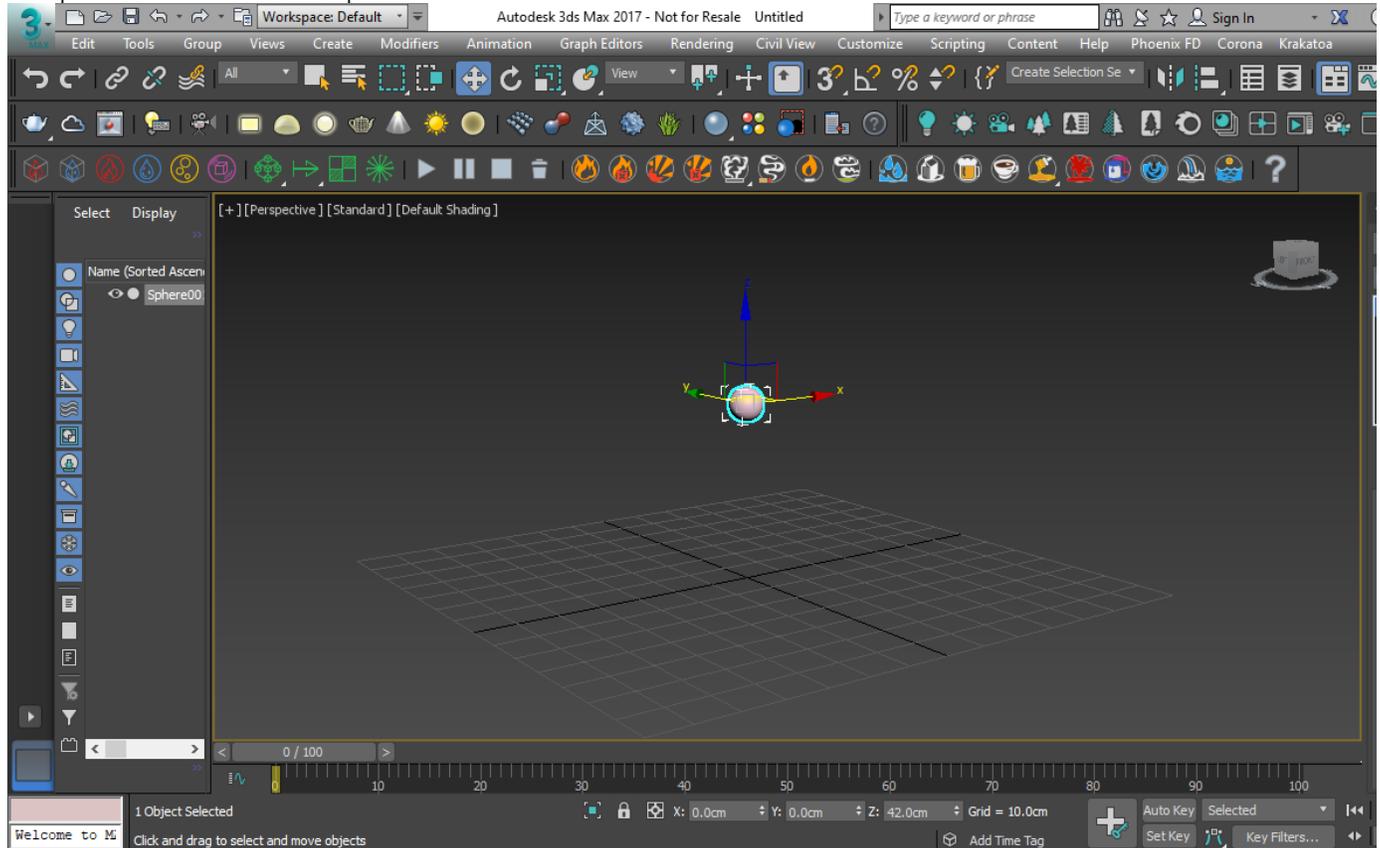
In the following steps we will show how to create a basic liquid simulation with the Phoenix Tap Water Preset.

After that we will explain how to set the simulation manually step-by-step for greater flexibility and control.

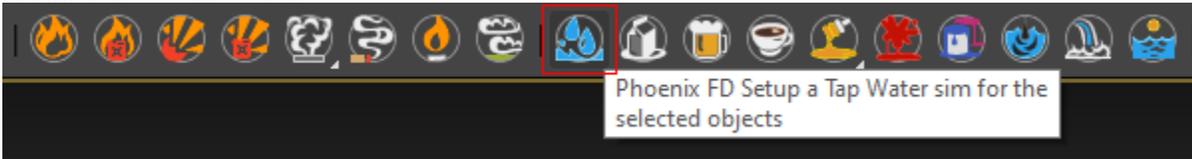
Tap Water Preset

We will begin by using a quick water preset.

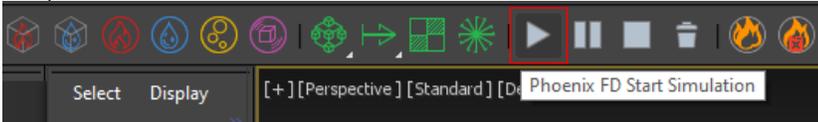
Create a **Sphere** with a **Radius** of about **4.0** centimeters.
This sphere will be the source of the liquid.



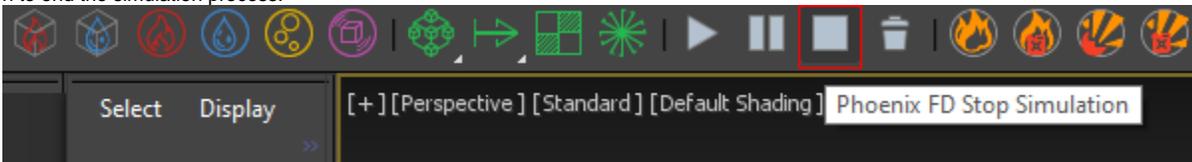
With the sphere selected, click on the **Tap Water Preset** button from the **Phoenix Toolbar**.



Now press the **Start Simulation** button in the Toolbar.



