Rapport du projet concernant la modélisation d'un dragon en OpenGL.

Université de Bourgogne
UFR Sciences et techniques
Licence 3 - Informatique

Projet Synthèse d'image

Modélisation d'un dragon

Eddy DRUET – Clément GILI – Groupe TP 02

TABLE DES MATIERES

INTRODUCTION

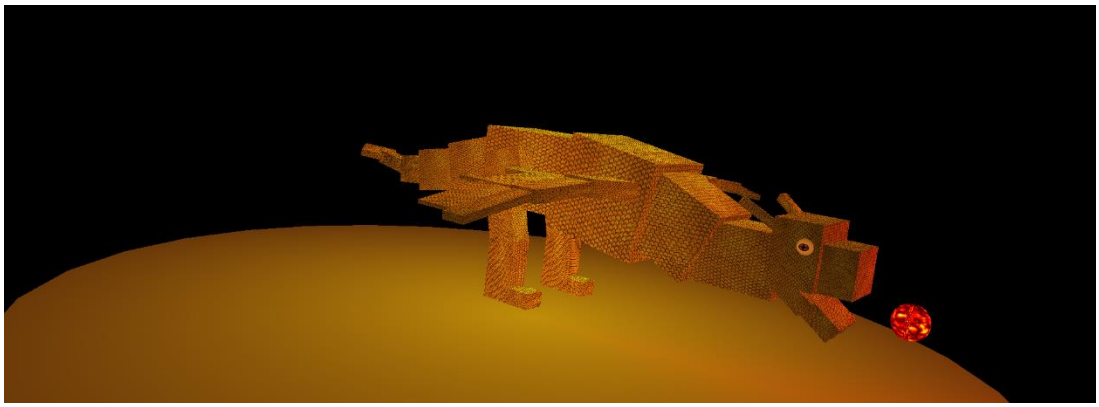

Figure 1 : Aperçu du dragon modélisé avec son animation de boule de feu.

CONTEXTE

Ce projet de synthèse d'image nous demandait de modéliser un dragon avec OpenGL. Nous avons réalisé toutes les tâches qui étaient demandées, que voici :

- Modélisation d'un dragon dans le style low-poly ;
- Modélisation d'une sphère à partir de sa représentation paramétrique ;
- Utilisation de trois textures :
  - Écailles du dragon plaquées sur les faces du corps du dragon ;
  - Œil enroulé autour des sphères constituant les yeux du dragon ;
  - Feu enroulé autour de la sphère constituant la boule de feu.
- Utilisation de deux types de lumières :
  - Un spot lumineux jaune éclairant le centre de la scène du haut vers le bas ;
  - Un point lumineux omnidirectionnel rouge se déplaçant avec la boule de feu.
- Contrôles pour zoomer et tourner la caméra à gauche, à droite, en haut et en bas ;
- Une animation manuelle permettant de tirer une boule de feu ;
- Une animation automatique concernant les ailes du dragon.

CONTROLES DISPONIBLES

Ci-dessous, la liste des contrôles disponibles pour gérer la caméra, activer des animations ou des options de débogage.

Contrôle	Description
Z / z / molette haut / molette bas	Zoomer / dézoomer la caméra.
Flèche gauche	Tourner autour du dragon par la droite.
Flèche droite	Tourner autour du dragon par la gauche.
Flèche haut	Tourner autour du dragon par le bas.
Flèche bas	Tourner autour du dragon par le haut.
Clique gauche + déplacer souris	Tourner autour du dragon.
Espace	Jouer l'animation de la boule de feu.

p	Afficher le mode plein.
f	Afficher le mode fil de fer.
s	Afficher le mode sommets seuls.
d / D	Activer / désactiver le test de profondeur.

Tableau 1 : Contrôles disponibles.

GESTION DE LA CAMERA

Le zoom de la caméra est effectué en modifiant l'angle FOV de la perspective de la caméra. La rotation de la caméra est effectuée en faisant une rotation de celle-ci autour de l'axe X et Y.

