Overview

Physically Based Shading brings with it different ways of considering lighting for levels within the CRYENGINE. This document will take you through the workflow that is recommended when lighting levels using Physically Based Shading introduced in CRYENGINE 3.6.

- Important Changes to Time of Day Properties
- Step-by-Step Workflow for Lighting Levels
  - Examples
- Further Considerations
  - Local CubeMaps
  - Resolution
  - Why is Luminance so Important?
  - How does Tone Mapping Work?

Important Changes to Time of Day Properties

The goal of using Physically Based Shading is maintaining the integrity of materials. As such, in CRYENGINE 3.6 we dropped some lighting approaches that were not conducive to PBS.

These dropped lighting approaches are:

- Constant ambient term.
- Hemi-spherical ambient term.

The reasoning behind this is that these lighting approaches would flatten the material as they only contain diffuse and no specular contribution.

What this mean in application is that some settings within Time of Day are no longer used and have been removed. These values are:

- Sky Color.
- Sky Color Multiplier.
- Ambient Ground Color.
- Ambient Ground Color Multiplier.
- Ambient Min Height.
- Ambient Max Height.

You will note that in the following tutorial all ambient lighting is done through the use of Environment Probes as they contribute to specular and also apply directional colored ambient lighting locally to the probe.

Step-by-Step Workflow for Lighting Levels

To complete this tutorial you will need to be using CRYENGINE 3.6 and already have a new level created.

1. To begin, we must first set the correct luminance for our level.

2. Open the Console in Sandbox and type the following command "r_HDRDebug=1" this command disables the in engine tone mapper and displays the average luminance for the scene in debug text as seen in the image

   **Average Luminance (cd/m2):** 12851.56
   **Estimated Illuminance (lux):** 95937.5
   **Exposure:** 1.00  **SceneKey:** 0.40

3. To get an accurate value of what luminance your level is currently at you must place a pure flat white plane that is rotated to be perpendicular to the sun as seen in the image. It doesn't have to be perfect, but note that the value will change significantly if the light from the sun hits the plane at a glancing angle.

   You can find the plane object in: Objects/default/primitive_plane_small.cgf

   Assign a pure white material with no specular color. Use `textures/defaults/white.tif` in the material.

   To get the angle correct to the sun, watch the shadow of the plane. It will be biggest in width and length when it's perfectly perpendicular thus, rotate the plane whilst observing the shadow with the goal of having the largest shadow possible.
4. Next, zoom your camera in so that the light side of the white plane takes up the entirety of your view-port.

5. Note the Estimated Illuminance (lux) value.

6. Now we must decide on the actual amount of luminance we want for the level. Do this by referring to the following table which lists some Real-World Values.

<table>
<thead>
<tr>
<th>Real World Setting</th>
<th>Amount</th>
<th>Unit</th>
<th>Ratio</th>
<th>Artistic Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Moon</td>
<td>0.25</td>
<td>LUX</td>
<td>0.00005</td>
<td>--</td>
</tr>
<tr>
<td>Living Room</td>
<td>50</td>
<td>LUX</td>
<td>0.01</td>
<td>--</td>
</tr>
<tr>
<td>Sunrise with Clear Sky</td>
<td>400</td>
<td>LUX</td>
<td>0.08</td>
<td>--</td>
</tr>
<tr>
<td>Office</td>
<td>500</td>
<td>LUX</td>
<td>0.1</td>
<td>--</td>
</tr>
<tr>
<td>TV Studio</td>
<td>1000</td>
<td>LUX</td>
<td>0.2</td>
<td>--</td>
</tr>
<tr>
<td>Overcast Day</td>
<td>15000</td>
<td>LUX</td>
<td>3.0</td>
<td>Approximately 1.5</td>
</tr>
<tr>
<td>Indirect Sunlight (IN SHADOW)</td>
<td>20000</td>
<td>LUX</td>
<td>4.0</td>
<td>Approximately 2.0</td>
</tr>
<tr>
<td>Direct Sunlight (DIRECTLY LIT)</td>
<td>100000</td>
<td>LUX</td>
<td>20.0</td>
<td>Approximately 10.0</td>
</tr>
</tbody>
</table>

7. Once you've established which value you want to represent open the Time of Day Tool.

8. In Time of Day there are a four values that will directly affect this. They should be used to achieve the desired luminance.

```
Sun
   Sun color
   Sun intensity (lux) 120000.0

Sky Light
   Sun intensity
   Sun intensity multiplier 200.0
```

9. Now that we've set our luminance values to mimic real world values it's time to setup our ambient lighting. You can now re-enable the tone mapper by entering the following command into the console "r_HDRDebug=0".

10. As all ambient lighting is now done through the use of image based lighting probes, otherwise known as environment probes, place one into your level now. The entity can be found under Misc -> EnvironmentProbe.
11. Environment Probes act very similarly to lights wherein they have a radius, however, in the case of CRYENGINE we use a box size to ensure the accuracy of the probes reflections. Set the parameters **BoxSizeX**, **BoxSizeY** and **BoxSizeZ** of this first probe to encompass the entirety of your level.

