

34-Utilisation des textures

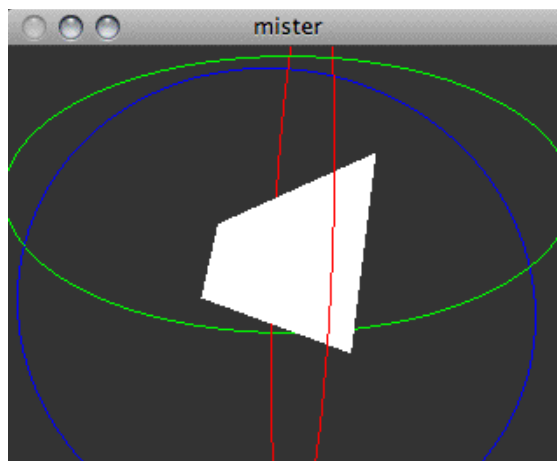
Ce didacticiel vous montre comment créer et appliquer des textures aux données de géométrie 3D générées par le groupe GL. Il couvre la création d'une texture nommée avec *jit.gl.render*, l'affectation d'une texture nommée à un objet GL, l'utilisation de couleurs en conjonction avec des textures, la conversion de données image/vidéo en une texture, et diverses façons d'envelopper la géométrie avec une texture.

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

Vous verrez un parallélogramme blanc, mais il s'agit en fait d'un tétraèdre dessiné par l'objet *jit.gl.plato*. L'objet *jit.gl.plato* est capable de restituer plusieurs solides de Platon, notamment les tétraèdres, les hexaèdres (également appelés cubes), les octaèdres, les dodécaèdres et les icosaèdres. Comme l'éclairage n'est pas activé et qu'aucune texture n'est appliquée au tétraèdre, il est difficile de dire qu'il s'agit en fait d'une forme 3D.

- Utilisez la souris pour faire pivoter le tétraèdre avec l'objet *jit.gl.handle*, comme indiqué dans le didacticiel 32.

Cela devrait illustrer qu'il s'agit en fait d'une forme 3D, mais en appliquant une texture à l'objet *jit.gl.plato*, cela deviendra encore plus apparent.



Rotation du solide platonique.

Qu'est-ce qu'une texture?

Une texture est essentiellement une image superposée à une géométrie. Tout comme les autres images de Jitter, les textures ont une composante alpha, rouge, verte et bleue. Afin d'utiliser la composante alpha, le mélange doit être activé. Le mélange est traité en détail dans le didacticiel 33.

Dans Jitter, une texture a un nom et appartient à l'objet *jit.gl.render*. Les autres objets qui sont attachés au contexte de dessin associé à un objet *jit.gl.render* donné peuvent utiliser n'importe laquelle des textures nommées appartenant à cet objet *jit.gl.render*.

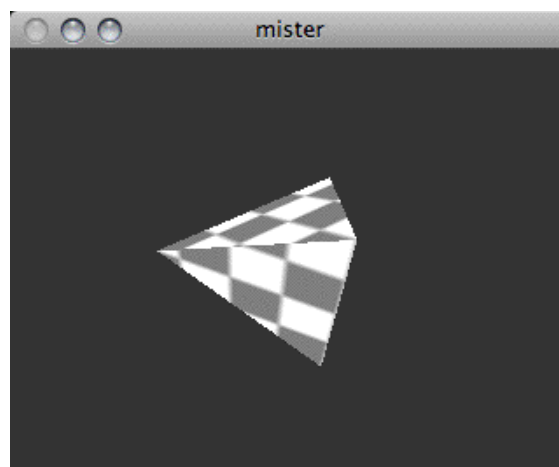
Pour des raisons d'efficacité, la plupart des implémentations OpenGL exigent que les dimensions des données de texture soient des puissances entières de deux: 16, 32, 64, etc. Cependant, Jitter autorise des textures de taille arbitraire. Il existe une taille minimale de **1** par **1** et une taille maximale en fonction de l'implémentation d'OpenGL, généralement **1024** par **1024**.

