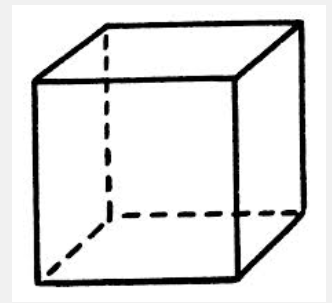
|  |
| --- |
| Laboratoire 8B: WebGL - Les objets 3D complexes |

Théorie

1. La création d’objets 3D complexes : les sous-objets

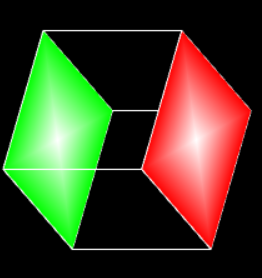
Jusqu’à date, tous les objets que nous avons dessinés étaient des objets très simples. De plus, c’était des objets 3D plats. En effet, tous ces objets n’avaient aucune profondeur. Tous ces objets étaient dessinés sur le plan Z = 0. Pour dessiner réellement des objets en 3D, il faut toujours utiliser l’axe des Z.

Pour créer un objet 3D complexe, nous allons créer des sous-objets. Tout objet 3D sera un assemblage de sous-objets (à la manière d’un jeu LEGO).

Pour illustrer cela, considérons un cube. Nous pouvons visualiser le cube comme étant un assemblage de 6 sous-objets (6 carrés). Ces 6 carrés ne sont pas situés au même endroit dans le plan cartésien, mais c’est quand même un assemblage de sous-objets.

La manière que nous allons procéder pour créer les vertex de nos objets 3D, c’est que nous allons créer autant de tableaux de vertex qu’il y a de sous-objets. Nous allons procéder de la même manière pour les couleurs. Par exemple, si nous voulons créer ce cube, nous pouvons créer 6 tableaux de vertex et 6 tableaux de couleurs (un tableau par face). Lorsque nous allons dessiner le cube, nous allons dessiner chacun des 6 sous-objets l’un après l’autre.

Pour résumer, un objet 3D va être créé à l’aide d’un tableau de sous-objets.

Sur la page Web **8-B-1 Web GL Création et dessin d’un cube.html**, vous apercevez un cube. Présentement, vous voyez le cube de côté car nous lui avons fait subir des rotations. Nous allons étudier les rotations des objets 3D dans un autre laboratoire.

Ce cube est composé de cinq sous-objets :

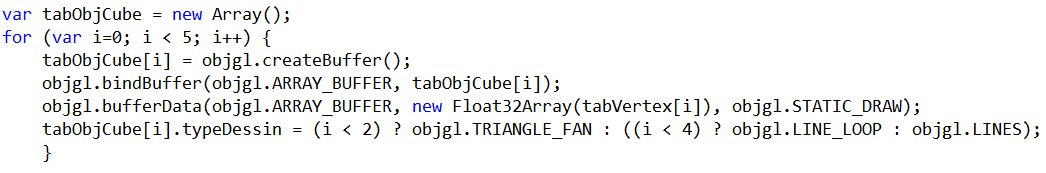
* un carré plein qui a un dégradé radial rouge qui est situé à l’avant (Z=1)
* un carré plein qui a un dégradé radial vert qui est situé à l’arrière (Z=-1)
* un carré vide blanc (pour le contour) qui est situé à l’avant (Z=1)
* un carré vide blanc (pour le contour) qui est situé à l’arrière (Z=-1)
* quatre droites blanches qui relient les sommets des deux carrés précédents.

Voyons voir la manière que nous avons créé ce cube.

