Advanced Effects, Rendering, & Compositing

Step 1: Introduction

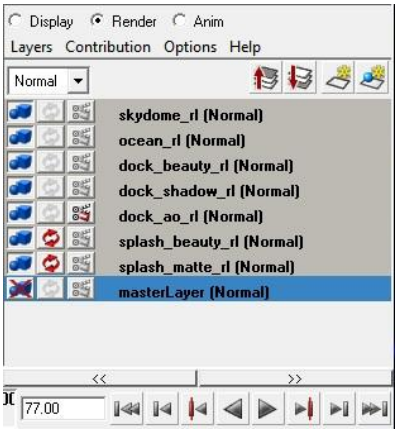
Effects are awesome because, as you will see, we get to do a little of just about the whole pipeline. We model, shade, and even animate. It is a pain because we learn to be patient the hardway. As you know, effects take a looooonnng time, so be prepared. As a renderer, we are at the end of the pipeline, our job is to render out the hard (or not so hard) work of those before us, make it look good, and not piss them off by screwing it up. So if it looks horrible, send it back before tackling it, making it look good because you are awesome at your job, and then pissing people off.

For this tutorial, I'll admit, we have terrible models and shaders. The lighing is pretty amateur too. But there is no time to send it back and you have to do the best with what you got, without messing with the work of those before you in the pipeline (i.e. no reshading or modeling, etc.). There enlies your challenge!

For this tutorial, we will be using one of the more advanced features new to Maya 2009, namely nParticles. Yay, you've graduated! To ensure we are saving all our stuff the the right place, since we will be working with multiple nCaches and render layers, so set your project to the directory in which you will be working FIRST, which should be located on the network. This saves many headaches later. For a refresh, you simply go to File->Project->Set... to setup your project folder.

Step 2: Defining Your Render Layers

Since we are in charge of compositing, rendering, and effects, it is our job to divide up the render layers in a sensible way. In your channel box, select "Render" for your layer type and under "Layers," we are going to create a few empty layers which we will then place objects into later in the tutorial. Don't worry if you don't understand what all these layers are used for at the moment, their usage will become obvious later in the tutorial. We are going to use the naming convention "something"_rl, where rl stands for "render layer." Go ahead and create the following layers:



We will now add objects to each of these render layers. Throughout the tutorial we will be adding more items as well. To add an object to a render layer, simply select the object or group and right-click on the render layer. Then, select "Add Selected Objects."

Add the following objects/groups to the skydome_rl:
- skydome_geo1
- skydome_geo2
- "lights" group

Add the following objects/groups to the ocean_rl:
- ocean_surface_geo
- "lights" group

Add the following objects/groups to the dock_beauty_rl:
- "dock_geo" group
- "lights" group

Add the following objects/groups to the dock_ao_rl:
- "dock_geo" group

Add the following objects/groups to the dock_shadow_rl:
- "dock_geo" group
- "lights" group

Add the following objects/groups to the splash_beauty_rl:
- "dock_geo" group
- "enviro_geo" group
- "lights" group

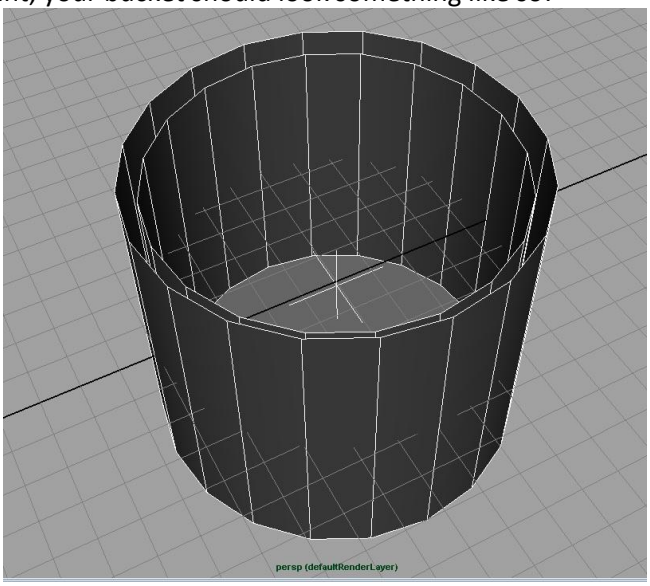
Add the following objects/groups to the splash_matte_rl:
- "dock_geo" group

