37-Géométrie sous le chapeau

> *Remarque*: certaines techniques décrites dans ce didacticiel sont dépassées. Il est recommandé aux utilisateurs de se servir de l'objet *jit.gl.mesh* au lieu de *jit.gl.render* pour rendre les matrices de géométrie.

Ce didacticiel présente le support de bas niveau de Jitter pour spécifier les données géométriques comme des matrices et les rendre avec l'objet *jit.gl.render*. Puisque les données sont contenues dans une matrice Jitter ordinaire, cela ouvre un monde de possibilités où des opérateurs matriciels arbitraires peuvent être utilisés pour générer et/ou traiter les matrices géométriques. Le didacticiel couvrira l'attribut ***matrixoutput*** du groupe GL, l'organisation des données dans des matrices géométriques, un exemple de la façon dont les matrices géométriques peuvent être traitées à l'aide d'opérateurs matriciels, et introduira diverses primitives de dessin supportées par l'objet *jit.gl.render*.

• Ouvrez le patch du didacticiel et cliquez sur l'objet *toggle* intitulé ***Start Rendering***.

Sortie matricielle

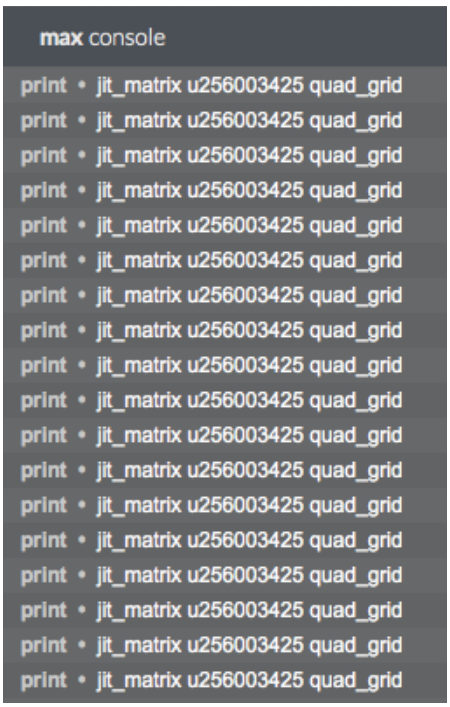
• Cliquez sur le *toggle* intitulé *Turn matrixoutput on/off.*.

Attendez une minute - la sphère qui était juste là il y a un instant a disparu.
Que se passe-t-il?

Certains des objets du groupe GL prennent en charge l'attribut ***matrixoutput*** et *jit.gl.gridshape* en fait partie. Cet attribut détermine si la géométrie de l'objet est rendue directement dans le contexte de dessin associé à l'objet ou si l'objet envoie une matrice contenant des données géométriques à sa sortie gauche. La géométrie n'est pas visible car elle est envoyée à une gate qui est fermée.

• Sélectionnez l'élément de menu ***print*** dans l'objet *umenu* intitulé «Matrix Destination»

Dans la fenêtre Max, vous devriez voir une série de messages comme "print: jit_matrix u26300000007 quad_grid". Ceci est similaire à ce que vous avez vu dans les didacticiels précédents comme sortie d'objets qui transmettent des données matricielles, à l'exception qu'il y a un élément supplémentaire. Plutôt que d'envoyer le message ***jit_matrix [matrix-name]***, comme vous devriez le savoir, l'objet *jit.gl.gridshape* envoie le message ***jit_matrix [matrix-name]*** [***drawing-primitive***]. Dans ce cas, la primitive de dessin est ***quad_grid***, ce qui signifie interpréter la matrice comme une grille de quadrilatères.



La sortie de l'objet *jit.gl.gridshape* dans la console Max

> Au moment où ce didacticiel a été écrit, les seuls objets qui supportent l'attribut ***matrixoutput*** sont les objets *jit.gl.gridshape*, *jit.gl.nurbs* et *jit.gl.plato*. Cependant, au moment où vous lirez ces lignes, il se peut que d'autres objets supportent ce mode de fonctionnement.

Il n'y a toujours pas d'excitation dans notre fenêtre, mais cela est facilement résolu.

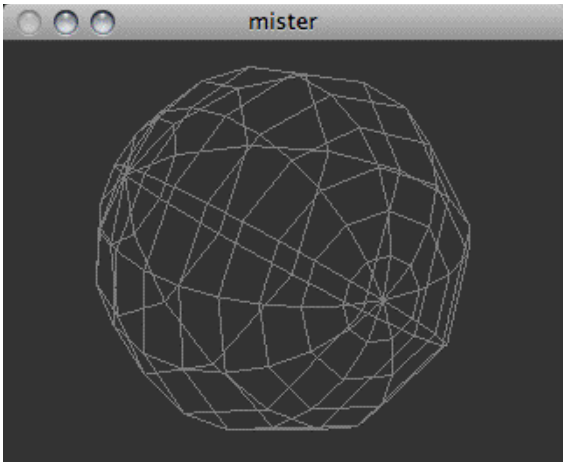
• Sélectionnez l'élément de menu ***direct*** dans l'objet *umenu* intitulé *Matrix Destination*.

