

PROJET SI 2021 : MODELISATION D'UN DRAGON

Dans ce projet nous allons voir la réalisation d'un dragon en C++/OpenGL. Plusieurs contraintes nous sont imposées. Nous devons :

- Modéliser au moins une primitive à partir de sa représentation paramétrique
- Utiliser au moins deux textures : une plaquée sur une face et une autre enroulée autour d'une primitive du 2
- Gérer au moins deux types lumières
- Les touches 'z' et 'Z' permettent de zoomer sur l'objet
- Les touches flèches directionnelles permettent de modifier la vue de l'objet
- Réaliser au moins une animation avec les touches du clavier
- Réaliser au moins une animation automatique

Le dragon que nous avons choisi de modéliser est celui du jeu vidéo « Minecraft ». Nous allons donc voir dans un premier temps la modélisation de la structure du dragon, la création des animations, puis l'application des textures et enfin l'ajout de lumières.

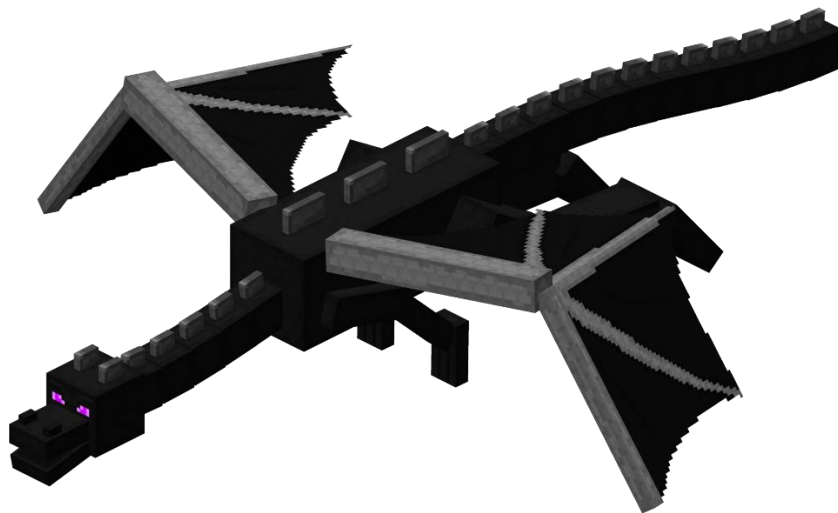


Photo du dragon que l'on souhaite réaliser

I/ La Modélisation

a) Les primitives utilisées

Comme on peut le voir, la structure du dragon est assez simple. Elle est uniquement composée de pièces rectangulaires. Nous allons donc utiliser la primitive d'un cube afin de recréer la majeure partie du dragon. Cependant afin de rendre cela un peu plus intéressant nous utiliserons également la primitive d'un cylindre et d'une sphère pour réaliser les ailes.

La primitive du cube est celle vue en TD/TP. On répertorie tous les indices de points par face dans un tableau, puis dans un autre on stocke les coordonnées de ces points. Il ne nous reste plus qu'à appeler lesdits points dans deux boucles imbriquées. La première pour créer une face à partir de 4 points, la seconde pour créer les 6 faces. Cette primitive est stockée dans la fonction cube().

PROJET SI 2021 : MODELISATION D'UN DRAGON

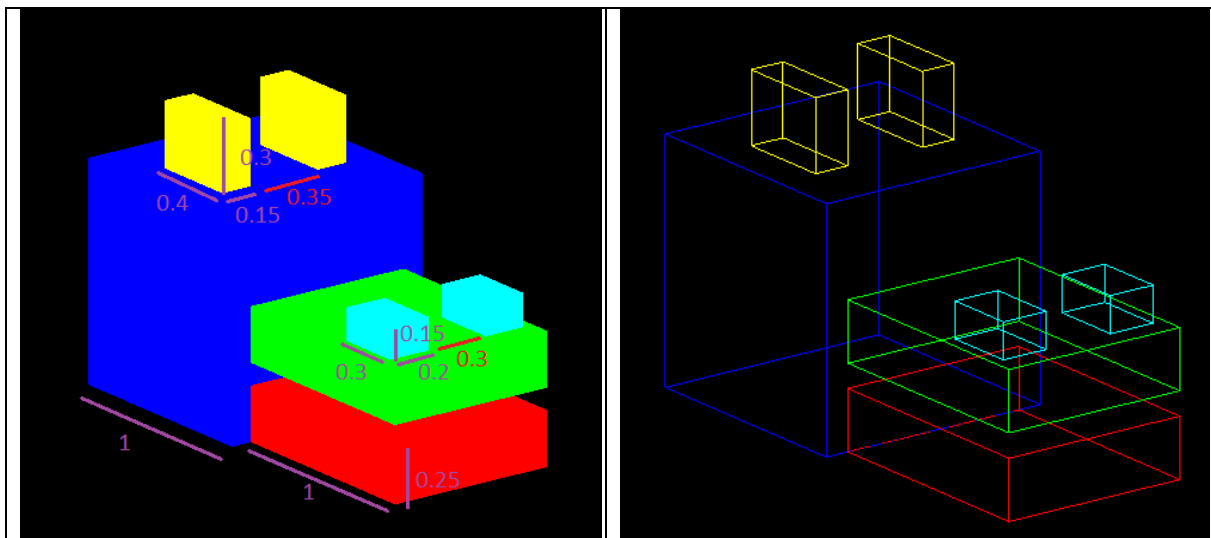
Pour la primitive du cylindre on commence par stocker les coordonnées x et y dans des tableaux $x[n]$ et $y[n]$ (avec n le nombres de points utilisés pour discrétiser le cercle). Puis dans une boucle on crée les couvercles du cylindre en prenant les points $(x[i], y[i], \pm h/2)$, $(x[(i+1)\%n], y[(i+1)\%n], \pm h/2)$, et $(0, 0, \pm h/2)$, puis on crée les faces en prenant les points $(x[i], y[i], h/2)$, $(x[i], y[i], -h/2)$, $(x[(i+1)\%n], y[(i+1)\%n], -h/2)$ et $(x[(i+1)\%n], y[(i+1)\%n], h/2)$. Cette primitive est stockée dans la fonction void `MonCylindre(float r, float h, int n)` avec r le rayon du cylindre.