Step 3: Preparing Colliders for nParticles

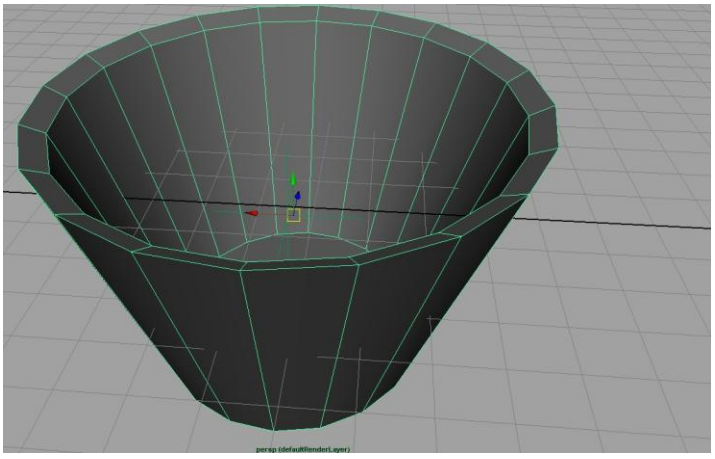
We'll get back to render layers later. Go ahead and move back to the masterLayer render layer for now. Feel free to turn off the enviro_geo_dl display layer (not render layer) to see our effects better. We will work from here to create our first nParticle system. Now, we are going to set up all our geometry that we would like the particles to collide with. This includes geometry that is not necessarily seen in the scene explicitly, but interacts with the nParticles to cause them to move dynamically the way we would like. In the case of a splash off the end of our dock, we will be using a bucket and a piston-like object to thrust the water and vapor nParticles into the air.

First we will create a bucket like piece of geometry that the nParticles will reside in initially. This can be done by creating a polygon cylinder and vertice snapping it around the effects locator that was placed in the scene for your aid (you can find this in the outliner in the "effects" group). In the inputs section of the channel editor, with the cylinder selected, click on "polyCylinder1" and set the "Subdivisions

Caps" to 0. Then extrude all the faces of your cylinder so that it becomes double-walled. Next, delete the top faces. At this point, your bucket should look something like so:



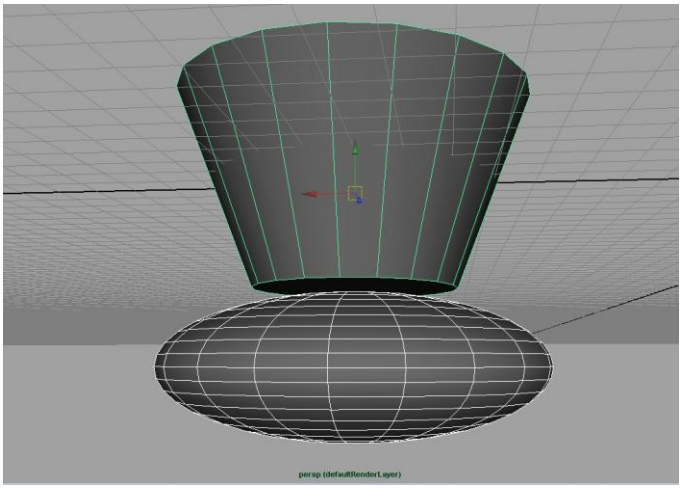
Finally, select the outer edge loop on the top of the bucket, extrude these edges to line up with the center section's edges, and merge the vertices to give our bucket's edge a top. Squeeze the bottom vertices inward so that the bucket is smaller at the bottom than the top. Finally, freeze the transofrmations on the bucket and rename your cylinder "splash_bucket_geo" in the outliner. The final bucket should look something like this:



Next, we are going to create a piston to cause our splash. Create a sphere that is approximately the same diameter of the splash bucket, squash it down, and place it as shown in the picture below. Name this object "splash_piston_geo" and freeze its transformations. At this point, it is a good idea to go ahead and delete your history.

NOTE: If you were working with a scene with rigged characters, MAKE SURE you say "Delete-by-type" at this point and have the bucket and piston selected as opposed to "Delete ALL by Type." Otherwise, with effects being typically after animation in the pipeline, you may break a rig! Just a word of caution

to avoid breaking things this late in the pipeline!



Place both your "splash_bucket_geo" and your "splash_piston_geo" objects in the display layer "splash_blobs_dl." Also, place both objects in the effects_group in the outliner if you haven't already.

