

36-Modèles 3D

Dans ce didacticiel, nous allons montrer comment charger des modèles 3D à l'aide de l'objet *jit.gl.model* et les rendre dans Jitter avec l'éclairage, les matériaux et la cartographie de texture.

Révision et configuration

- Ouvrez le patch du didacticiel. Cliquez sur le toggle intitulé «**Start Rendering**». Vous verrez un fond brun dans l'objet *jit.pwindow*. La couleur brune provient de l'attribut **erase_color** de l'objet *jit.gl.render*.

- Cliquez sur la boîte de *message* indiquant «**name ml, depthbuffer 1**» au-dessus de l'objet *jit.pwindow*.

L'attribut **name** permet à l'objet *jit.pwindow* d'être utilisé comme destination de dessin, bien que cela ait déjà été fait et enregistré avec le patch du didacticiel dans ce cas. Le message **depthbuffer 1** fait en sorte qu'un tampon de profondeur soit attaché à la destination, de sorte que la suppression automatique des surfaces cachées peut être effectuée.

- Cliquez sur la boîte de *message* indiquant «**texture grid 64 64**» au-dessus de l'objet *jit.pwindow*. Comme nous l'avons vu dans le tutoriel d'introduction aux textures, ce message envoyé à l'objet *jit.gl.render* entraîne la création d'une texture de 64 pixels en largeur et en hauteur, remplie d'un motif en damier par défaut. La texture peut être utilisée par tous les objets du groupe GL qui dessinent dans son contexte. Nous sommes maintenant prêts à charger quelques modèles et à les rendre.

Lecture d'un fichier modèle

L'objet *jit.gl.model* lit les descriptions d'objets 3D sous forme de collections de polygones. Ces descriptions sont stockées dans un format de fichier modèle. Parmi les formats de fichiers modèles courants pris en charge par *jit.gl.model* figurent *.obj* (Wavefront OBJ), *.dae* (Collada) et *.fbx* (Autodesk FBX). Ce didacticiel utilisera des fichiers *.obj*. Ce format est largement utilisé et présente l'avantage d'être lisible et modifiable dans les éditeurs de texte ordinaires.

- Cliquez sur l'objet *toggle* au-dessus de la boîte de *message* «**verbose \$ 1**» dans le coin supérieur gauche de la section «Draw a Model» du patch.

Cela envoie le message **verbose 1** à l'objet *jit.gl.model*, activant le mode verbose. Lorsque ce mode est activé, vous verrez des informations sur le fichier modèle que vous chargez d'imprimer dans la console Max. Il n'est pas bon de laisser le mode verbose activé lorsque vous effectuez un rendu graphique en temps réel, mais les messages peuvent être très utiles lorsque vous configurez et déboguez un patch.

- Cliquez sur la boîte de *message* contenant le message "**read mushrooms.obj**". Cet objet se trouve au-dessus de l'objet *jit.gl.model*.

Le fichier **mushrooms.obj** devrait avoir été installé dans le chemin de recherche de Max lors de l'installation de Jitter. Si, pour une raison quelconque, ce fichier a été déplacé, le message «**• error: jit.gl.model: can't find mushrooms.obj**» s'affiche dans la console Max. Si le fichier est trouvé correctement, vous verrez le message suivant dans la console Max:

```
reading mushrooms.obj .....
done.
node names
u137000467_mushroom
u137000467_mushroom5
u137000467_mushroom1
u137000467_mushroom6
u137000467_mushroom2
u137000467_mushroom7
u137000467_mushroom3
u137000467_mushroom8
u137000467_mushrooms.obj
u137000467_mushroom4
u137000467_mushroom9
jit.gl.model: reconstruction
groups: 10
matérials: 1
triangles: 10560
vertices: 7603
texture coords: 20
normals: 7603
```

Lorsque le fichier que vous spécifiez après que le message **read** est trouvé et déterminé comme étant un fichier de modèle valide, le message «**reading [file] .obj**» est imprimé. Ensuite, les points sont imprimés de temps en temps pendant le chargement du fichier. La lecture de grands modèles à partir de fichiers texte peut prendre un certain temps, il est donc agréable de voir les points apparaître afin de savoir que quelque chose ne s'est pas mal passé.

