|  |
| --- |
| Laboratoire 13A: WebGL : La projection |

Théorie

**1. Un peu d’animation**

Ouvrez la page **13-A-1 Web GL – Animation.htm**.

Appuyez sur les touches « Flèche-à-droite » ou « Flèche-à-gauche ». Le cube se déplace horizontalement de 0.1 unité (dans les sens de l’axe des X).

Appuyez sur les touches « Flèche-en-haut » ou « Flèche-en-bas ». Le cube se déplace verticalement de 0.1 unité (dans les sens de l’axe des Y).

Appuyez sur les touches « Home » ou « End ». Le cube recule ou avance de 0.1 unité (dans le sens de l’axe des Z). Présentement, vous ne voyez pas le cube avancer ou reculer mais si vous appuyez sur une des touches assez longtemps, vous allez finir par le voir disparaître. Nous allons expliquer la raison plus tard.

Appuyez sur les touches « Ctrl+Flèche-à-droite » ou « Ctrl+Flèche-à-gauche ». Le cube tourne dans le sens horizontal de 5 degrés (autour de l’axe des Y).

Appuyez sur les touches « Ctrl+Flèche-en-haut » ou « Ctrl+Flèche-en-bas ». Le cube tourne dans le sens vertical de 5 degrés (autour de l’axe des X).

Appuyez sur les touches « Ctrl+Home » ou « Ctrl+End ». Le cube tourne dans le sens de la profondeur (autour de l’axe des Z).

Appuyez sur les touches « Shift+Flèche-à-droite » ou « Shift+Flèche-à-gauche ». Le cube change de taille horizontalement (dans le sens de l’axe des X).

Appuyez sur les touches « Shift+Flèche-en-haut » ou « Shift+Flèche-en-bas ». Le cube change de taille verticalement (dans le sens de l’axe des X).

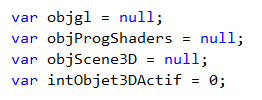
Appuyez sur les touches « Shift+Home » ou « Shift+End ». Le cube change de taille en profondeur (dans le sens de l’axe des Z). Présentement, vous ne voyez pas le cube changer de taille en profondeur mais si vous appuyez sur une des touches assez longtemps, vous allez finir par le voir disparaître. Nous allons expliquer la raison plus tard.

Appuyez sur la touche « V ». Maintenant, c’est la vrille qui est active. Vous pouvez transformer cette vrille de la même manière que vous avez transformé le cube.

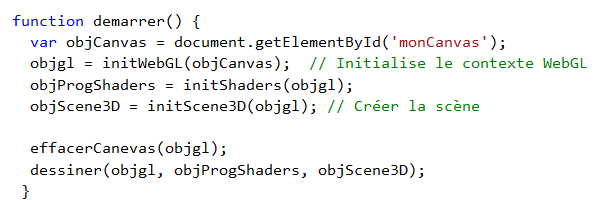
Pour que ce soit le cube qui soit de nouveau actif, appuyez sur la touche « C ».

Voyons voir la manière que nous avons programmé cela.

Tout d’abord, dans le canevas, nous avons programmé l’événement ***onkeydown*** (cet événement est appelé à chaque fois que l’utilisateur appuie sur une touche du clavier).

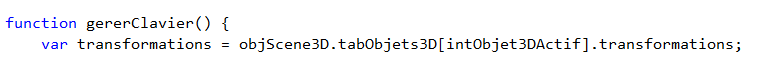


Pour que la fonction **gererClavier** ait accès aux variables importantes, nous les avons déclarées globalement. Ici, la variable **intObjet3DActif** contient le numéro de l’objet 3D qui est actif (au départ, c’est le cube).

Dans la fonction **demarrer**, nous initialisons la scène 3D (comme nous l’avons fait à la fin du laboratoire précédent).