Enfin pour la primitive de la sphère on crée 3 tableaux x, y, z de taille NM*NP (nombres de point pour discrétiser les cercles horizontaux et verticaux servant à faire la sphère). Comme pour le cylindre on y range les coordonnées, puis dans une double boucle on crée les différentes faces de la sphère. Cette primitive est stockée dans la fonction void `MaSphere(float r, int NM, int NP)` avec r le rayon de la sphère.

b) Les différentes parties du dragon

Pour une question de lisibilité nous avons créé plusieurs fonctions représentant chaque partie du dragon. Chaque partie est globalement similaire, on crée les primitive dont on a besoin, on les place à l'aide de translations et de rotations, puis, si nécessaire, on rajoute des structures à l'aide de `glBegin(GL_POLYGON)`. Nous ne détaillerons donc pas les constructions des différents éléments ici, nous indiquerons simplement les dimensions utilisées.

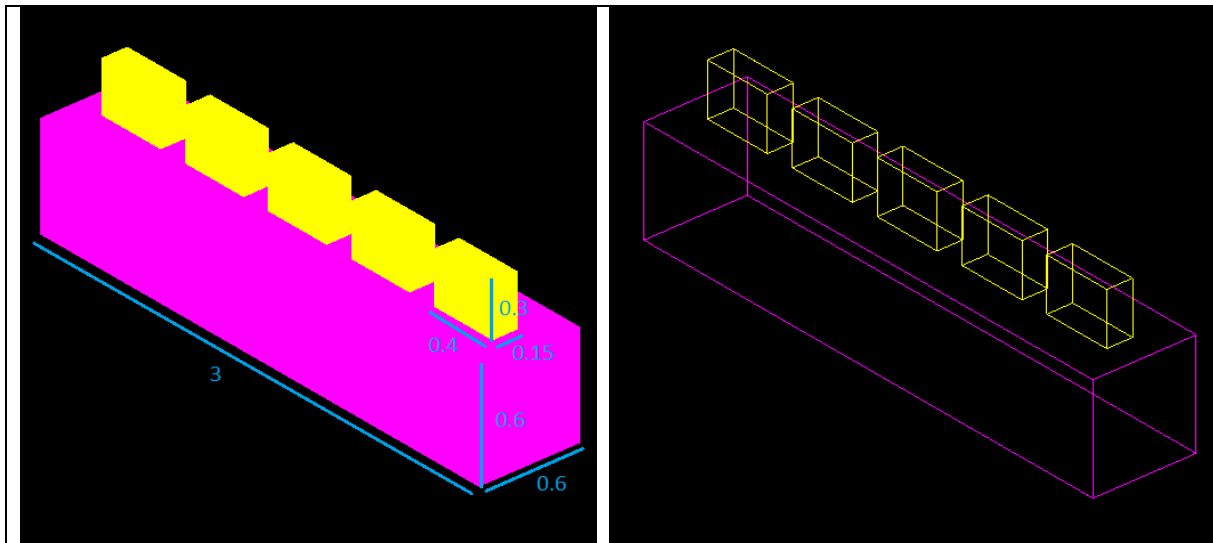
Pour la tête rien de spécial, nous n'utilisons que des cubes :



