

Le système solaire, c'est incroyable!

Loïc Huang

novembre - décembre 2024



Le vrai système solaire

1 Description de l'application développée

L'application développée est une modélisation du système solaire. Le système solaire est composée de planètes (telluriques ou gazeuses), d'étoiles et d'objets spatiaux (comètes, astéroïde, météoroides, ...). L'objectif principal de cette application est de simuler le mouvement et les interactions de ces objets dans l'espace en tenant compte de divers propriétés physiques tels que la masse, la vitesse, la distance au soleil ou encore la composition des objets. En utilisant des concepts de la mécanique céleste et des lois de la gravitation, les objets spatiaux peuvent se déplacer, interagir entre eux et réagir à différents événements comme l'entrée dans l'atmosphère d'une planète ou le passage proche du soleil.

Les planètes orbitent autour du Soleil et reçoivent de l'énergie. La gravité est aussi spécifique à chaque planète. Parmi elles, y a les planètes telluriques avec une composition physique (et liquide pour la Terre), avec une température qui varie. Et les planètes gazeuses avec une composition gazeuses et un nombre d'anneaux qui diffère selon la planète.

Les objets spatiaux se déplacent dans l'espace et peuvent interagir avec des planètes ou des étoiles. Parmi eux, les comètes sont en orbite elliptique autour du Soleil et peuvent passer plus ou moins proche de lui lors de sa période, ainsi que les astéroïdes qui peuvent entrer en collision avec un impact plus ou moins important et les météoroides qui peuvent entrer dans l'atmosphère d'une planète.

Les étoiles fournissent de l'énergie et brillent avec une intensité lumineuse variable.

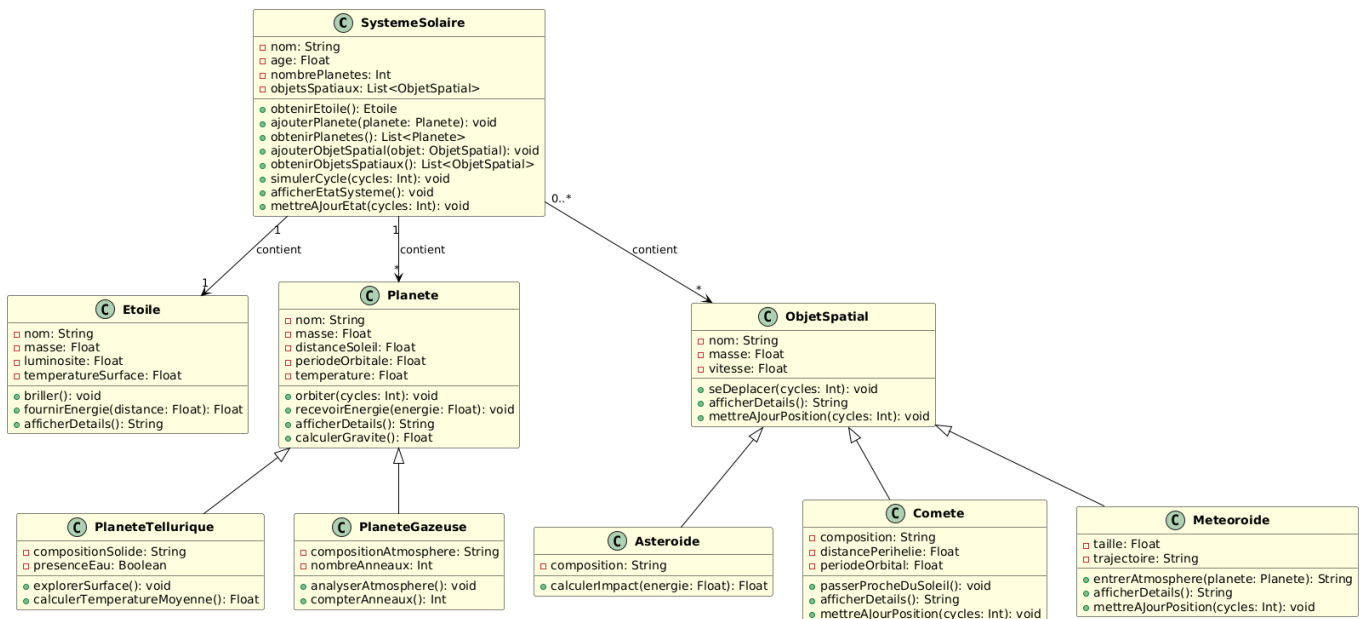
Le système solaire est constitué de tous ces corps célestes et modélise leurs comportements à chaque cycle.