Création d'une texture

- Créez une texture nommée **grid** en cliquant sur la boîte de *message* intitulée **texture grid 64 64** dans la section du patch intitulée **Textures**. Ce message est envoyé à l'objet *jit.gl.render*.

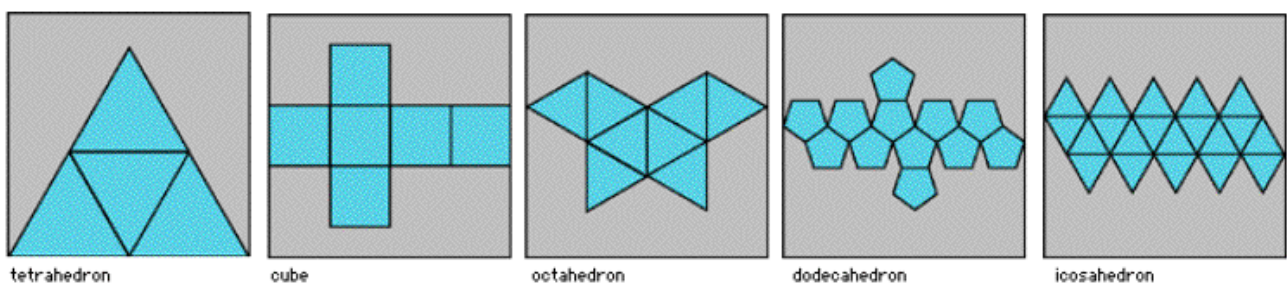
Le premier argument du message de texture est le nom de la texture, et les deux arguments suivants spécifient la largeur et la hauteur de la texture. Cela crée une texture **64 x 64** nommée **grid**, et la remplit avec le motif par défaut (un damier blanc et gris). Vous ne verrez pas encore de résultats pour l'instant, car la texture n'a pas encore été appliquée à la géométrie.

- Appliquez la texture au tétraèdre en cliquant sur la boîte de *message* intitulée **texture grid** dans la section du patch intitulée **Platonic Solid**. Cela définit l'attribut de texture de l'objet *jit.gl.plato*, et lors du dessin, il utilisera la texture nommée **grid**. Vous devriez maintenant voir un tétraèdre en damier.



Tétraèdre avec appliqué une texture en damier.

L'objet *jit.gl.plato* utilise une stratégie d'"emballage cadeau" pour appliquer la texture au tétraèdre. Dans le fichier d'aide *jit.gl.plato*, le sub-patch **texture_maps** illustre exactement comment les différents solides platoniques sont empaquetés.



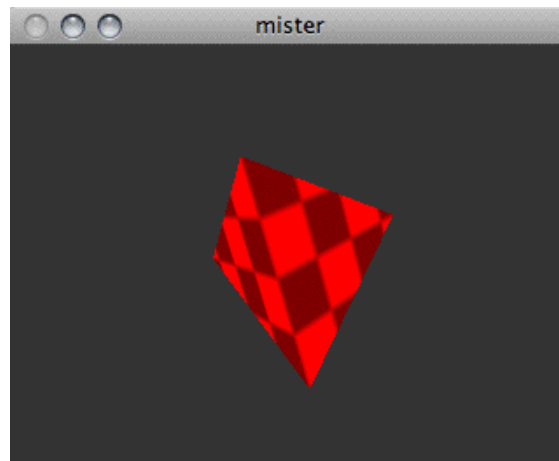
Comment les cartes de texture sont appliquées à différents solides platoniques.

Remarque importante: vous remarquerez que les objets *jit.gl.render* et *jit.gl.plato* utilisent le message de texture de différentes manières. L'objet *jit.gl.render* utilise ce message pour créer une texture, tandis que l'objet *jit.gl.plato* et les autres objets GL utilisent ce message pour **appliquer** une texture. L'objet *jit.gl.render* possède également un message permettant d'appliquer l'une de ses textures nommées à des données géométriques brutes transmises sous forme de matrices Jitter. Ce message est *usetexture*. La capacité de l'objet *jit.gl.render* à rendre des données géométriques brutes transmises sous forme de matrices Jitter sera abordée dans le **didacticiel 37**.