Tête du Dragon

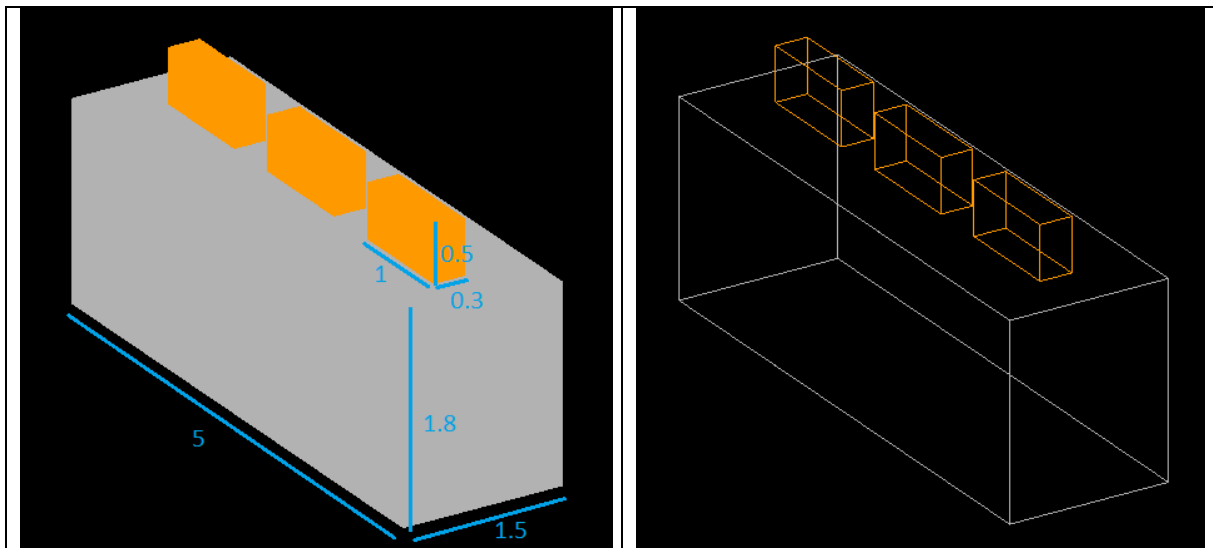
Pour le cou on utilise également que des cubes, cependant pour les écailles présentent au-dessus on automatise tous cela dans une boucle for qui prend en compte la taille du cou, la taille des écailles et le nombres d'écailles souhaiter pour les répartir de manière homogène. La formule utilisée est : $-L + \frac{L-l*n}{n+1} + l * (i - 1) + \frac{l}{2}$, avec L la longueur du cou, l la longueur d'une écaille, n le nombre d'écailles et i la i-ème étape (la boucle commence à i=1 et finie à i=n). Ici nous prenons n=5.

PROJET SI 2021 : MODELISATION D'UN DRAGON



Cou du Dragon

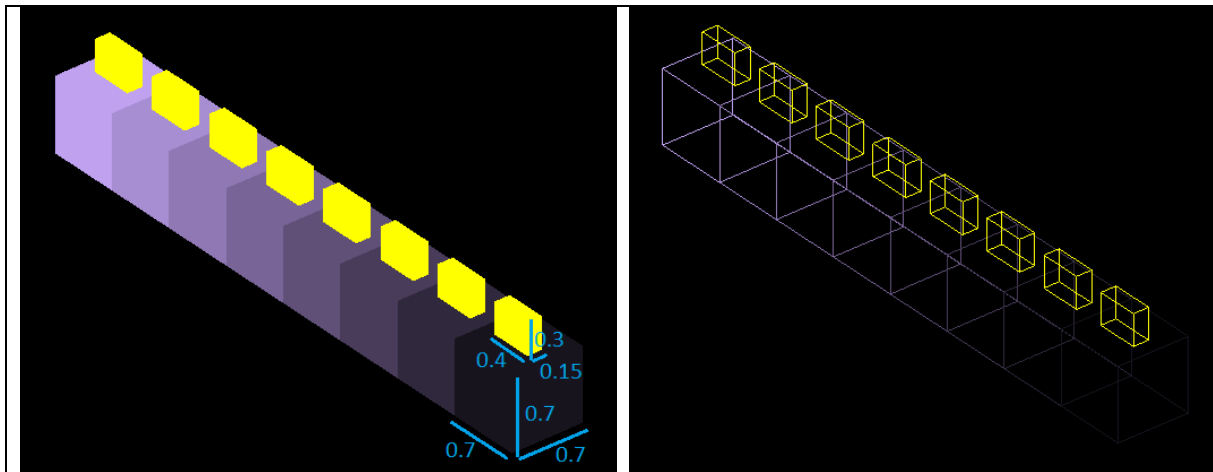
Pour le corps on effectue exactement le même travail que pour le cou en prenant cette fois ci $n=3$.