Les *Node Names* répertorient les nœuds d'animation à utiliser avec le système d'animation Jitter. Ceux-ci peuvent être assignés à l'attribut **name** d'un objet *jit.anim.node* pour une animation détaillée du groupe. Si vous envoyez à *jit.gl.model* un message **copynodestoclipboard**, puis collez le presse-papiers dans un patch, vous obtiendrez la hiérarchie complète des nœuds du modèle.

Lorsque le chargement du modèle est terminé, «**jit.gl.model: rebuilding**» est imprimé. Certaines opérations, en plus de la lecture d'un nouveau modèle, entraînent la reconstruction interne de la représentation du modèle - toutes ces opérations affichent ce message pour que vous sachiez ce qui se passe.

Le message «**groups: 10**» vous indique qu'il existe onze groupes de polygones dans le modèle. Un modèle n'est pas nécessairement constitué d'un seul objet comme une pomme ou un champignon, il peut s'agir d'un groupe d'objets ou d'une scène entière. L'affectation de différents objets du modèle à différents groupes de polygones permet de les manipuler indépendamment dans votre patch.

Le message «**matérials: 1**» vous indique combien de définitions de matériaux il y a dans le fichier du modèle. Une définition de matériau peut spécifier les composantes ambiante, diffuse et spéculaire de la couleur de la surface d'un objet, comme nous l'avons vu dans le *didacticiel 35*. Un fichier de modèle peut comporter plusieurs définitions de matériau qui peuvent être appliquées à un ou plusieurs groupes de polygones du modèle. Ce modèle, cependant, n'en possède aucune.

Nous pouvons également voir que ce modèle est constitué de 10 560 triangles joignant 7603 sommets, 20 coordonnées de texture et 7603 normales. Il se peut qu'il y ait moins de coordonnées de texture et de normales que de sommets, car ces nombres sont référencés dans un grand index

lorsque le modèle est lu, et peuvent être réutilisés lorsque les triangles sont construits.

Compatibilité des fichiers de modèles .obj

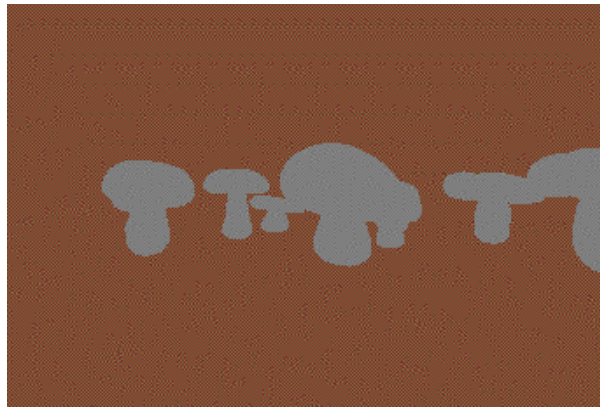
(.obj Model File Compatibility) Il existe de nombreuses extensions du format de fichier de base *.obj*. Parmi celles-ci, citons les surfaces NURBS et paramétriques. Pour l'instant, seules les sommets, les polygones, les groupes, les matériaux et les coordonnées de texture sont pris en charge.

Attributs du modèle

Si vous avez suivi les instructions, vous avez lu tout ce texte aride sur les messages de la console Max, tandis que onze champignons planent de manière alléchante dans le patch du didacticiel. Maintenant, examinons les attributs qui affectent la façon dont le modèle est rendu.

Éclairage et ombrage

Nous voyons les champignons dans l'espace, mais leurs contours sont simplement remplis avec le gris moyen qui est la couleur par défaut pour les objets du groupe GL.



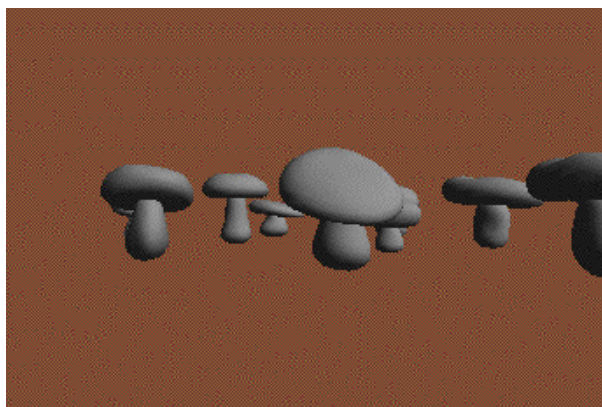
Les champignons à l'ombre plate.