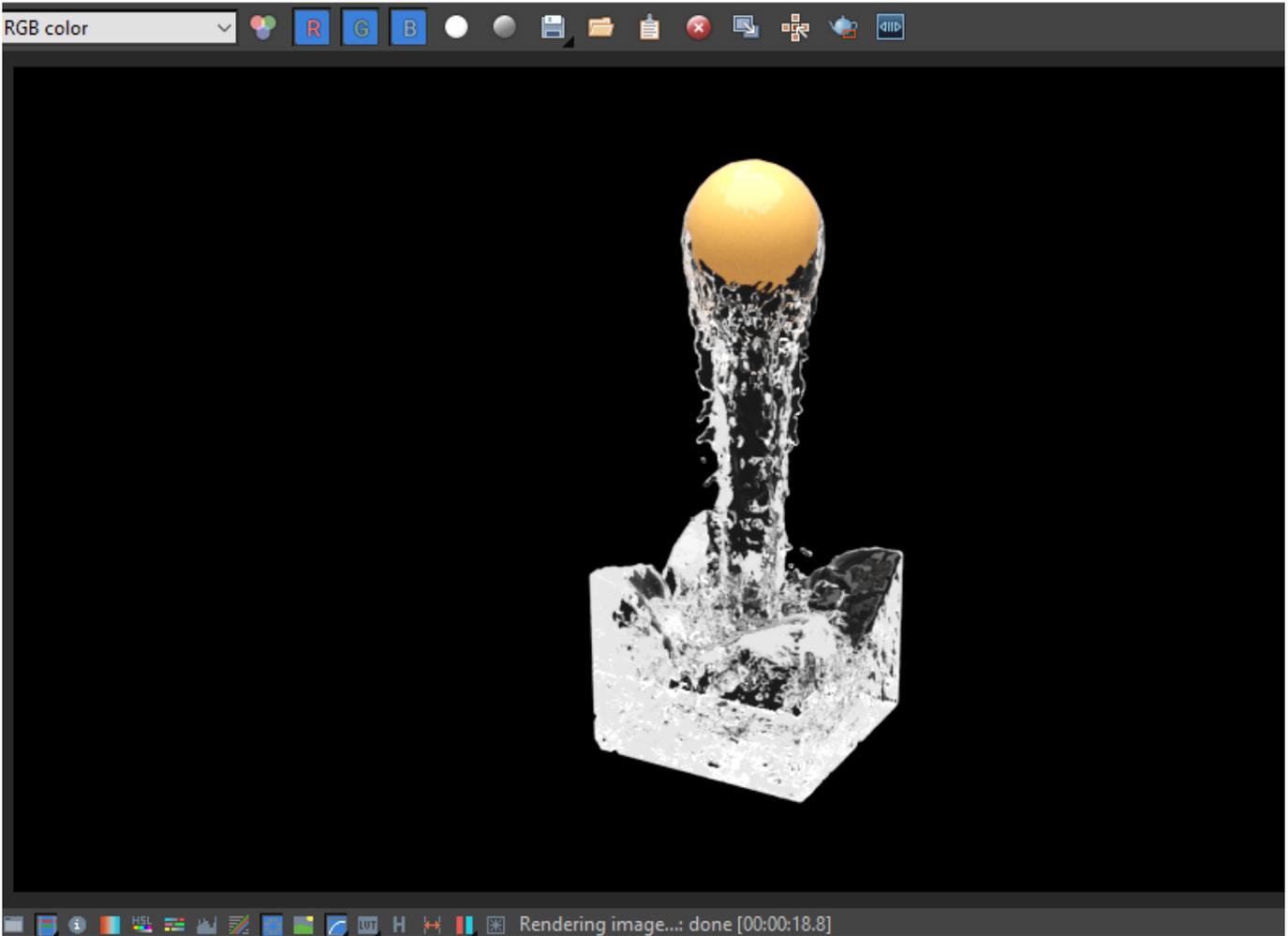
Let the simulation run for a few seconds and you will have a chance to see the results in the viewport, click the **Stop** button to end the simulation process.



Render your current frame to see the simulation results.

Note that the liquid fills the simulation volume container as it flows.

V-Ray frame buffer - [100% of 910 x 512]

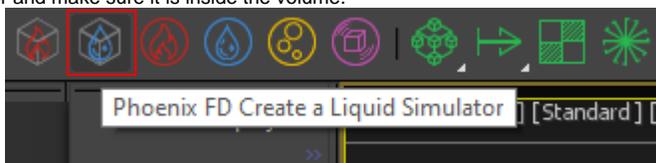


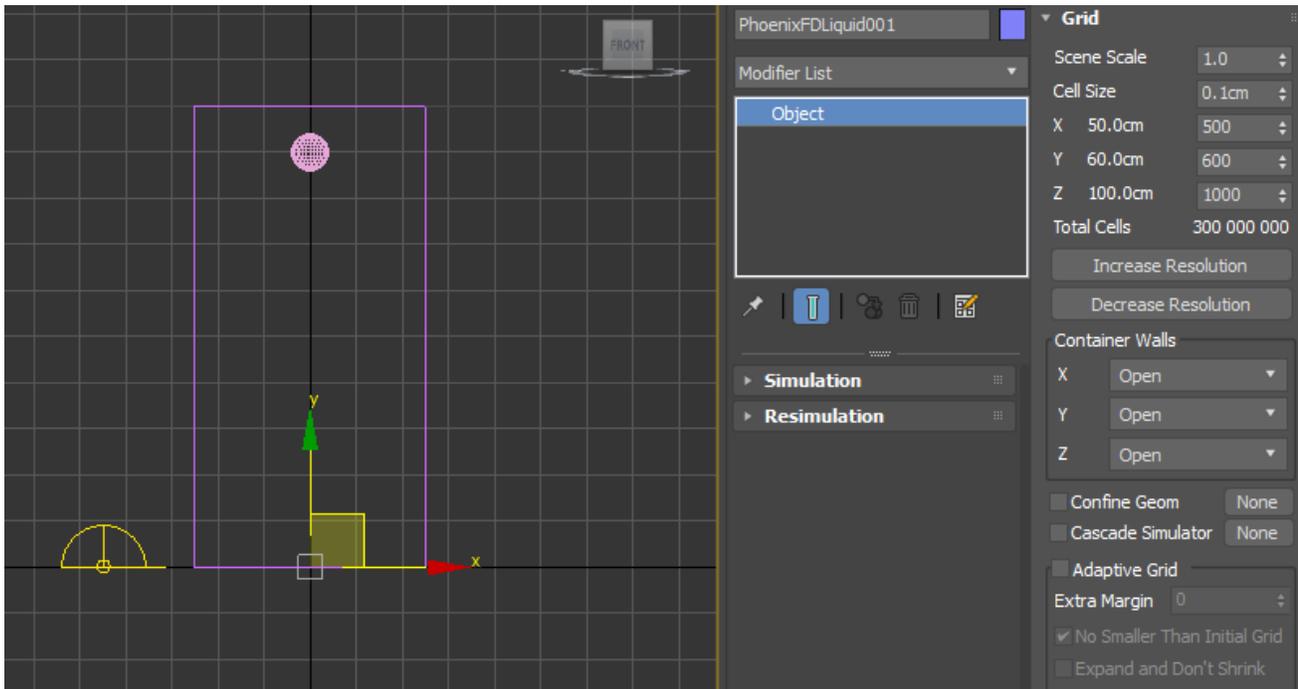
Manual Water Setup

Let's take a look at setting up a water simulation manually.

If you are continuing from the Preset section above, select the Liquid Simulator and Source objects and delete them from your scene. If not, create a **Sphere** with a **Radius** of about **4.0** centimeters.

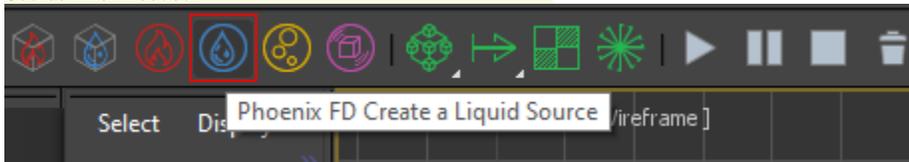
Click on the **Create a Liquid Simulator** button in the **Phoenix Toolbar** and click and drag in the viewport to create a volume to encompass the sphere, while leaving some room for the liquid to fall. Move the sphere near the top of the Simulator and make sure it is inside the volume.