Corps du Dragon

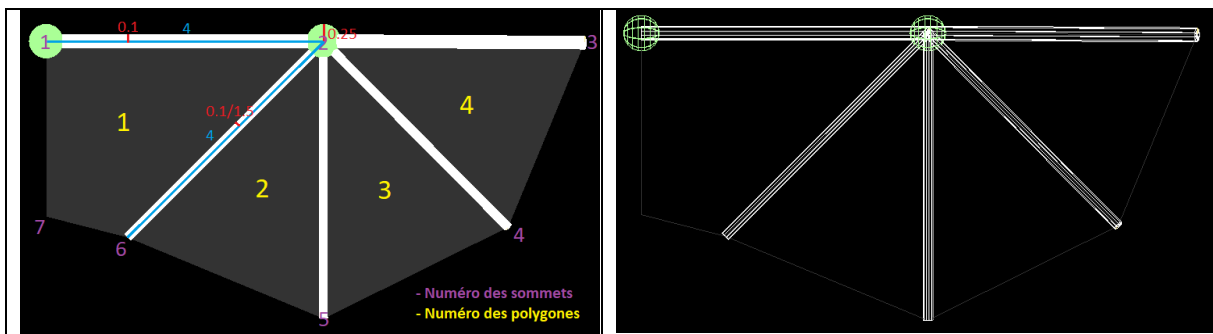
Pour la queue nous choisissons une manière plus technique. Cette dernière est composée de 8 petits cubes sur lesquels on retrouve 8 petites écailles. De cette manière on pourra l'animer de façon plus élégante.

PROJET SI 2021 : MODELISATION D'UN DRAGON



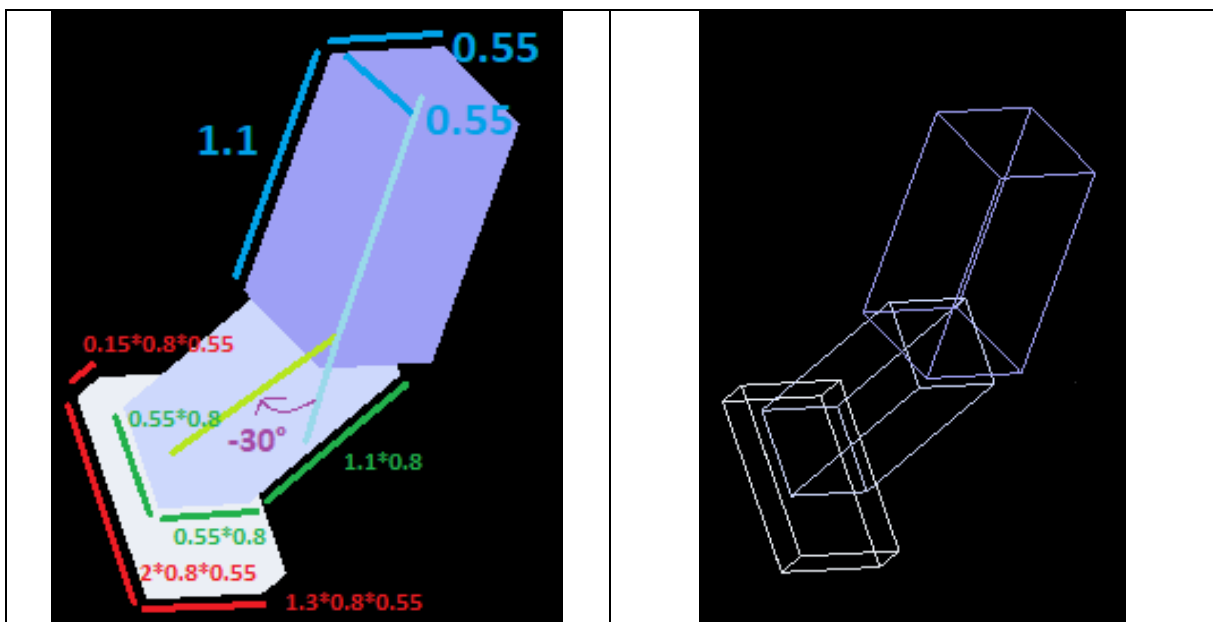
Queue du Dragon

On modélise ensuite les ailes. Pour ce faire on utilise des boules au niveau de articulations sur lesquels on fixe des cylindres qui représentent les os. On y joint ensuite des polygones pour représenter la peau. Enfin on effectue une rotation de 20° selon x des arête 2-3 2-4 et des parties 3 4.