- Cliquez sur le *toggle* au-dessus de la boîte de *message* indiquant "**lighting_enable \$ 1**" pour envoyer le message "**lighting_enable 1**" à l'objet *jit.gl.model*.

Maintenant, le modèle a une apparence plus solide car certaines zones polygonales sont plus claires et d'autres plus sombres. Mais chaque polygone a sa propre couleur de manière uniforme sur toute sa surface. Le résultat est plutôt grossier.

- Cliquez sur le *toggle* au-dessus de la boîte de *message* indiquant "**smooth_shading \$ 1**" pour envoyer le message "**smooth_shading 1**" à l'objet *jit.gl.model*.

Lorsque l'ombrage lisse activé, l'ombre de chaque polygone du modèle est mélangée de sommet à sommet. Cela masque les bords des polygones et donne une apparence plus lisse - nous commençons à avancer, si le réalisme est notre objectif.



Champignons avec éclairage et ombrage lisse activés.

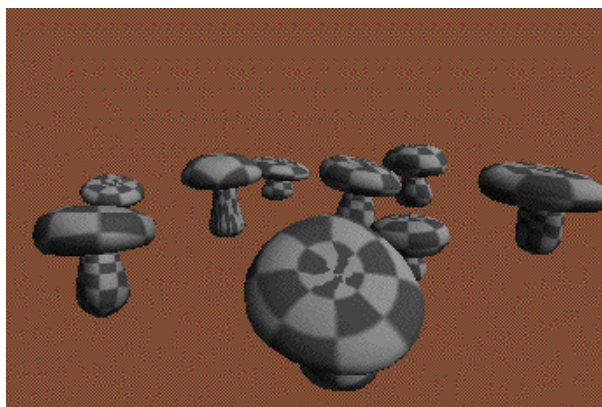
Mappage des textures

Comme pour les autres objets du groupe GL, vous pouvez définir la texture appliquée à l'ensemble du modèle à l'aide du message «texture».

- Cliquez sur la boîte de message indiquant "**grid**" au-dessus de l'objet "**prepend texture**" pour envoyer le message "**texture grid**" à l'objet *jit.gl.model*.

Cela définit la texture du modèle. Puisqu'il y a un objet *jit.gl.handle* connecté à l'objet *jit.gl.model* dans ce patch, nous pouvons faire tourner et déplacer le modèle comme indiqué dans le *Tutoriel 32*, afin d'avoir un meilleur aperçu de la façon dont la texture est appliquée.

Remarquez le soin avec lequel la texture est enroulée autour des champignons. Les chapeaux sont couverts radialement, les tiges cylindriquement, et il y a de petites variations dans l'apparence de chaque champignon. Ces coordonnées de texture ont été créées dans un programme de modélisation 3D par le fabricant du modèle. Les coordonnées spécifient le point de la texture 2D à utiliser à chaque sommet.



Champignons cartographiés avec soin.

Pour voir à quoi ressemble le modèle avec les coordonnées générées automatiquement, vous pouvez utiliser le message «**tex_map**». Comme indiqué dans le *didacticiel 34*, les modes **tex_map** de 1, 2 et 3 spécifient respectivement les mappages linéaires d'objets, de sphère et d'oeil. «**tex_map 0**» utilise les coordonnées spécifiées dans le modèle.

- Expérimentez ces modes si vous le souhaitez, en modifiant la boîte de *nombre* au-dessus de la boîte de *message* indiquant «**tex_map \$ 1**». Lorsque vous avez terminé, rétablissez les coordonnées de texture intégrées au modèle en définissant la boîte de *nombre* sur 0.

En plus des fonctions standard des groupes GL pour appliquer des textures, l'objet *jit.gl.model* peut être utilisé pour appliquer une texture différente à chaque groupe de polygones du modèle.

- Cliquez sur la boîte de *message* indiquant "0" au-dessus de l'objet "**prepend texture**" pour désactiver la texturation du modèle.
- Cliquez sur la boîte de *message* «**texgroup 1 grid**».