La sphère est à nouveau visible, car le message ***jit_matrix [matrix-name]*** [***drawing-primitive***] est envoyé à l'objet *jit.gl.render*. En réponse à ce message, l'objet *jit.gl.render* dessine la matrice fournie en utilisant la primitive de dessin spécifiée. Si aucune primitive de dessin n'est spécifiée, la primitive de dessin actuelle de l'objet *jit.gl.render* sera utilisée. Les primitives de dessin valides sont: ***points***, ***lignes***, ***ligne_strip***, **l*igne_loop***, ***triangles***, ***tri_strip***, ***tri_fan***, ***quads***, ***quad_strip***, ***polygone***, ***tri_grid*** et ***quad_grid***.

> La primitive de dessin actuelle de l'objet *jit.gl.render* peut être définie en envoyant le message [***drawing-primitive***] où [***drawing-primitive***] est l'une des primitives de dessin valides mentionnées ci-dessus.

De la même manière que vous pouvez modifier les dimensions des données vidéo, vous pouvez modifier les dimensions de la matrice de sortie en envoyant à l'objet *jit.gl.gridshape* le message ***dim***. Par défaut, les dimensions de la matrice produite par l'objet *jit.gl.gridshape* sont de 20 x 20.

• Cliquez sur le *toggle* intitulée ***Wireframe***.
• Modifiez la boîte de nombre intitulée ***Matrix Dimensions***.
• Essayez de faire tourner la sphère filaire avec la souris



"Les plans Death Star ne sont pas dans l'ordinateur principal."

Vous avez peut-être remarqué que dans ce patch, l'objet *jit.gl.handle* communique avec l'objet *jit.gl.render* plutôt qu'avec l'objet *jit.gl.gridshape*. Cela a pour effet de faire pivoter et de positionner la scène entière. L'argument ***@inherit_transform 1*** est nécessaire afin de faire cela correctement, sinon la rotation serait à nouveau composée avec la rotation de la scène, provoquant une confusion majeure.

Détails de la matrice géométrique

La vidéo dans Jitter est typiquement représentée par des données **char** à 4 plans, mais comment les données géométriques sont-elles représentées?

Chaque sommet de la géométrie est généralement représenté sous la forme de données ***float32*** avec 3, 5, 8, 12 ou 13 plans. Les plans 0 à 2 spécifient la position x, y et z du sommet. Les plans 3 et 4 indiquent les coordonnées de texture s et t. Les plans 5 à 7 indiquent le vecteur normal nx, ny et nz utilisé pour calculer les effets de l'éclairage sur la géométrie. Les plans 8 à 11 spécifient la couleur *rouge*, *vert*, *bleu* et *alpha des sommets*. Le plan 12 spécifie l'indicateur de bord e.

La matrice de sortie de l'objet *jit.gl.gridshape* a 12 plans, mais puisque nous n'appliquons pas de texture à la géométrie, et que l'éclairage n'est pas activé, les coordonnées de texture et les vecteurs normaux sont ignorés.

Traitement de la matrice géométrique

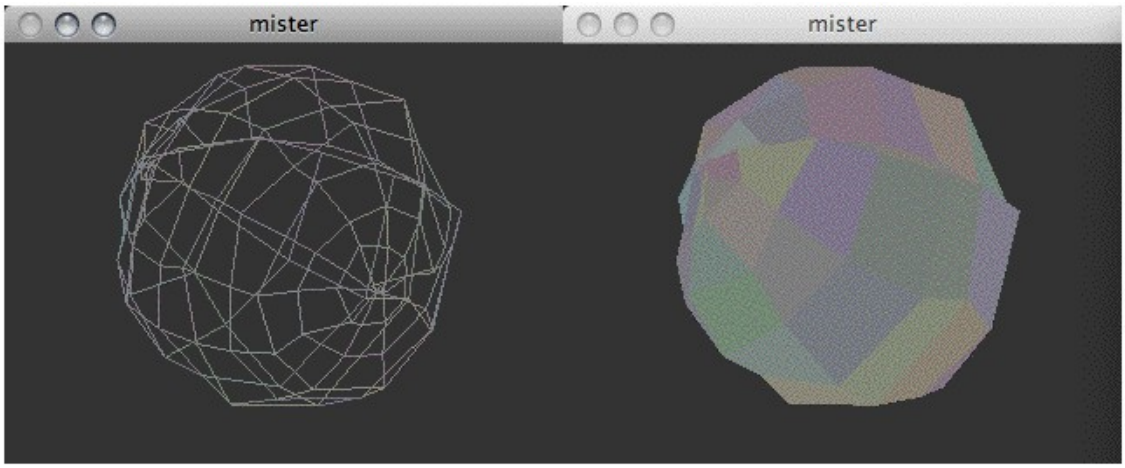
Au lieu de simplement rendre la géométrie inchangée, comme vous l'avez fait jusqu'à présent, il est possible de traiter cette géométrie avec des opérateurs matriciels.