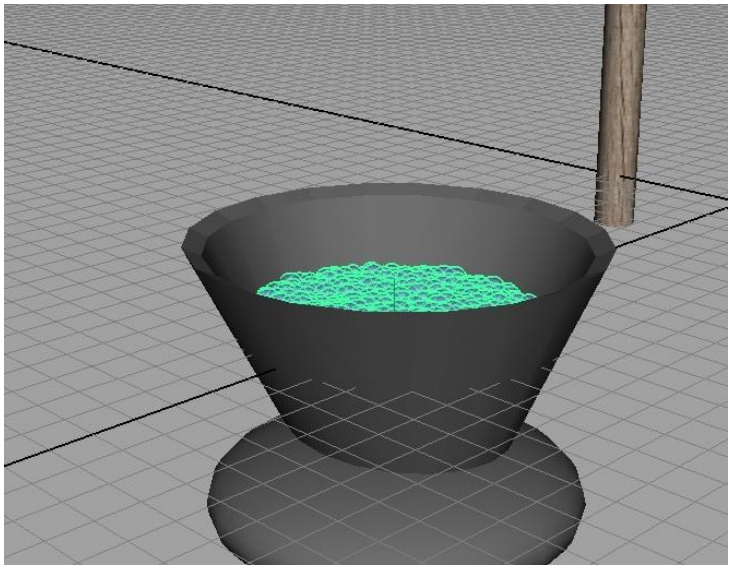
Step 4: Adding water nParticles

To add water nParticles to our scene, we will be filling up our splash bucket. To do so, we will be working with the "nDynamics" menu set, NOT the Dynamics menu set (they are different!). nDynamics are derived from the theories of nClothe, where objects and particles can collide and even merge much easier than past dynamic systems. Navigate to nParticles --> Create nParticles and check "Water" if it is not already checked. Then, with the splash_bucket_geo selected, navigate to nParticle--> Create nParticles->Fill Object (Options). Set the following parameters in the menu options:

Resolution = 35
Max Y = 0.6
Double Walled: Check

The resolution is simply how many nParticles that will be insterted into the selected geometery. The Max Y (as well as the like parameters) is a ratio that determines how much of your solid will be filled. In this case, we are only filling the bucket to 60% of its height. Double walled means that the object will be filled within both walls (as opposed to inside the bucket geometry itself). With these options selected, click apply. Lastly, make sure to rename your new nParticles "splash_blob_particles" and move them into the effects group in the outliner. Rename your nucleus as well to "splash_nucleus." This is not in your outliner, but can be easily reached by selecting your water_splash_particles and then selecting it's top tab in the attribute editor.

NOTE: Depending on the size of your splash bucket and piston, you may need to edit these values a bit. For instance, if your bucket is bigger than mine, then you may want to up the resolution a few units.



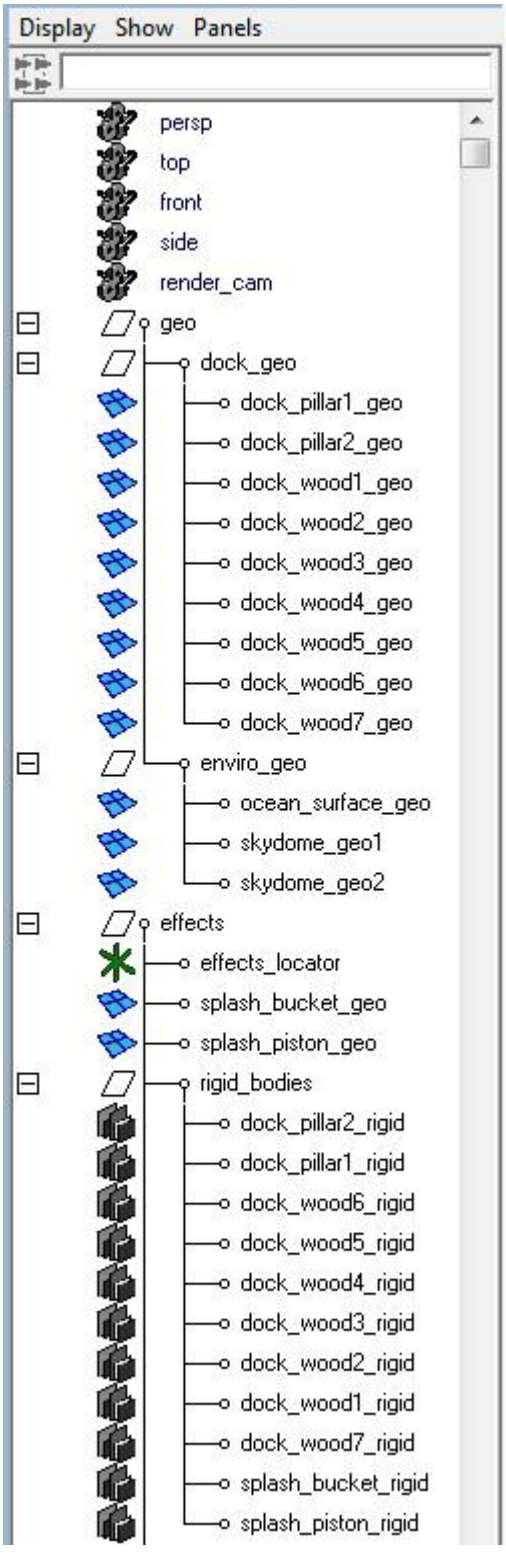
If you run the simulation now or render, you will notice three problems (well, many problems, but lets focus on three for now). One, the nParticles look nothing like a cohesive body of water, just a bunch of similar small spheres. Two, the nParticles fall straight through the bucket. Three, their shading doesn't even closely resemble water. Well the shader is an easy fix. In your Render Settings, select Render Using "mental ray." One down, two to go! Remember rigid bodies and particle options from previous tutorials? nParticles are a bit different, but still right up the same alley. Time to get crackin'!

