

Animating dawn over the sea...

Mastering 3D Graphics ~ issue #2

with the environment shader

SoftImage 3D

~ Maicol Borghetti

(c) 1999 Imago Edizioni (<http://www.imagonet.it>)

In this tutorial we will show how to create a complex and stunning scene without using image maps, but by relying exclusively on procedural textures. This spectacular dawn is pure maths and has been created without the help of any photographic image!



Defining Parameters

These are the main parameters that I used to create the shader for the scene. But you may have to experiment many times to reach the desired results. However, using a pre-formulation of the settings can help.

In other words before completing any render tests, it is fundamental to have a clear idea of what we plan to do.

Consequently, it is necessary to estimate the main parameters. We have to evaluate if the atmosphere is dense or not, if the planet has a large or small radius, if the sun will have a crisp or a hazy outline, if the moon will be visible or not, if our clouds will be thick and clean or light and hazy...

Knowing, in the preliminary phase, all these aspects, the use of the shader becomes easier and the number of test renders will be far fewer.

Sky shader (defining settings)

In the Model panel of SoftImage, let's create a cardinal sphere using the default settings.

Scale it, so the camera is inside the sphere. In our scene the sphere is scaled using 80 units on the three axes.

Now we can set the shader parameters in the Matter panel. Choose a constant material for the sphere, using the default values; then select "material shader" and choose the shader that best suits the scene that we want to produce. In this case I started with the SUNSET1 shader because it has the best default settings to create this kind of atmosphere in the scene. The shader is very complex, but because of this feature it is possible to define the best settings to obtain great results.

We will analyze the main parameters to see how they change the shader. Let's activate the automatic refresh on the shaderball preview so you can see the changes to the shader immediately, see [figure 1](#).



Figure: 1.

Global Parameters ([figure 2](#))

View Position: the parameters that define the main aspect of the shader are: I (inclination) A (Azimuth) Al (height). Varying these parameters makes it

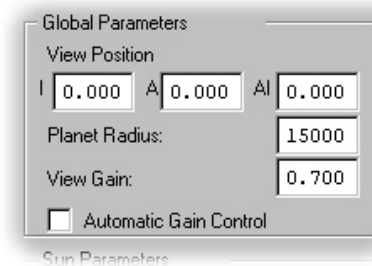


Figure: 2.

possible to determine from which inclination, azimuth and height the sky is being viewed.

Planet Radius: this defines the radius of the planet, the default setting is the same as that for planet earth: 6,000,000 m.

View Gain: this one controls the global brightness of the scene; through this parameter you can define the brightness of the shader itself. (Activating the automatic gain control gives a value to the View Gain according to the kind of map set in the Ground Map Parameter.)

Sun Parameters (figure 3)

These are the main parameters in our scene because they define the colors of the shader and therefore the final visual appearance:

Size: defines the size of the sun, the value being defined in degrees.

Inclination: defines the height of the sun above the horizon, the value being defined in degrees. A value of zero is equivalent to the sun being on the line of the zenith, whilst a value of 90 means that the sun is on the line of the Horizon.

Azimuth: this value expresses the position of the sun on the XZ level; zero is equivalent to a parallel position to the X-axis, while a value of 90 equates to a parallel position to the Z-axis.

Intensity: with this parameter you can change the brightness of the sun in relation to the other elements of the shader. This allows us to control the general appearance of the sun.

Halo size: defines the size of the halo created by the sun. It is possible to have direct control over the shading of the light-disc produced by the size parameter, and therefore enlarge or reduce it.

Attenuation: this parameter is similar to the View Gain parameter, but it affects only the brightness of the sun, giving it specific attenuation.

Color: is one of the most important parameters of the shader. This one, in fact, controls the general atmosphere of the scene.

Clouds Parameters (figure 5)

These parameters define the shape and amount of clouds in the sky. This

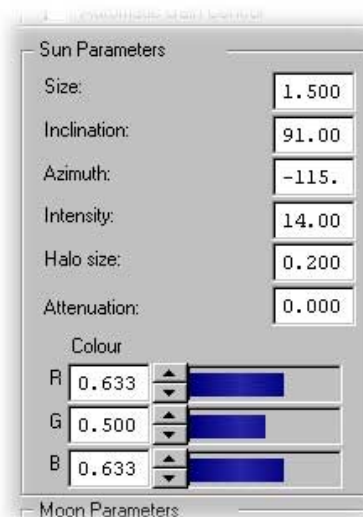


Figure: 3.

is a fractal noise algorithm and therefore you should be familiar with its control settings.