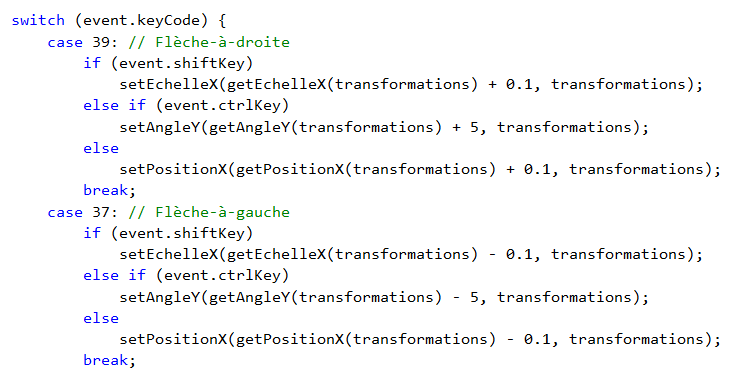
Sur la scène 3D, il y a 2 objets 3D : le cube (l’objet #0) et la vrille (l’objet #1).

Dans la fonction **gererClavier**, tout d’abord, nous allons chercher les transformations actuelles de l’objet3D actif.



Puis nous appliquons les transformations (la transformation appliquée dépend de la touche sur laquelle l’utilisateur a appuyée).

Voici les instructions pour les touches « Flèche-à-droite » et « Flèche-à-gauche ». Tentez de comprendre ce qu’elles font.



A la fin complètement de la fonction **gererClavier**, nous effaçons le canevas puis nous redessinons la scène au complet.

**2. La projection**

Tel que déjà énoncé, les transformations du modèle servent à déplacer, mettre à l’échelle ou faire tourner les objets 3D. En fait, ce ne sont pas les objets qui se transforment mais le plan cartésien (mais l’utilisateur ne s’en rend pas compte).

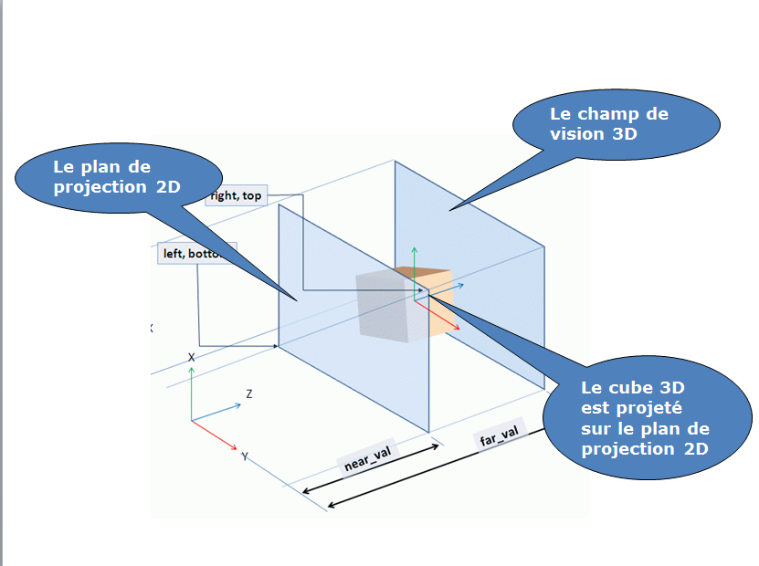
Les transformations de la projection, tout aussi importantes, servent à projeter les objets 3D sur une surface 2D (dans le cas qui nous concerne, les objets 3D sont projetés sur le canevas).

Il existe essentiellement deux types de projection :

* La projection orthographique
* La projection en perspective

2.1 La projection orthographique

Dans la projection orthographique, le champ de vision 3D **est un parallélépipède**.



Tous les objets 3D (en partie ou en totalité) qui sont situés à l’intérieur de ce parallélépipède sont projetés sur le plan de projection 2D (dans le cas qui nous concerne sur le canevas).

Tous les objets 3D (en partie ou en totalité) qui sont situés à l’extérieur de ce parallélépipède ne sont pas projetés sur le plan de projection 2D (ils ne sont pas visibles).

