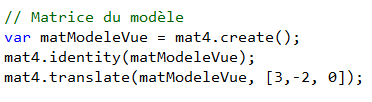
|  |
| --- |
| Laboratoire 9A: WebGL - Les transformations sur le modèle |

Théorie

En 3D, les transformations sur le modèle sont très importantes car c’est par l’intermédiaire de ces transformations que nous allons modifier les positions des vertex des objets 3D. Du moins, en apparence!!!

Il existe plusieurs types de transformations mais les principales sont la translation, la mise à l’échelle et la rotation.

1. La translation



Nous allons utiliser la translation pour déplacer nos objets 3D.

|  |
| --- |
| **Avant le déplacement** |
| **Après le déplacement** |

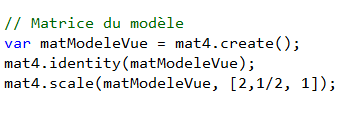
Sur la page Web **9-A-1 Web GL Translation d’un cube.html,** le cube s’est déplacé. Nous avons utilisé la matrice du modèle pour déplacer le cube (méthode **.translate**). Ici, le cube s’est déplacé de 3 unités sur l’axe des X et de -2 unités sur l’axe des Y.

En fait, ce n’est pas tout à fait vrai. En réalité, le cube ne s’est pas déplacé. C’est le plan cartésien qui s’est déplacé. Sur le plan cartésien, le cube est toujours situé à la même position.

Pour bien comprendre, écrivez votre nom sur une feuille transparente et déplacez la feuille transparente. Il va de soi que votre nom va sembler se déplacer car il a été écrit sur une feuille transparente. Mais en réalité, votre nom ne se déplace pas, c’est la feuille qui se déplace. Ici, la feuille transparente représente le plan cartésien.

**La translation déplace le point d’origine du plan cartésien**. Il va de soi que, lorsqu’on déplace le plan cartésien, que tout ce qui est situé sur le plan cartésien se déplace également.

1. La mise à l’échelle



Nous allons utiliser la mise à l’échelle pour modifier la taille de nos objets 3D.

|  |
| --- |
| **Avant la mise à l’échelle** |
| **Après la mise à l’échelle** |

Sur la page Web **9-A-2 Web GL Mise à l’échelle d’un cube.htm,** le cube a grossi dans le sens de l’axe des X mais s’est aminci dans le sens de l’axe des Y. En fait, cela ne ressemble plus du tout à un cube. Nous avons utilisé la matrice du modèle pour modifier la taille du cube (méthode **.scale**). Ici, le cube est 2 fois plus large mais 2 fois moins haut.

En fait, ce n’est pas tout à fait vrai. En réalité, le cube n’a pas modifié sa taille. C’est le plan cartésien qui a modifié sa taille. Sur le plan cartésien, le cube a toujours la même taille.

Pour bien comprendre, écrivez votre nom sur un ballon transparent et gonflez le ballon. Il va de soi que votre nom va paraitre plus gros car vous l’avez écrit sur le ballon. Mais en réalité, votre nom ne grossit pas, c’est le ballon qui grossit. Ici, le ballon représente le plan cartésien.

**La mise à l’échelle modifie de manière proportionnelle la distance qu’il y a entre chacun des points du plan cartésien par rapport au point d’origine** (le point d’origine ne change pas de place). Tout nombre entre 0 (exclus) et 1 (exclus) amincit l’objet, tout nombre plus grand que 1 grossit l’objet, le nombre 1 conserve la taille originale de l’objet et le nombre 0 fait disparaître la taille de l’objet (par exemple, si on met 0 pour l’axe des X, l’objet n’a plus de largeur).

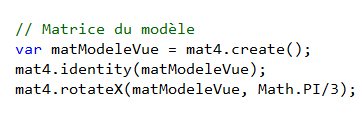
Au niveau de la mise à l’échelle, les nombres négatifs inversent le sens des axes. Si on considère l’axe des X, ce qui situé à droite va se dessiner à gauche et ce qui est situé à gauche va se dessiner à droite (comme dans un miroir).

Par exemple, si on tape les instructions suivantes, l’objet va se déplacer à gauche (car le sens de l’axe a été inversé).

mat4.scale(matModeleVue, [-1, 1, 1]); // Inversion de l’axe des X

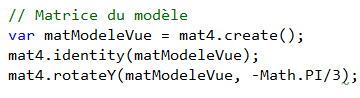
mat4.translate(matModeleVue, [3, 0, 0]); // Déplacer de 3 sur l’axe des X

1. La rotation

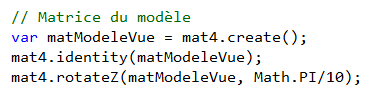


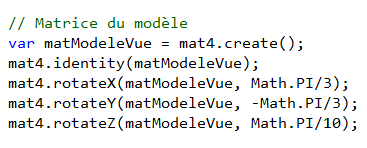
Nous allons utiliser la rotation pour faire tourner nos objets 3D.

Sur la page Web **9-A-3 Web GL Rotation X d’un cube.htm,** le cube a tourné autour de l’axe des X de 60 degrés (π/3 radians).



