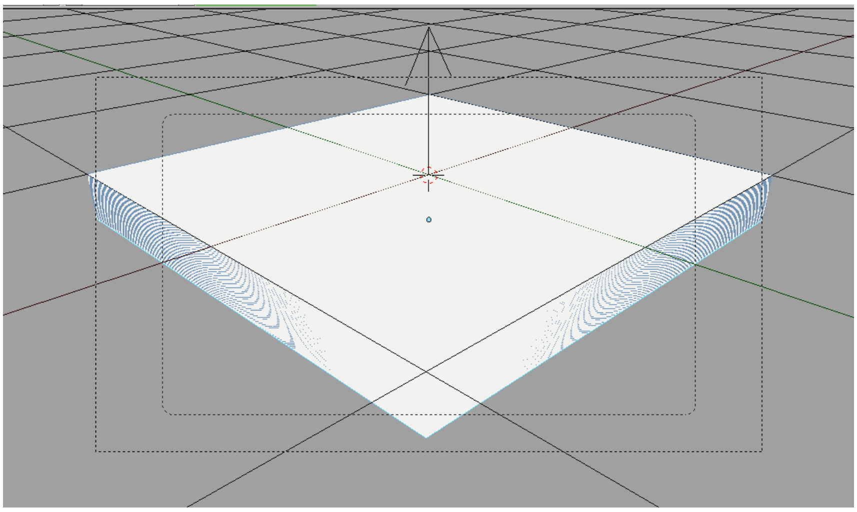
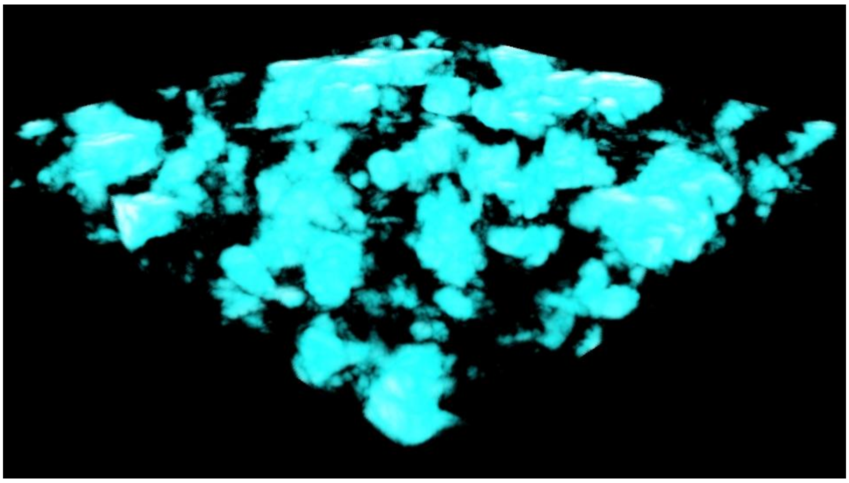
This tutorial will help you create the above star using Blender 2.49, or something close enough anyway. You can find the finished .blend here. I originally made the star waaay back when raytracing was not even in a stable Blender release, so it can easily be adapted to other Blender versions. It uses the classic "dupliframes" to fake volumetrics technique, and might become obsolete when 2.5 is released. However, it does render nice and fast. I'm going to assume basic familiarity with using Blender, but assume no prior knowledge of the techniques. You'll need to know how to move and scale objects, how to set keyframes, and how to create materials with textures.

Working with Dupliframes

If you're not already familiar with the dupliframes technique, it's really wonderfully simple, and shame on you. If you are, skip ahead to page 4. The idea is this : use a series of objects, usually planes, very close together with object (or globally) oriented procedural textures, which are mapped to the alpha (transparency) as well as colour. An illustration is probably better.

