

## Annexe 2-Le format de matrice OpenGL

### Matrices, vidéo et OpenGL

OpenGL est une norme qui définit une machine virtuelle permettant de transformer des listes de nombres en images 2D. Les listes peuvent contenir les emplacements 3D des sommets à connecter avec des lignes ou des polygones. Elles peuvent également contenir d'autres données telles que des coordonnées pour le mappage de textures, des normales pour l'éclairage, des couleurs et des indicateurs de bord.

Jitter traite la vidéo comme un type de données qui peuvent être traitées dans son système matriciel général. Lors de la manipulation de données vidéo dans Jitter, chaque cellule d'une matrice représente un pixel. Les quatre plans de la matrice stockent les composantes A, R, G et B de ce pixel. C'est la convention de Jitter pour stocker la vidéo.

Jitter a également une convention, utilisée par tous les objets du groupe GL, pour stocker les descriptions d'images compatibles OpenGL dans une matrice. Comme les données vidéo, ces descriptions peuvent être lues, traitées et produites par divers objets. Ce document spécifie la convention de Jitter pour le stockage des données OpenGL dans des matrices et la syntaxe des messages destinés à l'objet *jit.gl.render* contenant ces matrices.

#### *Quand vous avez besoin de cette référence*

Les objets du groupe GL envoient des matrices au format décrit ici lorsque leurs attributs *matrixoutput* sont à 1. Pour traiter ces géométries par le biais de vos propres patches, vous devez donc connaître ce format. Si vous voulez créer des géométries directement en utilisant l'objet *jit.op* et d'autres opérateurs matriciels, vous devrez également connaître ce format. Mais si les objets GL de Jitter sont suffisamment flexibles pour vos besoins de dessin, vous pouvez continuer à laisser ces objets dessiner en arrière-plan sans aucune connaissance du format de matrice OpenGL.

### Référence à la matrice GL

#### Format des messages

Les données OpenGL peuvent être transmises à l'objet *jit.gl.render* dans des messages Max de forme suivante

*jit\_matrix* [matrice géométrique].

*jit\_matrix* [matrice géométrique] [primitive de dessin].

et

*jit\_matrix* [matrice géométrique] [matrice de connexions] [primitive de dessin].

#### Primitive de dessin

La primitive de dessin spécifie comment dessiner les connexions entre les sommets dans la matrice de géométrie. Lorsqu'une primitive de dessin n'est pas transmise à l'objet *jit.gl.render* avec la matrice de géométrie, la primitive actuelle de l'objet *jit.gl.render* est utilisée. La primitive courante peut être modifiée en envoyant un message contenant uniquement la primitive. Lorsqu'une primitive de dessin accompagne la matrice de géométrie, elle est utilisée uniquement pour dessiner la géométrie donnée et la primitive courante reste inchangée.

La primitive de dessin peut être l'un des symboles suivants :

- *points* : dessine chaque sommet comme un point unique.
- *lines* : relie chaque autre paire de sommets par une ligne. Étant donné une matrice de géométrie avec des sommets A, B, C et D, elle dessine les segments de ligne AB et CD.
- *line\_strip* : connecte chaque paire de sommets avec une ligne. Étant donné une matrice de géométrie avec les sommets A, B, C et D, elle dessine les segments de ligne AB, BC et CD.
- *line\_loop* : comme *line\_strip* mais un segment de ligne est dessiné pour relier le dernier sommet de la matrice au premier.
- *triangles* : dessine des triangles non reliés entre eux. Étant donné les sommets A B C D E F, dessine les triangles ABC et DEF.
- *tri\_strip* : dessine une bande de triangles connectés. Étant donné les sommets A B C D E F, dessine les triangles ABC, CBD, CDE et EDF. Notez l'ordre des sommets, qui est interverti pour que tous les triangles soient orientés dans le même sens (voir Tutoriel 33 : Modes de polygone, couleurs et fusion).
- *tri\_fan* : dessine un éventail de triangles. Étant donné les sommets A B C D E F, dessine les triangles ABC, ACD, ADE et AEF.
- *quads* : dessine des quadrilatères non reliés entre eux. Avec les sommets A B C D E F G H, dessine les quadrilatères ABCD et EFGH.
- *quad\_strip* : dessine des quadrilatères connectés. Étant donné les sommets A B C D E F, dessine les quadrilatères ABCD et CDFE.
- *polygon* : dessine un seul polygone en utilisant tous les sommets de la matrice. Si le polygone n'est pas simple et convexe, les résultats sont indéfinis.
- *tri\_grid* : si la matrice de géométrie a deux dimensions, des triangles sont dessinés qui relient chaque sommet à ses voisins dans la matrice pour former une surface.
- *quad\_grid* : si la matrice géométrique a deux dimensions, on dessine des quadrilatères qui relient chaque sommet à ses voisins dans la matrice pour former une surface.