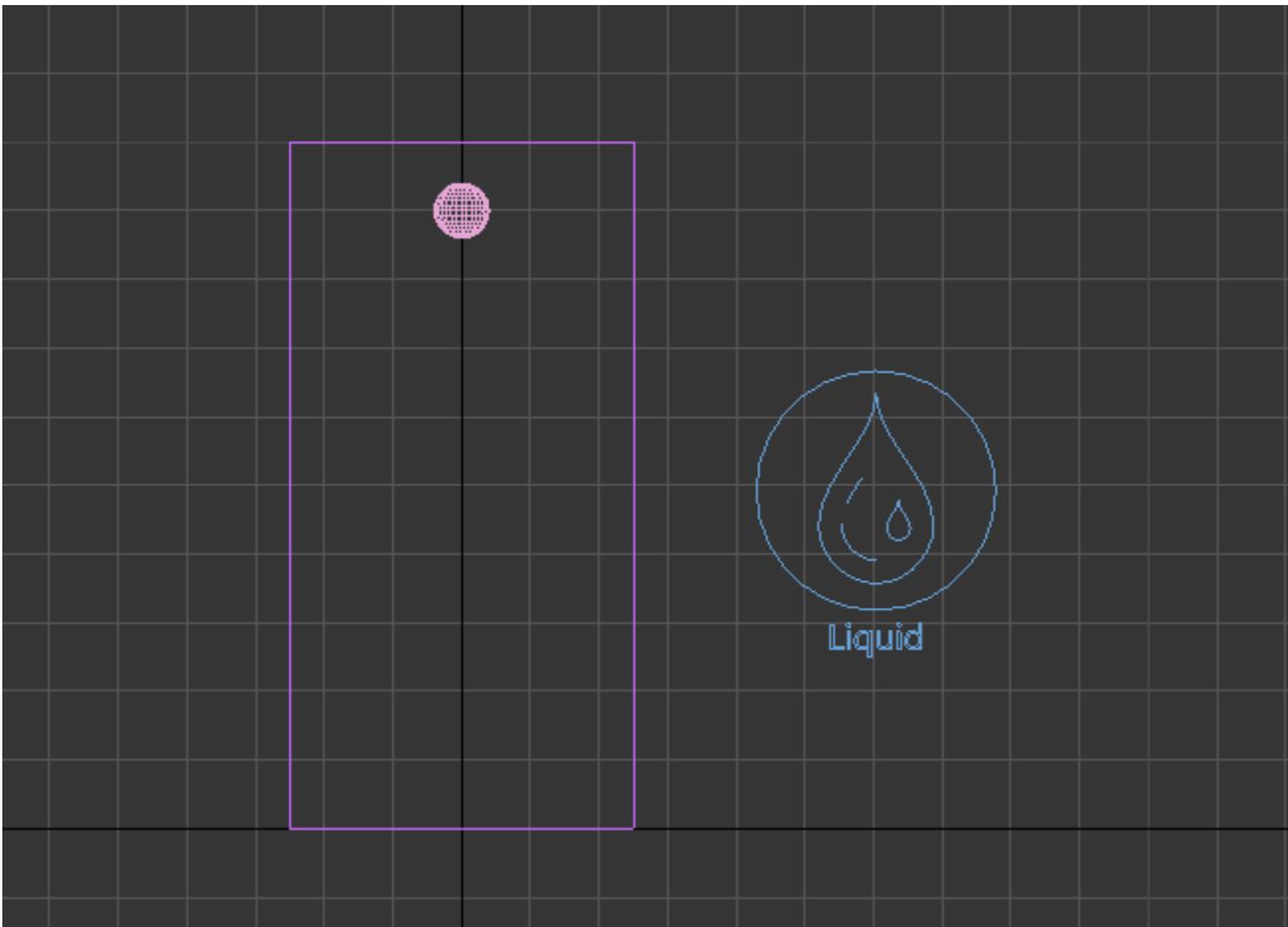




Click the **Create a Liquid Source** button on the **Phoenix Toolbar** and place the source in your scene.

The Source object is a 2D non-renderable icon, so size and placement are only important for being able to select the Source when needed.

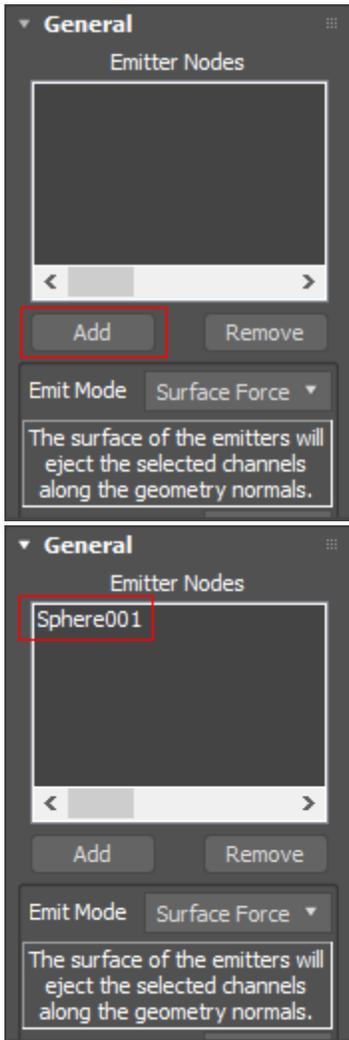




With the **Liquid Source** object selected, go to the **Modify Panel** and click the **Add** button located under the **Emitter Nodes** list, then select the **Sphere**.

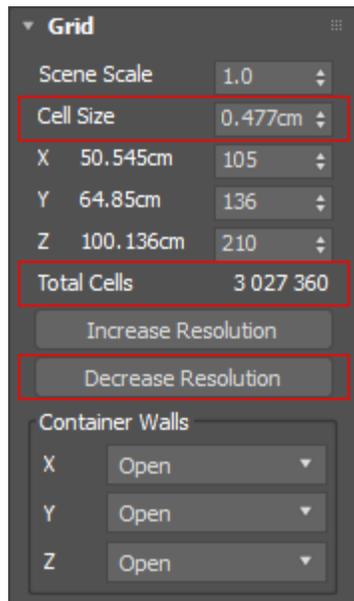
This way the sphere will be selected as an emitter.

The Source node is where we control the Emitter properties, like the Emit Mode and the Outgoing Velocity. The sphere is the actual emitter of the fluid inside the Simulator.

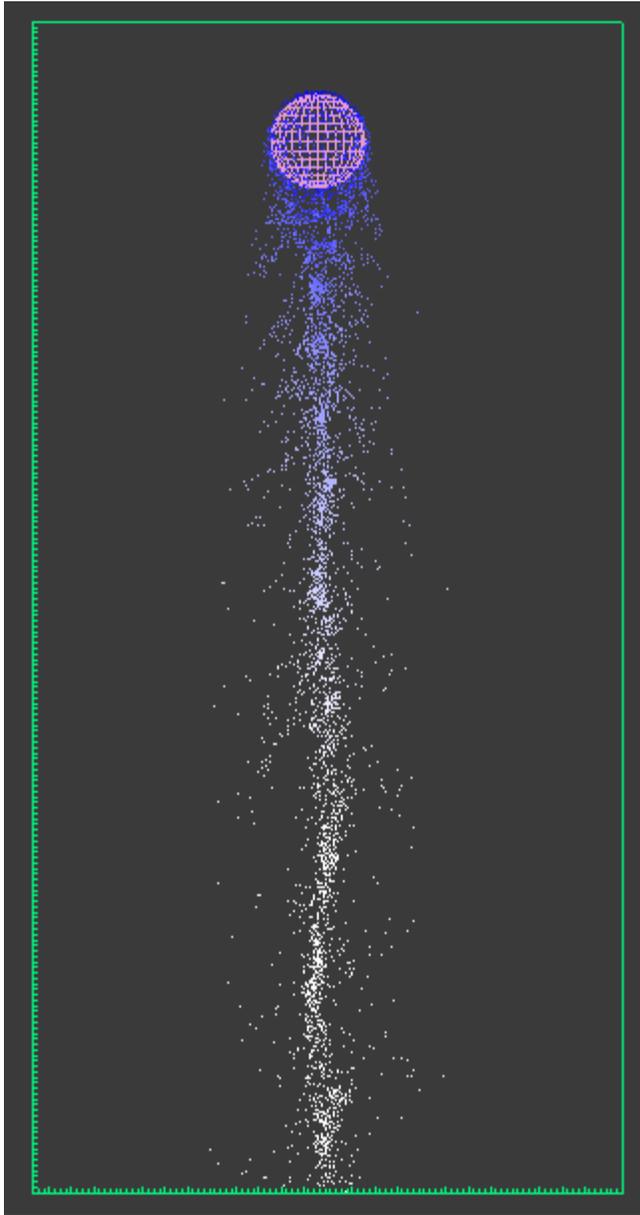


Select the Simulator. In the **Modify Panel** under the **Grid** rollout are located the parameters to control the resolution of the volume grid, which greatly affects the look and performance of the simulation.