Amplitude: the amplitude of the fractal noise that creates the clouds.

Centre: this one is an important parameter, it can vary between 0 and 1 and defines the cloud amount in the sky. Setting it to zero will result in a cloudless sky, whilst setting the value to 1 will cause the sky to be totally covered in clouds.

Ratio: varies the fractal ratio.

Iterations: with this value you can vary the number of repetitions of the fractal noise. The lower the value the more “soft” the clouds will appear and vice versa.

Altitude: defines the height of clouds in our simulated sky.

Turbulence: with this flag activated, you will obtain a more turbulent fractal noise.

Animated/Time: allows the possibility of animating the fractal noise with the time parameter.

Colour: defines the color of clouds through the use of the RGB controls.

Scales: defines the size of the fractal noise in relation to the other elements of the shader. Through experimentation, you can obtain clouds of different sizes.

Offset: this parameter is mostly used for animation; varying it with two keys can simulate the effect of wind on the clouds.

Rotation: this parameter is used in animation in order to eliminate the repetition effect of the shader.

Giving the right atmosphere to the scene using the Depth fading (figure 7)

If you try to set up the parameters used on the sky shader and render the scene you will obtain a different result compared

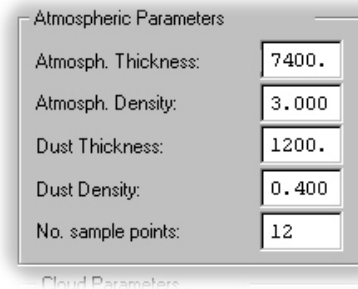


Figure 4.

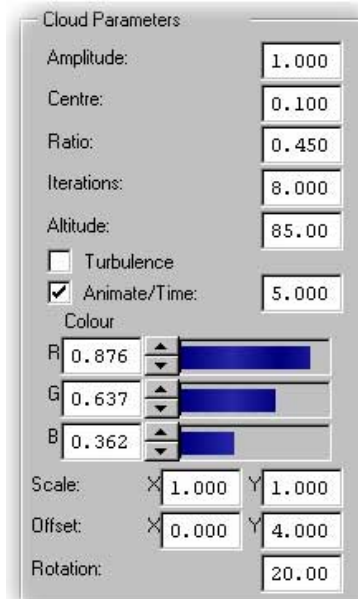


Figure 5.

with that you can see in the final render. In that image the main color is always somewhat orange and is obtained using the Depth-Fading.

Before setting up the depth-fading values you have to measure the distance of the camera from the horizon (the horizon corresponding to the end of the sky sphere). So make sure the camera is visible and then select INFO>DISTANCE (see [figure 6](#)).

With this measure you will be able to set the depth fading in a less accidental way, and it will be a reference measure to eliminate errors when setting values

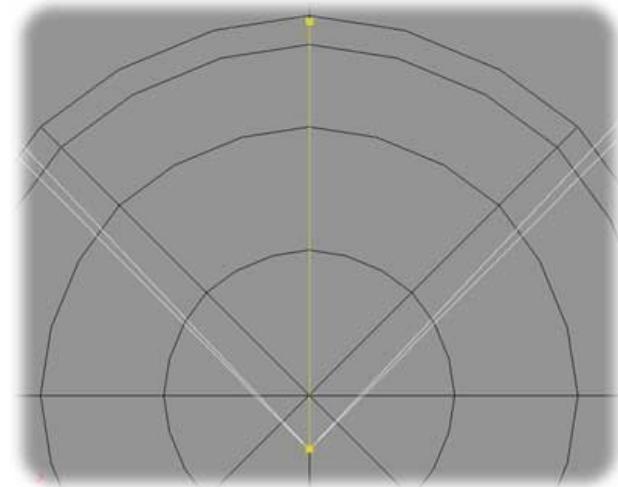


Figure: 6.

Starting Distance: defines the camera distance value where the depth-fading begins to influence the image.

Ending Distance: defines the distance – always starting from the camera – where the depth-fading effect finishes.

Transparency: this variable establishes how much the depth-fading influences the scene. With the value set to 1 its effect is not remarkable. We set the value to 0.4, which gives the desired color over the scene, but still allows you to appreciate the tones of the shader. If we had used a smaller parameter we would have risked covering the entire scene.

