Compte Rendu Projet de Webgl

Participants : Schill, Pesrotel, Ikorivyiza

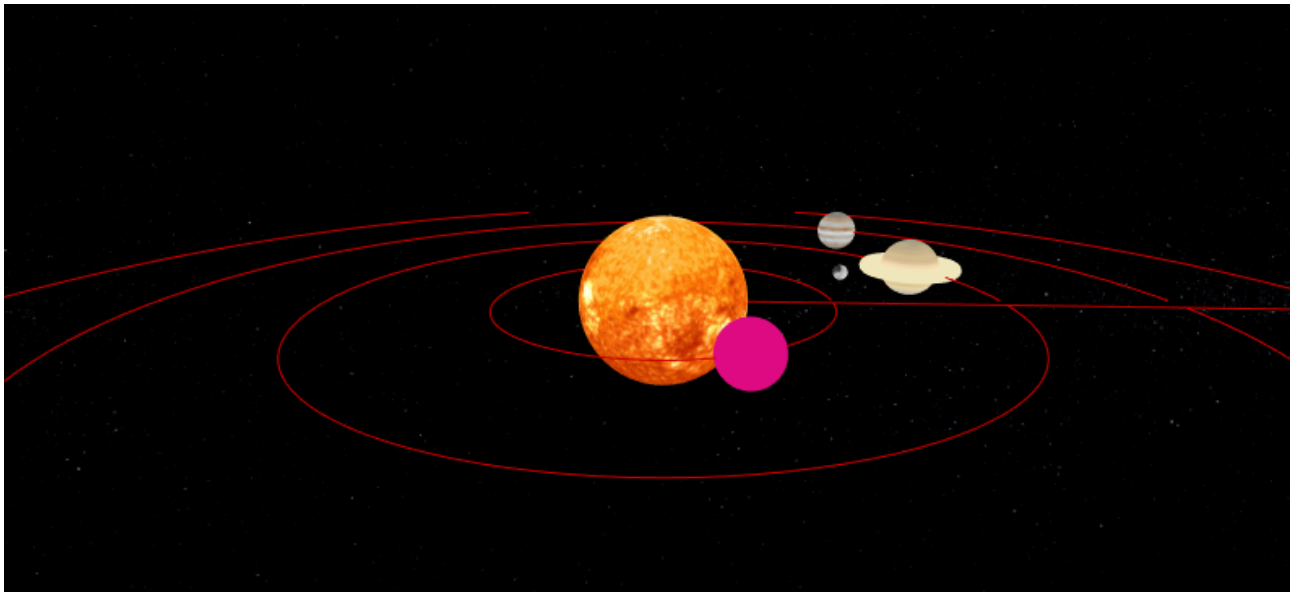
I-Découpage

On peut découper le projet en plusieurs parties :

-une partie lumière et toutes les fonctions usuelles du render.
-une partie objet (planètes, lunes…), leur texture, fonction de contrôle.
-une partie shader et gui.