Click **Decrease Resolution** until the **Total Cells** value is around **3-4 million** cells. You can also set the **Cell Size** to change the number of **Total Cells**.



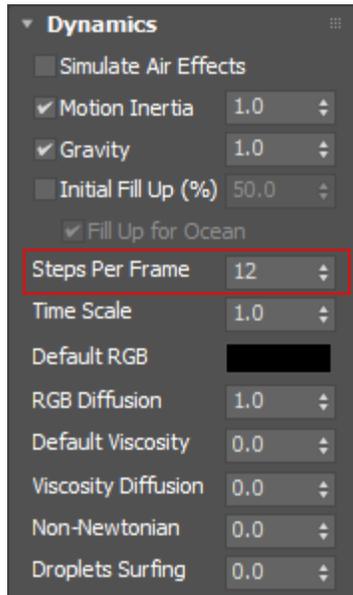
Click the **Start Simulation** icon in the Phoenix Toolbar to confirm that the sphere is emitting the liquid inside the Simulator.



Now let's get this simulation to look more like what we had with the preset.

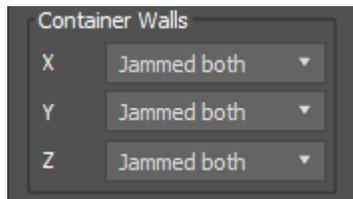
Right now the edges of the pour are pretty jagged.

Expand the **Dynamics** rollout and adjust the **Steps Per Frame** to **12**. This provides for a more natural look but will make the simulation run slower. The higher the **Steps Per Frame**, the longer the calculations, so try to find the lowest setting for the effect you need.



At this point the liquid is flowing right through the floor and disappearing. This means that the walls of the Simulator are open. Let's close them.

Open the **Grid** rollout, you can see the **Container Walls** are indeed all set to **Open**. Set the **X/Y/Z** values to **Jammed Both**.



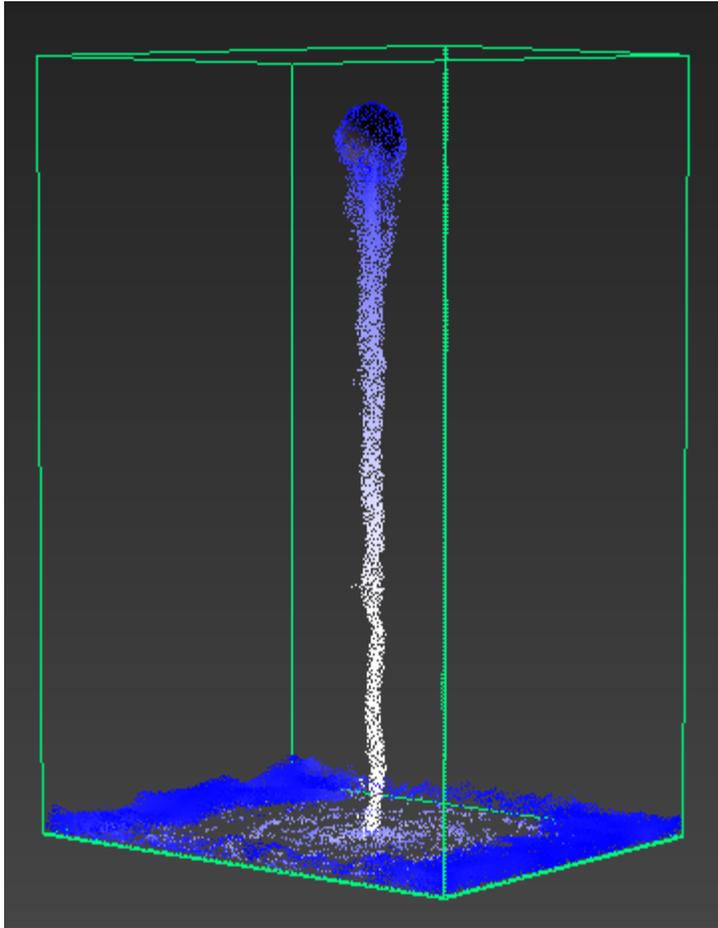
Start the simulation again. As the simulation runs, the liquid now looks smoother and collides with the bottom and the side walls of the Simulator.

It is, however, running slowly, so we can adjust the resolution by increasing the Cell Size or decreasing the Total Cells.

Stop the simulation and navigate to the **Grid** rollout. We currently have 3-4 million cells to calculate, so click **Decrease Resolution** until **Total Cells** amount is about **half a million** cells.

Start the simulation again and you'll see it is running faster. While the simulation is still running, go to the **Dynamics** rollout and set the **Steps Per Frame** to **1** to speed it up some more.

Feel free to experiment with different settings.



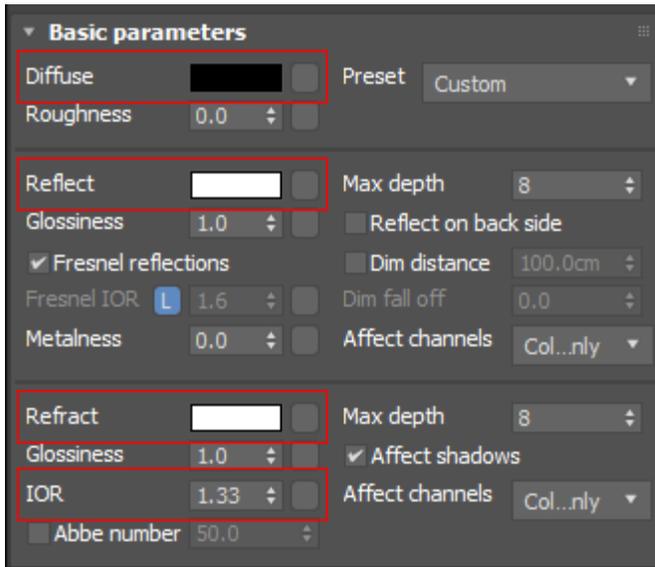
It's time to create a water material and apply it to the Simulator.

Open the **Material Editor** and create a new **V-Ray Material**.

Set the **Diffuse** color to **RGB [0, 0, 0]**.

Set the **Reflect** and **Refract** colors to **RGB [255, 255, 255]**.

Set the **Fresnel IOR** to **1.33**.



Assign the material to the Simulator and render a frame.

Feel free to adjust the material and the lighting in the scene to achieve the liquid look you're going for.

