

Rapport de projet



UNIVERSITÉ DE NANTES

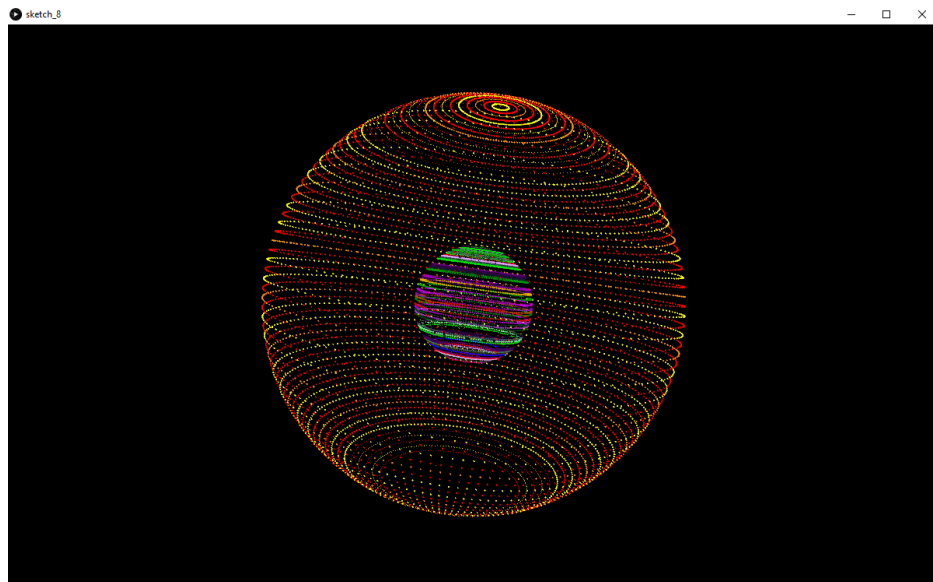
Introduction :

Ce document a pour but de décrire le déroulement de notre projet, depuis la première conception, jusqu'à la réalisation finale.

Nous avons choisi de réaliser une forme géométrique interactive et attrayante par sa couleur. Après réflexions, en se renseignant d'avantage sur internet et en prenant en compte ce que nous avons appris à faire, nous sommes partis sur l'idée de créer une Sphère Coloré avec un noyau.

Réalisation :

Dans un premier temps, Nous avons vu qu'il était possible d'utiliser la librairie peasy que l'on a importée afin d'utiliser la caméra PeasyCam ce qui nous permet de visualiser notre sphère.



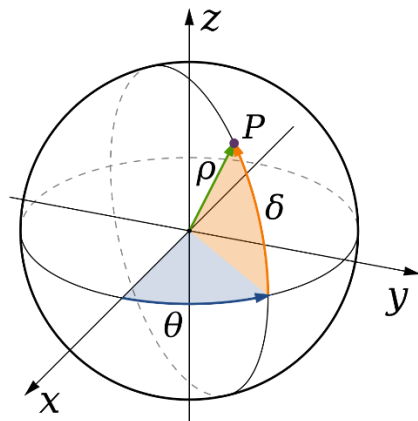
Cette caméra permet une certaine interactivité avec la souris afin de visualiser la sphère depuis différents angles et on peut zoomer ou dézoomer avec la molette ou le clic droit.

Nous avons donc commencé par chercher à représenter une sphère et avons trouvé que l'on pouvait passer des coordonnées cartésiennes aux coordonnées sphérique en utilisant les formules :

$$\begin{cases} x &= \rho \cos \delta \cos \theta \\ y &= \rho \cos \delta \sin \theta \\ z &= \rho \sin \delta \end{cases}$$