Step 5: Refining Our nParticle Simulation

Colliders

To refine our simulation, we must first create a few rigid bodies that the nParticles collide with. You can make colliders out of multiple objects at the same time. In our case, shift-select each part of the dock as well as the splash piston and splash bucket and then create your rigid bodies. As you may remember, to create a rigid body, we simply select the mesh/meshes we would like to collide with and navigate to nMesh --> Create Passive Collider(Options). Make sure that the "Solver" option is set to "blobby_splash_nucleus," otherwise your particles will not collide with the object/objects and you may begin to create extraneous solvers! Rename all your new nRigids accordingly. For example, if the nRigid corrisponds to "dock_pillar2_geo," rename its corrisponding nRigid to "dock_pillar2_rigid." Group your rigid bodies in a group named "rigid_bodies" and move this group into the effects group.

At this point your outliner should look something like this:



Select dock_wood1_rigidShape in the attribute editor (you can navitage there easily by selecting the dock_wood1_geo and then tabbing over). Under the Collisions Drop-Down Menu, set the "Friction" value to 0.8 and Thickness value to 0.001 respectively. Repeat this for each of the dock wood pieces and poles associated rigid bodies.

Go ahead and run the simulation. The particles should now be contained in the bucket and not go through the bottom as before.

Particle Settings

Now we must edit a few settings on our nParticles to make them look and act more like water drops from a splash. Each of the particle parameters can be found by selecting the water_splash_particlesShape (NOT simply the water_splash_particles) and looking in the channel editor or in the attribute editor. Being that there are practically a million attributes however, it is easier to access them in the attribute editor, which are conveniently subdivided in drop-down menus. Remember, these are the values I used to get the ball rolling, but feel free to experiment and see if you can create something more stylistically fitted for *Nebbish*! Note that some of these values are keyed to ensure the propper behavior at different times of the simulation. As this effect is happening in time, feel free to experiment with when these animated settings occur as well.

The following dictate the size of our particles:

In the "Particle Size" Drop-Down Menu:

Radius = 0.25

In the Radius Scale Sub Drop-Down Menu:

Radius Scale Input = Randomized ID

Radius Scale Randomize = 0.150

The following attributes contribute to how the particles interact and "blob" together:

In the "Collisions" Drop-Down Menu:

Collision Width Scale = 1.0

Friction = 0.15

Stickiness = 0.3

By keying the following attribute, this causes forces to initially not affect the particles, allowing them to settle and not be prone acting sporatic and jumping out of the bucket on initialization:

In the "Dynamic Properties" Drop-Down Menu:

Damp (Frame 1) = 1.0

Damp (Frame 3) = 0.0

The following attributes control how the particles move and form via there liquid characteristics. This simulation profile is selected due to the fact of us selecting "water" earlier when filling our bucket:

In the "Liquid Simulation" Drop-Down Menu:

Viscosity = 0.1

Liquid Radius Scale = 1.0

We will not need shadows in most of our layers involving our particles. For the shadow layer, we will run an override for the following options. For now, however, set the following attributes as such:

In the "Render Stats' Drop-Down Menu:

Casts Shadows = False (Unchecked)
Receives Shadows = False (Unchecked)