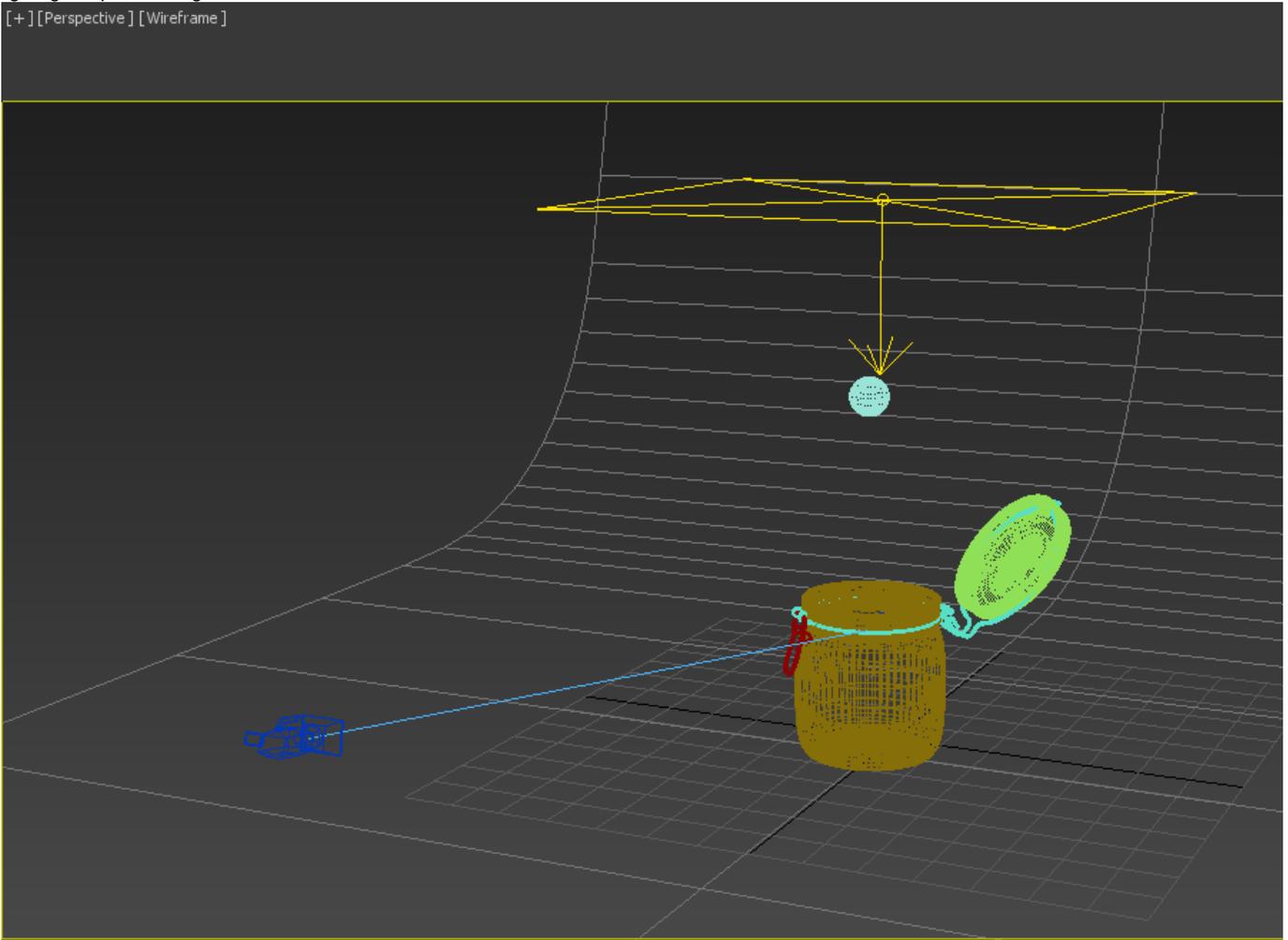


Example Scene

Now let's make the liquid collide with and fill an actual object rather than just the Simulator's grid.

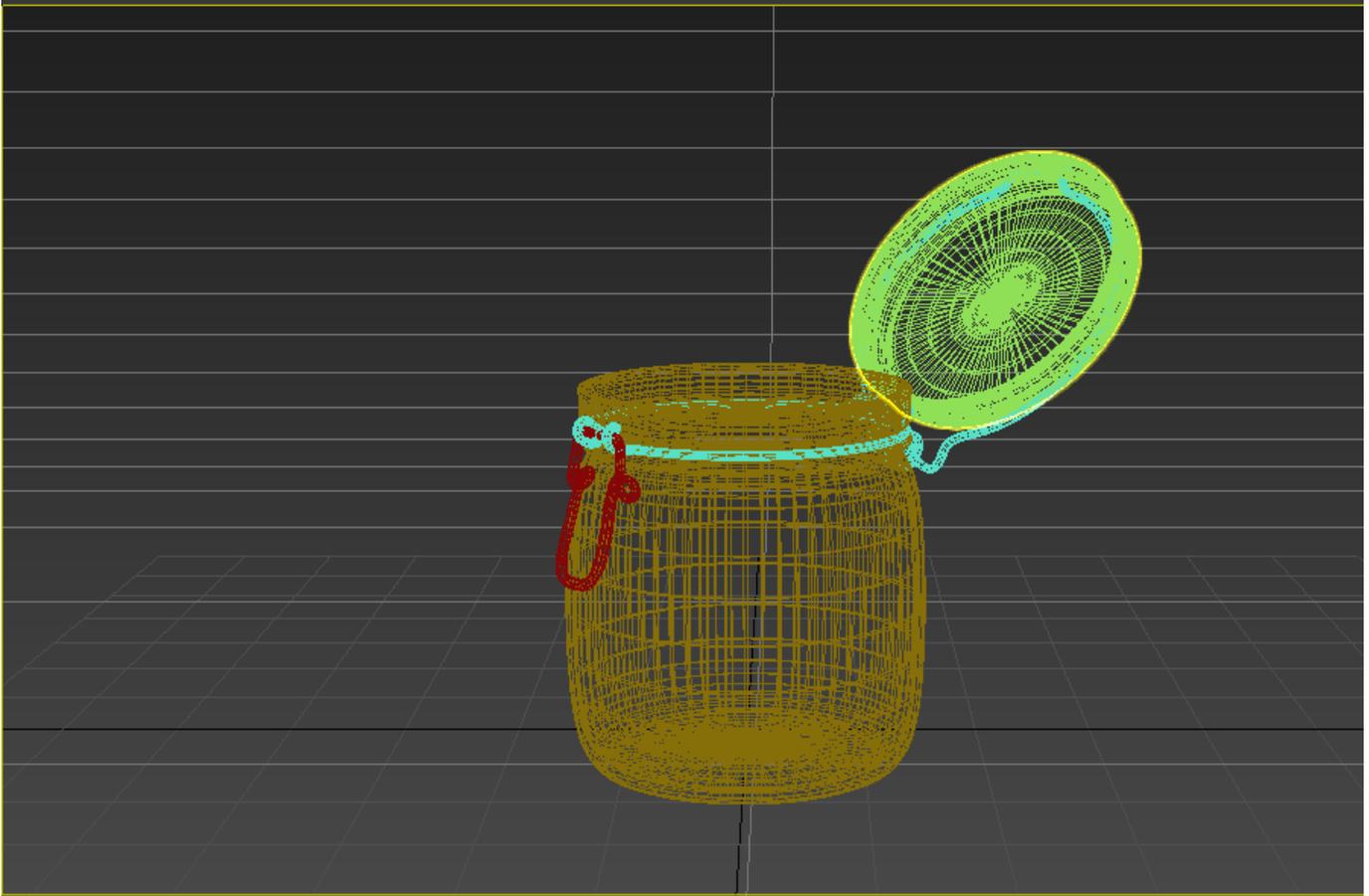
The following scene file can be downloaded from the Overview section above. It consists of a jar with a studio lighting setup and background.

[+] [Perspective] [Wireframe]



Open the provided scene and switch the view to **PhysCamer a001** to see what will be rendered.