Textures et couleurs

Lors de l'application d'une texture à une géométrie, OpenGL prend également en compte les informations de couleur et d'éclairage, de sorte que les valeurs de couleur et d'éclairage actuelles seront multipliées par l'image de la texture lors du dessin. Si la couleur est blanche et que l'éclairage est désactivé, les couleurs de la texture ne seront pas modifiées.

- Dans la section du patch intitulée *Platonic Solid*, définissez la couleur du tétraèdre sur le rouge en définissant la boîte de *nombre red* sur *1*, la boîte de *nombre green* sur *0* et la boîte de *nombre bleue* sur *0*.



Manipulation de la couleur de l'objet rendu.

- Remettez la couleur du tétraèdre sur blanc (**1. 1. 1.**) pour notre prochaine section.

Convertir une image ou une vidéo en une texture

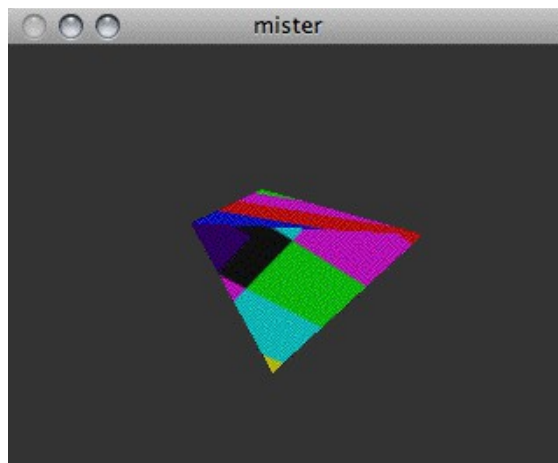
Bien qu'il s'agisse d'une illustration, vous voudrez généralement utiliser des textures autres que le damier gris et blanc par défaut. Cela peut être accompli en chargeant une image ou une vidéo dans les objets *jit.movie* ou *jit.matrix* et en envoyant le message **texture [texture-name] [matrix-name]** à l'objet *jit.gl.render*. Si la texture spécifiée par **[texture-name]** existe déjà, la matrice entrante sera ré-échantillonnée aux dimensions actuelles de la texture. Si aucune texture portant ce nom n'existe, une nouvelle texture sera créée. Ses dimensions seront la plus proche puissance de deux ou égale aux dimensions de la matrice Jitter.

- Cliquez sur la boîte de *message* contenant **texture-picture 128 128** dans la section du patch intitulée *Textures*. Cela crée une texture **128 x 128** nommée **picture**, et, comme précédemment, la remplit avec le motif de damier blanc et gris par défaut.
- Cliquez sur la boîte de *message* contenant **read colorbars.png,bang** pour charger l'image **colorbars.pict** dans l'objet *jit.movie*, et l'envoyer sur son chemin vers la texture nommée **picture**.

Vous ne verrez encore aucun résultat, car l'objet *jit.gl.plato* utilise toujours la texture nommée **grid**.

- Cliquez sur la boîte de *message* contenant **texture-picture** dans la section du patch intitulée *Platonic Solid*.

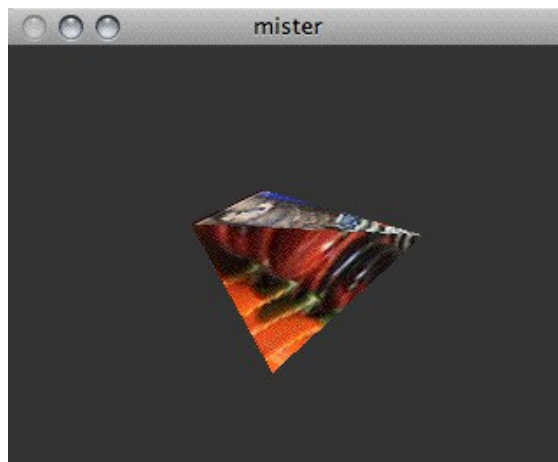
Vous devriez maintenant voir l'image des barres de couleurs enroulée autour du tétraèdre.