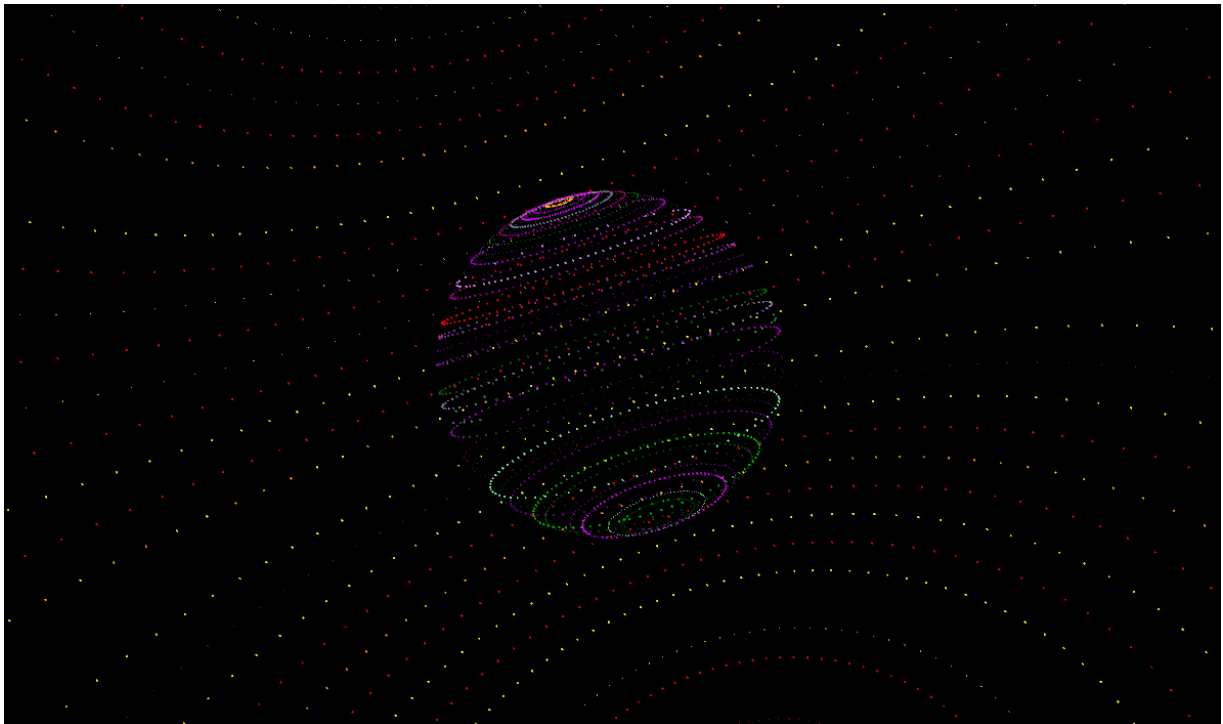
- ρ désigne la distance du point au centre du repère ;
- θ désigne la longitude, mesurée depuis l'axe des x généralement entre -180° et 180° ($-\pi \leq \theta \leq \pi$)
- δ désigne la latitude, l'angle depuis le plan équatorial, entre -90° et 90° ($-\frac{\pi}{2} \leq \delta \leq \frac{\pi}{2}$).

Le schéma ci dessous permet de mieux comprendre la relation entre des coordonnées cartésiennes et leur représentation en coordonnées sphérique :



Nous avons donc écrit une fonction drawSphere ainsi qu'une fonction drawSketch utilisant drawSphere permettant de donc de dessiner une sphère avec un nombre de points souhaité.

Pour la sphère nous avons décidé de fixer 3 couleurs (Rouge, Orange et Jaune) en fonction de la latitude, et pour son noyau nous avons décidé d'utiliser plus de couleurs elles aussi en fonction de la latitude mais aussi grâce à la fonction frameCount en fonction du nombre d'image affiché à l'écran depuis le début de l'affichage, ce qui permet d'avoir une variation de couleur en fonction du temps. Pour ce faire nous avons écrit une fonction « couleurs » qui permet de gérer les couleurs des points du noyau, ainsi qu'une fonctions « couleursFix » pour la gestion des couleur de la sphère extérieur.



Enfin, dans la fonction draw, nous avons simplement fixé les valeurs des variables :

- « *totale* » qui représente le nombre de points
- « *r1* » qui correspond au rayon de la sphère
- « *r2* » correspondant au rayon du noyau.

Conclusion :

Pour conclure, nous aurions aimé passer plus de temps sur ce projet, afin d'ajouter plusieurs fonctionnalités d'interactivité comme une touche permettant d'augmenter ou diminuer le nombre de points, une autre permettant d'influencer les couleurs ou encore une autre permettant de changer de forme géométrique.