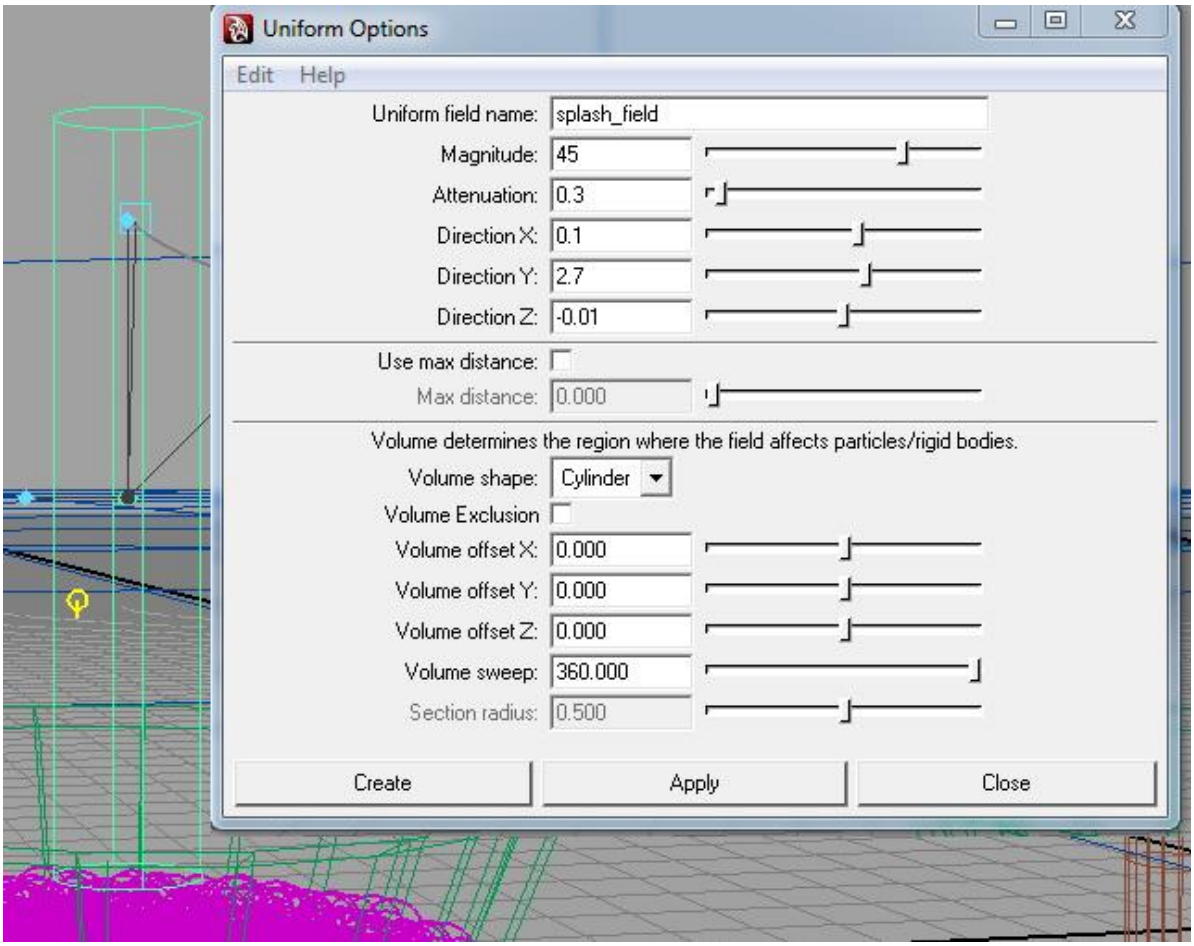
In addition to these settings, we will be upping the substeps in our splash_nucleus to 20. This setting can be found in the splash_nucleus' attributes under the "Solver Attributes" Drop-Down Menu. This will aid in making our nParticles collisions look more fluid and water-like, but will slow our simulation render time. That's okay, we have all the time in the world, right?

Step 6: Final Touches: Animating, Adding Fields, and nCaching

Animating
To cause our nParticles to simulate a splash, animate the splash_piston_geo thrusting into the bottom of the bucket. It should only require a few keyframes tops, spread out over a period of about half a second MAX (Remember, we're effects artists, not animators. We leave as much of our motion up to a physics engine as possible :).

Adding Fields
I found that adding the following field is helpful in making our splash more believable. Feel free to experiment to see if adding others is benificial, but keep them to a minimum to ease your life. Adding too many fields can get confusing and complicating really quick. Like good lighting and utilizing a small number of lights, maximize your effective work using simple rigs with only a few fields. By adding a cylindrical, uniform field to the middle of our splash, this will eject water particles near the center quicker than around the outskirts to give it that classic "plop" look. Navigate to Fields -> Uniform Field(Options).

Set up the initial attributes as shown below and place it in the middle of the bucket. You can easily modify the magnitude, attenuation, etc. at a later time dynamically by selecting the field and clicking "t" on your keyboard. This mode allows quick adjustment to the field's parameters and see just what direction the field is opperating in quickly and easily. You can cycle through the attributes by clicking on the power-button looking selector shown in yellow in the picture below. If your simulation is running too slow, you can nCache now to get a better idea of how your particles move over time (see **nCaching** below).



Here is where we iterate (and where we MAKE SURE we save constantly)! Refine the particle settings, bucket position, piston animationð etc. as desired to achieve a splash that looks the way we like (as in it supports our story and is believable, not necessarily *realistic*). To avoid the heartache associated with the particles interacting with the dock, key the particle size to go from there initial setting (~0.25) to zero, where there size falls off quickest as the particles begin to fall. You can also animate the threshold and opacity values of your particles found in the Shading Drop-Down Menu in the particle's attributes. The threshold value controls at what size the particles are rendered. Animating this value allows your particles to disappear at different times. You can play with these values now or after creating an nCache, it is up to you (nCaching only saves velocity and/or position information of the particles). If needed, edit the position of your bucket to avoid the majority of the water blobs colliding with the dock (this is a headache that is not worth the effort!).

At this point you should be getting the effect to look something like this, if not better:

nCaching

Once we have our nParticles acting the way we would like (or think we like), it is time to create an nCache. With the blob particles selected, go to nCache -> Create nCache (Options). nCache out frames 1 - 100 of your splash. Name your nCache something like "splash_blob_cache_v001." At anytime, if you would like to reevaluate your Cache, you can delete your cache by selecting nCache -> Delete nCache with the correct particles selected, or you can simply say Create nCache again with your updated attributes. Here, you will be prompted to either delete your old cache and create a new one or replace your cache. I recommend keeping your caches for safety sake and keeping version numbers as you go along. You can always revert to an earlier cache by the "Attach Existing Cache File..." option in the nCache menu.