La particularité de ce type de projection, c’est que les tailles et les angles des objets 3D sont intacts. Si vous faites avancer ou reculer (sur l’axe des Z) un objet 3D, vous n’aurez pas l’impression qu’il recule ou qu’il avance. Cela s’explique que tous les points (x,y,z) pour un z donné frappent le plan de projection au même endroit. C’est comme s’il n’y avait pas d’axe des Z.

De la même manière, si vous modifiez la taille en profondeur (sur l’axe des Z) d’un objet 3D, vous n’aurez pas l’impression que sa taille est modifiée.

Par contre, le champ de vision est respecté. Si vous faites avancer ou reculer l’objet 3D assez longtemps, il va finir par disparaître car il va sortir à l’extérieur du champ de vision (à l’extérieur du parallélépipède).

De la même manière, si vous grossissez en profondeur l’objet 3D assez longtemps, il va devenir tellement gros qu’une partie de l’objet 3D va sortir du champ de vision (il va devenir plus gros que le parallélépipède lui-même).

Vous pouvez tester cela avec la page **13-A-1 Web GL – Un peu d’animation.htm**.

Il est possible de visualiser le plan de projection comme étant une fenêtre 2D. Ce qu’il y a à l’intérieur du champ de vision est projeté directement sur la fenêtre.

Pour être réaliste, cette fenêtre (ou ce plan de projection) doit respecter la proportion hauteur/largeur du canevas.

Par exemple, sur la page **13-A-2-1 Web GL – Fenêtre plus large que haute.htm**, les objets 3D sont plus minces qu’ils ne devraient être en réalité. La raison c’est que le rapport largeur/hauteur du canevas est de 1 (640/640).



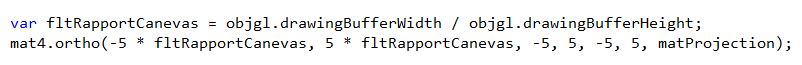
Par contre le rapport largeur/hauteur de la fenêtre est de 2 (20/10). Les deux rapports ne correspondent pas.

Il en va de même pour la page **13-A-2-2 Web GL – Canevas plus large que haut.htm**, les objets 3D sont plus larges qu’ils ne devraient être en réalité. La raison c’est que le rapport largeur/hauteur du canevas est de 4 (1280/320).



Par contre, le rapport largeur/hauteur de la fenêtre est de 1 (10/10). Les deux rapports ne correspondent pas.

Pour éviter cela, il suffit d’ajuster le rapport largeur/hauteur de la fenêtre pour que ce dernier corresponde à celui du canevas comme c’est le cas sur la page **13-A-2-3 Web GL – Fenêtre proportionnelle au canevas.htm**



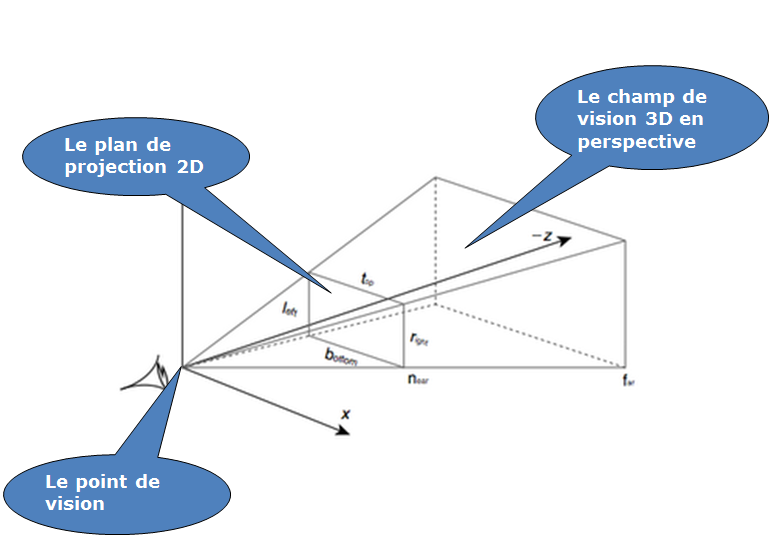
En ajustant la taille de la fenêtre (du plan de projection) sur celui du canevas, les objets 3D dessinés sont réalistes.