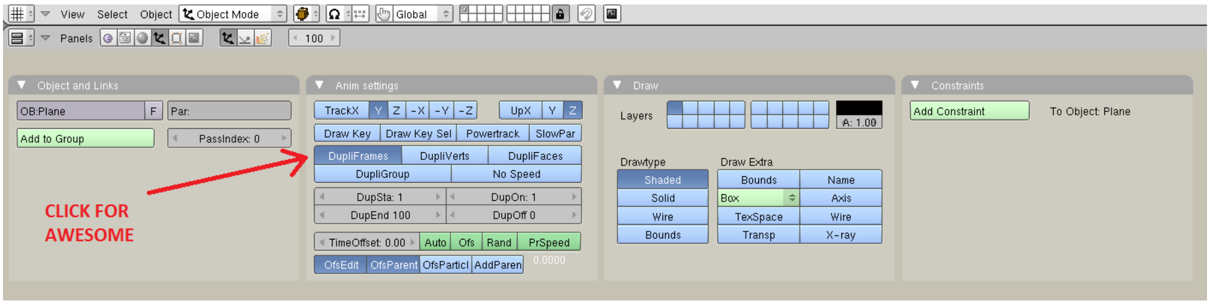




Above is the 3D view; below, the rendered output. The .blend can be found here. Note that you don't *have* to use dupliframes, you can use real objects too. Dupliframes are just a convenient way of creating lots of objects with the same material close together. The reason they have to be close is because that way they look like a solid continuous volume. The more objects the better – the closer the objects are to each other, the greater the part of the volume they appear to fill. Of course, there are limitations. Render a plane edge-on and you'll see zilcho.

The reason for using an empty to orient the textures, rather than the standard "orco", is twofold. Firstly, Blender's procedural textures are 3D. Using some fixed point in 3D space to orient them (rather than the object itself) means that each plane intersects a different part of the texture. Orco mapping would just result in exactly the same texture on each plane. Secondly, empties are handy easy fine control over the texture scaling – scale the empty and the texture scales with it.

Dupliframes are simple to set up. In this example all I did was add two keyframes for the plane with the plane just moving vertically. Dupliframes are then enabled by the button in the "object buttons" panel :



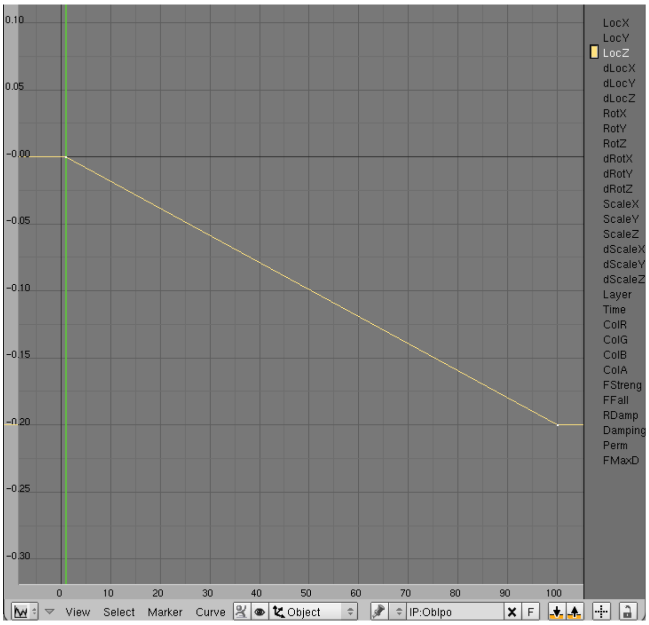
OB:Plane
Par:
PassIndex: 0
100
DupSta: 1
DupEnd 100
DupOn: 1
DupOff 0
TimeOffset: 0.00
0.0000
A: 1.00
To Object: Plane

CLICK FOR AWESOME

The "DupSta" and "DupEnd" buttons tell Blender which frames to enable dupliframes for. This means that if your object is animated over frames 1-100, but you set DupSta and DupEnd to frames 50 and 60, you'll see the object duplicated for frames 50-60. You'll see the duplicates for whatever frame you're on, mind you. The main use of these is to control the number of duplicate objects.

For this example I used the full 100 frames animated, so we get 100 planes stacked on top of each other. This is overkill. Usually you can get away with half that (using dupliframes it's much easier to alter things like this later on, so don't be tempted to manually duplicate objects instead).

