Painkiller sky tutorial

Using sky model

The sky model for PainEditor is a set of meshes modeled in Maya and exported as regular .mpk using our exporter. Maya's scene and *.mpk* file of sky model is independent of level map, so you can use one sky model in many levels or easily change the sky in a finished level without modifying the level map. The *.mpk* sky file should be saved in the *Data/Maps* directory, the same where the level maps are located. In PainEditor sky model is imported and set up via *SkyDome* section of level properties (Fig. 1). All sky settings are saved in text .*Clevel* file (*Data/Levels/Nameoflevel directory*), so you can also edit them with any text editor.

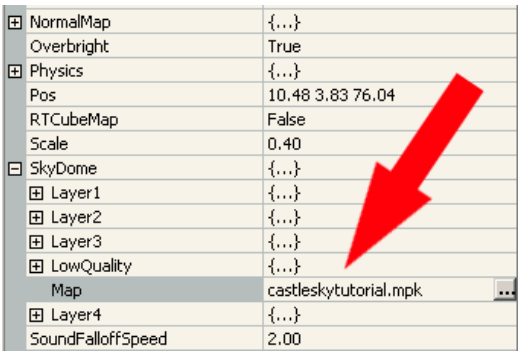

Fig 1. Sky parameters in PainEditor

You can use any sky model from Painkiller levels with your own textures and animations, so let's start by examining the structure of one of them and change its parameters. Run the PainEditor and load *C3L4_Castle* level, to see sky shown on figure 2.


Fig. 2. One of the skies from Painkiller

Sky model can contain up to 4 different meshes called *layers* and represented in PainEditor by *Layer1*, *Layer2*, *Layer3* and *Layer4* fields in *SkyDome* section. On every layer you can mix two different textures with blending mask and use lightmap where it's needed. In the *Castle* level, the sky is constructed with 3

hemispherical layers, as on figure 3. Such model is used in most of Painkiller levels.

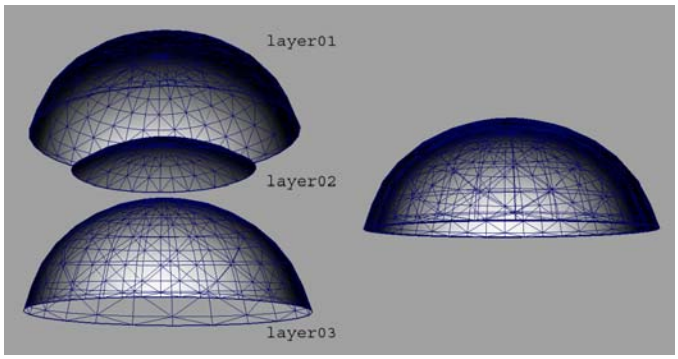

Fig. 3. Three meshes (on the left) are put together to create different layers of sky and clouds (on the right)

Loading textures onto sky layers

On the first, external surface (*layer01*), the main texture is projected and on two internal meshes (*layer02* and *layer03*) are placed moving semi-transparent clouds.
In the simplest case, when you don't want to create sophisticated animation or lighting effects, you'll need only the first layer with basic texture. So, temporarily, we can turn off all other layers and change main texture. Open *Layer2* and *Layer3* rollouts in PainEditor and load *_alpha_zero* texture from *Data/Textures/Skies* to *Tex1* and *Tex2* slots in both rollouts. This is a completely transparent texture (alpha = 0), so those layers become invisible for now. After this soft moving clouds disappear from the sky. Save level properties, using disk icon on main toolbar, so we can always go back to this starting point.
Now open the *Layer1* rollout and load any new texture to *Tex1* slot, replacing existing one (You'll see something like on figure 4).