Utiliser une image comme texture.

Dans de nombreux cas, vous n'aurez besoin d'utiliser que des images fixes comme textures, mais Jitter prend également en charge l'utilisation de vidéos en mouvement comme textures en copiant de manière répétée la sortie de l'objet *jit.movie* dans la texture nommée.

- Cliquez sur la boîte de *message* contenant **read dish.mov** pour charger *dish.mov* dans l'objet *jit.movie*.
- Cliquez sur l'objet *toggle* connecté à l'objet *metro* pour commencer à copier la vidéo dans la texture nommée **picture**.



Cartographie de texture à l'aide d'un film.

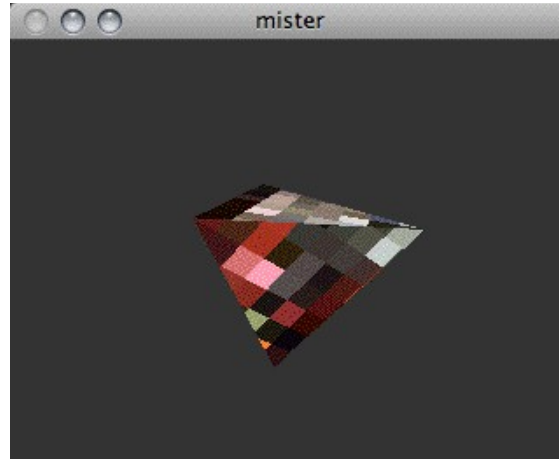
Interpolation et taille de la texture

Par défaut, l'interpolation de texture est activée, donc les pixels de l'écran qui se trouvent entre des pixels de texture utiliseront une valeur interpolée de leurs voisins dans l'image de texture. Cela a pour effet d'estomper ou de lisser la texture. Pour appliquer des textures sans interpolation, l'interpolation peut être désactivée en utilisant le message **interp** de l'objet *jit.gl.render*. Ce message n'affecte que la texture actuelle de l'objet *jit.gl.render*, donc avant d'envoyer le message **interp**, vous pouvez envoyer le message **usetexture [texture-name]** pour que la texture actuelle soit celle spécifiée par l'argument [**texture-name**].

- Cliquez sur l'objet *toggle* connecté à la boîte de message contenant **use texture picture, interp \$1** pour que l'objet *jit.gl.render* utilise la texture nommée **picture**, et ensuite pour activer et désactiver l'interpolation pour la texture nommée **picture**.

Une fois qu'une texture a été créée, la taille de la texture peut être modifiée, en envoyant le message **texture [texture-name] [width] [height]** où **[width]** et **[height]** spécifient les nouvelles dimensions.

- Réglez la boîte de *nombre* intitulée **Resize texture** sur **16**. Cela enverra à l'objet *jit.gl.render* le message **texture picture 16 16**, redimensionnant l'image de la texture en une image **16 x 16**.



Utilisation d'une texture non interpolée.

Les textures peuvent être supprimées afin de libérer de la mémoire en envoyant à *jit.gl.render* le message **deletetexture [texture-name]**.