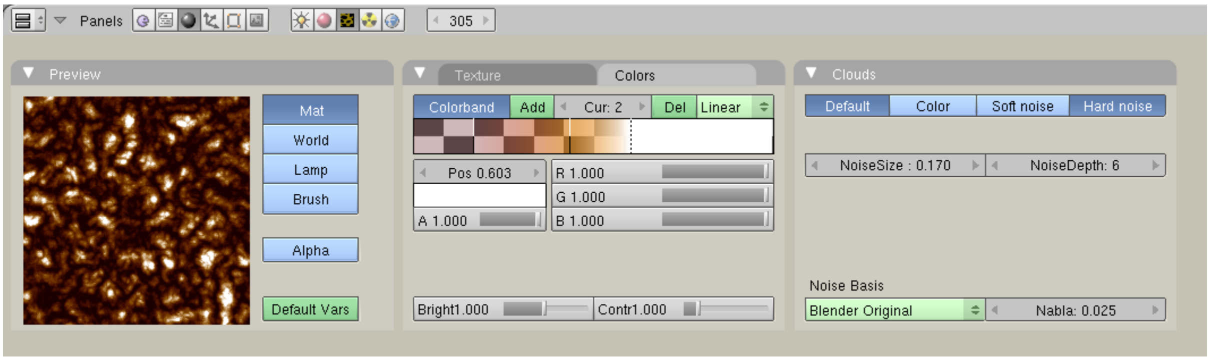
A final note on dupliframes – you probably want to set the IPO interpolation to linear, or your objects will be bunched up. This is easy. In the IPO window, select the curve you need (in the example it's the LocZ curve), press tab, then V. Done. Alternatively, select the curve, then hit "curve", hover over "interpolation mode", hit "linear". Done. It should look like this :



Of course if you were to render now you'd see nothing of interest to anyone. For that, on to the materials settings. The material colour is pure white, the alpha 1.0, and the texture mapped to colour and alpha using the "multiply" rather than default "mix" mode. Ensure that "ZTransp" is enabled. The texture is just a simple procedural clouds texture with some rudimentary colourband settings (see .blend and below). I used shadeless here – it IS possible to get nice results with shaded mode and "TraShadow enabled", but this can be excruciatingly slow, even on a fast machine. So don't do that. Or else.

Behold, the Stars !

Now we're ready to create a star (only in CG can you say that and not need to add maniacal laughter afterwards). Start with a nice fresh new .blend and delete everything. First it will need a surface so that we can't see right through it, so create a sphere and give it a procedural clouds texture. Here's the settings I used if you want to slog through it, otherwise, just look at the .blend.



305

Cur: 2

Pos 0.603

R 1.000

G 1.000

B 1.000

A 1.000

Bright1.000

Contr1.000

NoiseSize : 0.170

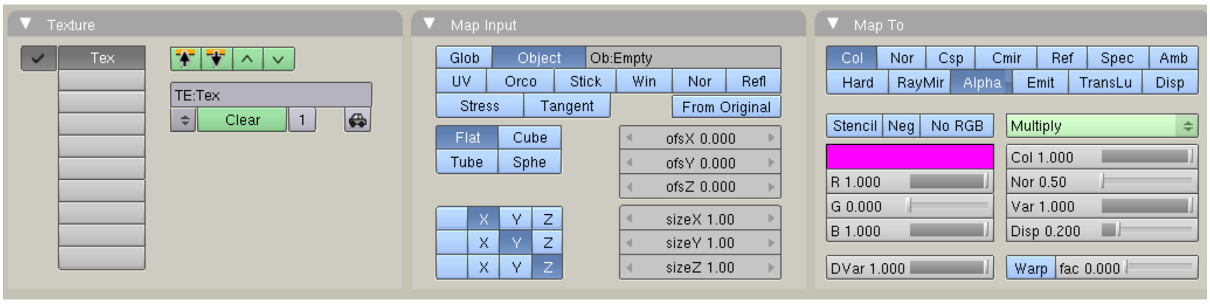
NoiseDepth: 6

Noise Basis

Blender Original

Nabla: 0.025

Material settings are as before, should again be shadeless but you can probably get away with alpha values less than 1.0 if you're so inclined. Be sure to also create an empty of about the same size as the sphere, and use that as the object to orient the textures.