Le message **texgroup [group-number] [texture-name]** définit la texture du groupe de polygones [**group-number**] du modèle avec la texture [**texture-name**]. À la suite du message qui vient d'être envoyé, le groupe de polygones 1 devrait être réglé sur la texture de grille, et donc l'un des champignons est couvert de grillage tandis que les autres restent gris uni. Ce message peut être utilisé conjointement avec les méthodes de mise à jour des textures décrites dans le didacticiel 34 pour appliquer une image ou un film différent à chaque groupe de polygones du modèle.

Si un message **texgroup** est reçu pour un groupe de polygones donné, ce groupe utilise la texture nommée dans le message, au lieu de toute texture qui aurait pu être envoyée à l'ensemble du modèle à l'aide du message "**texture**". Si un message **texgroup [group-number] 0** est reçu, la texture nommée n'est plus utilisée pour ce groupe, et le groupe revient à la texture globale du modèle.

- Cliquez sur la boîte de *message* indiquant «**texgroup 1 0**» pour supprimer la texture de la grille du groupe de polygones 1.

Dessin des groupes

En plus d'être texturés différemment, les groupes de polygones d'un modèle peuvent être dessinés séparément. Vous pouvez contrôler ce dessin à l'aide du message **drawgroup**.

- Cliquez sur l'objet **bang** au-dessus de l'objet *counter* sur le côté droit du patch du didacticiel.

Cela incrémente le *counter*, ce qui entraîne l'envoi du message «**drawgroup 1**» à l'objet *jit.gl.model*. Les groupes de polygones sont indexés à partir du nombre 1. Lorsque l'objet *jit.gl.model* reçoit le message **drawgroup [n]**, il ne dessine que le groupe de polygones n du modèle, jusqu'à nouvel ordre. Vous devriez donc maintenant voir un champignon isolé dans le cadre.

En cliquant plusieurs fois sur l'objet **bang**, vous pouvez faire défiler les champignons les uns après les autres. Vous pouvez utiliser cette technique pour charger une animation 3D, en stockant une image de l'objet animé dans chaque groupe de polygones du modèle.

Lorsqu'il reçoit le message **drawgroup 0**, l'objet *jit.gl.model* réactive le dessin de tous les groupes de polygones.

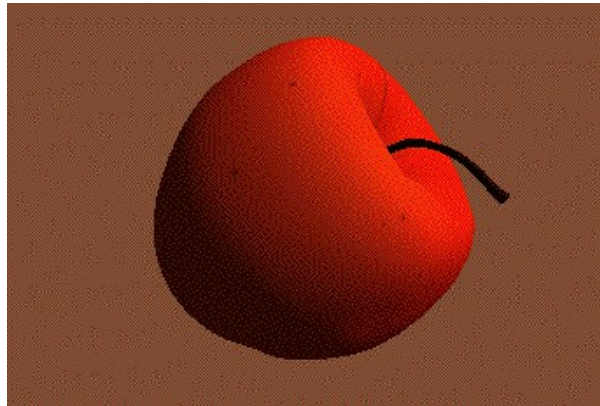
- Cliquez sur la boîte de *message* indiquant «**drawgroup 0**» pour réactiver tous les groupes de polygones.

Modèles de matériaux

Les modèles peuvent contenir des définitions de matériaux, mais ce modèle particulier n'en a pas. Chargeons-en un qui en a une.

- Cliquez sur la boîte de *message* contenant le texte «**read apple.obj**» pour charger le modèle de la pomme dans le dossier Tutorial.

Ce modèle est moins compliqué que le précédent, il se chargera donc beaucoup plus rapidement. En lisant la sortie de l'objet *jit.gl.model* dans la console Max, nous pouvons voir qu'il a 4 452 triangles. Nous pouvons également voir dans la console Max qu'il a 3 matériaux. Dans l'image rendue du modèle, nous pouvons voir les effets de ces matériaux. Même si aucune texture n'a été appliquée, la pomme est rouge avec une tige brune.