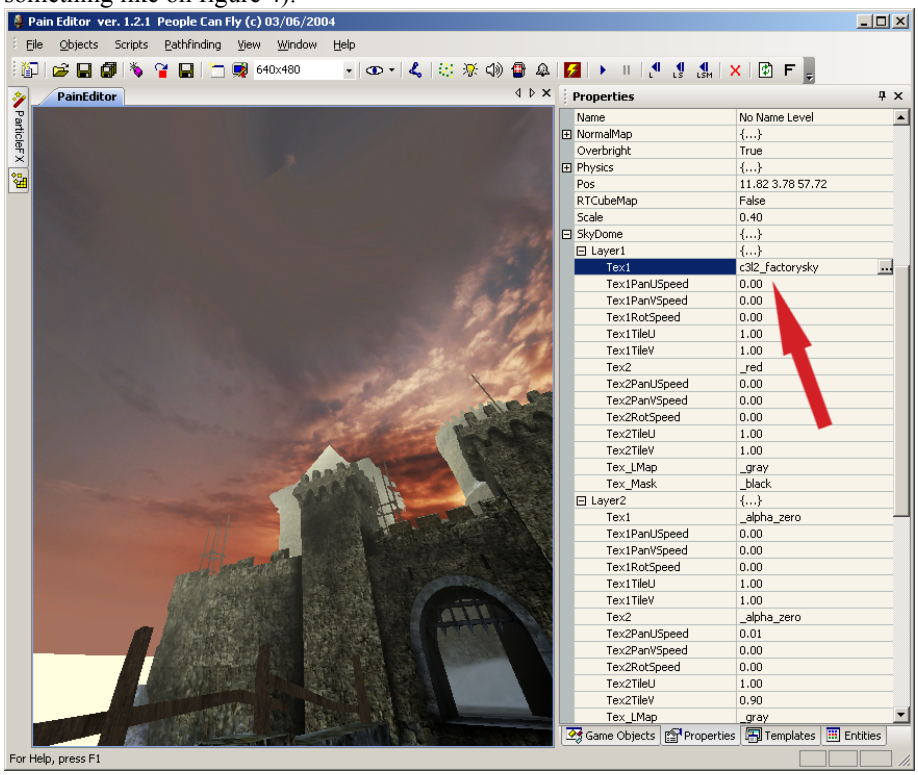

Fig. 4. Replacing the main sky texture on Layer1

Fog color

Using method described above you can load any sky texture into your level and stop thinking about sky, but at least one thing needs further consideration. As you see in left bottom corner of screen on figure 4 (as well as on top of castle towers), the new sky texture doesn't fit to the existing level's fog color. So you have to change setting in *Fog* rollout to give the fog more proper color. After changing fog color to 163,111,92 everything should look much better (Fig. 5).

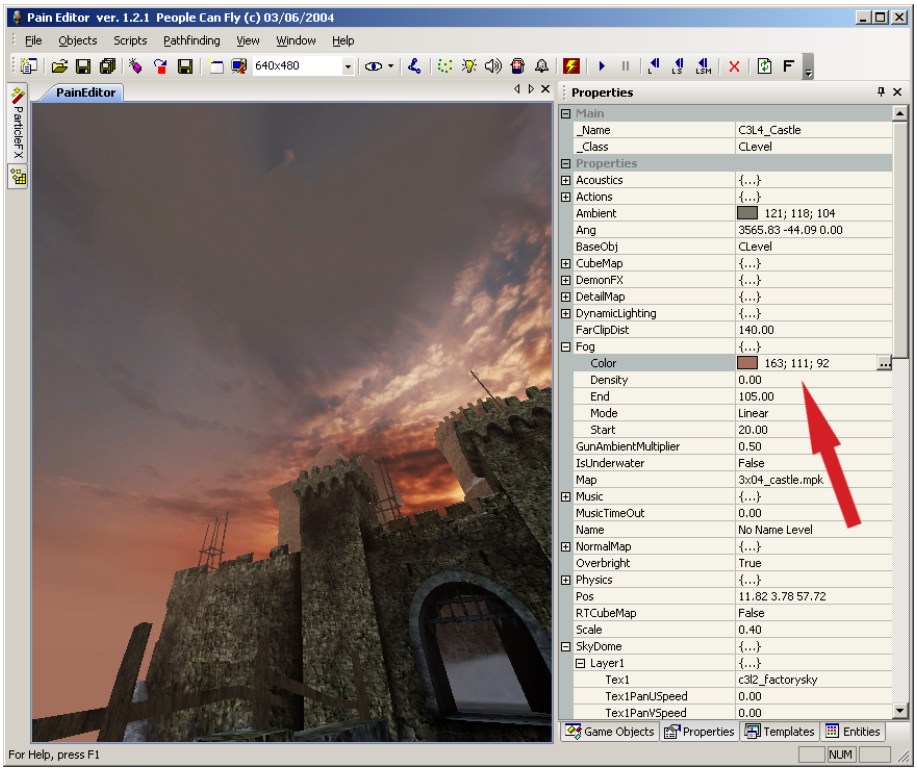

Fig. 5. Fitting the fog color to sky.

Hint: if you need to copy exact fog color from other level to new one, you can read it from .Clevel file (it's written as *o.Fog.Color* parameter, the first 3 numbers in brackets are RGB values to copy).