TE:Tex

1

Ob:Empty

ofsX 0.000

ofsY 0.000

ofsZ 0.000

sizeX 1.00

sizeY 1.00

sizeZ 1.00

Multiply

R 1.000

G 0.000

B 1.000

DVar 1.000

Col 1.000

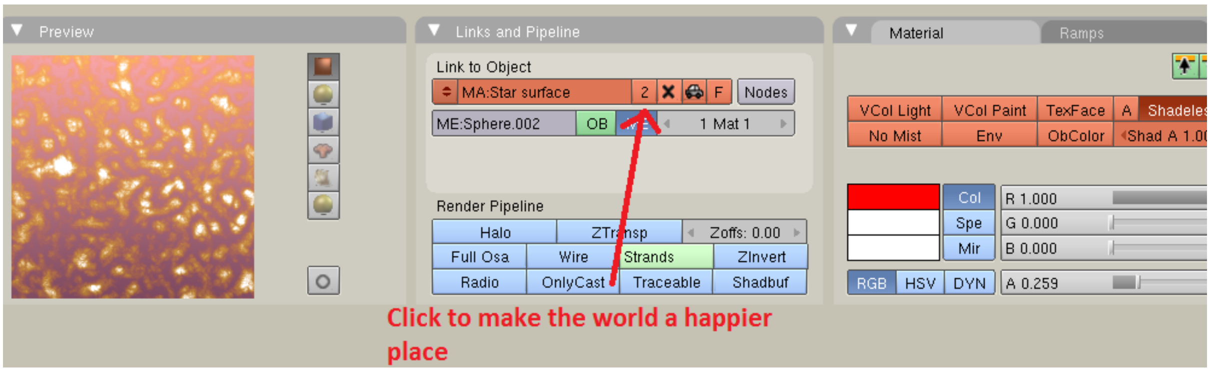
Nor 0.50

Var 1.000

Disp 0.200

fac 0.000

Next we need the key factor – the atmosphere. Duplicate the sphere and put it on another layer for ease of organisation. Make the material a single-user copy, i.e, click this button (the little 2) :



MA:Star surface 2 F

ME:Sphere.002

1 Mat 1

Zoffs: 0.00

Shad A 1.00

R 1.000

G 0.000

B 0.000

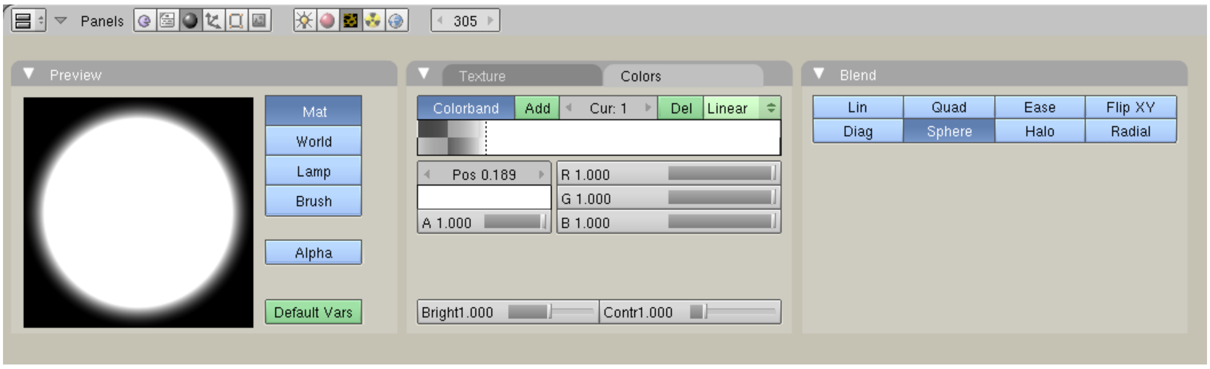
A 0.259

Click to make the world a happier place

You need to make the texture(s) single-user copies too. It's very important that the procedurals have the same noise and size settings on both the surface and the atmosphere. That way, it will look as though the bright parts on the star's surface extend outwards into space. If the settings are different, they won't match up and it'll look like the worst disaster since the Hindenburg.