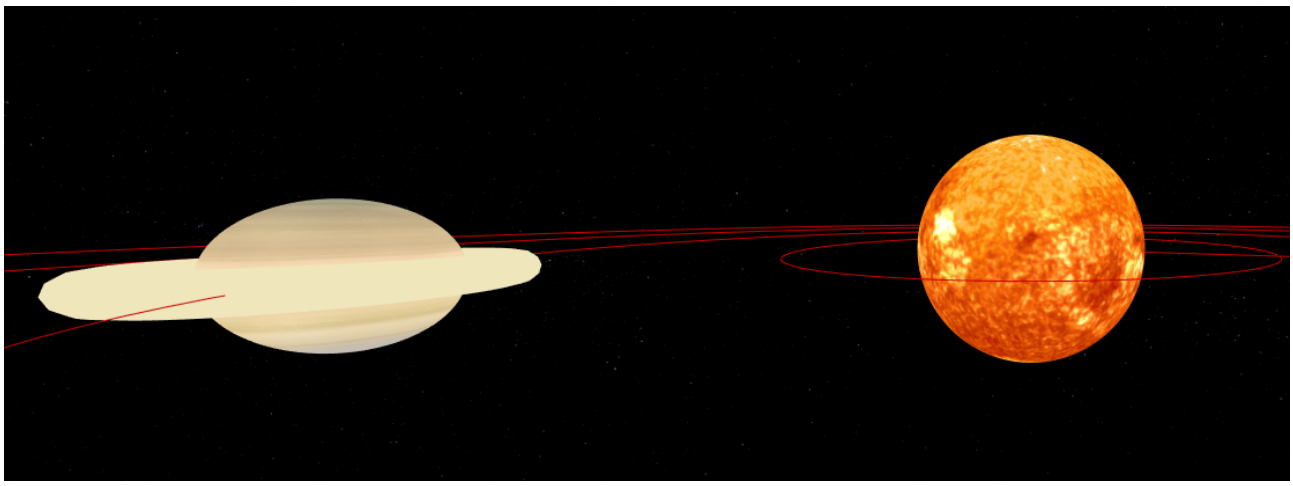
II-Lumières et render

Tout d'abord, le render est fait des fonctions usuelles (petite particularité pour le fond de la scène qui est revêtit d'une texture de ciel étoilé pour rendre un effet plus réaliste à la scène). Nous avons ajouté plusieurs fonctionnalité comme l'OrbitControl qui nous permet de pouvoir naviguer au bon vouloir dans la scène. En ce qui concerne la caméra, la seule chose à noter est la distance qui est de 1500 pour que l'on puisse voir la scène d'assez loin car l'orbite des planètes élargie de manière significative la scène, nous reviendrons sur la caméra dans une prochaine partie (gui). Pour le choix de la lumière nous avons choisie une hemisphereLight qui s'approche le plus à une lumière pouvant simuler notre soleil.



III-Objets, Textures et Contrôles

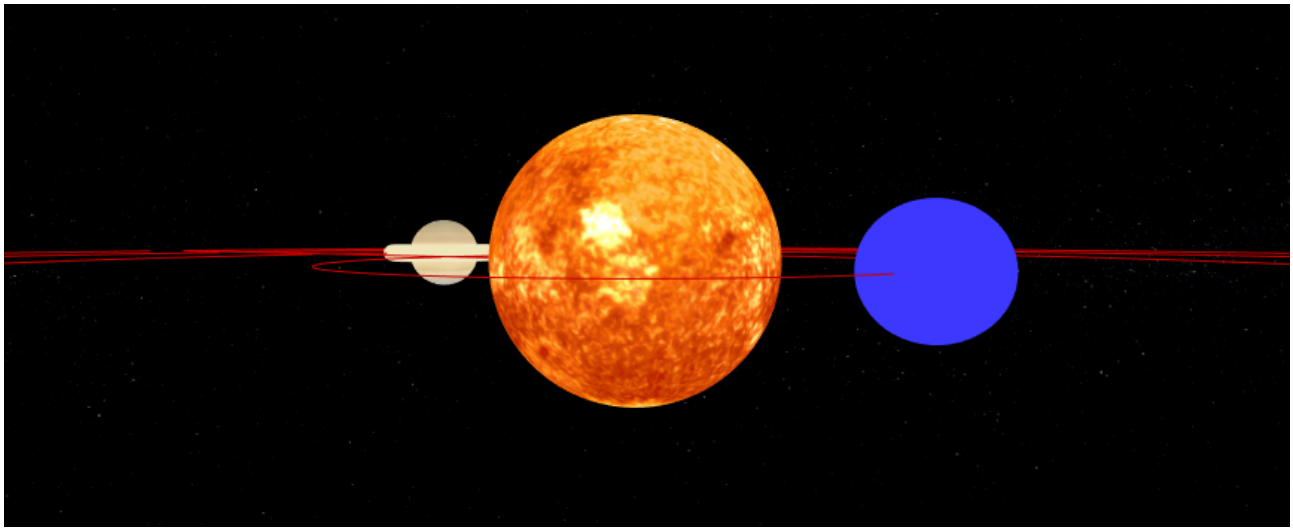
Nous abordons la partie la plus fournie, pour simuler un système solaire nous avons créé 7 sphères (1 soleil, 4 planètes, 2 lunes), 4 cercles pour simuler la trajectoire des orbites des planètes, 1 Tor pour simuler les anneaux d'une des planètes et de nombreux groupes/Pivots pour les rotations de nos objets. Pour chacune des planètes nous avons utilisés des groupes pour pouvoir les lier au soleil, tout en étant indépendant des autres planètes pour modifier leurs vitesses de rotations. Un problème survint au moment de créer les lunes, un oubli de modifier la position du centre de rotation des lunes, c'est à ce moment que passer par des pivots nous a paru moins brouillon et moins gourmand que de créer des groupes dans des groupes. Ainsi de cette manière nous avons implémenté les lunes et l'anneau de Saturne. Nous avons une planète avec des textures faites par des shaders, le reste sont des textures de type objectLoader prise avec une base de texture du système solaire libre de droit.