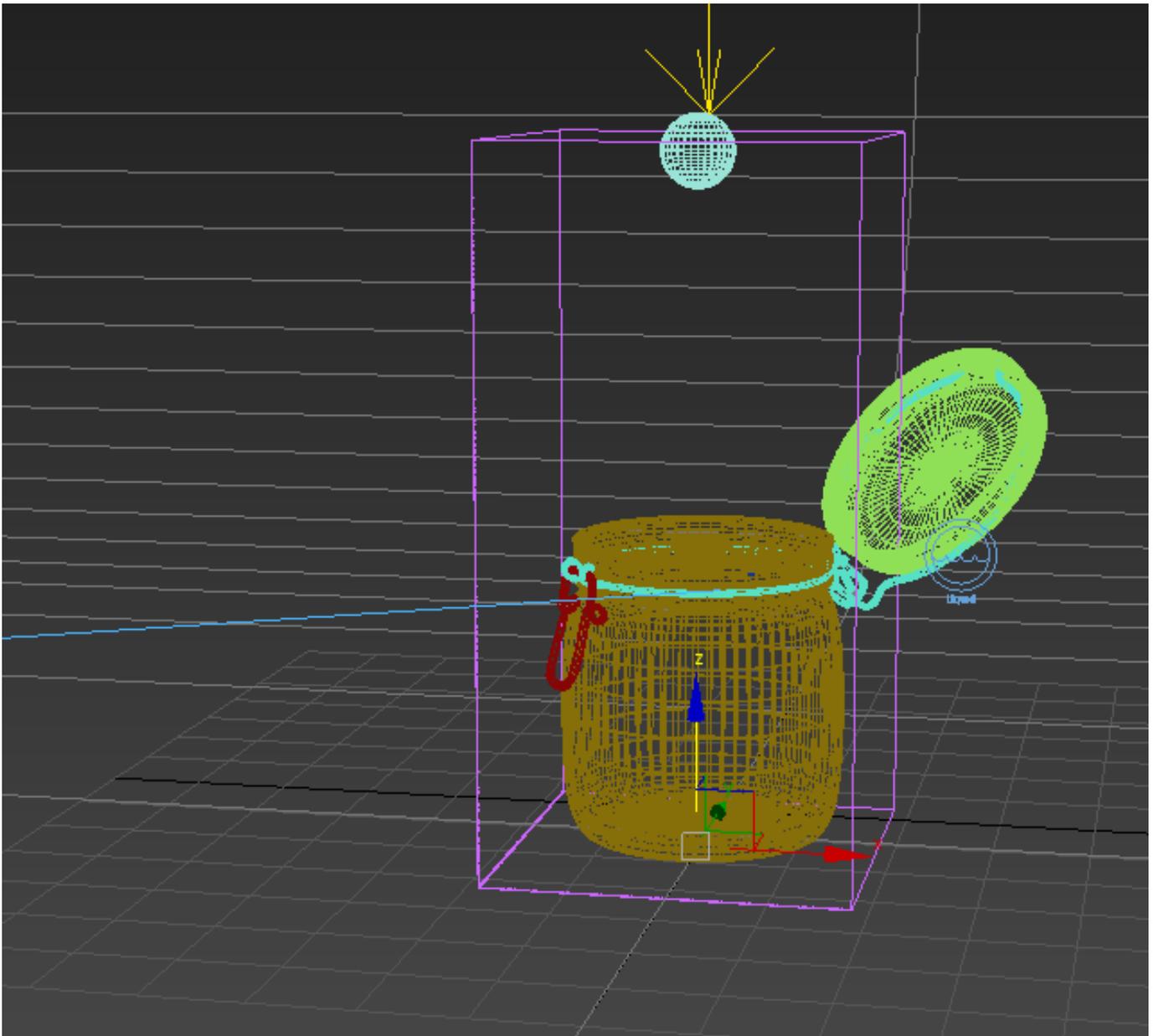
[+] [PhysCamera001] [Wireframe]



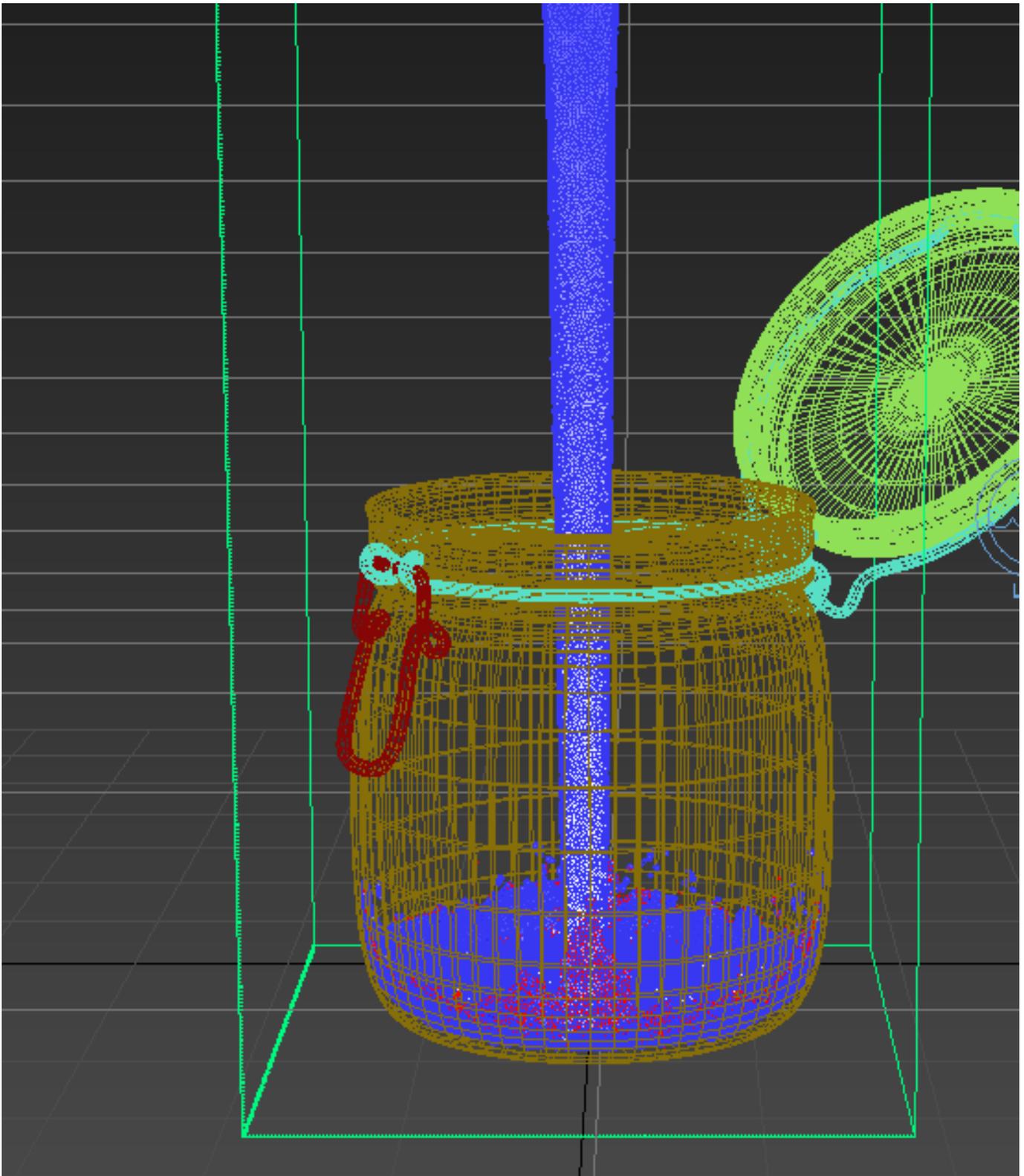
Return to the Perspective view and select the **sphere**. In the **Phoenix Toolbar** click the **Tap Water Preset** like we did before.