Step 7: The Vapor nParticle System

Using what you have learned from the earlier nParticle system, create a new particle system that will simulate more disturbed, airy water particles. Name the particles splash_vapor_particles and place this new system in the "splash_vapor_dl" display layer. A few guidelines:

- Feel free to duplicate the bucket and piston geometry from the first simulation. Get these new particles to burst up primarily around the blobby particles
- Use a new solver to generate this effect, not the same one as the blobs!
- Make sure to create a new display layer and label it accordingly with this new effect
- Keep everything organized in the outliner and named appropriately, including that pesky solver/nucleus!
- Turn off the "splash_blobs_dl" display layer while working with this new particle system. It is a pain in the ass to run multiple systems at the same time. Save yourself the time and maya crashes.
- Check out the "Cloud" Particle Render Type under the "Shading" Drop-Down Menu in the particle's attributes. You may have to play with the shader to get it to look formidably.
- Animate the airy particles to spring up quicker than the blobs, to approximately half the height of the blobs, and then fall slowly. To do so, think about animating the nucleus' Gravity attribute, found in the Gravity and Wind Drop-Down Menu.
- USE REFERENCE!!!!

Step 8: Preparing the Render Layers

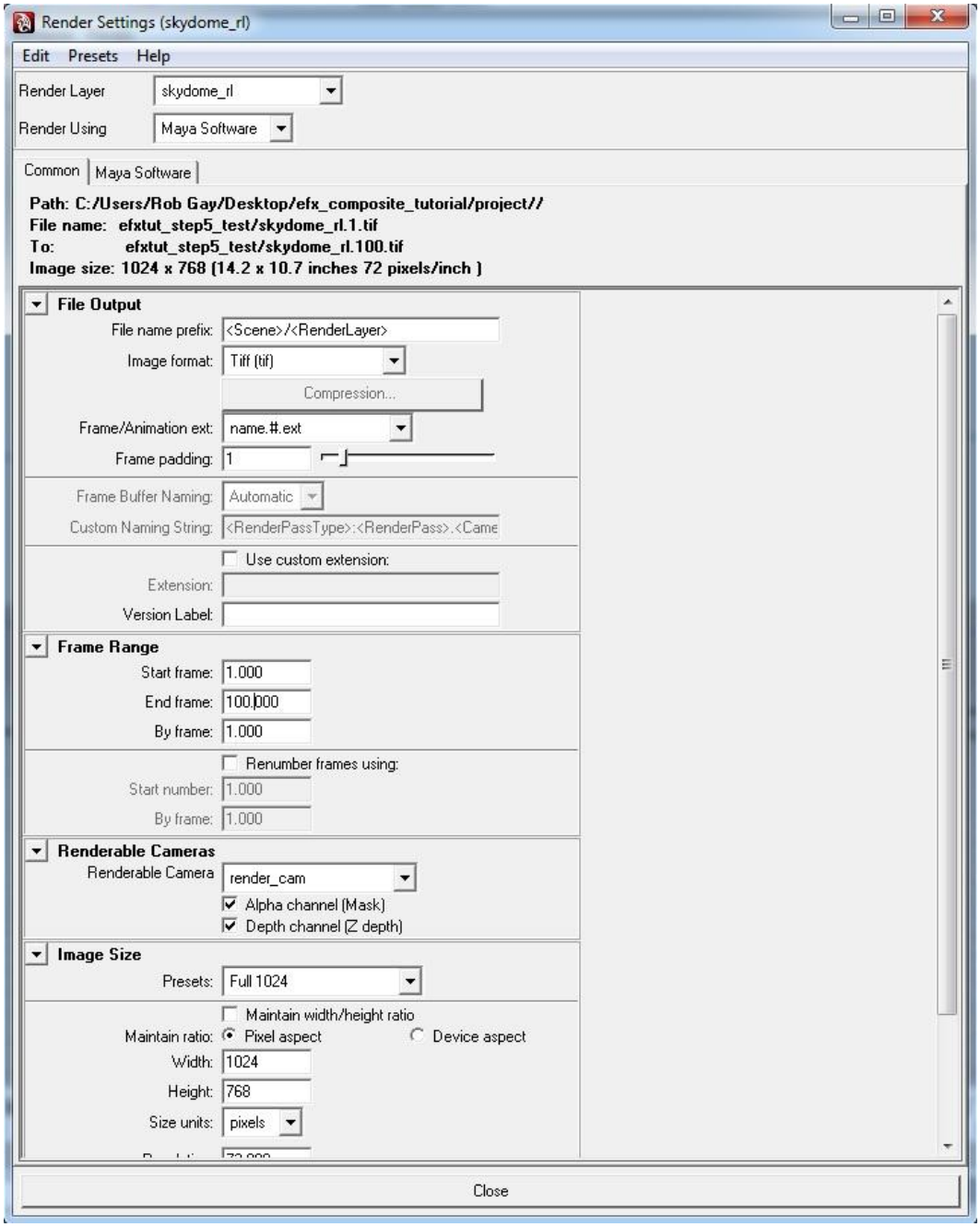
Once you have your nParticles all pretty and cached, the next step is to set up those render layers way back from the beginning.