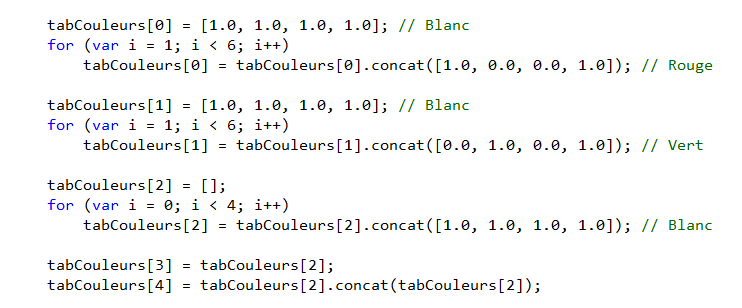
Tout d’abord, nous avons créé 5 tableaux de vertex : deux (2) tableaux pour les faces avant et arrière, deux (2) autres pour les contours avant et arrière et un dernier pour les 4 droites blanches qui relient les sommets.

|  |  |  |
| --- | --- | --- |
|  |  |  |

Par la suite, nous avons créé 5 tampons de vertex (1 tampon par tableau de vertex) :

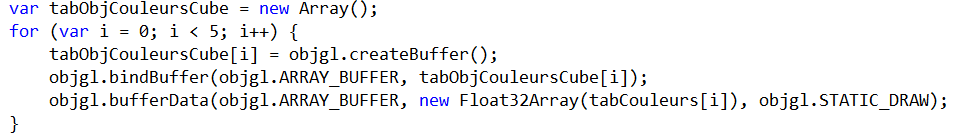


Observez que le type de dessin pour les faces avant et arrière est ***TRIANGLE\_FAN*** (c’est de cette manière qu’on obtient le dégradé radial), que le type de dessin pour les contours avant et arrière est ***LINE\_LOOP*** et que le type de dessin pour les droites qui relient les sommets est ***LINES*** (***LINES*** sert à dessiner une droite entre 2 vertex; le nombre de vertex doit être pair).

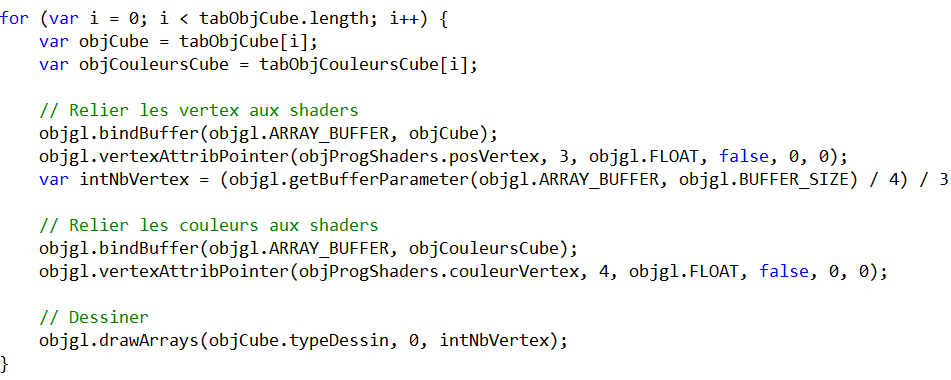


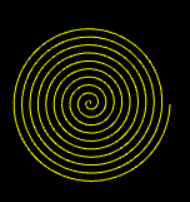
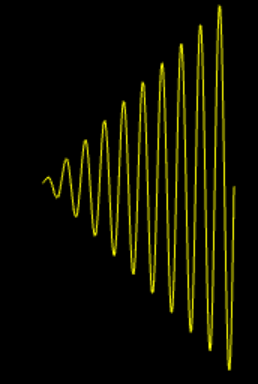
Nous procédons de la même manière pour les couleurs. La couleur de la face avant est un dégradé radial qui va du blanc au rouge; la couleur de la face arrière est un dégradé radial qui va du blanc au vert, la couleur des contours et des droites qui relient les sommets est blanche.

Par la suite, nous avons créé 5 tampons de couleurs (1 tampon par tableau de couleurs) :



Au niveau du dessin, nous dessinons chacun des sous-objets l’un après l’autre (dans un boucle **for**).





