

DOCUMENTATION

Sommaire :

Introduction :	1
Création d'un fichier de description de graphe de scène :	1
Création d'un noeud de géométrie :	1
Création d'un noeud de groupe :	3
Création d'un noeud de transformation :	4
Création d'un noeud de transformation (animation) :	4
Exemple de structure :	5

Introduction :

Notre grammaire est un langage de description de graphe de scène s'inspirant d'un langage structurant type XML. Chaque noeuds est ainsi représentés par l'ouverture d'une balise contenant son type et par la fermeture de cette même balise. Les balises indiquant l'ouverture et la fermeture d'un noeud peuvent contenir 3 types différents :

- une transformation
- un groupe
- une géométrie

Ainsi afin d'insérer un fils à ce noeud il faut créer la nouvelle balise dans la première avant la fermeture de celle-ci. Chaque balises détiennent plus ou moins d'attributs à définir lors de leurs initialisations tous séparés par des points virgules.

Un noeud de transformation ne peut être utilisé que pour représenter un noeud de l'arbre de description du graphe de scène.

Un noeud de groupe peut être utilisé en tant que noeud de l'arbre mais aussi en tant que feuille dans le cas où le noeud utilisé pointe sur un noeud qui à déjà été initialisé dans la description.

Un noeud de géométrie ne peut être utilisé que pour représenter une feuille de l'arbre de description du graphe de scène.

Exemple de structure :

```
<Transformation; ... ;>  
    <Geometry; ... ;>  
    </Geometry>  
</Transformation>
```

Création d'un fichier de description de graphe de scène :

Afin de créer votre fichier de de description d'un graphe de scène, il est tout d'abord nécessaire d'ouvrir une balise racine. Cette balise ne contient aucun attributs et permet de signaler le début de la description. L'ouverture de cette balise est représentée par "<Root>", de la même manière, sa fermeture est représentée par "</Root>".

Racine de la description :

```
<Root>  
    ...  
</Root>
```

Création d'un noeud de géométrie :

Il existe 2 manières de représenter une géométrie dans la description, nous pouvons tout d'abord utiliser la même forme que pour tous les noeuds en ouvrant une balise `<Geometry; ... ;>` puis en la fermant avec `</Geometry>`. Mais sachant qu'une géométrie ne peut être qu'une feuille de l'arbre alors nous avons décidé d'implémenter une autre forme au noeud de géométrie : `<Geometry; ... ;/>`. Cette balise représente à la fois une balise d'ouverture et de fermeture d'une géométrie.

`<Geometry; ... ;>` `</Geometry>` équivalent à `<Geometry; ... ;/>`

Comme énoncé rapidement dans l'introduction, chaque noeud détient un ou plusieurs paramètres. Dans le cas du noeud de géométrie, il faut obligatoirement renseigner le type de géométrie qui peut être : "Cone", "Sphere", "Cylinder", "Cube", "Teapot", "Tetrahedron" ou "Torus". Chaque type de géométrie détient un ou plusieurs attributs qui doivent être renseignés à la suite du type entre accolade et séparés par une virgule.

Exemple des attributs nécessaires à la création de chaque type de géométrie :

```
<Geometry; type = Cone{base, height, slices, stacks}; ... ;/>
<Geometry; type = Sphere{radius, slices, stacks}; ... ;/>
<Geometry; type = Cylinder{base, top, height, slices, stacks}; ... ;/>
<Geometry; type = Cube{size}; ... ;/>
<Geometry; type = Teapot{size}; ... ;/>
<Geometry; type = Tetrahedron{}; ... ;/>
<Geometry; type = Torus{radius,oradius, sides, rings}; ... ;/>
```

Un autre paramètre obligatoire est le nom de la géométrie. Ce paramètre permettra de rendre unique la géométrie créée et de la réutilisée à plusieurs reprises.

Exemple d'initialisation d'un nom de géométrie :

```
<Geometry; type = Cube{15}; nom = "cube1"; ... ;/>
```

Enfin le dernier paramètre nécessaire à la création d'une géométrie est l'association d'un matériau. Pour cela, après avoir indiqué le nom de la géométrie nous créons un paramètre "Material :" et indiquons à la suite tous les attributs pour l'initialisation d'un nouveau matériau.

Exemple d'initialisation d'une géométrie :

```
<Geometry; type = Cube{15}; nom = "cube1"; Material : nom = "mat1", Ka{x,y,z}, Kd{x,y,z},
Ks{x,y,z}, d = x, Ns = x, textureScale{x,y}, textureFile = "...";>
```

Après l'initialisation d'un matériau il est possible de récupérer ce même matériau sur une autre géométrie sans avoir à réinitialiser toutes ses données en indiquant seulement le nom du matériau. L'initialisation d'un matériau de même nom qu'un matériau déjà existant remplacera l'ancien par le nouveau.