Additional layers and animations

As we can see on figure 4 and 5, there is a strange distortion of sky texture on top of the screen. It the result of applying cylindrical texture mapping on spherical mesh, so a second layer was added especially to fix this problem. Texture on this layer is blended to transparency and masks distortion on top of first layer. So reload level to earlier saved version and see the distortion of sky above your head. Now load the *c3l4_castleskytop* texture into *Layer2/Tex2* slot (Fig. 6).

Fig. 6. Distortion on sky texture (on the left) is masked under additional layer of clouds (on the right).

The advantage of the structure discussed here is the possibility to animate an additional layer – as you see in editor, the clouds on top of sky are slowly moving in virtual wind direction and blend seamlessly into surrounding environment (instead of cylindrical mapping of texture on *Layer1* we could use planar mapping and avoid distortions, but in that case animation shown here is not possible). You can try different animation effects by changing values of *Tex2* texture's parameters:

1) *Tex2PanUSpeed* – speed in U direction (horizontally in texture coordinates space)
2) *Tex2PanVSpeed* - speed in V direction (vertically in texture coordinates space)
3) *Tex2RotSpeed* – speed of texture rotation
4) *Tex2TileU* – number of texture's repetitions in U direction
5) *Tex2TileV* - number of texture's repetitions in V direction

Texture blending

The effect of transparency in the second layer is accomplished by using a special mask, which creates transition from base cloud texture (*Tex2 = c3l4_castleskytop*) to completely transparent texture (*Tex1 = _alpha_zero*). The mask of transparency is an additional texture, *c3l4_castleskybulbmasker*, which is loaded to *Tex_Mask* slot of *Layer2*. Alpha channel of this texture contains gradient transition from white to black, what results in blending textures from *Tex2* to *Tex1* on *Layer2* surface (for comparison: in *Layer1* we have loaded a completely black mask, which means that only one texture is visible, so it doesn't matter what texture is loaded to second slot, because it's never shown with this mask). In figure 7 the effect of blending is illustrated.

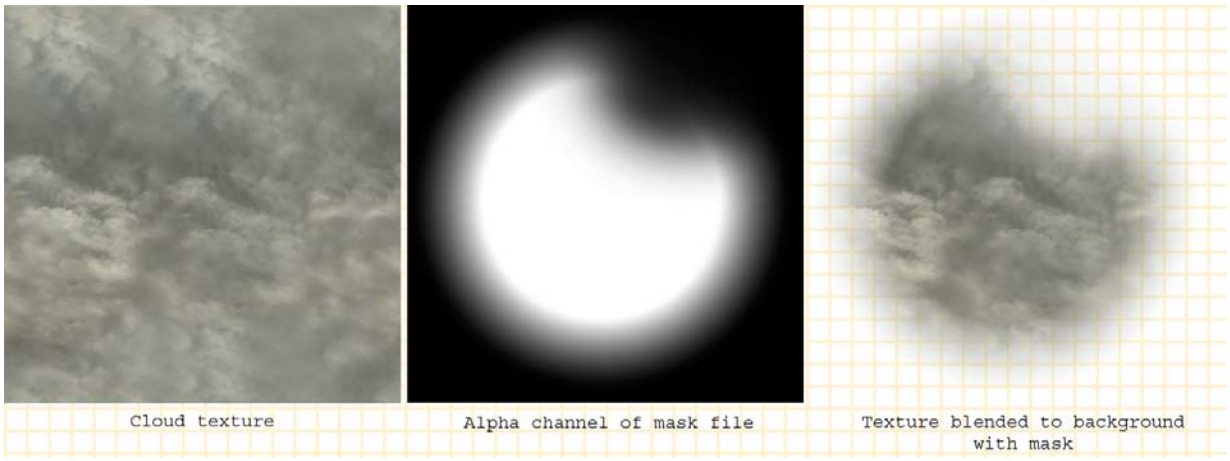

Fig. 7. Blending texture to background with mask

It's important to remember that blending textures with alpha channels is something different than blending them with RGB colors (which is also possible). In this example blending is based on the alpha channel of the mask so RGB part of the mask is completely white. If you load it into picture viewer, you'll see nothing but white screen – to see alpha channel you usually need specialized software, as Photoshop. Also note that mask doesn't move even if all textures are animated – mask is projected from second UV set of mesh, which can't be animated.