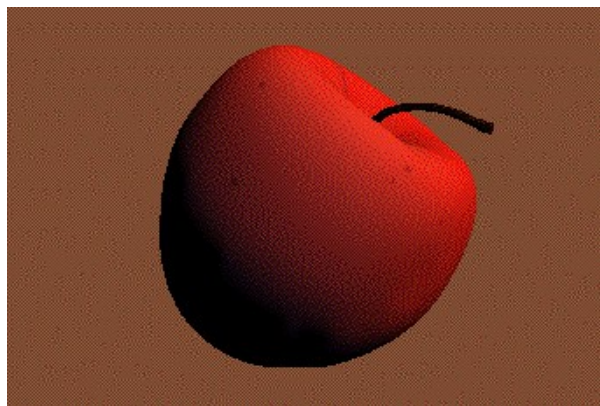


Le modèle pomme montrant deux matériaux.

L'attribut **material_mode** de l'objet *jit.gl.model* détermine dans quelle mesure les matériaux spécifiés dans le fichier du modèle seront utilisés pour le rendu du modèle. Pour l'instant, le mode de matériau est 1, par défaut.

- Définissez la boîte de *nombre* au-dessus de la boîte de *message* indiquant «**material_mode \$ 1**» à la valeur **2**.

Vous remarquerez que la peau de la pomme prend un aspect plus brillant et légèrement plus bleu, tandis que la tige reste dans le même brun plat. Cela est dû au fait que les composantes spéculaires des matériaux sont maintenant utilisées dans le rendu. Lorsque nous avons chargé la pomme pour la première fois avec le paramètre par défaut de l'attribut **material_mode** de 1, seuls les composantes diffuses des matériaux étaient utilisés dans le rendu.



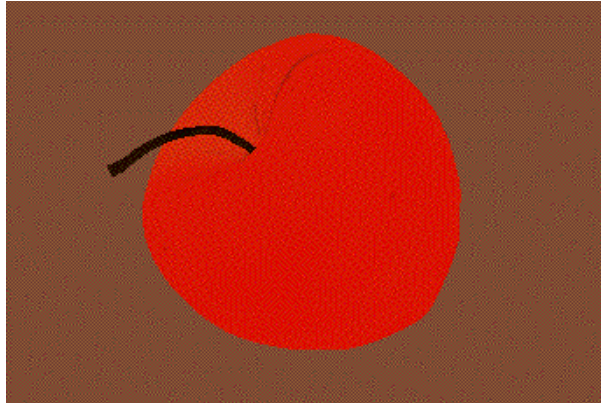
La pomme avec des composantes de matière spéculaire activée.

- Définissez la même boîte de *nombre* à la valeur 0.

A présent, aucune des propriétés matérielles du modèle n'est utilisée dans le rendu. Au lieu de cela, l'attribut de couleur de l'objet est utilisé pour déterminer la couleur diffuse de l'objet. Ainsi, le modèle apparaît dans le gris par défaut. Essayons un autre réglage pour voir comment les différents modes de matériaux peuvent être utiles.

- Désactivez l'objet *toggle* au-dessus de la boîte de *message* «**Lighting_enable \$ 1**». Ensuite, essayez tous les différents paramètres de l'attribut **material_mode**.

Vous constaterez que lorsque l'éclairage est désactivé, les matériaux de l'objet n'ont d'effet que dans le mode de matériau 1. Comme la composante diffuse du matériau est utilisée pour déterminer la couleur de l'objet, le dessin en aplat prend une couleur différente pour chaque groupe de polygones.



La pomme à l'ombre plate avec un mode matériel de 1.

Voici un bref récapitulatif des paramètres de l'attribut **material_mode**:

material_mode 0 n'utilise pas les matériaux du fichier modèle. L'attribut de couleur de l'objet détermine la couleur diffuse, et la couleur plate si l'éclairage n'est pas activé.

material_mode 1 utilise la composant matériel diffus du fichier modèle pour déterminer la couleur diffuse ou la couleur plate du modèle rendu.

material_mode 2 utilise tous les composants matériels du fichier modèle.

Sommaire

Nous avons vu comment charger des modèles 3D complexes en plusieurs parties à partir de fichiers au format *.obj.* sur le disque. Les attributs de l'objet *jit.gl.model* peuvent être utilisés pour affecter l'éclairage et l'ombrage du modèle et pour déterminer comment ses matériaux affectent le rendu. Le fichier modèle peut contenir différents groupes de polygones qui peuvent être dessinés et texturés individuellement.

Merci à Oliver French <http://o.ffrench.free.fr/meshbank> pour l'utilisation de ces élégants modèles freeware.