• Sélectionnez l'élément de menu ***xfade*** dans l'objet *umenu* intitulé *Matrix Destination*.

Maintenant, la matrice est envoyée à travers l'objet *jit.xfade* pour effectuer un fondu enchaîné de la matrice géométrique avec une matrice de bruit. Tout comme l'objet *jit.xfade* peut être utilisé pour un fondu enchaîné entre des matrices vidéo, il peut être utilisé pour effectuer un fondu enchaîné entre des matrices géométriques.

• Cliquez sur l'objet *button* étiqueté "Generate Noise"
• Modifiez progressivement la boîte de *nombre* intitulée "Crossfade Value" à ***0,1***.
• Activez et désactivez le rendu filaire en cliquant sur le toggle intitulé *Wireframe*. Remarquez comment le bruit déforme à la fois la géométrie et la couleur de la forme.

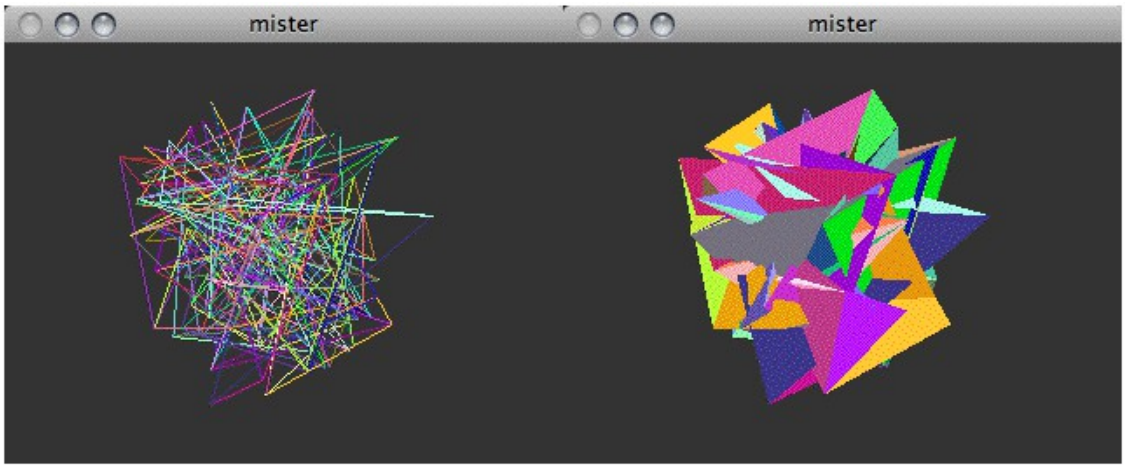
Cela déforme subtilement la sphère avec le bruit, ainsi que l'introduction de couleurs aléatoires. Les coordonnées de texture et les vecteurs normaux sont également affectés, mais cela n'est pas visible car aucune texture n'est appliquée et l'éclairage n'est pas activé.



Une sphère légèrement déformée (filaire et remplie)

• Cliquez plusieurs fois sur l'objet *button* intitulé «Generate Noise» ou activez le *toggle* intitulé «Rapid Noise» pour générer une nouvelle matrice de bruit à chaque fois que la géométrie est dessinée.
• Faites passer progressivement la boîte de *nombre* intitulée «Crossfade Value» à ***1.0***. Encore une fois, activez et désactivez le rendu filaire pour voir comment le bruit est visualisé.

Maintenant, la géométrie dessinée est un pur bruit.