Aile du Dragon

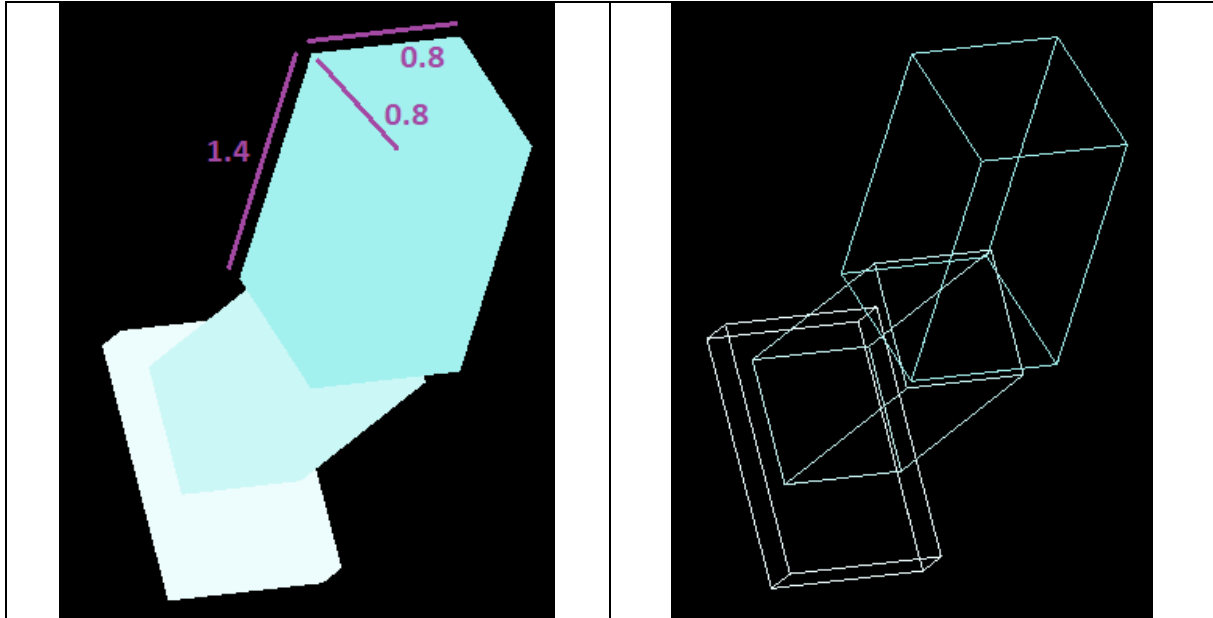
On continue avec les pattes avant. Nous n'utilisons que des cubes, mais nous rajoutons une rotation θ entre l'avant-bras et le bras.



Patte avant du Dragon

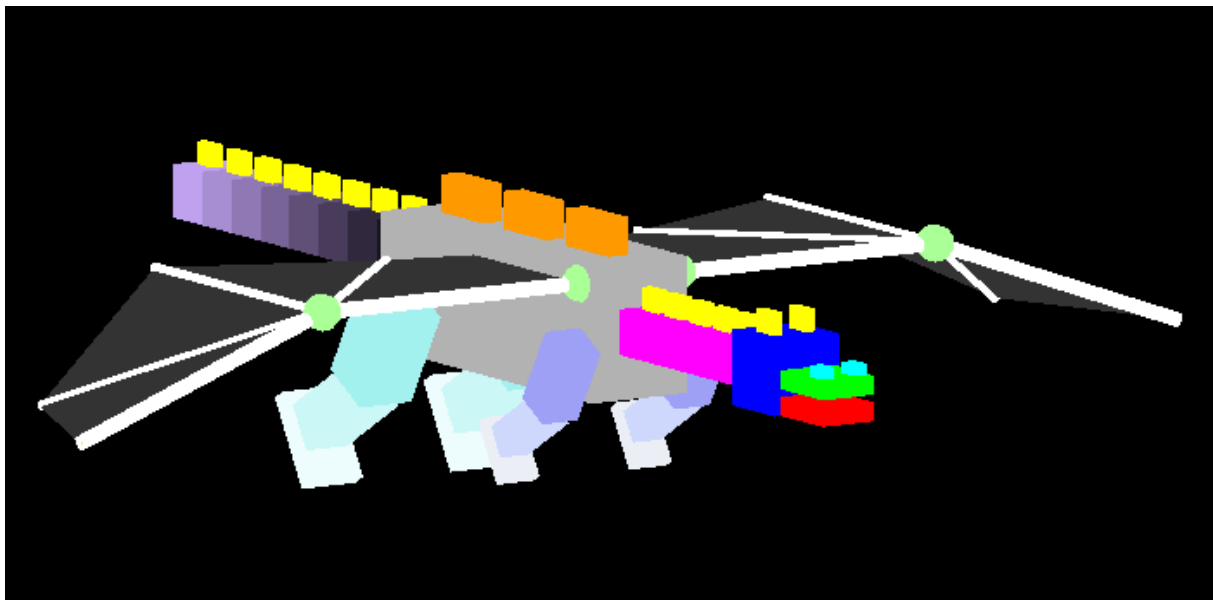
PROJET SI 2021 : MODELISATION D'UN DRAGON

Pour finir on modélise la patte arrière. Il s'agit de la même que la patte avant mais en plus grosse. Comme chaque élément est proportionnel au bras on ne note que les proportions de ce dernier.



Patte arrière du Dragon

Une fois assemblé on obtient donc le résultat suivant :



Dragon

II/ Les Animations

a) Animations clavier

Pour les animations claviers, en plus d'ajouter les fonctionnalités demandées, à savoir le mouvement de caméra avec les flèches directionnelles et le zoom/dézoom avec les touches 'z' et 'Z',

PROJET SI 2021 : MODELISATION D'UN DRAGON

on ajoute également un déplacement selon l'axe x avec les touches 'a' et 'A' (pour avancer), un déplacement selon l'axe y avec les touches 'h' et 'H' (pour haut) et un déplacement selon l'axe z avec les touches 'c' et 'C' (pour coté). Enfin on rajoute une rotation de la mâchoire avec les touches 'r' et 'R'.