Next, use this new sphere to create the "volumetric" atmosphere. Set a size keyframe on frame 1, advance to a different frame (50-100 should do nicely) scale the object up until it extends as far as you want the atmosphere to, then set another size keyframe. Enable dupliframes just like before.

Again, you'll want to use the empty to orient the texture. But here you'll also need a second texture. This one needs to be a spherical blend texture, something like this :



Cur: 1

Pos 0.189

R 1.000

G 1.000

B 1.000

A 1.000

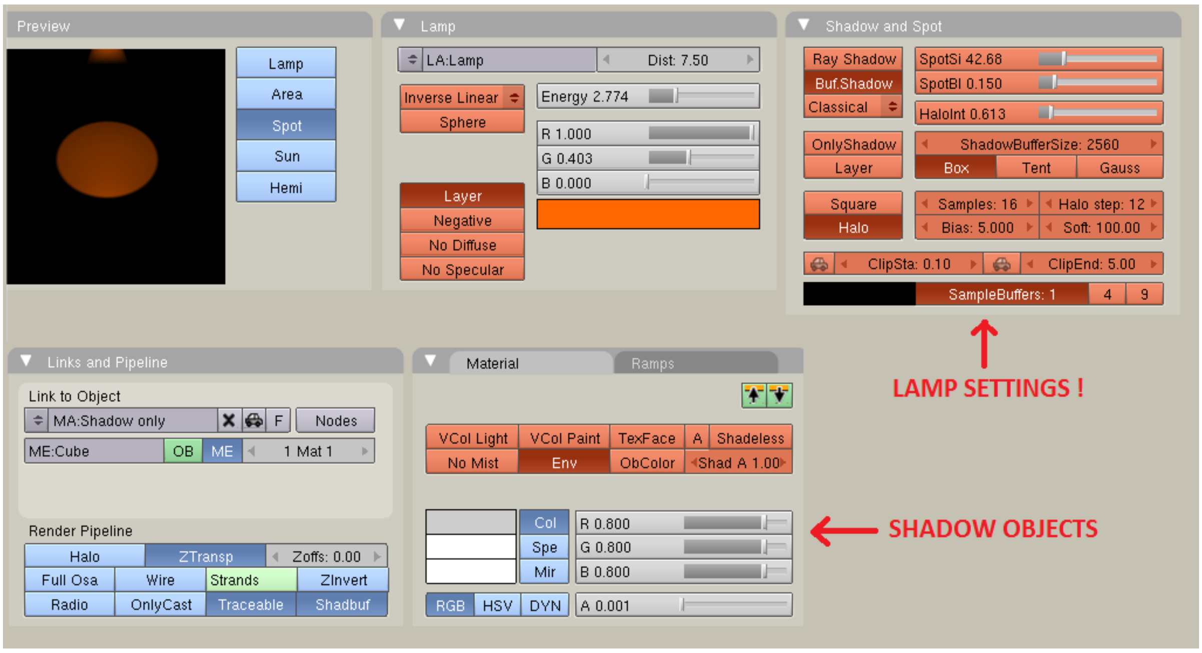
Bright1.000

Contr1.000

This too must be mapped to the empty (or a different empty if you prefer) and set to only multiply the alpha – nothing else. What it does is to make the atmosphere gradually fade out, so that it doesn't come to an abrupt and stupid-looking halt. You'll need to fiddle with the precise colourband settings as well as the empty's scale – unless, sensibly, you skip all that melarky and download the .blend. Ideally, you want the texture scaled so it sets the alpha to be 1.0 at the star's surface, and 0.0 at the edge of the atmosphere.

And that's basically it. You'll see in the file I also added a simple little particle system to give the thing a bit more random variation. That's nothing fancy and I'm sure there are at least 50 billion particle tutorials floating around the aether. I also added a spotlight with a volumetric halo and some invisible, shadow-casting objects to give a decent "beaming" effect.

If you're not familiar with spot halos, they're lovely. Make sure the spot points at the camera and adjust the "SpotSi" angle to small enough to just illuminate the shadow-casters (same applies to "ClipSta" and "ClipEnd" – have to use the old "Buffer shadows" for this to work). Increase "Halo Step" to 12. Here are the appropriate Lamp settings and the material settings for the shadow-casters, for those control freaks who like to do everything their own way. But again, the .blend is probably the best way to see what's going on.