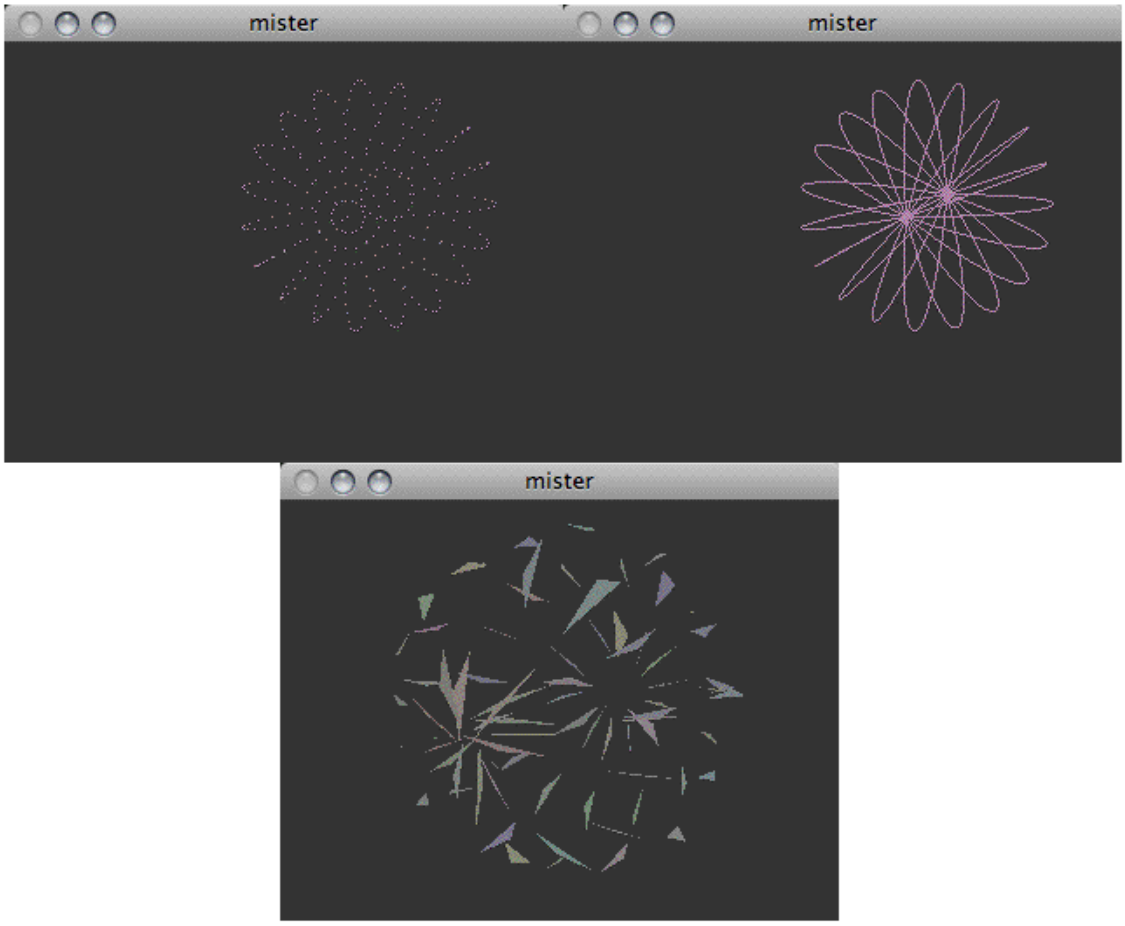
Bruit blanc exprimé en géométrie (filaire et rempli)

• Remettez la boîte de *nombre* intitulée "Crossfade Value" sur ***0***.

Dessin des primitives

Vous remarquerez que malgré le fait que *jit.gl.gridshape* produise le message ***jit_matrix [matrix-name] quad_grid***, nous ajoutons ***quad_grid*** à la sortie de *jit.xfade*. En effet, la plupart des opérateurs matriciels ignorent l'argument de la primitive de dessin et produisent simplement le message ***jit_matrix [matrix-name]***. Par conséquent, nous devons ajouter le nom de la primitive de dessin au message. Cela offre une bonne occasion d'expérimenter diverses primitives de dessin.

• Essayez de sélectionner les différents éléments de menu disponibles dans *l'umenu* intitulé *Primitive de dessin*.



Utilisation de différentes primitives de dessin: ***points*** *(gauche),* ***line_strip*** *(centre) et* ***triangles*** *(droite)*

Résumé

En plus de dessiner directement dans leurs contextes de dessin associés, certains objets du groupe GL supportent l'attribut ***matrixoutput***. Lorsqu'il est activé, la matrice géométrique est envoyée à la sortie gauche de l'objet avec le message ***jit_matrix [matrix-name]*** [***drawing-primitive***].

Les matrices géométriques sont généralement des données ***float32*** avec un nombre de plans de 3, 5, 8, 12 ou 13 plans et peuvent être traitées à l'aide d'opérateurs matriciels arbitraires d'une manière similaire au traitement des matrices vidéo.

L'objet *jit.gl.render* prend en charge diverses primitives de dessin pour rendre ces matrices géométriques: ***points***, ***lignes***, ***ligne_strip***, ***ligne_loop***, ***triangles***, ***tri_strip***, ***tri_fan***, ***quads***, ***quad_strip***, ***polygone***, ***tri_grid*** et ***quad_grid***.