At this point you should treat your initial environment probe as a global one. What this means is that it will provide the entire level with ambient lighting information, however, realize that this is calculated from the probes location.

We will add smaller more localized probes later which will override the global one when they are nested inside of the global bounds.
12. Set the Diffuse Color, Diffuse Multiplier and Specular Multiplier to normalized values RGB: 255,255,255 and 1 as in the image above.

13. Next we can now generate our Cubemap. Do this by clicking Generate Cubemap.

When you generate a cubemap there are three textures that will be created in the directory: Textures\cubemaps\MYLEVELNAME\...

One texture is the reflections and specular contribution another is the diffuse component and finally the source TIF for both. The name of the texture will be set to the same name as the environment probe entity.

14. The major difference you will notice will be in the shadows as with the addition of the diffuse lighting from the probe they will not longer be pure black.
Examples

In the following images you can see the difference between active and inactive environment probes. Note that the contents of the scene are baked into and then re-projected out from the environment probe and thus the variety of indirect lighting and it’s apparent effect will vary greatly.

15. It is highly recommended to place a reflective and perfectly smooth sphere near each environment probe in the level. This is to visually ensure that the cubemap is still accurate. Should it look different than the environment around it, it would likely need regenerating. You can easily do this by placing a geom entity with the following parameters:

- **Model:** Objects/default/primitive_sphere_noproxy.cgf
- **Material:** Materials/references/black_reflective
- Under the entity properties of the GeomEntity, tick **HiddenInGame** so they don’t appear.

16. The last step is to generate the cubemap a second time. The reason for the second generation is that you will be able to bake in **retro reflections** from the original cubemap. Basically reflections of the original cubemap on objects will be baked in in this second cubemap generation.

17. Congratulations you have now setup a physically based lighting situations and can start adding assets, lighting, and more to your scene. Remember to re-bake your cubemaps as your level progresses.

Further Considerations

As the preceding tutorial is based on creating a new level there will be situations in which you have a much more populated level and there are some points that need to be addressed when working in more complete levels.

Local CubeMaps
The purpose of a local cubemap is to more accurately represent the lighting condition in smaller areas. Ensure that all areas within your level have some amount cubemap coverage.

Some additional parameters need to be considered in these situations:

- **Box Size X,Y,Z** need to be set to encompass the area you need.
- **Specular** and **Diffuse** values should be set to maintain consistency and as such a value of 1 at pure white for both is recommended.
- **Specular** and **Diffuse** values **lower than 1**. The CRYENGINE shading model relies heavily on consistency and accuracy and as such it highly recommended not to use values any lower than 1 for local cubemaps. If you find that lighting looks incorrect then consider the following:
  - Specular is too high/low:
    - Check the cubemap for errors.
  - Diffuse it too high/low:
    - Check cubemap texture for errors.
    - Re-check luminance values for the level.
    - Add a separate light entity to provide additional ambient light and re-generate cubemap.
- **Sort Priority** Any Overlapping cubemaps need to have a sorting priority assigned to them. The recommended way to do this is by basing sort priority on visual interest which mean give more obvious noticeable cubemaps (say with a specific object reflected in it) higher priority in those areas.
- **Fall Off** When using local cubemaps it’s recommended to always adjust the falloff value so that all areas are 100% affected by the cubemap. A balance will have to be struck between a smooth falloff (and thus transition) for each probe against its coverage.
  - **AttenuationFalloffMax** is used to control how the influence of the cubemap falls off.
    - A value of 0 means that the box shape will have hard edges and there is no falloff (cheaper performance).
    - A value of 1 means the falloff will begin at the center of the box and blend out to the box extents (most expensive on performance).
    - A value of 0.8 means the falloff begins at 80% of the extents of the box shape.

**Resolution**

In most cases it’s not required to have much more than a 256px resolution for your cubemaps. Currently the Environment Probe system supports the generation of 256px cubemaps only, and the renderer has been tuned for this values to achieve the best results without spending unnecessary resources on higher resolution textures with little benefit.

**Why is Luminance so Important?**

The Purpose of luminance is to create a balance of ratios between bright and dark objects in the area and get the right amount of auto-exposure.

The luminance ratios are very important to get natural looking results out of our shader and tone-mapping models, and you should adhere to them as close as possible, whilst of course still taking composition into consideration. It should also be noted that real world values are sometimes not possible in the engine so a balance is really what’s important here.

**How does Tone Mapping Work?**

The scene key, exposed to Time of Day in the HDR settings interface, controls the amount of exposure and determines if the tone mapped image appears relatively bright or dark
It is estimated automatically from the average scene luminance, that’s why it is important to use standard luminance levels.

A low scene key of 0.05 gives a subjectively dark image whilst a high key of 0.8 results in a very bright image.

**A value of 0.18 gives the impression of a moderate illumination.**

The key (y-axis) is mapped to the average luminance (x-axis) using the following function.

As can be seen from the graph, an average luminance of approximately 0.3 will give a moderate illumination (scene key 0.18).
\[ 1.03 - \frac{2}{(2 + \log(x + 1)/\log(2))} \]