2 Utilisation des contraintes

- 8 classes : Il y a au total 9 classes qui sont SystemeSolaire, Etoile, Planete, PlaneteTellurique, PlaneteGazeuse, ObjetSpatial, Asteroide, Comete et Meteoroides.
- 3 niveaux de hiérarchie : Il y a 5 héritages, PlaneteTellurique et PlaneteGazeuse héritent tout les deux de Planete, Asteroide, Comete et Meteoroides héritent tout les trois de ObjetSpatial. Il n'y a pas 3 niveaux de hiérarchie successives (c'est-à-dire A hérite de B hérite de C) mais il y a plus de 3 fois un niveau de hiérarchie.
- 2 fonctions virtuelles différentes et utilisées à bon escient : mettreAJourPosition et calculerTrajectoire dans ObjetSpatial sont virtuelles car les mouvement des comètes et des astéroïdes sont influencés par le Soleil tandis que le mouvement des météoroides est influencé par les planètes, leurs trajectoires sont aussi différentes avec les comètes elliptiques, les astéroïdes circulaires et météoroides chaotiques;
- 2 surcharges d'opérateurs : opérateur == pour comparer si deux objets spatiaux sont les mêmes et opérateur + dans planète pour fusionner deux planètes avec des propriétés combinées
- 2 conteneurs différents de la STL : utilisation de vecteur pour les planètes car elles sont en nombre fixe et cela permet d'accéder à eux plus facilement; utilisation de liste pour les objets spatiaux car l'ajout et la suppression sont plus fréquentes, cela permet de gérer la mémoire plus facilement.
- diagramme de classe UML complet : voir section suivante Diagramme UML
- commentaire du code : ok
- pas d'erreur avec Valgrind : ok

- rendu par dépôt git, adresse à envoyer par mail avec dans le sujet le motif [MAIN 4 C++ Projet] : ok
- pas de méthode/fonctions de plus de 30 lignes : ok
- utilisation d'un Makefile avec une règle "all" et une règle "clean" ou autre outil de build automatique : ok
- utilisation de test unitaire : ok

3 Diagramme UML



4 Exécution du code

Il n'y a pas de librairie à installer. Le code s'exécute en créant chaque objet et en les ajoutant dans les listes qui leur sont attribuées puis on utilise la méthode `simulerCycle` de la classe `SystemeSolaire` pour modéliser les comportements de chaque objet instancié. On peut aussi utiliser la méthode `afficherDetails` pour voir les détails d'un objet de n'importe quelle classe.

Compilation:

```
g++ -std=c++11 -o simulation main.cc Etoile.cc Planete.cc
PlaneteTellurique.cc PlaneteGazeuse.cc Asteroide.cc Comete.cc
Meteoroides.cc ObjetSpatial.cc SystemeSolaire.cc
```

Execution:

```
./simulation
```

5 Je suis fière

- Je suis fière d'avoir utiliser des connaissances en mécanique céleste pour modéliser le comportement des corps célestes, notamment le passage proche des comètes du Soleil, la fluctuation de l'intensité lumineuse des étoiles, l'entrée dans l'atmosphère des météoroides, l'orbitation des corps et la réception de l'énergie des planètes, malgré que ce soit des modélisations très peu précises et éloigné de la réalité, ceux ci donnent un léger appercue pour modéliser le systeme solaire.
- Je suis fière d'avoir eu l'idée d'implémentée des variable random pour modéliser au mieux le systeme solaire car beaucoup d'informations sont inconnues et beaucoup de facteurs sont a prendre en compte pour avoir une simulation parfaite. L'utilisation de facteur random permet faire varier le comportement des objets même si cela diffère totalement dans le vrai systeme solaire.
- Je suis fière d'avoir implémenté un code dans le thème de l'espace et d'avoir trouver des classes avec des méthodes spécifiques pour chacun en respectant l'utilisation des contraintes demandées malgré que la simulation soit complètement éloignée de la réalité.