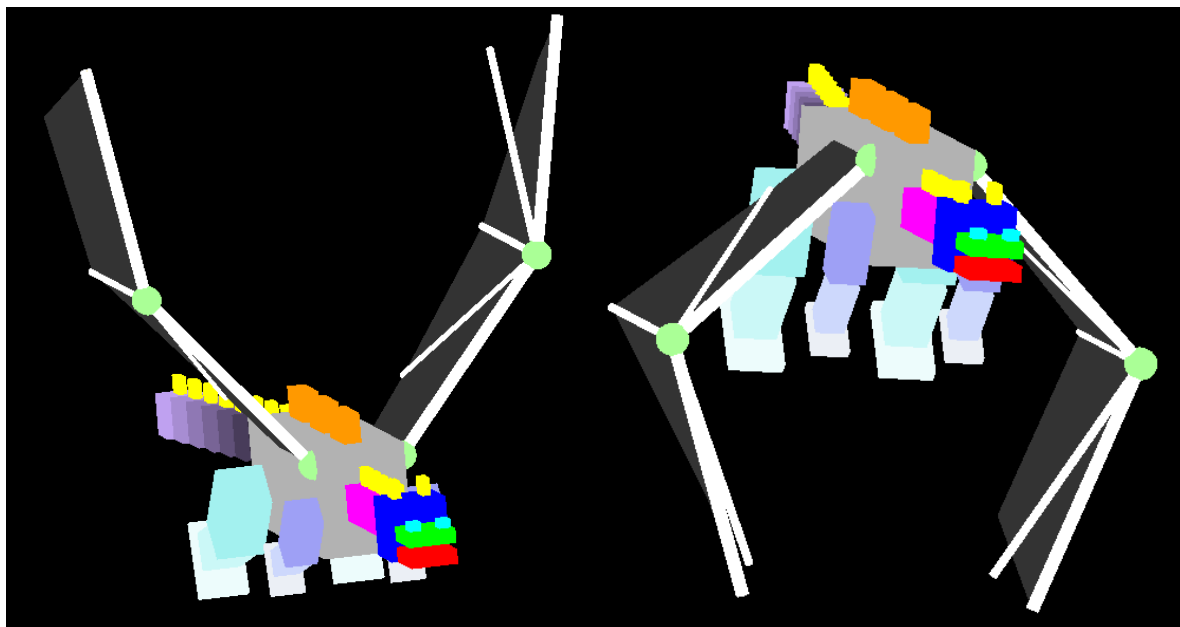
b) Animations automatiques

Tout comme les parties du corps du dragon on réalise une fonction animation qu'on appellera dans l'affichage pour animer le dragon. Cette dernière est « contrôlée » par la barre espace du clavier. En effet la barre espace permet de faire changer d'état un booléen « activateAnimation ». Une fois cette condition validée on peut lancer les animations.

Pour ce faire on fait varier deux variables dh et dr (pour hauteur et rotation). dh varie de -h à h (qui vaut 0.2 ici), et dr de -r à r (qui vaut 5 ici). Une fois que dh atteint une des valeurs extrêmes de l'intervalle on change d'état un booléen mouve (respectivement rot), ce qui a pour effet de faire incrémenter ou décrémenter dh (dr) de $\frac{h \text{ (ou } r)}{nb_mvt}$, avec nb_mvt = 80.

On décide donc d'animer l'entièreté du dragon en le faisant tourner de dr sur son axe x, puis en le faisant translater de 3*dh le long de y. Une fois ces animations basiques réalisées on s'attaque à une tâche plus complexe. Pour animer la queue on décale chaque petit cube de dh*i (avec i le numéro du cube). On anime également les pattes d'une rotation de -3*dr autour de z au niveau de la face supérieure des bras. Pour finir on anime les ailes sur les 2 boules, chacun ayant une rotation de -10*dr autour de x.

On obtient alors le résultat suivant :



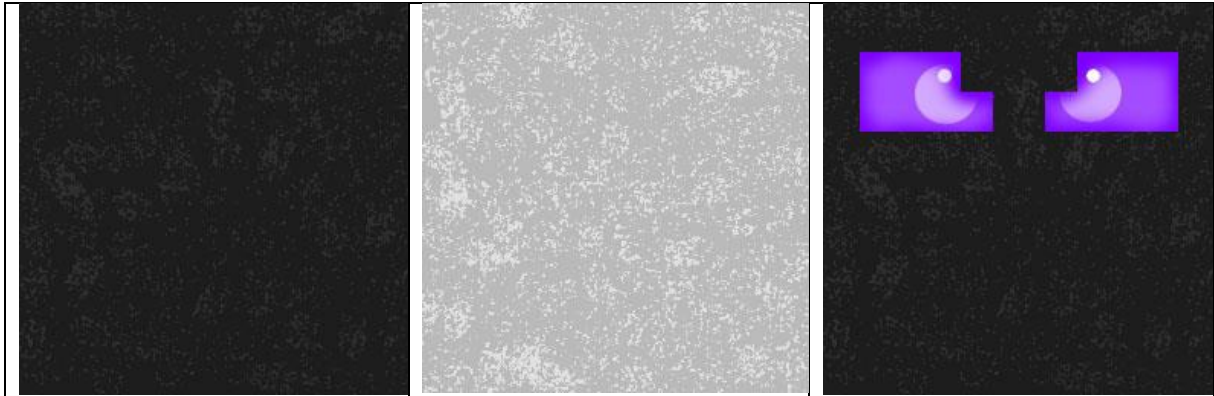
*Dragon animé**

(*) On remarque que l'arête entre le point 2 et 4 ne suit pas le même mouvement que le polygone. Je n'ai malheureusement pas réussi à corriger ce problème.