Color: the choice of color is used to obtain an orange atmosphere over the entire scene.

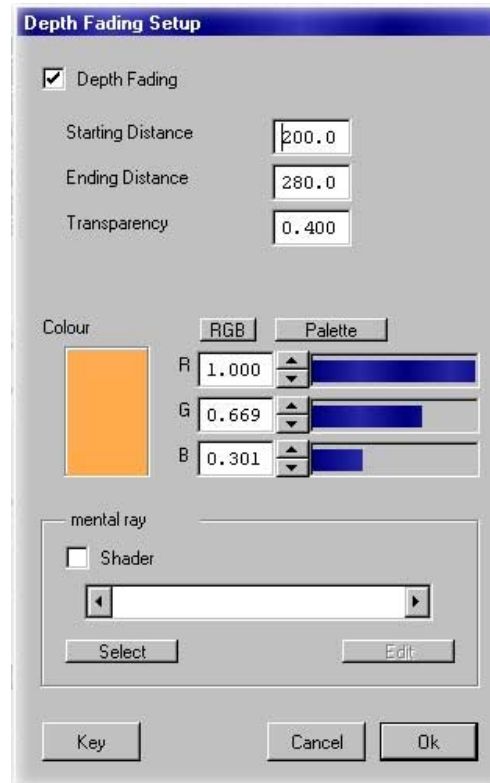


Figure: 7.

Water: defining the material, the texture and using the Lume Tools

Now let's create the water. In the Model panel select GET>PRIMITIVE>GRID and create a polygonal grid using the default settings.

As for the creation of the sky sphere, you have to scale the grid until it matches the sphere size. In this scene it has been scaled to 90 units.

When defining the material and the texture it is necessary to point out that we used one of the best SoftImage plug-ins for the creation of the water – the LumeTools. This plug-in allows – without significantly increasing the rendering times – the creation of great marine scenes, both over and under the surface.

The Lume shaders used in this scene are:

Material shader: water surface, it helps to simulate the

reflections of the water as you view it from different angles (see [figure 8](#)).

Volume shader: submerge, helps to simulate the loss of intensity of light under water and therefore, it also helps to vary the transparency of the water according to the depth (see [figure 9](#)).

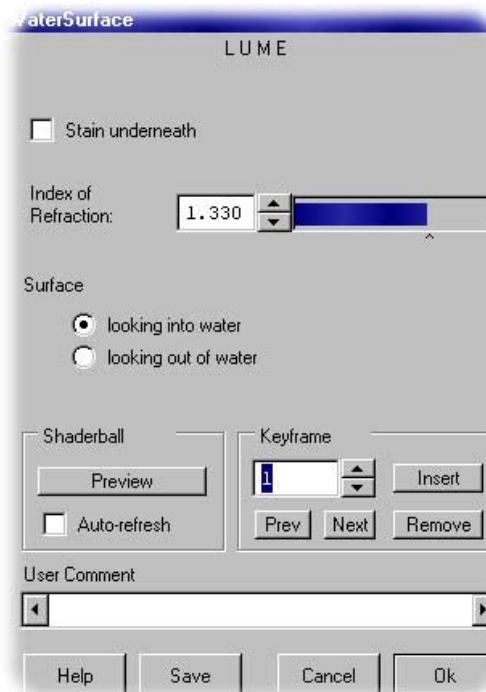


Figure: 8.

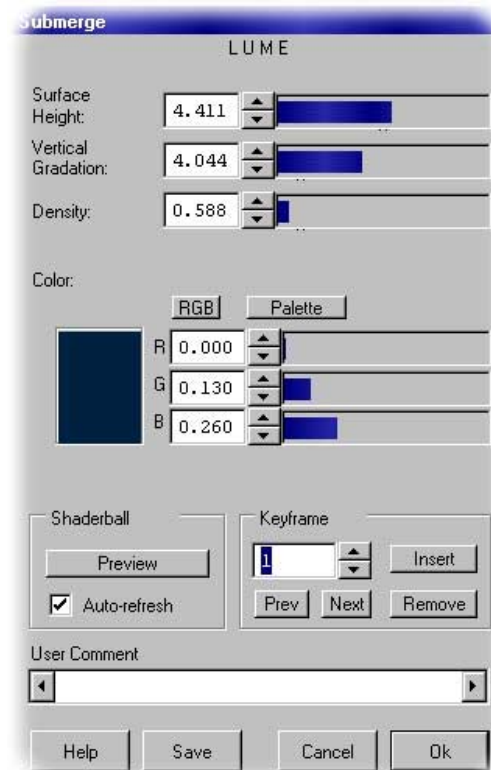


Figure: 9.