We will need to adjust the **Simulator Grid Size** so that it covers both the sphere and the jar. Again, use the **X/Y/Z size** parameters in the **Grid** rollout.

Use the Size attributes of the Simulator as opposed to simply scaling the Simulator container with 3ds Max's Scale tool, as that may adversely affect the simulation.



Start the simulation. Once it is finished, go back to the camera view. Here is how the simulation looks so far on frame 23.



Next, we will create materials for the water and the glass jar.

Open the **Material Editor** and create a new **V-Ray Material**.

Set the **Diffuse** color to **RGB (0, 0, 0)**.

Set the **Reflect** and **Refract** colors to **RGB (255, 255, 255)**.

Set the **Fresnel IOR** to **1.33**.

Assign this material to the Simulator.
[blocked URL](#)

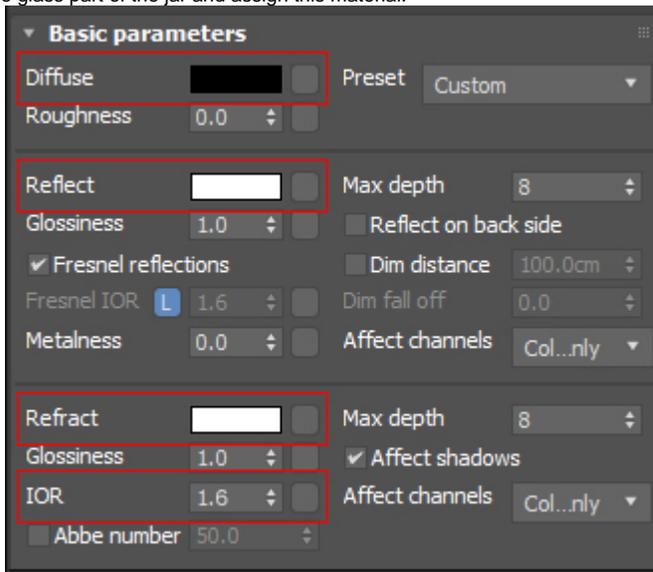
Create another **V-Ray Material** for the glass jar.

Set the **Diffuse** color to **RGB (0, 0, 0)**.

Set the **Reflect** and **Refract** colors to **RGB (255, 255, 255)**.

Set the **Fresnel IOR** to **1.6**.

Select the glass part of the jar and assign this material.



Make sure V-Ray is set as your Renderer in the Render Setup window.

In the V-Ray tab:

Set the **Image sampler Type** to **Progressive**.

Keep the **Max. subdivs** set to **100**.

Set the **Render time** to **1**.

Set the **Noise threshold** to **0.005**.

Target: Production Rendering Mode

Preset: No preset selected

Renderer: V-Ray 5, hotfix 2 Save File ...

View to Render: Quad 4 - Camera001

Common V-Ray GI Settings Render Elements

▶ Frame buffer

▶ Global switches

▼ IPR options

Start IPR

Fit resolution to VFB Force progressive sampling

▼ Image sampler (Antialiasing)

Type Progressive ?

Render mask None <None>

Min shading rate 6

▼ Progressive image sampler

Min. subdivs 1 ?

Max. subdivs 100

Render time (min) 1.0 Ray bundle size 128

Noise threshold 0.005

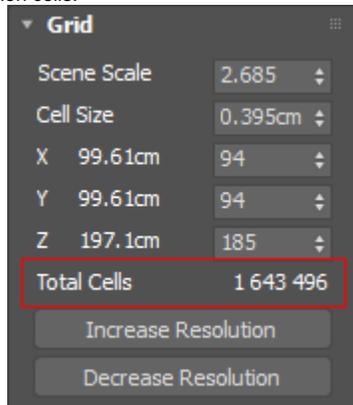
Render a frame to see how the fluid is interacting with the jar. Any geometry inside a simulation container will automatically interact with the emitting fluid. In addition, geometry can be excluded from interactions, and you can even create non-solid geometries, which are geometries that a simulation can flow through, but still be used as sources or forces acting on the simulation. You can find more about this topic in our [Solid and Non-Solid Bodies QuickStart tutorial](#).



With the current simulation the fluid will eventually overflow the jar. To control how much fluid is discharged by the emitter, select the **Liquid Source** object and locate the **Outgoing Velocity** parameter. Animating this value will control how much liquid is created, which can be used to start and stop the pouring of water.

Before proceeding, let's lower the simulation resolution.

Select the **Simulator** and navigate to the **Grid** rollout. Click the **Decrease Resolution** button three times to lower the **Total Cells** to around **1.6 million** cells.



We will animate the pour next, so scrub through the timeline to return to the start. Select the Liquid Source and set the **Outgoing Velocity** to **16**.

Note that if this value is set to **0** on the first frame, nothing will be emitted even if you start the simulation.

Next, scrub ahead to frame **23** and click the **Auto Key** button.



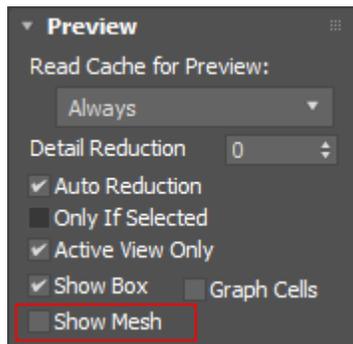
Set the **Outgoing Velocity** to **0** at frame **23**. With the current keys the Outgoing Velocity will gradually decrease from 16 to 0, which is not the effect we want.

Click and drag the key at frame 0 over to frame 20, so the liquid starts with a steady pour and then abruptly stops between frame 20 and 23.

