

Rendering Sprites with Distinct Textures

It is possible to render sprite particles so that each sprite has a distinct texture applied to it. The texture of a given sprite can remain constant for all frames, or it can change from frame to frame. The general idea is to use a *File 2D Texture* node to define an texture sequence, and to control which texture of that sequence will be applied to a specific sprite via the 'spriteNumPP' particle system attribute.

The general steps to set up per-sprite texturing are:

1. Prepare an texture sequence for use in a *File 2D Texture* node. It is required to provide a tdl texture file sequence. *3Delight for Maya* will accept a texture sequence that contains zero-padded frame numbers, such as 'smoke_001.tdl'. However, it is recommended to use non-padded file names, such as 'smoke_1.tdl' so Maya can properly display the textured sprites in the scene view. The first image sequence is expected to have the frame number 1.



To convert a complete image sequence to tdl textures, a MEL procedure like the following can be used:

```
proc convertTextures(string $texture_files)
{
    string $file = basename($texture_files, "");
    string $dir = substring($texture_files, 1, size($texture_files) - size($file));

    string $textures_to_process[] = 'getFileList -filespec $texture_files';

    for($curr_tex in $textures_to_process)
    {
        string $curr_file = $dir + $curr_tex;
        string $curr_tdl = DL_convertTextureName($curr_file, 1);
        print("Converted " + $curr_file + " to: " + $curr_tdl + "\n");
    }
}
```

So if the original image sequence begins with 'smoke_1.iff' in IFF format in the '~/Images/' directory, one can call the above procedure to produce a tdl version of the complete image sequence by invoking:

```
convertTextures("~/Images/smoke_*.iff");
```

The resulting tdl files will be produced in the *3Delight Textures* folder; see [3Delight Data Locations](#).

2. Create an Hypershade shading network that contains a *File 2D Texture* node. Set its *Image Name* attribute to frame 1 of the tdl texture sequence produced in the previous step. Turn on the *Use Image Sequence* attribute. Assign this shading network to the particle shape.
3. Select the particle system shape and make sure the *Particle Render Type* attribute is set to 'Sprite'.
4. The particle system needs to have a *spriteNumPP* attribute. If none exist, click the *General* button in the *Add Dynamic Attribute* group of the *Attribute Editor*, then select 'spriteNumPP' in the list displayed in the *Particle* tab and click *Add*.
5. Set the 'spriteNumPP' initial values as desired. For instance, if the image sequence contains 10 images, each sprite can be assigned a random image by defining an expression similar to:

```
particleShapel.spriteNumPP = rand(1, 10);
```

6. If the sprites must change texture from frame to frame, it is possible to define how the 'spriteNumPP' attribute will change for each frame by defining a 'Runtime before dynamics' expression for this attribute. For instance, the following expression will have a sprite use the next image in the sequence as a texture for each frame, and loop through a 10-frames image sequence:

```
particleShapel.spriteNumPP = 1 + ((particleShapel.spriteNumPP + 1) % 9);
```

7. If the 'spriteNumPP' is not guaranteed to only use values for which a texture is defined in the image sequence, add a 'SpriteCycleLength' float attribute (not per-particle) to the particle system shape and set it to the number of textures in the image sequence. The shader will automatically wrap frame values so they are constrained between 1 and the value defined in 'SpriteCycleLength'. Note that these two attributes must begin with a capital "S" as they are special Maya attributes and will be listed under the *Sprite Attributes* section of the *Attribute Editor*.
8. If desired, add a 'SpriteAnimation' bool attribute (not per-particle) to the particle system shape. It can be used to turn off the per-sprite texturing.
9. Using the *Assignment Panel* or the *3Delight Relationship Editor*, create a *Geometry Attributes* node and assign it to the particle system shape.
10. While this *Geometry Attribute* node is displayed in the *Attribute Editor*, add the *Particle System Variable* attribute, listed under 'Geometry -> Particles'.
11. In the *Particle System Variable*, select 'spriteNumPP' in the right-hand side column and click *Add*. Also add the *SpriteCycleLength* and the *SpriteAnimation* attributes if they have been defined in the previous steps.

12. It is recommended to use a particle disk cache when rendering with motion blur or if the expressions driving the above attributes can produce different results when a frame is rendered multiple times (for instance, this would be required if 'rand()' is used for one of the particle system attributes like in this example).