Texture shader: ocean, is the main shader in the scene, it gives a realistic bump texture to the grid and also the typical surface movements of a liquid.

Material

The definition of the material is simpler: the water is both a reflective and transparent material. The viewing angle defines the transparency and reflection values.

The use of a material shader as the water surface gives us a perfect equilibrium. The default settings are ideal for our purpose. We can simulate the loss of light intensity at depth under the water surface, and also the variation of transparency of the water using the volume shader submerge. This shader will help us during the creation of the beach, when we have to reproduce the transparency of the water at low depths in the deep sea.

Texture

In this panel you can set the ocean shader, the most important plug-in, simulating the surface.

The ocean shader generates a procedural bump on the grid, providing a realistic liquid surface.

After creating the grid and giving it the above-listed materials, select TEXTURE>2D GLOBAL and assign a

“dummy” image to the grid, then select shader and choose ocean. Now let’s see the shader settings:

Wave size

These values define the dimension and the number of waves (see [figure 10](#)).

Largest: increasing this value you will have a greater number of waves of larger dimensions.

Smallest: defines the number of small waves.

Number between: is the number of waves in-between the largest and the smallest ones.

Steepness: controls how calm or rough the sea is; increasing it will produce more rippling seas.

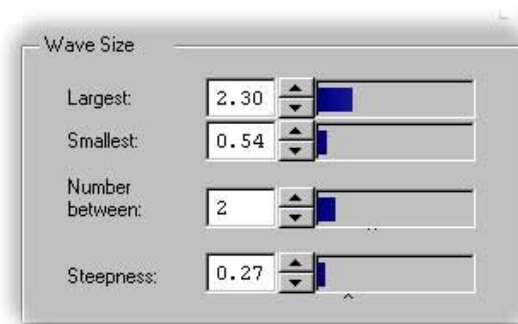


Figure: 10.

Wave position (figure 11)

Relative to: if the position of the waves is relative to world center you will not have any change even with the translation of the object on the ZX plane, while if they are relative to the center of the object the translation will move the waves with it.

Directionless: this parameter sets the direction of the waves and that can be indefinite (directionless) or be defined in degrees by the user.

Wave animation (figure 12)

Wave speed: increases or reduces the speed of movement of the waves.

Loop animation: gives a loop to the procedural animation of the shader, deciding when it has to repeat itself.

Shape Method (figure 13)

This can be of two kinds: a simple single bump on the tex-

ture, or a more complex one like a displacement map on the object. Using the second method it is necessary to have a grid with more vertices and so slowing everything down.

Flats (figure 14)

Across the sea's surface there are more or less rippled areas, activating flats you insert some doldrums and can simulate these areas. Up to this point, the rendering is reasonable, a beautiful orange sky rise above a realistic sea (figure 15). Now let's insert the beach and the rocks.



Figure: 11.

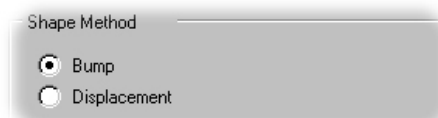


Figure: 13.

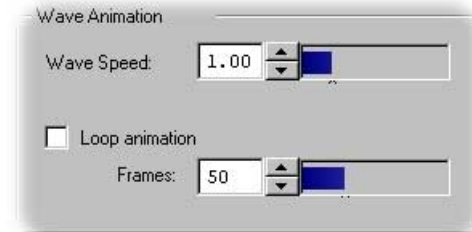


Figure: 12.

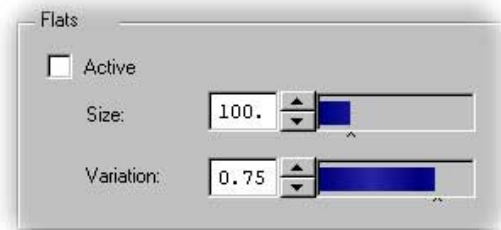


Figure: 14.