STRUCTURES DE DONNEES

Plusieurs structures de données ont été implémentées pour faciliter la création des formes, et les calculs.

POINT

La classe « Point » permet de stocker les 3 coordonnées d'un point dans l'espace. Elle est utilisée dans le calcul des formes primitives comme la sphère.

FACETTE

La classe « Facette » permet de stocker les coordonnées des 4 points constituant une facette. Cela permet de facilement dessiner les formes comme la sphère.

TTEXTURE

Une structure « TTexture » a été implémentée, celle-ci permet de stocker la taille et les pixels d'une image jpeg qui a été chargée. L'intérêt est de pouvoir charger plusieurs images et surtout de tailles différentes pour appliquer différentes textures à plusieurs objets.

FORMES PRIMITIVES

Le dragon étant low-poly, nous avons eu besoin que de deux primitives : le parallélépipède et la sphère. Ces formes sont dessinables en appelant les fonctions se trouvant dans le fichier « formes.h ».

Ces formes primitives n'ont pas de textures et se situent à l'origine de la scène. Libre à nous de les déplacer où l'on veut après dessin.

PARALLELEPIPEDE

Le parallélépipède peut être créé avec une longueur, une largeur et une hauteur précise. La création du parallélépipède est simple : un cube centré à l'origine de la scène et de taille 1 est dessiné, puis il est mis à l'échelle suivant la longueur, la largeur et la hauteur données.

Par ailleurs, nous avons généré les coordonnées de texture pour chacune des faces pour pouvoir appliquer des textures sur les parallélépipèdes par la suite.

Également, pour avoir une interaction avec la lumière cohérente, le vecteur normal de chaque face a été défini.

SPHERE

La sphère peut être créée avec un rayon précis. Cette forme est créée par sa représentation paramétrique que nous avons vu en TD.

D'abord, les coordonnées des points sont calculées en fonction du nombre de parallèles et de méridiens, puis les points constituant les facettes sont déterminés. Enfin, toutes les faces sont dessinées, en plus, de générer les coordonnées de textures.

Les coordonnées de texture de la sphère sont calculées de sorte à pouvoir appliquer des images de type « projection cylindrique centrale ».

MODELISATION DU DRAGON

De manière générale, les parties du dragon ont été modélisées en plaçant successivement des formes à l'aide de translation, et/ou de rotation.

Le dragon a été découpé en plusieurs parties : les pattes, le corps, les ailes, la tête et la boule de feu. Ces parties se trouvent dans des fichiers différents dans le répertoire « dragon » que l'on peut dessiner en appelant successivement leur fonction de dessin.

GESTION DES TEXTURES

Dans cette partie, nous verrons comment les textures ont été gérées, où nous les avons utilisés, et à quoi elles ressemblent. Trois textures au total ont été utilisées : deux enroulées autour d'une sphère, et une plaquée sur une face.

CHARGEMENT DES TEXTURES

Le chargement des images jpeg depuis le stockage est effectué avec la bibliothèque « jpeglib ». Il suffit de spécifier le chemin d'accès de l'image à la fonction « loadJpeg » dans le fichier « utils.h ». Cette fonction retourne une structure « TTexture » contenant les pixels de l'image que l'on peut ensuite donner à OpenGL.

À l'initialisation du programme, les trois textures sont chargées avec OpenGL pour pouvoir être utilisé ultérieurement lors de la modélisation du dragon.

UTILISATION DES TEXTURES

Le choix de la texture à appliquer est faisable en appelant la fonction OpenGL « glBindTexture » avec l'identifiant de la texture à utiliser.

Cependant, pour que cela soit plus simple à utiliser, nous avons créé une fonction « setTexture » où il suffit de spécifier l'identifiant de la texture, et une autre fonction « clearTexture » pour ne pas utiliser de texture. Ces deux fonctions se trouvent dans le fichier « textures.h ».