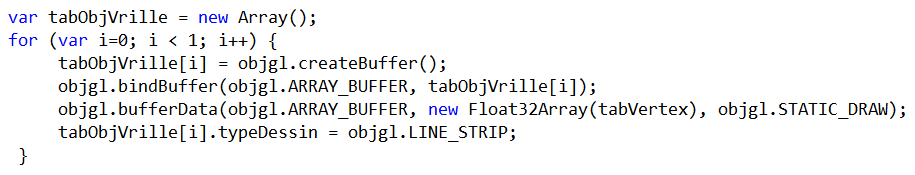
Sur la page Web **8-B-2 Web GL Création et dessin d’une vrille.html**, vous apercevez une vrille jaune. Présentement, vous avez l’impression que c’est une spirale. Vous avez l’impression que c’est une spirale parce que la vrille s’enroule autour de l’axe des Z. Par contre dans l’illustration ci-contre, vous pouvez vous rendre compte que c’est bel et bien une vrille.

Contrairement au cube que nous avons dessiné, la vrille n’est composée que d’un seul objet.

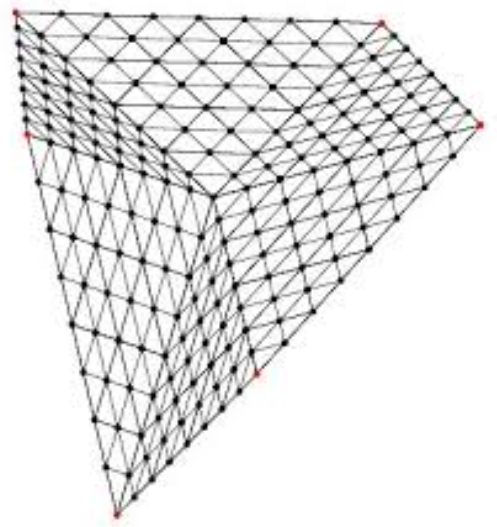
La seule différence entre la spirale et la vrille, c’est que la vrille possède des coordonnées sur l’axe des Z (les coordonnées en Z de la spirale sont toujours 0). En fait, la vrille s’enroule autour de l’axe des Z. La spirale est un objet 2D mais la vrille est un réel objet 3D.



Bien qu’il n’y ait qu’un seul objet 3D dans la vrille, pour conserver la même logique, nous avons placé cet objet dans un tableau de tampons de vertex. Ce tableau ne contient qu’un seul tampon de vertex. Il en va de même pour les couleurs.



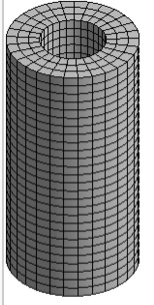
2. La création d’objets 3D complexes : le maillage



Une autre technique, pour créer des objets 3D est de créer un maillage.

Un maillage est un objet 3D défini par un tableau de vertex. A partir de ces vertex, on crée des cellules nommés polytopes.

Un polytope est une figure géométrique qui relie n vertex parmi l’ensemble de vertex. Par exemple, ici, les polytopes sont des triangles. Chaque triangle relie 3 vertex.

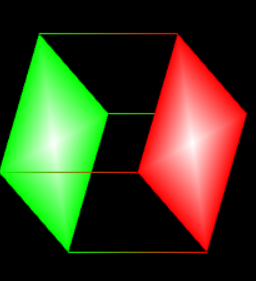
Il est possible d’utiliser d’autres polytopes. Par exemple, ici, les polytopes sont des quadrilatères (tétragones). Chacun d’eux relie 4 vertex.

Habituellement, on utilise des triangles car on ne peut pas plier un triangle en utilisant ses vertex.