Figure 15: An intermediate rendering. You have to place the rocks and animate the scene.

Sand grid (Defining the surface with the bump map)

The beach is very simple. In the Model panel insert GET>PRIMITIVE>GRID and choose a cardinal grid to obtain a softer model.

Insert points according the definition and the width of the beach. Use the parameter DEFORMATION>BUMPMAP + to obtain the dunes and the depressions.

Selecting a grayscale image, previously created in PhotoShop, I applied this tool to the grid and obtained a nearly perfect deformation.

Naturally if you don't get immediate and appreciable results, you can always make small modifications to the grid. As an example, scale it on the Y-axis, or rotate it until you get a nice shot of the dunes and depressions. The modifications can vary according to the scene shot.

Modelling and defining the materials & 3d texture of the rocks

Like the beach, we have another relatively easy object to model in the scene, the rocks. In the Model panel use GET>PRIMITIVE >SPHERE leaving the default settings.

You will obtain a polygonal sphere that is easy to deform. Pulling points and scaling it (free form modeling) you will obtain nice looking rocks (like those in the image) in less than 5 minutes.

Once you have a nice shape, and if you want a softer model, apply an EFFECT>ROUNDING 0.5 to smooth all the deformations we had previously created by manipulating the control points of the sphere.

Creating other rocks is easy, let's simply duplicate, scale and rotate the objects until they look different to the original.

Defining the material is also straightforward. There are no shaders, you have to select a kind of lambert material only and give it a color similar to that of the rocks in Diffuse.

Defining the texture is a little more complex. I avoided painting some “rocky” maps in Photoshop and instead used a shader on the 3d texture – the 3DMarble_Amber (figure 16). It is very effective and its controls allows us to obtain the desired results with very little in the way of rendering tests, thanks to the shaderball pre-view.

Before starting to set the shader, it is necessary to load a 3d texture already defined in SoftImage; that’s the ROCKS1 texture in the folder: ROCKS.

We used this texture because the ambient, diffuse and specular settings are at zero and the roughness is already set to an acceptable value for us. Select the shader and edit its settings. Using the default set-

tings we can get a nice result, but varying the parameters we can get the very best from it.

These settings define the color of the veins in the marble, their width and spread. Because we previously selected the rocks1 texture, the color of our rocks will not change. It will vary the color of the veins, but it will be only defined from that one chosen for the material. Any-

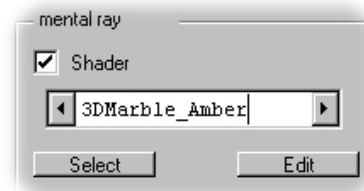


Figure: 16.

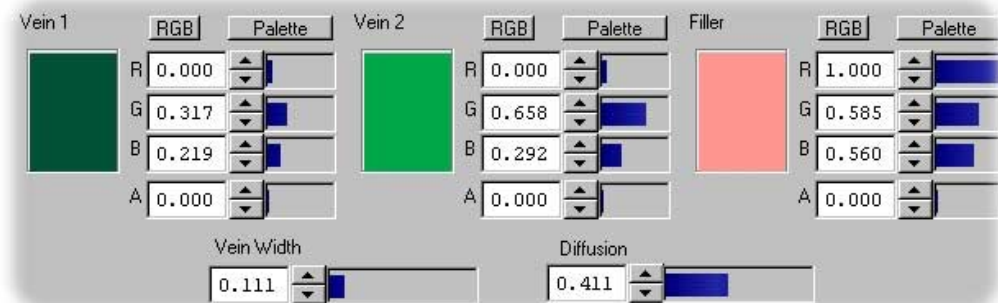


Figure: 17.

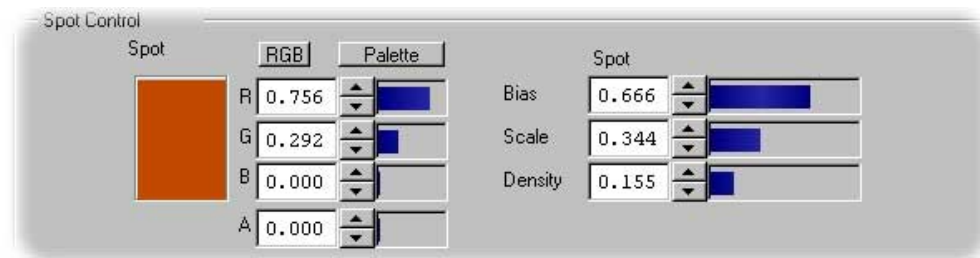


Figure: 18.

way, the value of these 3 parameters (vein 1, vein 2 and filler) remarkably changes the aspect of the texture modifying the roughness (figure 17). The spot creates some flat areas on our marble (figure 18). We can use these controls to create more or less smoothed areas on the rock.