PROJET SI 2021 : MODELISATION D'UN DRAGON

III/ Les Textures

L'application des textures reste assez basique. Cependant nous devons recourir à une petite astuce pour texturer correctement la face du dragon. On commence par créer les 3 textures suivantes, toutes de taille 256*256.



Texture noire, grise et face

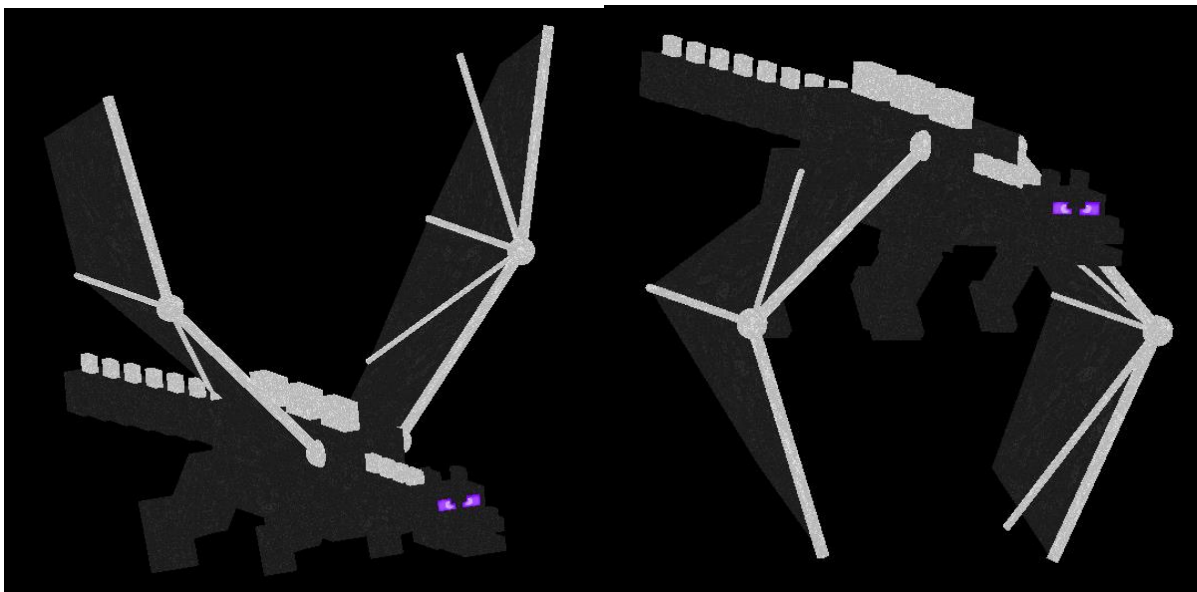
La texture noire servira pour la peau du dragon, tandis que la texture grise servira pour les os et les écailles. La face quant à elle ne servira qu'à texturer une face du cube représentant la tête.

On plaque les textures appropriées sur les cubes, mais on enroule celles sur les sphères et les cylindres (sauf sur les couvercles) à l'aide d'un jeu de coordonnées spécifié dans le code. Pour la face on crée un booléen global « face » qu'on passe à *true* lors de la création du cube représentant la tête. Dans la fonction *cube*, si *i* vaut 1 et *face=true* alors on change la texture utilisée, on tourne le cube de -90° autour de x, puis une fois le cube créé on fait l'action inverse. Ensuite on repasse *face* à *false*.

On charge donc ces 3 textures dans les tableaux *image1*, *image2* et *image3*, puis lorsque qu'on veut charger une texture spécifique on utilise la commande :

```
glTexImage2D(GL_TEXTURE_2D,0,GL_RGB,largimg,hautimg,0,GL_RGB,GL_UNSIGNED_BYTE,imagei);
```

On obtient alors le résultat suivant :



PROJET SI 2021 : MODELISATION D'UN DRAGON

IV/ Les Lumières

Avant de pouvoir ajouter de la lumière il nous faut calculer les normales des faces de nos primitives. Pour cela il nous suffit de prendre 3 points d'une face. On en tire 2 vecteurs dont on fait le produit vectoriel. On obtient alors un vecteur normal à la surface. On le divise alors par sa norme et on obtient le vecteur normal à la surface de norme 1.