Add your splash_blob_particles and splash_vapor_particles to the following render layers:

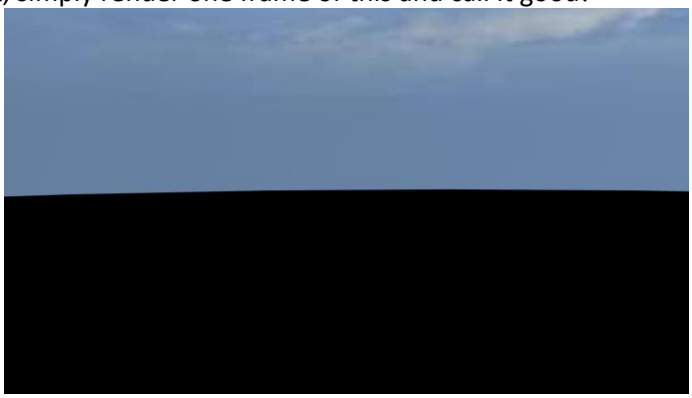
- dock_shadow_rl
- splash_blobs_beauty_rl
- splash_blobs_matte_rl.

Step 9: Rendering

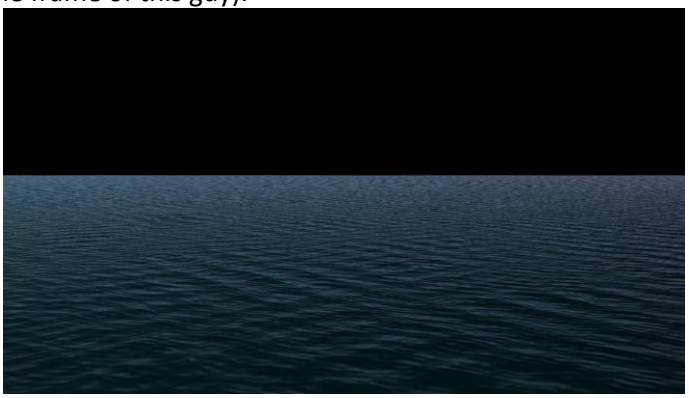
Most of the render layers will be done with Maya Software. Go ahead and set our default Render Settings to Maya Software and name it the following: "/images/<RenderLayer>." If you right-click and hold over the File name input box, the name "tags" appear, such as the one we are using, namely "<RenderLayer>." These are useful for future renders, just FYI. Also, render from the render_cam and render using high quality. If you are up to waiting a bit longer for your renders/ have the time, render in "Full 1024." Last, make sure you turn on all your display layers, including both nParticle systems. Use the settings shown below:



skydome_rl
This layer is the easiest, simply render one frame of this and call it good!



ocean_rl
Layer 2 is second easiest, render frames 1-100 of this guy and your good to go (the shader moves, so we cannot do simply one frame of this guy).

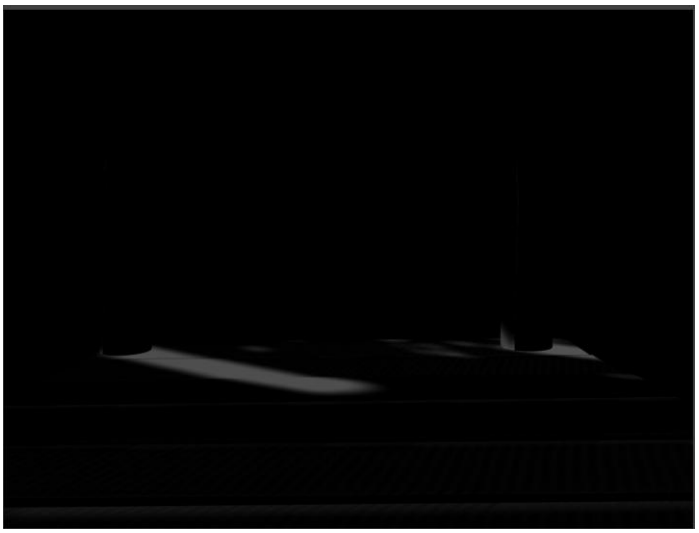


dock_beauty_rl
For the dock_beauty_rl, we are going to remove the shadows and have them render on a separate layer. Select each of the lights that emit depth map shadows; in our case this is the sun_light. In the Attribute Editor, we will be performing a Layer Override. Right-click over where it says "Use Depth Map Shadows" in the Depth Map Shadow Attributes Drop-Down Menu (not over the input box). Select "Create Layer Overide." The text should then turn orange. Finally, deselect the attribute. Only one frame of this guy needs to be rendered of this render layer.