This shader is a fractal noise and like all those we've previously used, is defined by some parameters. These are the main values that influence the final result, because they define the shape, the size and the accuracy of our rocks (figure 19).

Defining the color and position of the luminous sources

The luminous sources, their color and position are the most important aspects in a 3D scene. In fact if they are badly positioned, they can spoil many hours spent on modeling complex objects.

Imagine the scene that we are creating, illuminated as in figure 20.

With the help of lights we can try to match reality. Rather than going through a long technical speech we can reproduce realistic lighting with a little careful thought.

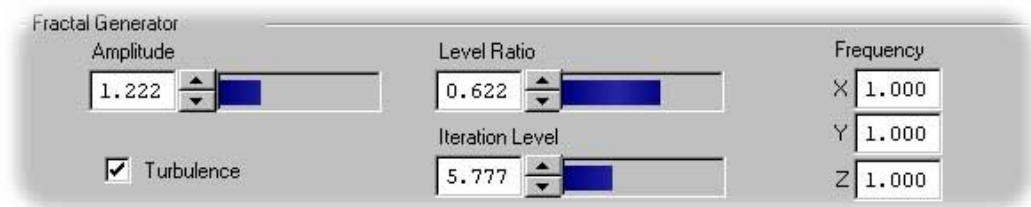


Figure: 19.



Figure 20: Using wrong lighting settings could endanger the final results.

The lighting of this scene is rather simple because the only light source is the sun. The sun, when is at this height on the horizon, creates an orange light with a strong contrast on the objects placed between us and the luminous source, so this kind of lighting can help us (figures 21 and 22).

In positioning this light source we must respect all the previous settings like the azimuth of the sun in the sky shader, so let's place the light at the point where the shader defines the sun. Now let see the final results.

Animation (sky shader, depth fading, sun)

After achieving a satisfying result on a single frame, we can animate the whole scene, conveying the idea of the sun rising and illuminating the scene.

Before setting the keyframes we have to select the panel PREFERENCES>ANIMATION and change the Fcurve Default Interpolation from spline to linear.

This operation is necessary to make the function curve created when we set the keyframe on the sky shader linear, for the light and the depth fading. This is necessary because the rotation of the earth is linear and therefore the change in height of the Sun is constant.

On the curve of the sky shader you will not get the desired effect. To make it linear you have to set it manually, selecting FCRVSELECT>MATERIAL>MATERIAL SHADER>ENVATMOS_SUNSET2>SUN_INCLINATION in the panel Motion.

Set up a frame rate of 12 (maybe be our animation could be used on a multimedia CD-ROM) calculating that 40 seconds equals 480 frames.

Our goals is to make the sun rise until it reaches the position in figure 24, so we can set a key in last frame of our animation of the sky shader, the light and the depth fading.

Now let's see the main settings to animate:

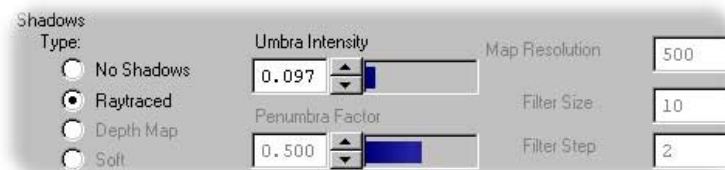


Figure: 21.

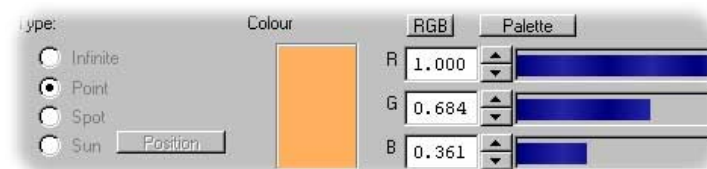


Figure: 22.

Sky shader: we previously said that this shader is a simulator, in fact animating only the parameter that varies the height of the sun on the horizon (inclination) we obtained the previewed results.

