

## Annexe 3-Le format de fichier JXS

### Le fichier de description du Jitter Shader

Jitter fournit un moyen d'encapsuler les paramètres et les programmes de *shaders* (nuances) par le biais d'un fichier de description de *shaders* basé sur XML (JXS). Ce fichier JXS offre un moyen facile de configurer tout état OpenGL requis pour obtenir un effet particulier, sans aucun travail supplémentaire lié au patch.

Le fichier JXS peut contenir une liste de paramètres et de textures, suivie d'un ensemble d'implémentations dépendant du langage pour les programmes de sommets et de fragment du *shader*.

Chaque implémentation de langage doit spécifier à la fois un programme de sommets et un programme de fragment. Ces programmes peuvent soit faire référence à des fichiers externes, soit être stockés en ligne dans une section CDATA ou un commentaire XML. Les programmes de chaque implémentation de langage seront chargés et liés dans l'ordre où ils sont listés dans le fichier JXS. Si une implémentation de langage particulière ne parvient pas à compiler, l'implémentation suivante sera utilisée, jusqu'à ce qu'un candidat réussi soit trouvé.

À l'intérieur de la description de l'implémentation du langage, tout paramètre de *shader* précédemment spécifié doit être identifié par une balise ***bind*** qui indique quel programme de *shader* doit recevoir les valeurs du paramètre.

La liste suivante présente le format d'un fichier JXS typique :

```
<jittershader name="myshader">
  <em>Description optionnelle</em>
  <description>C'est mon shader</description>

  <!-- liste optionnelle d'objets de texture à associer -->
  <texture file="mytexture.jpg"/>

  <!-- liste optionnelle des paramètres du shader -->
  <param name="myparam" type="vec3" default="3.0 4.0 5.0">
    <description>C'est mon paramètre</description>.
  </param>

  <br /><!-- La liste des implémentations de langage -->
  <language type="glsl" version="1.0">

    <!-- liste des cibles de liaison pour les paramètres de shader -->
    <bind param="myparam" program="vp"/>

    <!-- programmes de sommet et de fragment -->
    <programme name="vp" type="vertex" source="sh.passthru.xform.vp.glsl"/>
    <programme name="fp" type="fragment">
      <![CDATA[

        // point d'entrée
        void main()
        {
```

```

    gl_FragColor=vec4(0.5, 0.5, 0.5, 1.0) ;
  }
]]>
</programme>
</language>
</jittershader>

```

Notez que le document XML doit être bien structuré avec les balises d'ouverture et de fermeture correspondantes.

## Variables d'état des *shaders* JXS

Jitter fournit plusieurs variables intégrées uniformes qui présentent l'état OpenGL aux programmes de *shaders* JXS. On peut y accéder en spécifiant le nom de la variable dans l'attribut *state* de la balise XML *param*.

Par exemple :

```
<param name="itvmat" type="mat3" state="NORMAL_MATRIX " />
```

Les liens d'état de paramètres de *shaders* disponibles sont répertoriées ci-dessous.

### Modèles de vue et de projection des matrices

*WORLD\_MATRIX* (mat4)  
*VIEW\_MATRIX* (mat4)  
*MODELVIEW\_MATRIX* (mat4)  
*PROJECTION\_MATRIX* (mat4)  
*VIEW\_PROJECTION\_MATRIX* (mat4)  
*MODELVIEW\_PROJECTION\_MATRIX* (mat4)  
*PREV\_MODELVIEW\_PROJECTION\_MATRIX* (mat4)  
*NORMAL\_MATRIX* (mat3)

### Caméra

*CAMERA\_POSITION* (vec3)  
*CAMERA\_DIRECTION* (vec3)  
*VIEWPORT* (vec2)  
*INVERSE\_VIEWPORT* (vec2)  
*NEAR\_CLIP* (float)  
*FAR\_CLIP* (float)  
*FAR\_CORNER* (vec3)

### Lumière

*LIGHT\_VIEWPROJ\_MATRIX0-7* (mat4)  
*LIGHT\_RANGE0-7* (float)  
*LIGHT0-7\_POSITION* (vec3)  
*LIGHT0-7\_DIRECTION* (vec3)  
*LIGHT0-7\_AMBIENT* (vec4)  
*LIGHT0-7\_DIFFUSE* (vec4)  
*LIGHT0-7\_SPECULAR* (vec4)  
*LIGHT0-7\_CUTOFF* (float)  
*LIGHT0-7\_EXPONENT* (float)

## **Matériau**

*AMBIANT (vec4)*  
*DIFFUSE (vec4)*  
*SPECULAR (vec4)*  
*EMISSION (vec4)*

## **Texture**

*TEXTURE0-7\_MATRIX (mat4)*  
*TEXDIM0-7 (vec2)*

## **Transformations de matrices**

Les variables matricielles (variables de type mat3 et mat4) peuvent être transformées de manière spécifique en utilisant l'attribut **transform** de la balise XML *param* :

```
<param name="itvmat" type="mat4"  
state="VIEW_MATRIX"  
transform="INVERSE_TRANSPOSE " />
```

Les transformations matricielles disponibles sont les suivantes :

*IDENTITY*  
*TRANSPOSE*  
*INVERSE*  
*INVERSE\_TRANSPOSE*

## **Attributs de sommets**

Les programmes de *shaders* Jitter JXS peuvent accéder à des attributs de sommets personnalisés via la balise d'état *VERTEX\_ATTR* :

```
<param name="pvel" type="vec4" state="VERTEX_ATTR " />
```

et dans le programme des sommets :

```
attribute vec4 pvel ;
```

Pour définir les valeurs des attributs de sommets personnalisés à partir du patch, envoyez à *jit.gl.mesh* le message *vertex\_attr\_matrix* suivi du nom d'une matrice *jit.matrix* contenant les valeurs des attributs. L'exemple de patch *custom.vertex.attribute* en fait la démonstration.

En outre, il existe plusieurs attributs de sommets intégrés disponibles via les balises d'état suivantes :

*POSITION (vec3)*  
*NORMAL (vec3)*  
*TANGENT (vec3)*  
*BITANGENT (vec3)*