TEXTURE DES ECAILLES DU DRAGON (PLAQUEE)

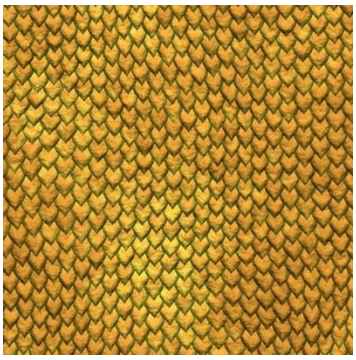

Figure 2 : Texture d'écailles plaquée sur l'ensemble des faces du corps du dragon (pattes, corps, tête, ailes).

TEXTURE DES YEUX DU DRAGON (ENROULEE)

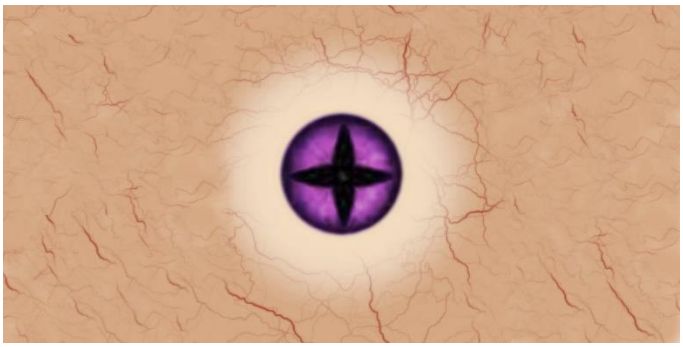

Figure 3 : Texture d'œil enroulée autour des deux yeux du dragon.

TEXTURE DE LA BOULE DE FEU (ENROULEE)

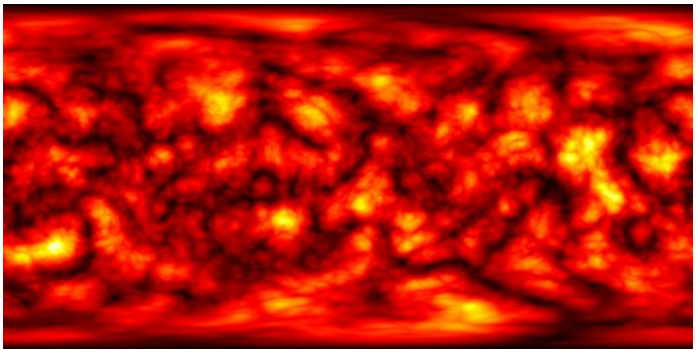

Figure 4 : Texture de feu enroulée autour de la boule de feu.

MATERIAUX

Nous avons voulu modéliser un dragon de feu, donc avec une ambiance très jaune, orange. Nous avons donc défini les propriétés des matériaux pour qu'ils diffusent une lumière jaune, avec une faible lumière d'ambiance blanche. Aucune lumière d'ambiance globale n'a été utilisée.

GESTION DES LUMIERES

Dans cette partie, nous verrons comment nous avons configuré les lumières. Nous avons utilisé deux types de lumières : un spot lumineux éclairant le centre de la scène du haut vers le bas, et un point lumineux omnidirectionnel se déplaçant avec la boule de feu.

SPOT LUMINEUX

Le spot lumineux a été positionné à l'origine de la scène, en hauteur (Y = 5). La source de lumière a été configuré en mode directionnel avec une direction pointant à la verticale en (0, -1, 0) donc de haut en bas depuis la source.

L'angle du faisceau de lumière est de 90°, celui-ci est suffisamment large pour éclairer l'entièreté du dragon. De même, le coefficient d'atténuation de la lumière aux bords du faisceau lumineux est de 5.

Le spot diffuse et a une ambiance jaune pour donner cet aspect « dragon de feu ».

BOULE DE FEU (POINT LUMINEUX OMNIDIRECTIONNEL)

Étant donné que c'est une boule de feu, il fallait que celle-ci diffuse une vive lumière rouge tout autour d'elle, et qu'elle se déplace en même que la boule de feu se déplace.

Nous avons donc utilisé une source de lumière omnidirectionnelle avec une position qui est translatée en même que celle de la boule de feu. La lumière se situe au même endroit que la boule de feu. La couleur de diffusion est rouge.