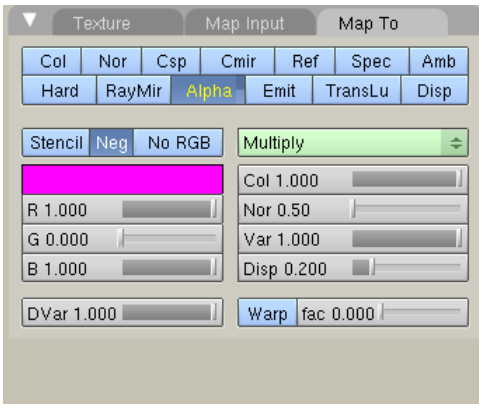
LA:Lamp Dist: 7.50
Inverse Linear
Energy 2.774
R 1.000
G 0.403
B 0.000
Classical
SpotSi 42.68
SpotBl 0.150
HaloInt 0.613
ShadowBufferSize: 2560
Samples: 16 Halo step: 12
Bias: 5.000 Soft: 100.00
ClipSta: 0.10 ClipEnd: 5.00
SampleBuffers: 1 4 9
MA:Shadow only
ME:Cube OB ME 1 Mat 1
Zoffs: 0.00
Shad A 1.00
R 0.800
G 0.800
B 0.800
A 0.001

LAMP SETTINGS !

SHADOW OBJECTS

HOWEVER, one final point remains : background stars ! Time and again I see renders where the background stars look BRIGHTER nearer the foreground star, in utter and crazed defiance of all laws of nature. It is rather irritatingly complicated to avoid, but it can be done. What you need to do is to use a texture mask to blank out the stars close to the star. If you want total realism and don't have any background stars, well huzzah, you can stop reading now. Away with you.

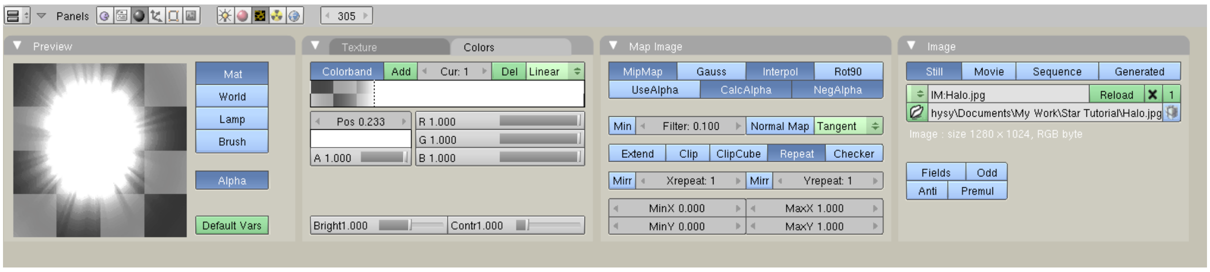
Firstly, render your background stars and save the image. Then render the spot halo layer from your main star file and save that image. Then, create a plane which fills the camera view (you might want to do this in a new .blend and use the final image as a background image in the main star file). Make the material shadeless and give it two textures, firstly the stars (mapped only to control colour) then the halo (controlling only negative alpha). Also select the "neg" option next to stencil :



Multiply
R 1.000
G 0.000
B 1.000
Col 1.000
Nor 0.50
Var 1.000
Disp 0.200
DVar 1.000
fac 0.000

To make sure the texture is facing the camera at all times use the "win" projection mode instead of the default "orco". "Ztransp" is not needed, but "shadeless" mode is.

The texture settings for the stars image does not matter very much, be sure only to lower the filtering value as much as possible. But for the halo, you also have to select "calc alpha" and "neg alpha" (this seems like a quadruple negative, but meh, it works). In fact you don't really need calc alpha unless you use the colourband, which I prefer to do because it gives you greater control over the apparent extent of the halo beam.



