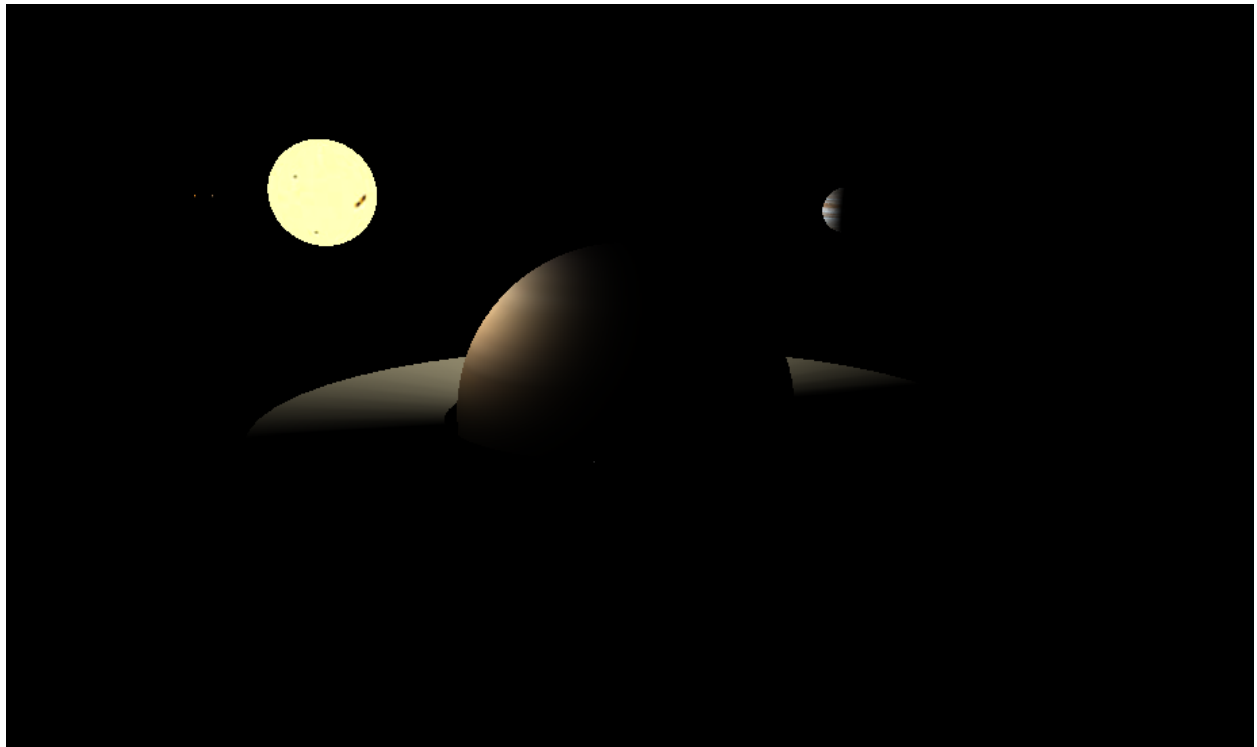


SOLAR SIS-TEM Visualizer

Rapport de Projet



[Introduction](#)

[Architecture](#)

[Fonctionnement](#)

[Résultats et gestion](#)

[Perspective d'améliorations](#)

Introduction

Ce projet dont l'objectif était de générer une représentation fidèle du système solaire a été réalisé en un mois. Durant cette période nous avons commencé par reprendre le TP sur le multitexturing comme base. Par la suite une longue partie de refactorisation du code a commencée de telle sorte à pouvoir facilement créer de nouveaux astres. Pour chaque amélioration une branche a été créée par son auteur et une revue du code à été faite par l'autre membre du binôme. Voici le lien GIT ou vous pourrez retrouver l'historique des branches et commits utilisées : <https://gitlab.com/ofghanirre/solarsistem>

(Comme Antonin n'avais pas de solution pour coder (macOs) je lui ai prêté un accès distant à mon serveur Linux ce qui fait que certaine des amélioration qu'il a réalisés sont sur des commit à mon nom)