Enfin, par défaut, l'atténuation de la lumière sur le dragon plus la boule de feu s'éloignait était trop faible, nous avons donc configuré les atténuations constantes (0), linéaires (0,4) et quadratiques (0,4) à tâtons, jusqu'à que le résultat rende bien.

Cependant, il y avait un dernier problème, la boule de feu n'était pas assez éclairée malgré la lumière. Il fallait qu'elle soit totalement lumineuse, nous avons donc désactivé la lumière sur la boule de feu avec « GL_LIGHTING ». Elle apparaît ainsi entièrement éclairée.

GESTION DES ANIMATIONS

Dans cette partie, nous verrons quelles sont les animations créées, et comment elles ont été conçues. Nous avons trois animations : les ailes qui battent (automatique), l'ouverture de la gueule du dragon suivi du lancement d'une boule de feu (manuelle).

ARCHITECTURE DES ANIMATIONS

Les animations sont gérées dans le fichier de la partie du dragon concerné. Les états courants de l'animation y sont stockés. Les états peuvent être la phase actuelle de l'animation, un angle de rotation, une position actuelle, etc.

Voici les fonctions implémentées pour gérer les animations :

Fonction	Description
playAnimation	Jouer l'animation de la partie concernée, les états de l'animation sont initialisés.
stopAnimation	Arrêter l'animation de la partie concernée, les états de l'animation sont réinitialisés.
toggleAnimation	Jouer ou arrêter l'animation de la partie concernée.
tickAnimation	Cette fonction est appelée à chaque temps d'inactivité du programme (idle), elle fait progresser les états de l'animation (angle, position, phase, etc.).

Tableau 2 : Fonctions permettant l'animation des parties.

Ainsi, dans la fonction de dessin de la partie du dragon, les valeurs des états de l'animation sont utilisées pour transcrire l'animation à l'écran.

BATTEMENT DES AILES

Les ailes effectuent un mouvement continu et automatique de battement d'un angle entre -20° et 20° sur l'axe Z.

L'animation est constituée de deux phases : la phase basse, lorsque les ailes sont au plus basses ; et la phase haute, lorsque les ailes sont au plus haut de leur battement. L'état stocké est l'angle courant des ailes qui est par défaut 0°.

L'angle de rotation sur l'axe Z est incrémenté ou décrémenté de 0,5 à chaque « tick ». Lorsque l'angle atteint 20°, la phase basse débute, tandis que la phase haute cesse. Lorsque l'angle atteint -20°, c'est l'inverse.

BOULE DE FEU

Cette animation est déclenchée lorsque la phase d'ouverture de l'animation de la gueule du dragon est terminée (cf. partie sur l'animation de la gueule du dragon, ci-dessous). TODO

L'animation de la boule de feu ne possède pas de phase, mais stocke deux états : la position actuelle de la boule de feu, et le « scale » actuel de cette dernière, les deux initialisés à 0 par défaut.

Le déroulement de l'animation est celui-ci :

1. La boule de feu est invisible, car son scale est à 0 ;
2. La boule de feu s'agrandit jusqu'à atteindre sa taille maximale de 0,25 ;

3. La boule de feu avance vers l'axe Z, et descend légèrement sur l'axe Y ;
4. Atteint une position Z = 5, l'animation est arrêtée.

À chaque « tick », le scale est augmenté de 0.01, la position Z avance de 0.008 et la position Y diminue de 0.0003.

GUEULE DU DRAGON

Cette animation manuelle est jouée par l'appui de la touche « Espace ». Elle précède l'animation de la boule de feu.

Il y a trois phases pour cette animation : la phase d'ouverture de la gueule, la phase où la boule de feu apparaît et son animation est déclenchée, et enfin la phase où la gueule se referme. Un état est stocké, l'angle courant d'ouverture de la gueule par défaut à 0°.