Modes de mappage

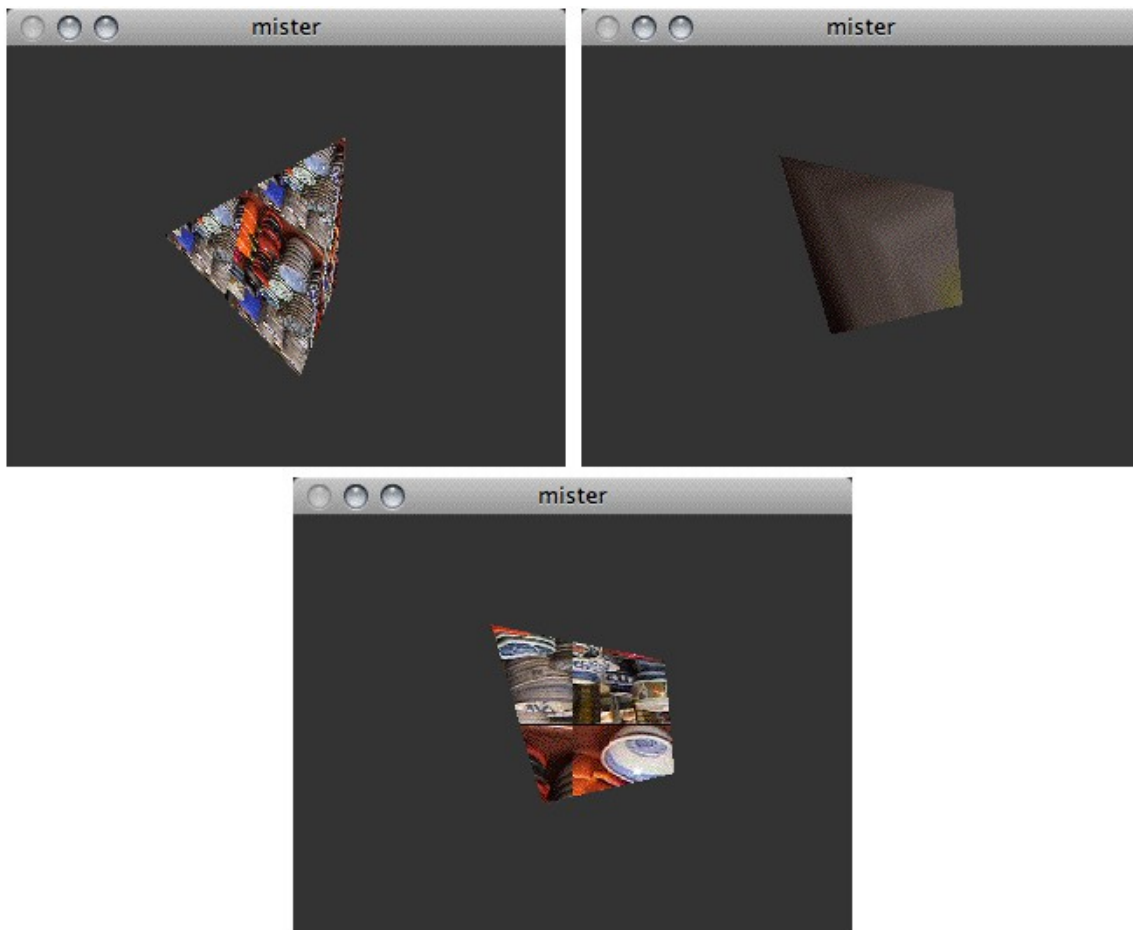
- Ouvrez le sub-patch **texture_mapping**.

Jusqu'à présent, nous n'avons abordé que le mappage de texture implicite que l'objet *jit.gl.plato* fournit. OpenGL fournit également quelques autres mappages de texture explicites pour appliquer des textures aux données géométriques. Il s'agit des modes de mappage **linéaire de l'objet**, **linéaire de l'œil** et de la **carte de sphère**.

Le mode de **l'objet linéaire** applique la texture d'une manière fixe par rapport au système de coordonnées de l'objet. Lorsque l'objet est tourné et positionné dans la scène 3D, le mappage de la texture reste le même. En revanche, le mode **linéaire de l'œil** applique la texture de manière fixe par rapport au système de coordonnées de l'œil. Lorsque l'objet est tourné et positionné dans la scène 3D, l'application de la texture à l'objet change. Enfin, le mode de **mappage de sphère** produira l'effet communément appelé «mappage d'environnement»; l'objet est rendu comme s'il reflétait le milieu environnant et suppose que la texture contient une image mappée par sphère de milieu environnant. Lorsque l'objet est tourné et positionné dans la scène 3D, l'application de la texture à l'objet change.