Exemple de réutilisation d'un matériau :

```
<Geometry; type = Cube{15}; nom = "cube1"; Material : nom = "mat1", Ka{x,y,z}, Kd{x,y,z},
Ks{x,y,z}, d = x, Ns = x, textureScale{x,y}, textureFile = "...";>
...
<Geometry; type = Teapot{15}; nom = "teapot1"; Material : nom = "mat1";>
```

De la même manière, chaque géométrie détient un nom unique, ce qui permet d'appeler une même géométrie plusieurs fois, toutefois si on initialise une géométrie avec le même nom qu'une déjà existante alors l'ancienne se voit remplacée par la nouvelle.

Exemple de réutilisation d'une géométrie :

```
<Geometry; type = Cube{15}; nom = "cube1"; Material : nom = "mat1", Ka{x,y,z}, Kd{x,y,z},
Ks{x,y,z}, d = x, Ns = x, textureScale{x,y}, textureFile = "...";>
...
<Geometry; nom = "cube1";>
```

Création d'un noeud de groupe :

Un noeud de groupe permet de regrouper n'importe quels autres noeuds. Afin d'initialiser un groupe 1 seul paramètre est obligatoire qui n'est autre que l'attribution d'un nom permettant le stockage et la réutilisation du groupe. Pour cela la construction de la balise reste la même que les autres.

Exemple d'initialisation d'un groupe :

```
<Group; nom = "grp1";>
...
</Group>
```

Un groupe peut être utilisé en tant que noeud de l'arbre de description. Dans ce cas il est composé de un ou plusieurs noeud. Ayant effectué un regroupement il est possible de l'utiliser en tant que feuille de l'arbre en appelant ce groupe déjà défini :

Exemple de réutilisation d'un groupe :

```
<Group; nom = "grp1";>
  <Geometry; type = Cube{20}; nom = "cube1"; Material : nom = "mat1";
  Ka{0.5,0.5,0}, Kd{0,0.2,0}, Ks{0.5,0.3,0}, d = 1, Ns=1, textureScale{1,1}, textureFile = "...";>
  <Geometry; type = Teapot{15}; nom = "teapot1"; Material : nom = "mat1";>
</Group>
...
<Group; nom = "grp1";> </Group>
```

Création d'un noeud de transformation :

Un noeud de transformation permet d'appliquer un type de transformation aux noeuds fils, c'est pourquoi ce type de noeud ne peuvent être des feuilles de l'arbre de description de la scène de graphe. Afin de créer une transformation, il faut obligatoirement renseigner le type de transformation qui peut être : "Translation", "Rotation" et "Scale". Chaque type de transformation détient un ou plusieurs attributs qui doivent être renseignés à la suite du type entre accolade et séparés par une virgule.

Exemple des attributs nécessaires à la création de chaque type de géométrie :

```
<Transform; type = Translation{x, y, z}; ... ;> ... </Transform>  
<Transform; type = Rotation{x, y, z, angle}; ... ;> ... </Transform>  
<Transform; type = Scale{x, y, z}; ... ;> ... </Transform>
```

Le deuxième et dernier paramètre obligatoire est le nom de la transformation. Ce paramètre permettra de stocker et de rendre unique la transformation créée.

Exemple d'initialisation d'un nom de géométrie :

```
<Transform; type = Translation{x, y, z}; nom = "translation1";> ... </Transform>
```

Création d'un noeud de transformation (animation) :

Chaque type de transformation détient un ou plusieurs attributs qui doivent être renseignés à la suite du type entre accolade et séparés par une virgule. Ces attributs seront mis sous la forme de String contenant une expression mathématiques basique (+, -, *, /) en utilisant que la variable t.

Exemple des attributs nécessaires à la création de chaque type de géométrie :

```
<Transform; type = Translation{"2*t+5/-2", "4", "t"}; ... ;> ... </Transform>  
<Transform; type = Rotation{"2*t+5/-2", "4", "t", "95"}; ... ;> ... </Transform>  
<Transform; type = Scale{"2*t+5/-2", "4", "t"}; ... ;> ... </Transform>
```

Le deuxième et dernier paramètre obligatoire est le nom de la transformation. Ce paramètre permettra de stocker et de rendre unique la transformation créée.

Exemple d'initialisation d'un nom de géométrie :

```
<Transform; type = Translation{x, y, z}; nom = "translation1";> ... </Transform>
```

Exemple de structure :

```
<Root>
  <Transform; type = Translation{100,100,115}; nom = "Benbou";>
    <Group; nom = "grp1";>
      <Transform; type = Rotation{1,1,0,90}; nom = "Rotation";>
        <Geometry; type = Cone{22.5,50,10,5}; nom = "cone1";
Material : nom = "mat1"; Ka{0.5,0.5,0}, Kd{0.8,0.2,0}, Ks{0.5,0.3,0}, d = 1, Ns=1,
textureScale{1,1}, textureFile = "feuille.png";/>

        </Transform>
        <Geometry; type = Cube{20}; nom = "cube1"; Material : nom =
"mat1";/>
        <Geometry; type = Teapot{10}; nom = "teapot1"; Material : nom =
"mat1";/>
      </Group>
    </Transform>
    <Transform; type = Translation{45,45,45}; nom = "Translation";>
      <Group; nom = "grp1"> </Group>
      <Transform; type = Scale{5,2,5}; nom = "Scale";>
        <Geometry; nom = "cube1";/>
      </Transform>
    </Transform>
  </Root>
```