2.2 La projection en perspective

Dans la projection en perspective, le champ de vision 3D est beaucoup plus près de l’être humain car l’être humain possède une vision en perspective.

C’est grâce à la vision en perspective qu’un objet éloigné semble plus petit qu’un objet proche; que deux parallèles semblent se rencontrer à l’infini. En fait, sans la vision en perspective, l’être humain aurait beaucoup de difficultés à évaluer la distance qu’il y a entre lui et un autre objet à l’extérieur de lui.

Dans la projection en perspective, le champ de vision 3D **est une pyramide tronquée**.

Tous les objets 3D (en partie ou en totalité) qui sont situés à l’intérieur de cette pyramide tronquée sont projetés sur le plan de projection 2D (dans le cas qui nous concerne sur le canevas).

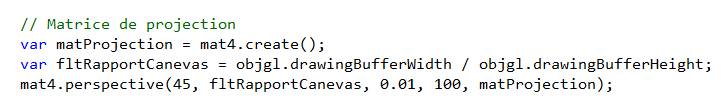
Tous les objets 3D (en partie ou en totalité) qui sont situés à l’extérieur de cette pyramide tronquée ne sont pas projetés sur le plan de projection 2D (ils ne sont pas visibles).

Le point de vision représente l’œil de l’être humain. En **WebGL**, cet œil est toujours situé sur le point d’origine du plan cartésien. De plus, la pyramide est toujours centrée autour de l’axe des Z négatifs.

Pour voir ce que donne la vision en perspective, ouvrez la page **13-A-2-4 Web GL – Projection en perspective (45 degrés).htm.**

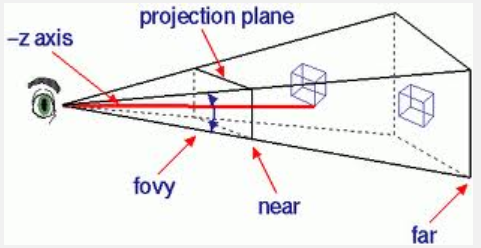
Au point de départ, vous ne voyez rien. La raison, c’est que le cube et la vrille sont situés à l’extérieur du champ de vision. Appuyez sur la touche **«**Flèche-à-droite » plusieurs fois. Cela va faire en sorte que le cube va se déplacer de la gauche vers la droite (au point de départ, il est situé à gauche). Lorsque vous voyez l’arrière du cube (un grand carré vert), appuyez sur la touche « Home » plusieurs fois pour faire reculer le cube. Vous allez voir la face avant du cube (un grand carré rouge). Continuez à faire reculer le cube. Observez que le cube devient de moins en moins gros. Si vous continuez à faire reculer le cube assez longtemps, à un moment donné, il va sortir du champ de vision.

En fait, le cube ne rapetisse pas lorsque vous reculez le cube. C’est la vision en perspective qui donne cette impression.



Pour réaliser la vision en perspective, vous devez utiliser la matrice **mat4.perspective.**

Cette matrice requiert 5 paramètres :

* L’angle de vision en hauteur (ici, l’angle de vision en hauteur est de 45 degrés).
* Le rapport largeur/hauteur de la fenêtre (on utilise habituellement le rapport largeur/hauteur du canevas).
* La distance entre l’œil et le plan de projection sur l’axe des Z négatifs (habituellement, cette distance est très courte; ici, il y a 0.01 unité de distance entre l’œil et le plan de projection).
* La distance entre l’œil et la fin du champ de vision sur l’axe des Z négatifs (habituellement, cette distance est grande pour éviter que les objets sortent trop rapidement à l’extérieur du champ de vision; ici, cette distance est de 100 unités.)
* La matrice de projection.