Go to frame 1 and after opening the sky shader window, set the inclination to a greater value than the previously used one and insert a key.

Depth fading: this control helped us to define an orange atmosphere, but if you want to represent a previous storm, you have to change some values. So darken the color and increase the transparency (important). In this way its influence on the scene will be smaller and lose the orange atmosphere.

Now let's define a key at frame 1 (figure 23).



Light: we animate the light by modifying the color, in fact using a darker color we can simulate exactly the change of light at that moment.

Its position, and above all its height, could be varied, but 40 seconds is too short a period to see substantial changes in the shadows. So we shall leave it there.

After changing the color, set a key at frame 1. How do we find the exact values to the parameters for these three aspects (sky shader, depth fading and sun)? The final result can be obtained only with these three aspects all together, so you have to experiment with some rendering tests.

Following the above discussion, you should be able to get some nice results in a short time.



Figure 23: (left) The first frame shows a darker scene with a blood red sky. Figure 24: (right) The final scene created following this tutorial. The image is also the final frame of the sun rising animation and starts from a darker environment that you can see (figure 23).



Preview

Refresh



Auto

Global Parameters

View Position

I A AI

Planet Radius:

View Gain:



Automatic Gain Control

Sun Parameters

Sun Parameters

Size: 1.500

Inclination: 91.00

Azimuth: -115.

Intensity: 14.00

Halo size: 0.200

Attenuation: 0.000

Colour



Moon Parameters

Atmospheric Parameters

Atmosph. Thickness:

7400.

Atmosph. Density:

3.000

Dust Thickness:

1200.

Dust Density:

0.400

No. sample points:

12

Cloud Parameters

Cloud Parameters

Amplitude:

1.000

Centre:

0.100

Ratio:

0.450

Iterations:

8.000

Altitude:

85.00

☐ Turbulence

☒ Animate/Time:

5.000

Colour

R 0.876



G 0.637



B 0.362



Scale:



1.000



1.000

Offset:



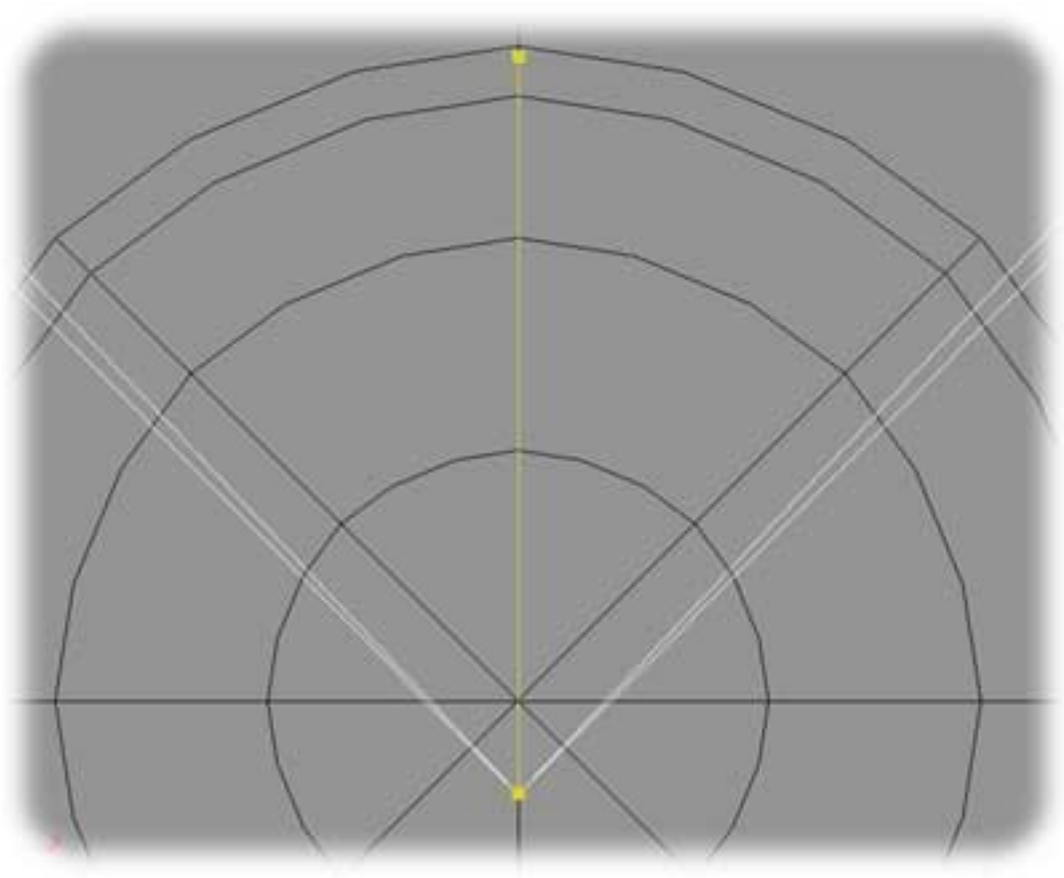
0.000



4.000

Rotation:

20.00



Depth Fading Setup

☒ Depth Fading

Starting Distance

Ending Distance

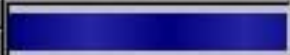
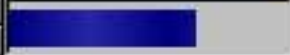
Transparency

Colour

RGB

Palette



R	1.000	 	
G	0.669	 	
B	0.301	 	

mental ray

☐ Shader

A horizontal bar with a left-pointing arrow on the left and a right-pointing arrow on the right, used for selecting a shader.

Select

Edit

Key

Cancel

Ok

L U M E

☐ Stain underneathIndex of
Refraction:

1.330



Surface

- ☒ looking into water
☐ looking out of water

Shaderball

☐ Auto-refresh

Keyframe

1



User Comment

L U M E

Surface
Height:

4.411

Vertical
Gradation:

4.044



Density:

0.588



Color:

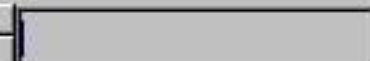
RGB

Palette



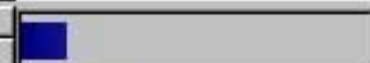
R

0.000



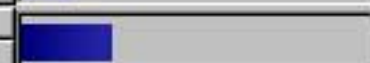
G

0.130



B

0.260



Shaderball

Preview



Auto-refresh

Keyframe

1



Insert

Prev

Next

Remove

User Comment

Help

Save

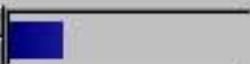
Cancel

Ok

Wave Size

Largest:

2.30



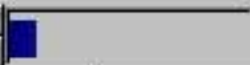
Smallest:

0.54



Number
between:

2



Steepness:

0.27



Wave Position

Relative to:

- ☒ World Center
☐ Object Center

Direction:

- ☒ Directionless
☐ Directed

Angle: °

Wave Animation

Wave Speed:

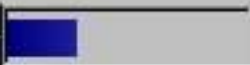
1.00



☐ Loop animation

Frames:

50



Shape Method



Bump



Displacement

Flats

☐ Active

Size:

100.



Variation:

0.75





mental ray



Shader

◀ 3DMarble_Amber ▶

Select

Edit

Vein 1



RGB

Palette

R

0.000

G

0.317

B

0.219

A

0.000

Vein 2



RGB

Palette

R

0.000

G

0.658

B

0.292

A

0.000

Filler



RGB

Palette

R

1.000

G

0.585

B

0.560

A

0.000

Vein Width

0.111

Diffusion

0.411

Spot Control

Spot



RGB

Palette

R

0.756

G

0.292

B

0.000

A

0.000

Bias

0.666

Scale

0.344

Density

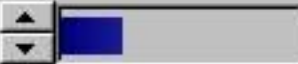
0.155

Spot



Fractal Generator

Amplitude

A horizontal slider bar with a blue fill. The value 1.222 is displayed in a text box to the left of the slider.

☒ Turbulence

Level Ratio

A horizontal slider bar with a blue fill. The value 0.622 is displayed in a text box to the left of the slider.

Iteration Level

A horizontal slider bar with a blue fill. The value 5.777 is displayed in a text box to the left of the slider.

Frequency

X

Y

Z



Shadows

Type:



No Shadows



Raytraced



Depth Map



Soft

Umbra Intensity

0.097



Penumbra Factor

0.500



Map Resolution

500

Filter Size

10

Filter Step

2

Type:

☐ Infinite

☒ Point

☐ Spot

☐ Sun

Position

Colour



RGB

Palette

R 1.000

G 0.684

B 0.361