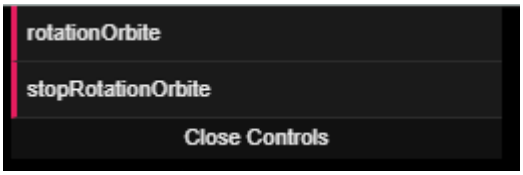
Pour plus de réalisme les différentes planètes ont une vitesse de rotation autour du soleil différente, de mêmes pour les lunes et les rotations des planètes sur elle même. Dans cette partie nous pouvons inclure les cercles qui simulent la trajectoire des planètes, faites à base de cercles elles prennent la distance entre le soleil et la planète concernée (rayon de rotation) et ont pour texture et mesh le type «Line» qui nous permet d'afficher le cercle vide en laissant que le périmètre de celui-ci.

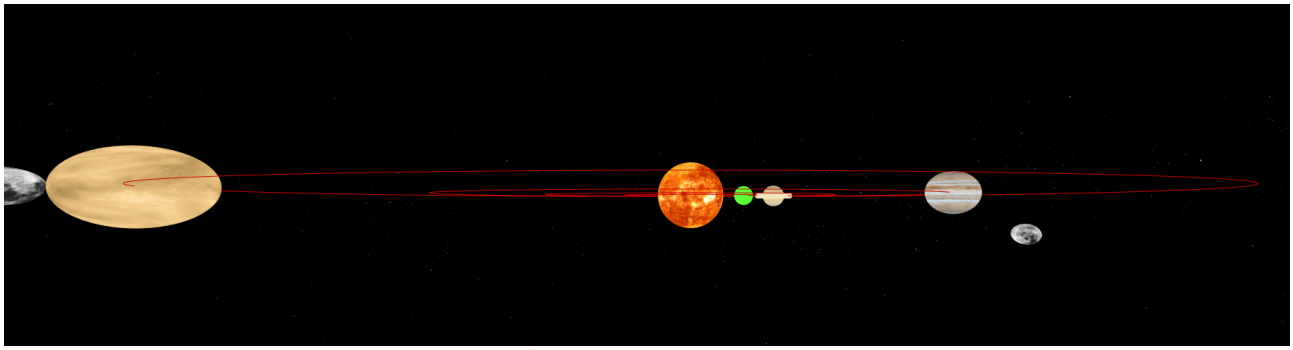
IV-Shader et GUI

Inspiré du tp1 sur shadertoy et combiné au vertex shader du cours nous avons implémenté la planète Mercure qui semble danser tout en changeant de couleur grâce à une variable uniforme passé en paramètre qui prend le temps réel pour être modifié dans nos fonctions. Nous récupérons aussi la position des sommets grâce à une variable de type « varying » pour pouvoir la passer dans le fragmentShader pour simuler les uv comme en shadertoy.



En ce qui concerne le GUI, il est utilisé pour donner un effet en mouvement de la caméra. Le GUI est là pour activer le mouvement de caméra ou le désactiver. Cette vue nous permet de voir toutes les planètes en mouvement à différentes distances.





V-Conclusion

En conclusion, le projet nous a laissé un grand choix de possibilité pouvant nous laisser porter par notre créativité mais la partie Shader nous limite de par sa difficulté pour simuler des choses réaliste du fait que l'on manipule des fonctions mathématiques qui rend les choses plus abstraites.

VI-Source

texture planètes : https://www.solarsystemscope.com/textures/
shader : http://www.unilim.fr/pages_perso/guillaume.gilet/Enseignement/InfoGraphique/WebGL/PipelineWebGL_2020.html
structure de nos objets : https://threejs.org/docs/