## La matrice de connexions

Une matrice de connexions doit être unidimensionnelle et contenir des données soit **long** soit **char**. Si elle existe, elle spécifie l'ordre dans lequel il faut connecter les sommets de la matrice de géométrie. En laissant la matrice de géométrie constante et en modifiant la matrice de connexions qui y est indexée, il est possible de dessiner un ensemble changeant de connexions entre les mêmes sommets.

## La matrice de géométrie

Les matrices de géométrie doivent contenir des données au format **long**, **float32** ou **float64**, **float32** étant le choix le plus courant. **Float64** offre plus de précision que ce qui est normalement nécessaire pour le rendu à l'écran. Le type de données **long** limite les valeurs aux nombres entiers, ce qui n'est généralement pas souhaité pour spécifier des coordonnées.

Chaque cellule de la matrice représente un sommet. L'image est rendue en dessinant des connexions entre les sommets, soit le long des lignes, soit le long des colonnes de la matrice. Une matrice d'une cellule en largeur sera connectée le long des colonnes (sa seule colonne). Une matrice d'une cellule en hauteur sera connectée le long des lignes. Si la matrice est supérieure à une cellule en largeur et en hauteur, l'attribut *geom\_rows* de l'objet *jit.gl.render* détermine si les lignes ou les colonnes seront respectées. Si les primitives *tri\_grid* ou *quad\_grid* sont spécifiées, les lignes et les colonnes de la matrice sont toujours connectées.

Il est possible d'utiliser des matrices comportant entre 2 et 13 plans. Les plans 13 et plus sont réservés pour une utilisation ultérieure et sont actuellement ignorés. Chaque plan représente une valeur différente dans l'un des cinq groupes suivants : sommets, coordonnées de texture, normales, couleur de sommet et bords. Si suffisamment de plans sont présents pour définir un groupe donné, ce groupe est utilisé pour le rendu de la matrice. Pour être utilisé, chaque groupe doit être totalement présent, à l'exception des sommets. Si seules les valeurs x et y des sommets sont présentes, la coordonnée z de tous les sommets est fixée à 0, et il en résulte une image 2D sur le plan xy.

plan 0 : coordonnée x du sommet

plan 1 : coordonnée y du sommet

plan 2 : coordonnée z du sommet

plan 3 : coordonnée s de la texture du sommet, si elle est texturée

plan 4 : coordonnée t de la texture au sommet, si elle est texturée

plan 5 : composante x de la normale de la lumière au sommet

plan 6 : composante y de la normale de la lumière au sommet

plan 7 : composante z de la normale de la lumière au sommet

plan 8 : composante rouge de la couleur du sommet

plan 9 : composante verte de la couleur du sommet

plan 10 : composante bleue de la couleur du sommet

plan 11 : composante alpha de la couleur du sommet

plan 12 : indicateur de bord pour la connexion au sommet suivant : si = 0, aucune ligne n'est dessinée.