The same way as earlier we can set up a third layer, adding more subtle clouds/haze effect. Third layer surrounds the whole sky, so the range of clouds on this layer is much bigger, but finally it also depends on the mask used in *Tex_Mask* slot of *Layer3*. This mask is basically the same as in *Layer2*. As you can see on figure 7, there is a notch in the circular white spot. It's made to make sure that clouds don't reach the sun area, because if the sun was to be blocked out by clouds, all lighting situation in level should naturally change, which is not possible with pre-rendered lightmaps used here. Note, that *c3l4_castleskyclouds* texture used here has it's own alpha channel, so those clouds are much more transparent, giving a very gentle moving haze on the sky.

Lightmaps

On every sky layer you can use lightmap to control lightness of textures, and where it's not needed, neutral gray texture should be placed into *Tex_Lmap* slot. If you see *Layer1* and *Layer2* settings, you'll see such *_gray* texture in their lightmap slots. But on *Layer3* we wanted to use lightmap, because fluffy clouds look much better when they are lighted in surrounding of sun or moon. Lightmap used for this layer is shown on figure 8. Where lightmap is pure grey, lightness of texture doesn't change, and in lighter and darker areas texture varies dependently on lightmap intensity and color. Lightmap, similarly as blending mask, is projected using second UV set, so it doesn't move with animated textures.


Fig. 8. Lightmap used for animated clouds in castle level

Modifying sky model with Maya

You can create your own sky models completely from scratch in Maya, or change example model attached to this tutorial — *CastleSkyTutorial.ma*. Meshes for given sky layer can have any given shape, but you must remember some things:

1) Name of meshes should contain *layer01*, *layer02*, *layer03* or *layer04* string. Only 4 layers are allowed, so any additional or wrong named geometry will be ignored by PainEditor.
2) If given layer is dedicated for transparent textures, its name has to contain *trans* string, so full name of layer is *trans_layer02* or something similar (Fig. 9).
3) If you don't plan to use 4 layers, export only those needed — even if layer has fully transparent textures, it's still rendered as invisible geometry, so it wastes GPU power for unnecessary operations.
4) Too many polygons in meshes results in beefier hardware requirements and slower renderings, but too low-poly meshes causes distortion in textures, especially during animation.
5) Normals of surface need to be reversed to interior of mesh, because during play you are looking on sky from within the model.
6) Sky model is exported from Maya in the same way as level map – select geometry to export, choose *File >> Export Selection*. In Export window choose *Files of type >> MeshPakExport* and navigate to *Data/Maps* folder, saving model with any given name.

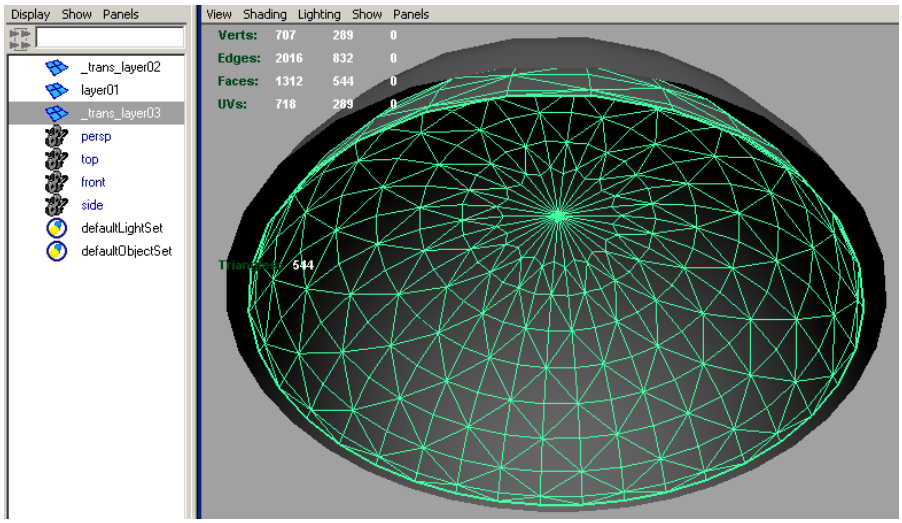

Fig. 9. Example sky model in Maya

Low quality skies

On weaker graphics card, as GeForce 2 or GeForce 4 MX, it is not possible to render all blended textures, masks and lightmaps, so in this case simpler sky model is used. It contains one layer with one texture and its parameters are set up in *LowQuality* rollout of *SkyDome* section. Aside from *Tex* and *Map* fields you have possibility to change *Angle* and *Height* of sky hemisphere over the level.