Sur la page Web **9-A-4 Web GL Rotation Y d’un cube.htm,** le cube a tourné autour de l’axe des Y de -60 degrés (-π/3 radians).

Sur la page Web **9-A-5 Web GL Rotation Z d’un cube.htm,** le cube a tourné autour de l’axe des Z de 18 degrés (π/10 radians).

Sur la page Web **9-A-6 Web GL Rotation XYZ d’un cube.htm,** le cube a tourné autour de l’axe des X de 60 degrés puis a tourné autour de l’axe des Y de -60 degrés puis a tourné autour de l’axe des Z de 18 degrés.

|  |
| --- |
| **Avant la rotation** |
| **Après la rotation autour de l’axe des Z** |

En fait, ce n’est pas tout à fait vrai. En réalité, le cube n’a pas tourné. C’est le plan cartésien qui a tourné. Sur le plan cartésien, le cube n’a pas bougé.

**La rotation fait tourner le plan cartésien autour du point d’origine** (le point d’origine ne tourne pas).

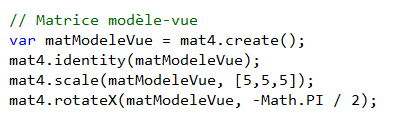
Il faut faire très attention à la rotation. Étant donné que la rotation fait tourner le plan cartésien, les axes changent d’orientation. Par exemple, si on fait tourner le plan cartésien de 90 degrés autour de l’axe des Z, cela va avoir comme conséquence que l’axe des X va prendre la place de l’axe des Y et que l’axe des Y va prendre la place des X. Cela peut être mélangeant comme on le verra plus loin.

Il faut également faire attention lorsqu’on crée nos objets. Si nos objets 3D ne sont pas créés autour du point d’origine du plan cartésien, l’objet 3D ne va pas tourner sur lui-même mais va tourner autour du point d’origine du plan cartésien qui est situé à l’extérieur du cube (un peu comme la terre qui tourne autour du soleil).

C’est pour cette raison qu’il est préférable de créer nos objets 3D autour du point d’origine du plan cartésien (autant que possible centrer l’objet 3D).

1. Le cumul des transformations

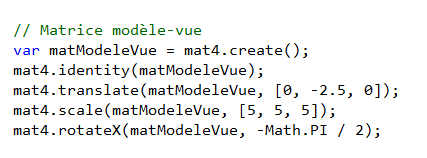
En 3D, les transformations sur le modèle sont cumulatives.



Sur la page Web **9-A-7 Web GL Echelle et rotation X d’une vrille.htm,** la vrille a grossi de 5 fois sa taille normale puis a tourné de -90 degrés autour de l’axe des X. Donc, ce qui a tourné, c’est la vrille qui a grossi de 5 fois (et non pas la vrille originale).

Ici, la vrille n’est pas centrée. La raison c’est que, lorsque nous avons créé la vrille, la pointe de vrille était située sur le point d’origine du plan cartésien et, puisque c’est le plan cartésien qui tourne, la vrille a tourné autour de sa pointe.

Si on veut solutionner ce problème, il faut absolument faire descendre le point d’origine du plan cartésien avant d’appliquer les autres transformations.



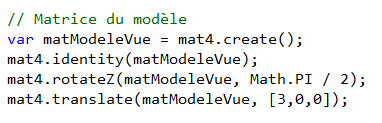
Sur la page Web **9-A-8 Web GL Transformation, échelle et rotation X d’une vrille.htm,** nous faisons descendre le plan cartésien (sur l’axe des Y) de -2.5 unités avant d’appliquer les autres transformations sur la vrille.

Notez qu’ici, la partie visible du plan cartésien va de -5 à 5 sur l’axe des X, de -5 à 5 sur l’axe des Y et de -5 à 5 sur l’axe des Z.



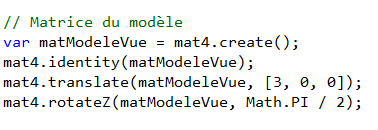
Si, lors de la création de la vrille, nous avions centré la vrille, nous n’aurions pas eu ce problème.

L’ordre des transformations est très important car les transformations ne sont pas commutatives.



Par exemple, si nous faisons tourner le plan cartésien de 90 degrés autour de l’axe des Z. Cela va avoir comme conséquence que l’axe des X va prendre la place de l’axe des Y. Si par la suite, nous déplaçons le point d’origine de 3 unités sur l’axe des X, l’objet va monter (car l’axe des X a pris la place des Y).

Voir **9-A-9 Web GL Rotation puis translation d’une vrille.htm**



Cela n’a pas le même effet si on fait l’inverse. Si, tout d’abord, nous déplaçons le point d’origine de 3 unités sur l’axe des X, l’objet va se déplacer vers la droite. Puis, si, par la suite, nous faisons tourner le plan cartésien de 90 degrés autour de l’axe des Z, l’objet va simplement tourner.

Voir **9-A-10 Web GL Translation puis rotation d’une vrille.htm**

C’est pour cette raison que plusieurs programmeurs adoptent habituellement la stratégie suivante :

* On applique tout d’abord la translation
* Puis on applique la mise à l’échelle
* Finalement on applique les rotations

Les rotations sont appliquées à la fin car elles ont la fâcheuse habitude de modifier l’orientation des axes.