Pendant la phase d'ouverture de la gueule, cette dernière effectue une rotation sur l'axe X d'un angle qui est incrémenté de 0.05 à chaque « tick ». Lorsque l'angle d'ouverture atteint 40°, la phase de la boule de feu s'enclenche et la phase d'ouverture de la gueule cesse.

La phase de la boule de feu démarre l'animation de la boule de feu, et passe immédiatement à la phase de fermeture de la gueule.

Similairement, à la phase d'ouverture, la phase de fermeture de la gueule diminue l'angle de 0.05, jusqu'à atteindre 0° pour que l'animation de la gueule s'arrête.

REPARTITION DU TRAVAIL

Nous nous sommes divisé les tâches en deux, l'un s'est occupé de faire la forme parallélépipède, ainsi que le corps, les pattes et les ailes (texturage et animation compris), tandis que l'autre s'est occupé de faire la forme sphère, la tête, la boule de feu et les lumières.

Nous avons effectué tous les deux le fichier « main.cpp », donc l'initialisation et la configuration d'OpenGL.

DOCUMENTATION

Une documentation Doxygen est disponible dans le répertoire « docs », fichier « index.html » pour avoir un aperçu global du projet concernant les fonctions, fichiers, classes et structures.

IMAGES

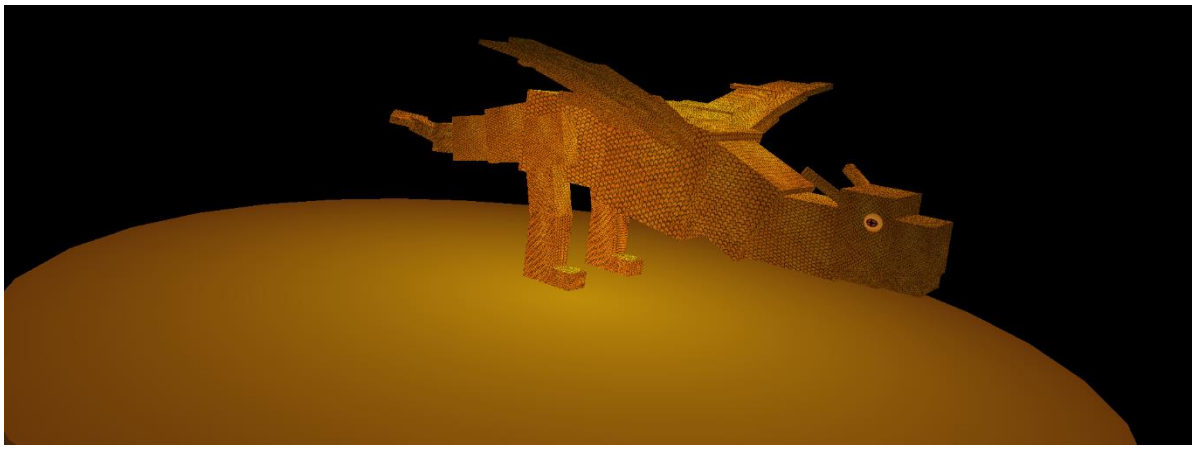

Figure 5 : Vue d'ensemble du dragon.

