4. Creating Materials

4.1 Types

Instead of creating materials for your self, like one usually does in 3d modeling programs, a couple of predefined materials are used. What needs to be filled out are the different maps used.

Settings for General
Use Alpha: TODO
Depth Test:
Value:

4.1.1 Flat

This material shows the diffuse map without any lighting.
Texture units: Diffuse

4.1.2 Additive

This transparent material is blended additively with the background.

Result = source_pixel + screen_pixel

It is not affected by lighting.
Texture units: Diffuse

4.1.3 Modulative

This transparent material is multiplied with the background.

*Result = source_pixel * screen_pixel*

It is not affected by lighting.
Texture units: Diffuse

4.1.4 Modulative x 2

This transparent material is multiplied with the background times two, like this:

*Result = (source_pixel * screen_pixel) * 2*

It is not affected by lighting.
Texture units: Diffuse

4.1.5 Alpha

Each texel on the texture is transperant according to the alpha channel on the texture.

It is not affected by lighting.
Texture units: Diffuse

4.1.6 Diffuse

The diffuse map with lighting.
Texture units: Diffuse and Illumination (optional).

4.1.7 DiffuseSpecular

The diffuse map with lighting and specular term. Does not have any support for specular map.
Texture units: Diffuse and Illumination (optional).

4.1.8 Bump

This material shows the diffuse map with a bump mapped surface.
Texture units: Diffuse, Normal-Map and Illumination (optional).

4.1.9 BumpSpecular

This material shows the diffuse map with a bump mapped surface and specular term. The alpha channel of the normal map is used as specular map.
Texture units: Diffuse, Normal-Map and Illumination (optional).

4.1.10 BumpColorSpecular

This material shows the diffuse map with a bump mapped surface and a color specular term.
Texture units: Diffuse, Normal-Map, Specular and Illumination (optional).

4.1.11 EnvironmentMapReflect

This material shows the diffuse map with a environment reflection from a CubeMap.
Texture units: Diffuse and CubeMap.

4.1.12 Water

This transparent material is blended additively with the background. It also has a bobbing surface.

Result = source_pixel + screen_pixel

It is not affected by lighting
Texture units: Diffuse.

4.2 Texture units

The different material types uses texture units to calculate the value of pixel. Each texture units have the following properties:

Type:	The type of texture, 1D, 2D and Cube can be used. Normally only 2D is used.
Mipmaps:	If mipmaps should be created for the texture and should be used for most textures.
Compress:	If hardware compression should be used if available. (Not supported yet).
Wrap:	The wrap mode to use when drawing the texture, “Repeat” means texture is tiled. “Clamp” means that uv-values above 1 and below 0 are clamped. Useful on for example flat objects with transparent edges. “ClampToEdge” is the same as “Clamp” but does not blend with the opposite edge.
Anim mode:	How to play an animation, looping or bouncing back and forth with oscillation.
Frame time:	How long an image should be displayed in an image sequence, 1 = 1 image / second, 2 = 1 image / 2 seconds, 0.2 = 5 images / second etc.

4.2.1 Diffuse

The r,g and b channels contain the color values. The alpha channel is used for transparent materials.

4.2.2 Normal-Map

This is a map showing the topology of a surface in tangent space. The r,g and b channels contain the compressed normal vectors. The alpha channel is used for specifying the specular term for specular material, 0= no specularity 1 = full specularity.

4.2.3 Illumination

This map is additively added after all lights has been rendered.

4.2.4 Specular

The r,g and b channels are used to create a specular effect. The whiter the shinier, black = no shine.

4.2.5 CubeMap

Cube map texturing is a form of texture mapping that uses a 3D direction vector (a fancy phrase that means nothing more than a direction) to index into a texture that is six square 2D textures arranged like the faces of a cube. See this cube map tutorial for extensive information on it's construction.

The CubeMap is used to create a “reflection” of the environment in a texture, for example the ice in the ice cave level in penumbra.

For the HPL engine you make the 6 images as separate images, and you name them:
nameofimage_neg_x
nameofimage_neg_y
nameofimage_neg_z
nameofimage_pos_x
nameofimage_pos_y
nameofimage_pos_z

In the material editor under Texture Units at the file prompt: you add one of the images and only use the “nameofimage” excluding “_neg_x”, the engine will then automatically locate and use the 6 images.

4.3 File format

Material files has the extension “.mat”, are made in XML and has the following format:

```
<Material>
<Main Type="[material type]"/>
<TextureUnits>
<[Texture unit] File="[file name]" Type ="[type]" Mipmaps="[true or false]"
Compress="[true or false]" Wrap="[wrap mode]"/>
[additional texture units…]
</TextureUnits>
</Material>
```

Instead of handwriting these files HplHelper can be used to create material files.

4.4 Physics material

This is gives the material physical properties as well. Read more about the different types under chapter 5.6. Note that this property is only useful for static scene geometry and is NOT used for entities, entities specify their physics material in the entity file.