dock_shadow_rl

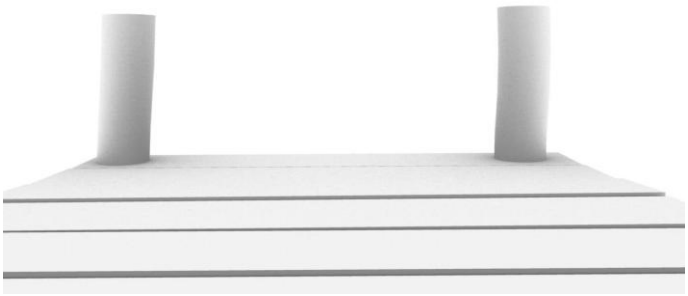
For the shadow render layer, we are going to use a Layer Override Preset. Right-click on the dock_shadow_rl and select "Attributes." Click and hold over the "Presets" button in the top left of the Attribute Editor and select "Shadow." To ensure this layer receives shadows from our particle systems, select the splash_blob_particles and in the Attribute Editor, under the Render Stats Drop-Down Menu, create a layer override for "Casts Shadows." Enable this attribute after creating the override by checking its box.

Repeat setting up this override for the splash_vapor_particles. To test, scroll to where the particles are high in the air over the dock and render one frame. ITS OK IF ITS BLACK! Select the white circle in the square in the "Render View" window to see the alpha channel of this frame. You should see white where the shadows are in the render, from both the dock pilars and the particles. Render frames 1-100 of this layer (alpha layer shown below, vapor particles not shown).



dock_ao_rl
In short, ambient occlusion is a technique of removing light where it would typically get trapped between objects close in proximity. In essense, quick and dirty contact shadows. It takes a while however, and isn't needed for all objects in the scene. Thus, we can cheat and render a single frame of its effects on the dock alone.

To do so, we are going to create a layer override by selecting the "Attributes" of our Render Layer as we did in our dock_shadow_rl, and select the Preset "Occlusion." After selecting this override, a surfaceShader should appear in the Attribute Editor. Rename it "dock_ao_override_shader." To the right of this object in the Attribute Editor, select the tab for the input node to this shader titled "mid_amb_occlusion." In the Parameters Drop-Down Menu, up the Samples to 32, and bring the Dark value to a dark, dark gray rather than black (remember, nothing in the real world is completely black). Render out a single frame of this layer.



Note: Notice the shadowing effect between the pilars. This can be reduced by playing with the Spread and Max Distance values. Feel free to experiment with these to try and remove these unwanted shadows. This can also be cleaned up in post-production.

splash_beauty_rl
Because we get these nice reflections and refractions off the particles (mainly the blobs), which are not possible rendering with Maya Software, we are going to render with mental ray. Thus, we are going to override the render software. Head to your render settings and right-click over the "Render Using" at the top. Create a layer override and select "mental ray." Move over to the Quality tab and set the "Quality Presets" to Preview (production without a farm is insanity. If you wish to try it on crunch, be my guest but no garuntees). Patiently render out frames 1-100. (Vapor not pictured below)

splash_matte_rl

We used the dock and environment to induce reflections and refractions on our particles, but we now need to scratch them from our scene as we have the lit versions the way we want them rendered in Maya Software. Thus, we will matte them out of our splash_beauty_rl. To do so, select all the dock geometry. Then, right-click on the splash_matte_rl and select Overrides --> Create Material Override --> Use Background. Render rrames 1-100. Again, we will be using the alpha of this information so do not worry about the shader of the particles looking severely sub-par. (alpha layer below with vapor particles not shown).

At this point, we have all the layers needed to composite our final shot.

Step 10: Compositing in After Effects

Learning the Basics
After completing your renders, we will be moving to After Effects to composite and learn the post production side of animation. At this point, if possible, set aside a couple hours to watch the tutorials at the following link: http://www.videocopilot.net/basic/

I could bore you with how to do this stuff in writing, but these video tutorials are awesome, and much more effective in my opinion (just don't feel obligated to buy his products).

Compositing your Scene
Beyond simple layering, the final version of your scene should have the final techniques applied:

1. Create precompositions for your particles, dock, and environment and add effects to stylize the effect.
2. Setup the layers to interact correctly (track matted, multiplied, etc.)
3. Color corretion (ex. getting the sky and water closer to similar hues and more believables brightness levels, adjust the shadows to have a slight coloring, adjust the weights of the ambient occlusion versus the direct shadows, etc.)
4. Tweak the timing of your simulation using time remapping (optional)
5. Add a 3D title to your scene using a camera and 3D movement
6. Render your scene as a high quality quicktime video

Turn-In Requirements:

Turn-in a folder with the following files (and ONLY the following files) on the network:

1. The final Maya 09 file
2. The final After Effects file
3. All associated texture files (sky and dock)
4. Your clean project folder (including rendered images)
5. Your final quicktime video
6. Your reference for your splash