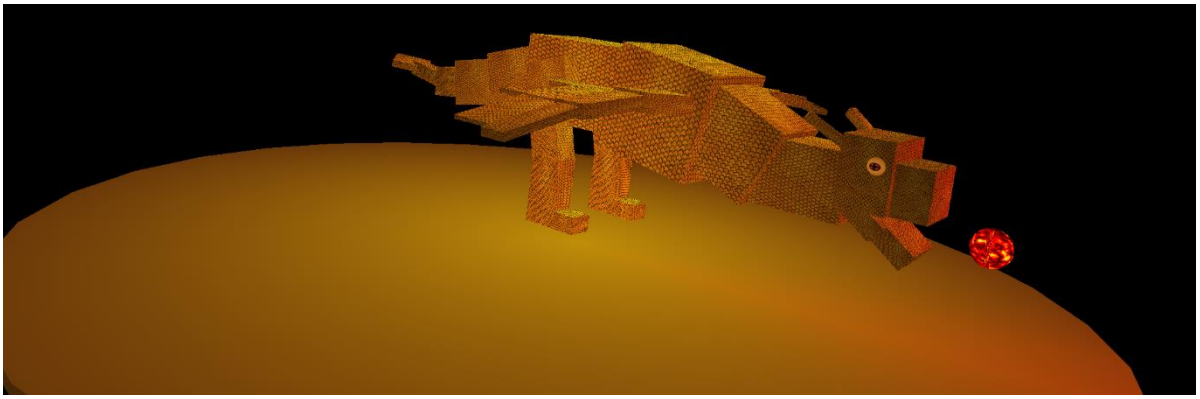

Figure 6 : Vue d'ensemble du dragon avec animation de la boule de feu.

Figure 7 : Vue du côté gauche du dragon.

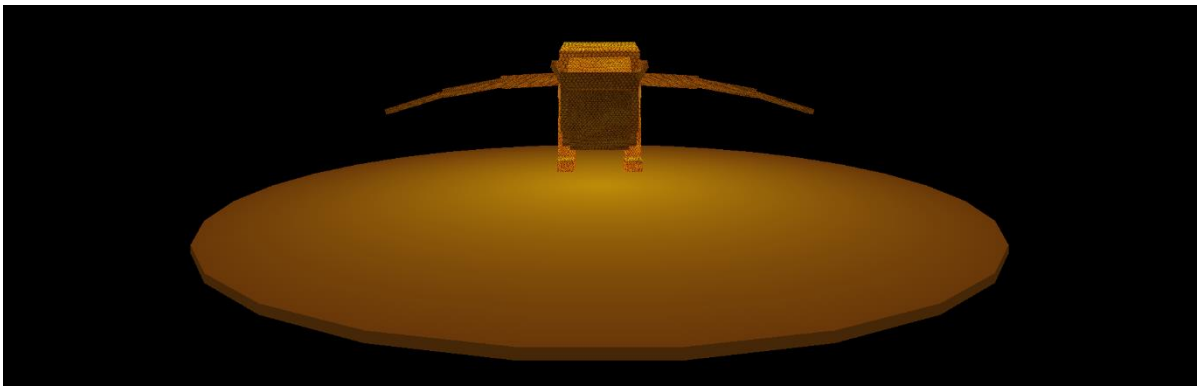

Figure 8 : Vue de face du dragon.

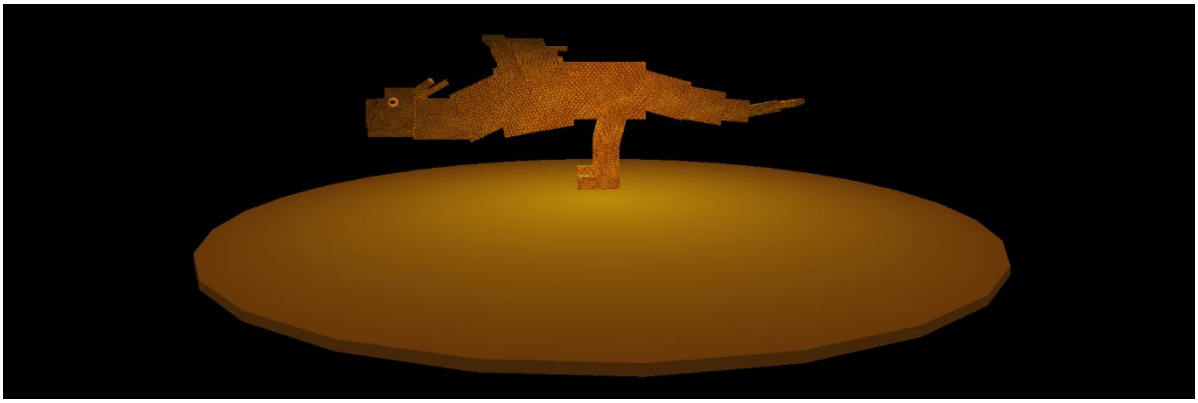

Figure 9 : Vue du côté droit du dragon.