305

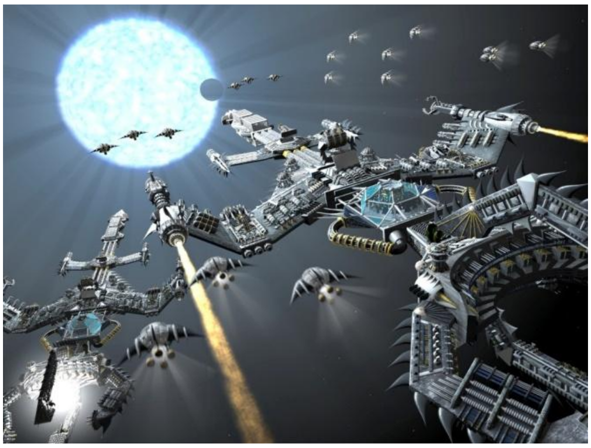
Cur: 1
Linear
Pos 0.233
R 1.000
G 1.000
B 1.000
A 1.000
Bright1.000
Contr1.000

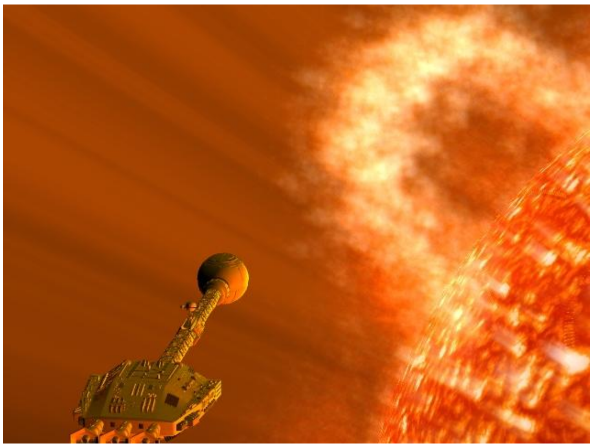
Filter: 0.100
Tangent
Xrepeat: 1
Yrepeat: 1
MinX 0.000
MaxX 1.000
MinY 0.000
MaxY 1.000

IM:Halo.jpg
1
hysy\Documents\My Work\Star Tutorial\Halo.jpg
Image : size 1280 x 1024, RGB byte

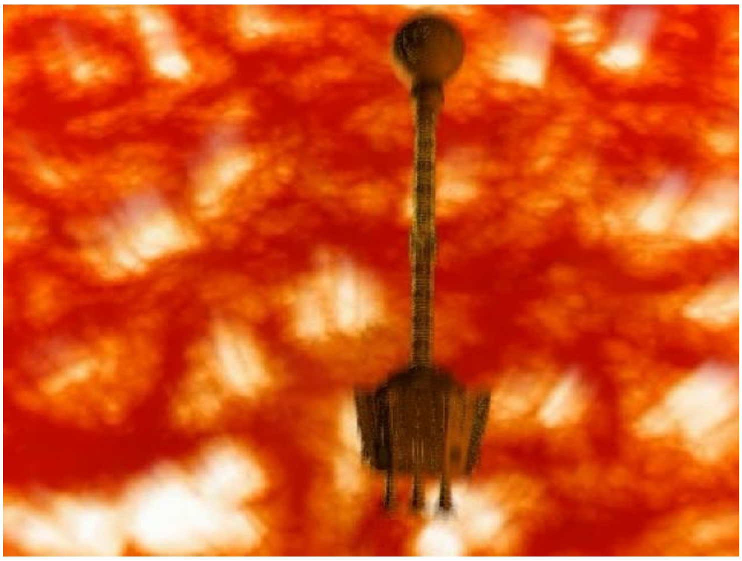
Then you can render your image, save it and use it in the background of your star. And then the stars will more correctly fade with proximity to it, rather than getting brighter. Since the Star.blend doesn't have any of this set up, here's another example file.

Example images – shameless self-promotion because it's MY tutorial and I can if I want to :





You can also use dupliframe planes to create rather nice close-up star surfaces too :



Guide to the Star.blend file :

Layer 1 : Particle system

Layer 2 : Star surface

Layer 3 : Spot halo

Layer 4 : Star atmosphere

The End.

Feedback ? Send me a PM on blenderartists (Rhysy 2), or email feedback@rhysy.net