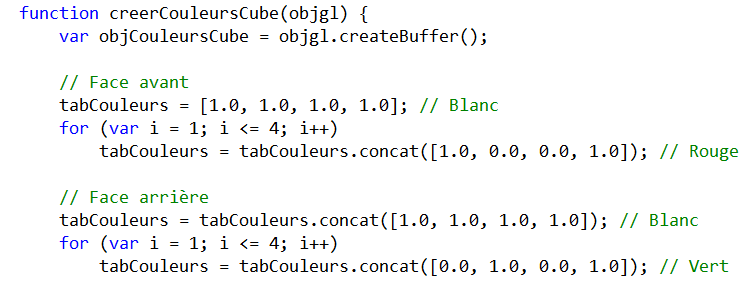
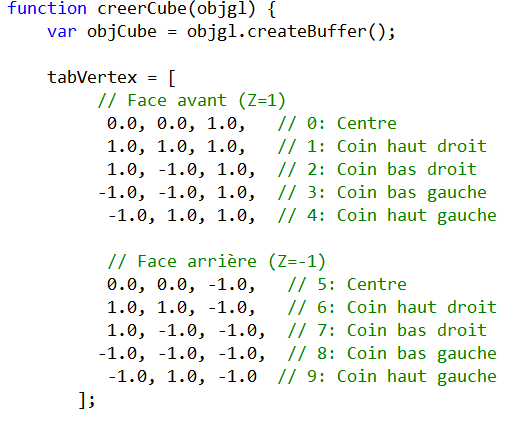
Un triangle est toujours bidimensionnel tandis qu’il est possible pour un quadrilatère d’être tridimensionnel.

La manière que nous allons procéder pour créer un maillage, nous allons tout d’abord créer tous les vertex présents dans le maillage. Puis, nous allons associer une couleur à chaque vertex. Finalement, nous allons créer le maillage en tant que tel, c’est-à-dire les polytopes.

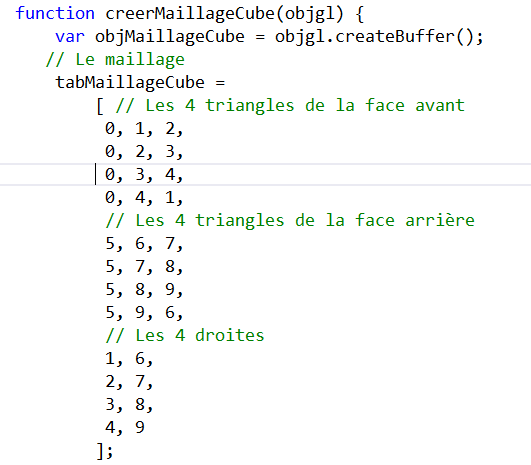
Au niveau du dessin, nous allons dessiner ces polytopes.

Sur la page Web **8-B-3 Web GL Création et dessin d’un cube par maillage.html**, vous apercevez un cube (j’ai créé des rotations au niveau du cube).

Dans ce cube, il n’y a qu’un et un seul tableau de vertex et qu’un et un seul tableau de couleurs.



Ici, le cube a 10 vertex. L’ordre des vertex est très important car cela va déterminer le numéro du vertex que nous allons utiliser plus tard lorsque nous allons créer le maillage.

Dans la fonction qui crée le maillage, nous avons 2 types de polytopes : les triangles qui relient trois (3) vertex et les droites qui relient deux vertex.

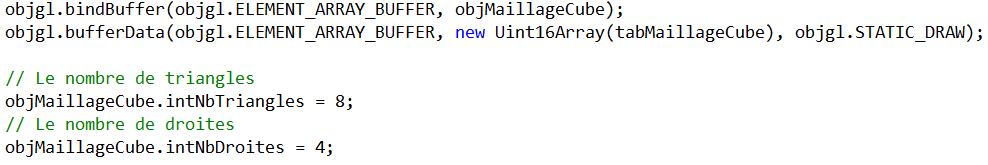
Pour relier les vertex, nous utilisons le numéro du vertex et non pas le vertex lui-même.

Par exemple, les vertex (0, 1, 2) seront reliés pour former un triangle. Les vertex (1, 6) seront reliés pour former une droite.

Dans le tableau, les droites sont placées à la fin des triangles.