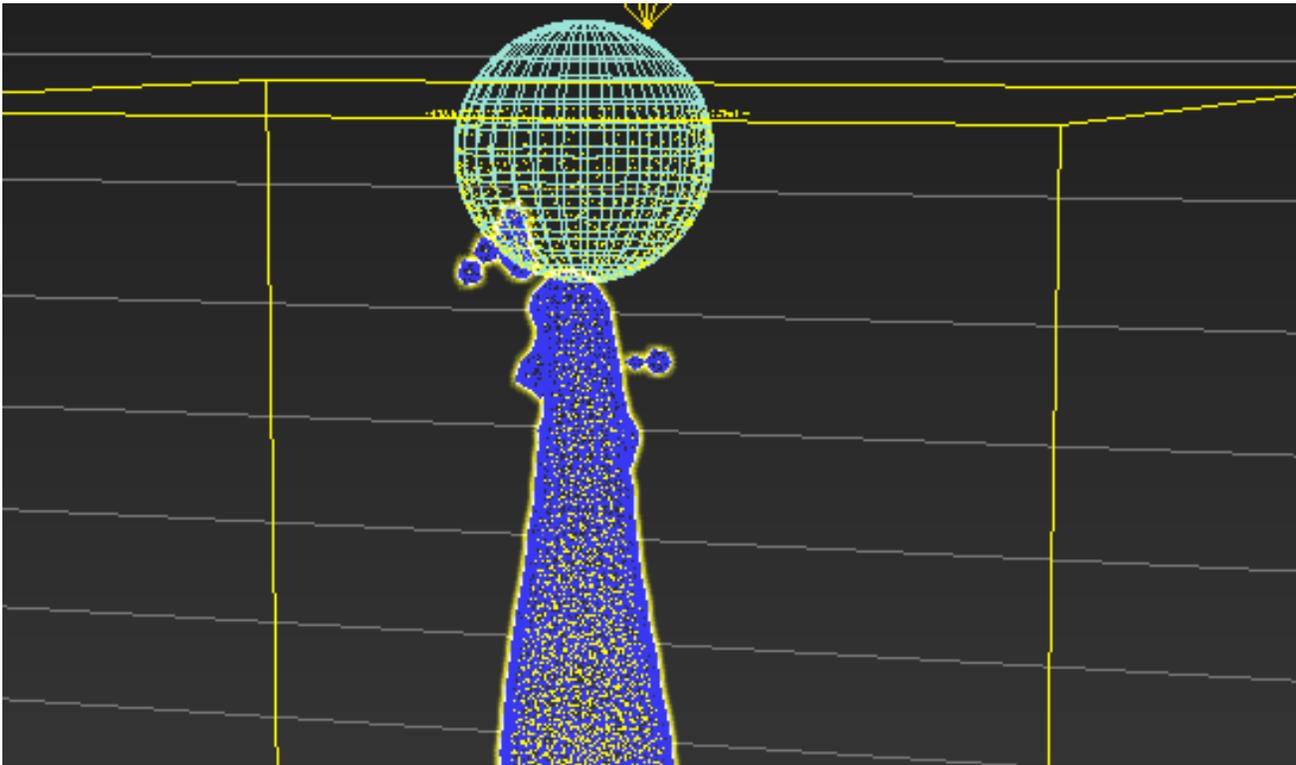


Now that the pour animation is set, turn off **Auto Key**. Go back to the beginning and start the simulation.

After the simulation resolves you may notice that the result appears a little choppy if you scrub through the timeline. To fix this, select the **Simulator**, go to the **Preview** rollout and turn off **Show Mesh**. Only particles for the liquid pour will be displayed and the simulation will scrub faster. At any point in the simulation you can turn **Show Mesh** back on to see the liquid better.

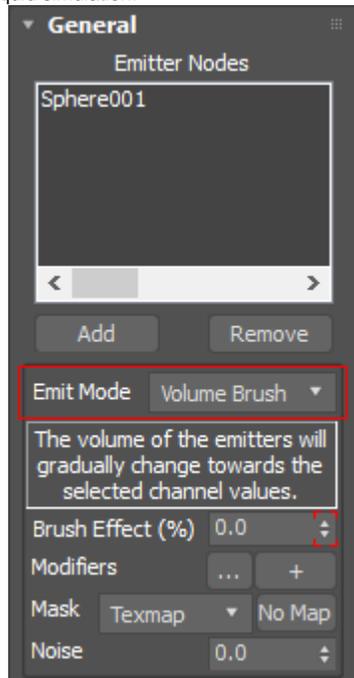


Notice at around frame 30, at the top of the pour, there is a large spread of drops coming out of the emitter. This is because the emission is coming from the surface of the sphere.

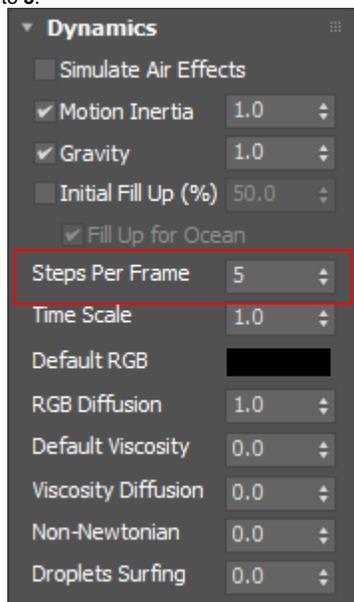


To fix this, select the **Liquid Source** object and change the **Emit Mode** to **Volume Brush**. What this means is that instead of emitting the liquid from the surface area of the sphere, it will now emit from the whole volume of the sphere.

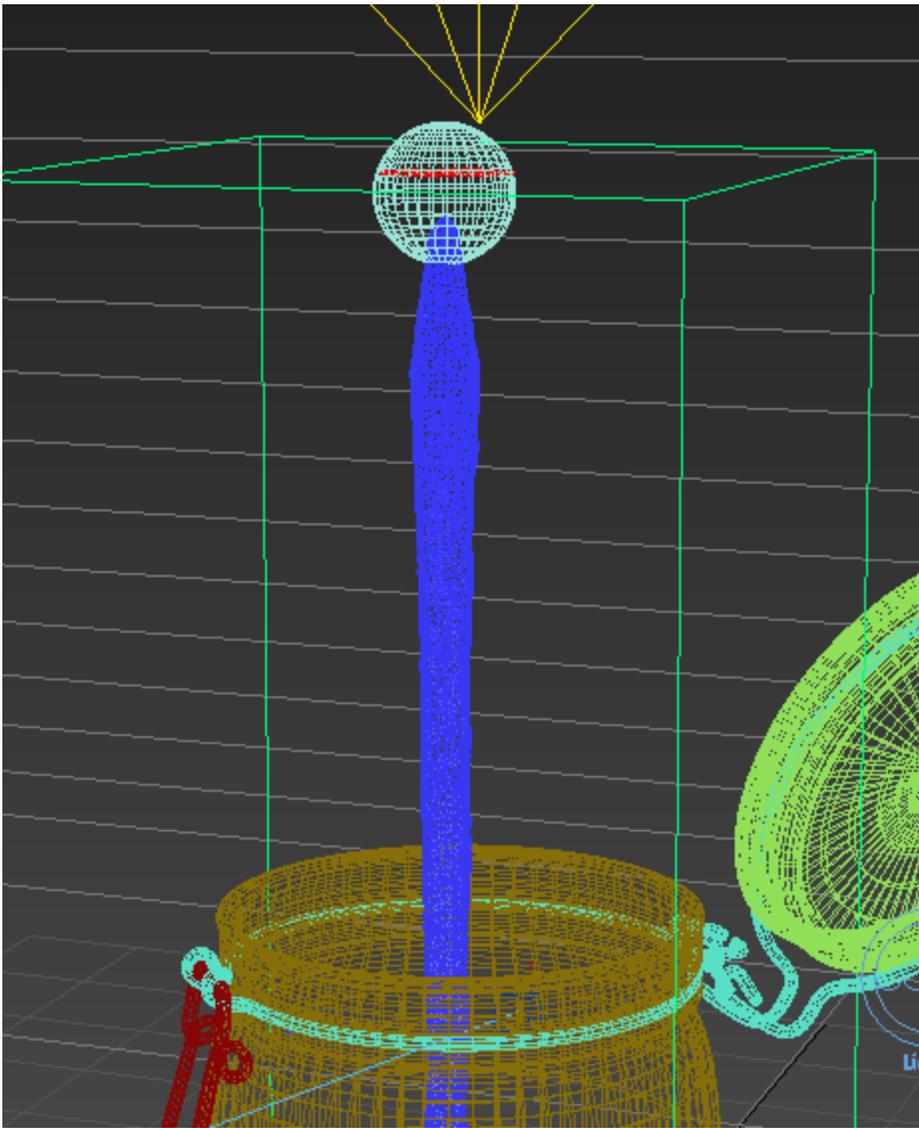
A window will ask to convert the emitter to a Non-Solid object. Select **Yes**. This will prevent the sphere geometry from interacting with the liquid simulation.



Finally, select the Simulator and under the **Dynamics** rollout reduce **Steps Per Frame** to 5.



Start the simulation. Notice that the spread out droplets are no longer an issue.



Return to the camera view and render a frame to see how the simulation looks.