Points : $A_1(a_1, b_1, c_1)$, $A_2(a_2, b_2, c_2)$, $A_3(a_3, b_3, c_3)$.

$$\text{Vecteur normal : } \vec{n} = \overrightarrow{A_1A_2} \wedge \overrightarrow{A_1A_3} = \begin{pmatrix} b_2c_3 - b_2c_1 - b_1c_3 - c_2b_3 + c_2b_1 + c_1b_3 = n_1 \\ c_2a_3 - c_2a_1 - c_1a_3 - a_2c_3 + a_2c_1 + a_1c_3 = n_2 \\ a_2b_3 - a_2b_1 - a_1b_3 - b_2a_3 + b_2a_1 + b_1a_3 = n_3 \end{pmatrix}$$

$$\text{Norme : } |\vec{n}| = \sqrt{n_1^2 + n_2^2 + n_3^2}$$

$$\text{Vecteur normal de norme 1 : } \vec{n} = \frac{\vec{n}}{|\vec{n}|}$$

Maintenant qu'on a notre vecteur normal nous pouvons définir les propriétés de nos matériaux et de nos lumières.

- Concernant les lumières on prend :

$$\text{Lumière 0 (directionnelle) : } \begin{bmatrix} \text{direction : } \{15.0, 5.0, 3.0, 1.0\} \\ \text{diffuse : } \{1.0, 1.0, 1.0, 1.0\} \\ \text{ambient : } \{1.0, 1.0, 1.0, 1.0\} \\ \text{speculaire : } \{1.0, 1.0, 1.0, 1.0\} \end{bmatrix}$$

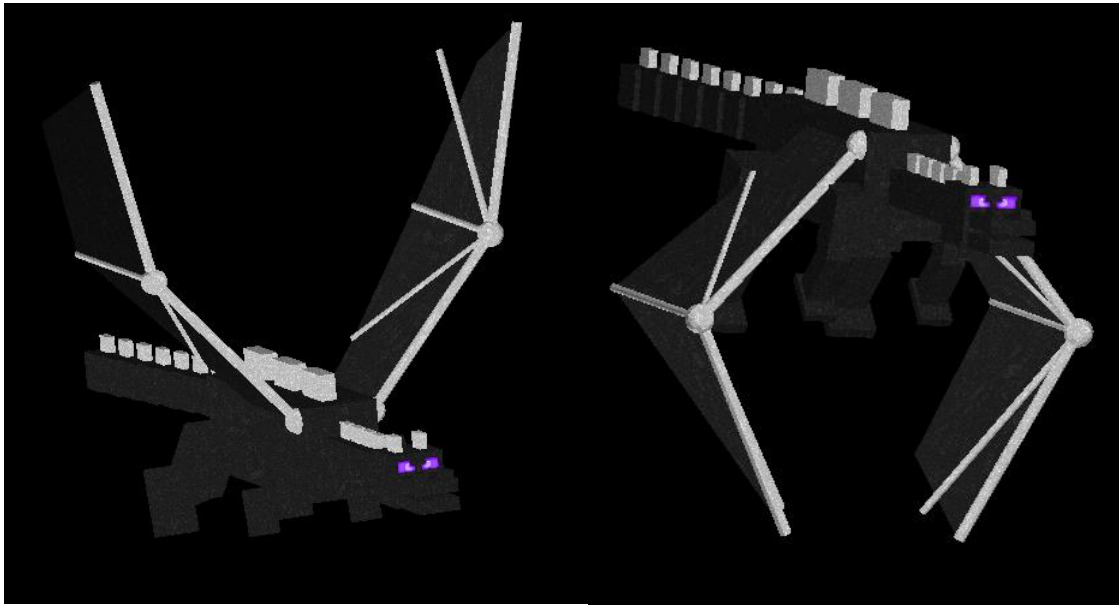
$$\text{Lumière 1 (cône) : } \begin{bmatrix} \text{direction : } \{5.0, -5.0, 0.0, 0.0\} \\ \text{diffuse : } \{1.0, 1.0, 1.0, 1.0\} \\ \text{ambient : } \{1.0, 1.0, 1.0, 1.0\} \\ \text{speculaire : } \{1.0, 1.0, 1.0, 1.0\} \\ \text{angle : } 60^\circ \\ \text{direction : } \{-1, 1, 0\} \\ \text{exposant : } 2.0 \end{bmatrix}$$

Le cône tourne autour du dragon en pointant sur le point (0,0,0).

- Concernant les matériaux on fait simple : On règle la lumière spéculaire à $\{1.0, 1.0, 1.0, 1.0\}$ et la brillance (shininess) à 50.

Une fois tous ces éléments réalisés on obtient le résultat final :

PROJET SI 2021 : MODELISATION D'UN DRAGON



Dragon final avec lumières

Conclusion

Nous avons donc réalisé le dragon du jeu vidéo « Minecraft », modélisation assez basique car cubique, en y rajoutant quelques petites difficultés comme des sphères ou des cylindres. Les animations appliquées ainsi que les textures et les lumières permettent un rendu assez proche du modèle original.

Certains points restent cependant imparfaits, comme l'arête 2-4 qui ne suit pas le même mouvement que la toile.