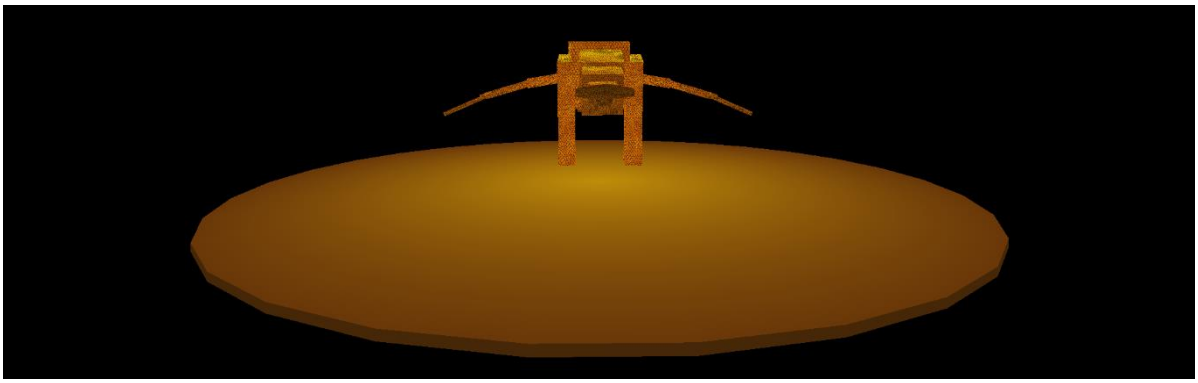

Figure 10 : Vue arrière du dragon.

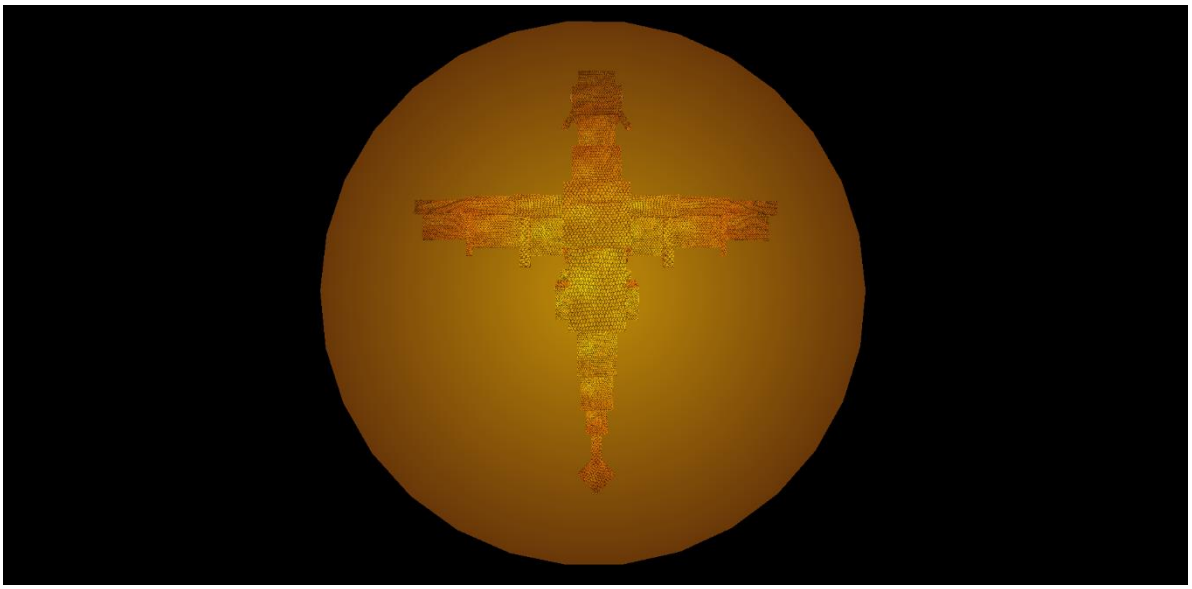

Figure 11 : Vue du dessus du dragon.