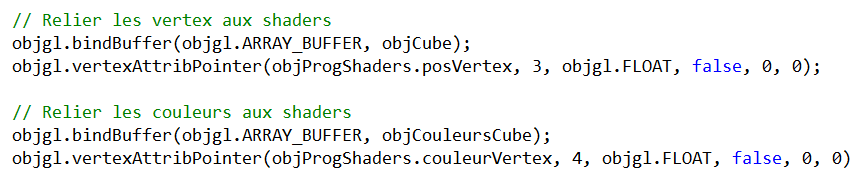
A la fin de la fonction, nous ajoutons ce tableau dans un tampon (de type **ELEMENT\_ARRAY\_BUFFER**).

Puis, nous indiquons combien de triangles doivent être créés et combien de droites doivent être créées.

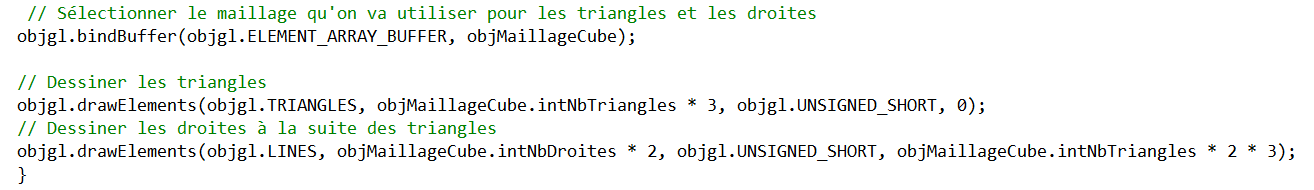


Ces nombres seront utilisés plus tard, lorsque nous allons dessiner.

Au niveau du dessin, on envoie les vertex et ses couleurs d’un seul coup au ***vertex shader*** comme on le fait habituellement. Contrairement à l’objet 3D qui a été créé selon la méthode des sous-objets, on le fait une seule fois car il n’y a qu’un seul tableau de vertex et un seul tableau de couleurs.

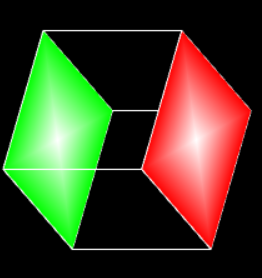


À la fin, on sélectionne le maillage puis on dessine avec **.drawElements()** au lieu de dessiner avec **.drawArrays()**.



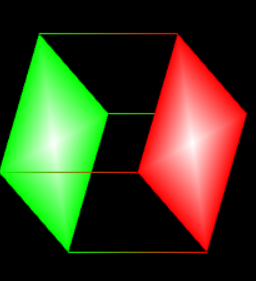
Observez ici qu’on utilise seulement deux (2) types de dessin: les triangles et les droites. Ce sont toujours les mêmes.

3. Sous-objets vs maillage

L’avantage d’utiliser des sous-objets, c’est qu’il est possible de se créer une banque d’objets 3D simples puis de les assembler pour former des objets 3D complexes. Cette technique peut être comparée au jeu LEGO.

Par contre, cette technique a l’inconvénient de créer plusieurs tableaux de vertex (et de couleurs). Il y a donc plusieurs appels **au *vertex shader***. Cela ralentit le système lorsque plusieurs objets 3D complexes sont présents.

De plus, le même vertex est répété plusieurs fois. Par exemple, dans le dessin du cube, le vertex d’un coin est répété 3 fois.

La technique du maillage requiert un seul tableau de vertex (et de couleurs). Il y a donc 1 seul appel au ***vertex shader***.

Par contre, cette technique a l’inconvénient qu’on doit visualiser l’objet 3D dans sa totalité. Il est plus difficile d’assembler des objets déjà construits.

Par contre, avec cette technique, on peut créer un tableau de vertex universel, c’est-à-dire un tableau de vertex à partir duquel plusieurs sous-objets 3D différents peuvent être créés.

La technique des sous-objets est une technique ascendante (du simple vers le complexe) tandis que la technique du maillage est une technique descendante (du complexe vers le simple).