Architecture

Notre objectif principal lors du développement de l'application a été de pouvoir créer et dessiner simplement les différents astres du système solaire. Pour cela nous avons créé une interface pour les astres implantés par de nouvelles interfaces : planète, planète avec anneaux et ainsi que par la classe du soleil. Les trois interfaces se trouvent dans "shared/stellarObject.hpp". Les différentes implémentations de ces interfaces se trouvent dans le répertoire **stellarObjects** avec les données de la Nasa écrites en dur dans le fichier include de ce même répertoire. Par la suite nous avons approfondi la gestion des événement qui se fait dans le fichier "shared/manageEvent.hpp", nous allons vous détailler les différents événements traités dans la partie suivante. Nous avons aussi ajouté dans la bibliothèque glimac les classes pour renvoyer les vertex d'un cercle, d'une ellipse, d'un anneau. Nous avons regroupés toutes ces formes avec la classe Sphère via une interface nommée "Shape". De plus chaque stellarObject contient la liste de ses satellites ainsi le soleil contient toute les planètes qui contiennent elles mêmes leurs satellites. Ainsi l'affichage se fait récursivement en appelant la fonction drawAll du soleil.

flèche haut et bas : zoomer et dézoomer

mollette : même chose que les flèche haut et bas

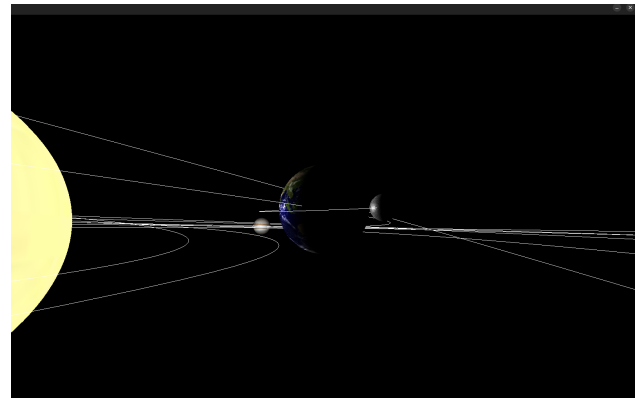
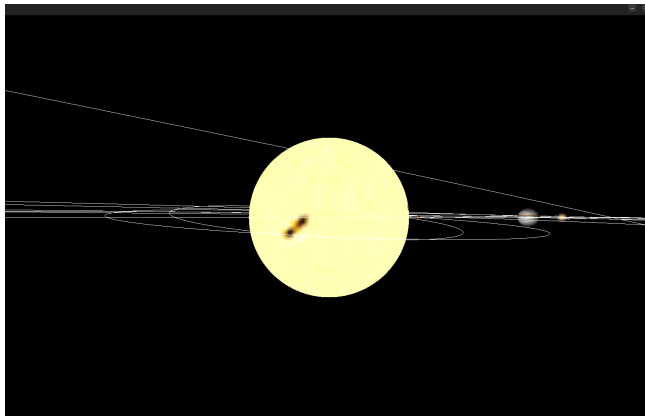
sourie : en maintenant appuyé le clique droit l'on peut tourner autour de l'astre que l'on regarde

Résultats et gestion

L'entièreté des éléments de base ont été implémentés. Mis à part la skyBox pour laquelle la texture de cube ne s'affiche pas, toutes les autres fonctionnalités fonctionnent parfaitement. En plus, nous avons ajouté le système de lumière au projet permettant un rendu plutôt réaliste malgré que les ombres ne soient pas affichées.

Le soleil a également un shader de lumière permettant un rayonnement périodique, nous avons jugé la fonctionnalité divertissante à faire et ajoutant un côté vivant à l'étoile.

En voici quelque exemple :



Perspective d'améliorations

Au vu du rendu final nous tenons à exprimer nos aspirations pour une potentielle suite sur ce programme. En effet nous avons comme ambition de réaliser la ceinture d'astéroïde présente dans le système solaire sous deux méthodes:

La première étant un anneaux texturé, permettant un affichage convenable de loin

La seconde étant une génération procédurale d'objets pseudo rond de taille aléatoire permettant un affichage dynamique pour une visualisation rapprochée de la ceinture.

Enfin nous avons pour ambition de rajouter une texture à la terre, permettant d'afficher les principales lumières artificielles dans la face sombre de la terre de nuit



Source : <https://www.cartesfrance.fr/geographie/cartes-satellite/carte-france-nuit.html>

Image référence de l'affichage souhaité.