Ces modes de mappage explicites peuvent être utilisés en définissant l'attribut **tex_map** du groupe GL. Une valeur **tex_map** de **0** est la valeur par défaut et utilise les coordonnées de texture implicites de l'objet GL. Une valeur **tex_map** de **1** utilisera le mode **d'objet linéaire** d'OpenGL. Une valeur **tex_map** de **2** utilisera le mode **carte de sphère** d'OpenGL. Une valeur **tex_map** de **3** utilisera le mode **linéaire de l'œil** d'OpenGL.

- Essayez de modifier la boîte de *nombre* reliée à la boîte de *message* contenant **tex_map \$ 1**. Positionnez et faites pivoter le tétraèdre avec votre souris, et voyez comment les différents modes affectent le mappage de la texture.

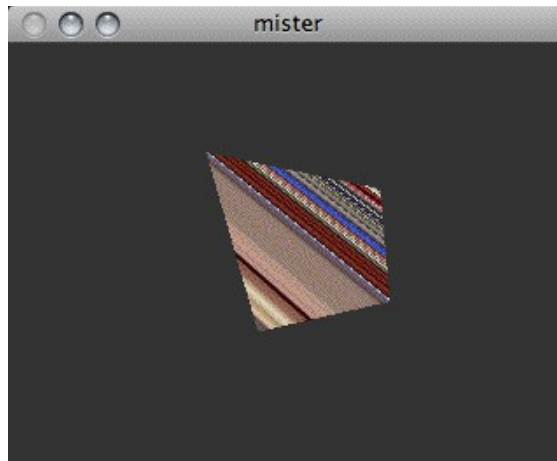


Utilisation de différents modes de mappage implicites: objet linéaire (gauche), carte de sphère (droite), linéaire de l'oeil (inférieur).

Les modes de mappage *linéaire* et *linéaire de œil* des objets d'OpenGL ont des paramètres supplémentaires qui affectent la façon dont ils s'appliquent à la texture. Ces paramètres sont définis avec les attributs **tex_plane_s** et **tex_plane_t** du groupe GL. Ces attributs sont chacun des vecteurs en coordonnées homogènes à 4 dimensions. Le produit scalaire de **tex_plane_s** et d'un point donné en coordonnées homogènes à 4 dimensions détermine la position horizontale de l'image de texture à appliquer. Le produit scalaire de **tex_plane_t** et d'un point donné dans des coordonnées homogènes à 4 dimensions déterminent la position verticale de l'image de texture à appliquer. Par défaut, **tex_plane_s** est égal à **(1. 0. 0. 0.)** et **tex_plane_t** est égal à **(0. 1. 0. 0.)**.

Une description plus détaillée de la façon dont les attributs **tex_plane_s** et **tex_plane_t** affectent le mappage de la texture est hors de portée de ce didacticiel, mais cela ne signifie pas que vous ne pouvez pas jouer avec pour générer des effets intéressants. Pour les curieux, veuillez consulter le *Livre rouge* ou le *Livre bleu* d'OpenGL.

- Expérimentez avec les objets de la boîte de *nombre* sous les étiquettes du *plan de texture s* et du *plan de texture t* (**tex_map** doit être réglé sur **1** ou **3** pour avoir un effet).



Expérimentation sur des paramètres supplémentaires.

Sommaire

Nous avons établi comment créer des textures et diverses façons de les appliquer à la géométrie créée par le groupe GL. Le message **texture** de l'objet *jit.gl.render* peut être utilisé pour créer, dimensionner et copier des données d'image ou de vidéo dans des textures nommées. L'attribut **texture** du groupe GL spécifie la texture nommée à utiliser, et l'attribut **tex_map** du groupe GL sélectionne soit le mode de mappage de texture implicite, soit l'un des trois modes de mappage de texture OpenGL explicites: *objet linéaire*, *linéaire de œil* ou *carte de sphère*.