Il est important de prendre note que plus que l’angle de vision en hauteur est grand et plus que le champ de vision est large. Dans la 2ième image, le champ de vision est plus large que dans la 1ère image. Un angle de vision en hauteur de 180 degrés ou plus est impossible (si cela était possible, on pourrait voir ce qu’il y a en arrière de nous).

Ouvrez la page **13-A-2-5 Web GL – Projection en perspective (170 degrés).htm.**

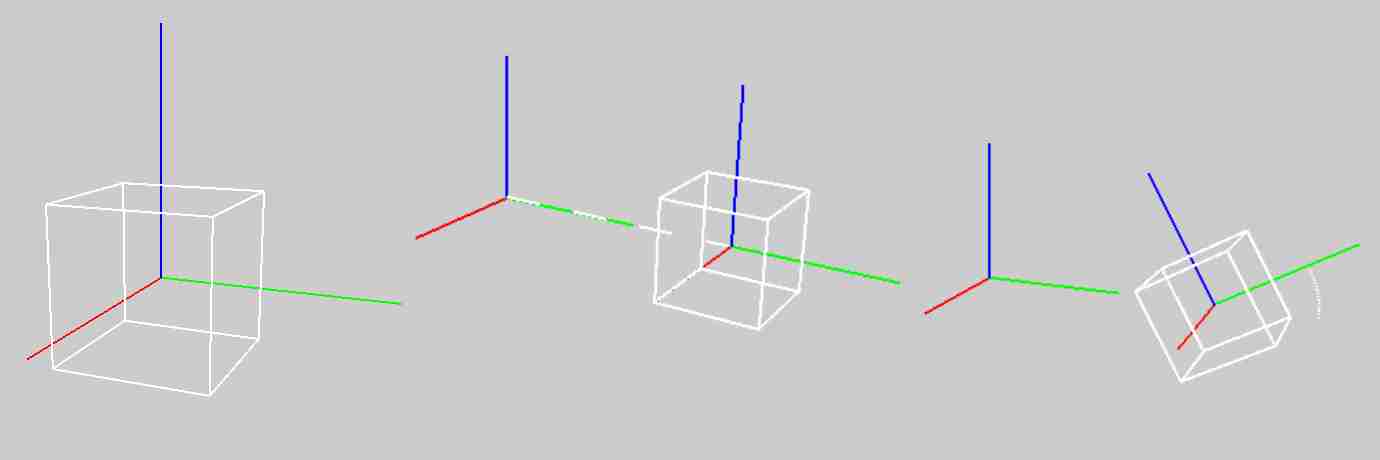
Ici, l’angle de vision en hauteur est de 170 degrés, ce qui donne un champ de vision très large. On voit le cube et la spirale mais ces objets sont très déformés dans le sens de la profondeur. Vous pouvez faire reculer ou avancer le cube, le faire tourner si vous voulez pour constater à quel point le cube se déforme (en réalité, le cube ne se déforme pas; c’est l’angle de vision qui donne cette impression). C’est ce que vous verriez si vous aviez un angle de vision à 170 degrés.

Dans la vision en perspective, l’avantage d’avoir un champ de vision plus large, c’est qu’on peut apercevoir plus d’objets 3D. Le désavantage, c’est que ces objets 3D sont très déformés dans le sens de la profondeur.

2.3 Le système de coordonnées du modèle et le système de coordonnées de la projection

La matrice du modèle et la matrice de projection sont deux matrices complètement différentes. Cela implique qu’il existe deux systèmes de coordonnées (deux plans cartésiens).

Au point de départ, les deux systèmes de coordonnées coïncident.

****

Lorsqu’on applique les transformations du modèle, les deux systèmes de coordonnées ne coïncident plus.

Pour visualiser les objets 3D, l’utilisateur utilise le système de coordonnées de la projection. Ce système de coordonnées est fixe. C’est pour cette raison que l’utilisateur a l’illusion que c’est l’objet 3D qui se transforme, alors, qu’en réalité, c’est le système de coordonnées du modèle qui se transforme.