Réponses aux Questions

Thibault Rieben
Youssef Saied

May 29, 2017

Abstract

Document contenant les questions poser le long du projet.

Question P1.1 Comment représentez-vous ces vecteurs ? Comment sont-ils organisés : quels attributs ? quelles méthodes ? quels droits d'accès ?

Les vecteurs sont représenté par 3 double (les coordonnées x,y,z) privée pour garder l'encapsulation avec des getters et setters, ansi que des méthodes pour les manipuler (norme, distance, normaliser etc...)

Question P4.1 Avez-vous ajouté un constructeur de copie ? Pourquoi (justifiez votre choix) ?

Pour le moment on n'as pas ajouté de constructeur de copie car celui par défaut (de copie) nous est suffisant, il copie coordonnée par coordonnée (c'est ce qu'on veut).

Question P4.2 Si l'on souhaitait ajouter un constructeur par coordonnées sphériques (deux angles et une longueur),

a] que cela impliquerait-il au niveau des attributs de la classe ?

On devrait ajouter 3 attributs supplémentaires (deux angles et une longueur) de type double à tout les vecteurs, et les calculer si les coordonnée cartésienne sont données et vice-versa

b] quelle serait la difficulté majeure (voire l'impossibilité) de sa réalisation en C++ ? (C'est d'ailleurs pour cela qu'on ne vous demande pas de faire un tel constructeur !)

Pour les deux représentations les attributs sont de type double, alors on ne pourrait pas distinguer la quelle des deux représentation on souhaite construire.

[Question P4.3] Quels opérateurs avez vous introduits ?

les operateurs *, chapeau , == Vecteur3D, == double (vérifie si la normé est égale au double) plus petit (égal) et plus grand (égale) qui vérifient les normes aussi.

Question P5.1 Comment avez-vous implémenté la masse des grains : comme attribut ou comme méthode ?

Méthode parce que c'est plus souple; on peut plus tard changer la manière de calculer le volume et au lieu d'avoir un attribut et en plus on n'a pas besoins d'une méthode get. C'est plus facile a initialiser.

Question P6.1 Comment proposez-vous de représenter les obstacles dans votre projet ? Cela modifie-t-il la classe Grain ?

Une classe Obstacle mère avec des sous-classes, par exemple Plan, Brique etc. Non, car c'est une classe extérieure à Grain

Question P8.1 En termes de POO, quelle est donc la nature de la méthode dessine ?

Méthode virtuelle abstraite, car chaque objet à sa manière d'être dessiné, qui n'as pas forcement de forme générale. Ainsi on "force" la redéfinition de la méthode par chaque classe.

Question P8.2 Peut-on faire comme proposé ci-dessus ou faut-il changer quelque chose ? Si oui, pourquoi ; si non, expliquez quoi.

ajouteforce() doit devenir virtuelle, pour qu'on puisse utiliser la méthode ajouteforce() définit dans la classe Grain.

Question P8.2.1 Quelle est la bonne façon de le faire dans un cadre de programmation orientée-objet ?

Heritage, polymorphisme et redéfinition des virtuel méthodes par rappot à leurs spécificité

Question P8.3 A quoi faut-il faire attention pour les classes contenant des pointeurs ? Quelles solutions peut-on envisager ?

Les constructeurs par copie (superficielle) et vers où les pointeurs pointe. Dans notre cas jusqu'à présent, le constructeur par copie superficielle nous est suffisant car on pointe vers une instance de la classe Medium qui est static et constante.

Question P8.4 Comment représentez vous la classe Système ? Expliquez votre conception (attributs, interface, ...)

La classe Système est une fille de Dessinable, avec comme attributs des tableaux de pointeurs de Grain, Obstacle et Source, ainsi qu'un pointeur de milieu. Un opérateur de affichage, une méthode dessine ainsi que des méthodes évolues.

Question P8.5 Pourquoi fait-on cela ? A quoi peuvent bien servir ces deux dernières méthodes ?

Car un Système est un objet "lourd", alors nous voulons éviter des copies de cet objet. Aussi parce que le Système est le lieu ou la simulation à lieu, alors au niveau de la conception on veux pas avoir des systèmes identiques.

Question P9.2 Quelle est la complexité de ces deux algorithmes ci-dessus ?

La complexité est (max(taille tab grain, taille tab obstacles) * taille tab grain) pour les deux algorithmes.Alors $\mathcal{O}(n^2)$ avec n le nombre de grain dans le système (normalement le nombre de grains est plus grand que le nombre d'obstacles

Question P11.1 A quel endroit vous semble-t-il plus approprié de mettre un tel test ?

Dans la méthode évolue de système.

Question P12.1 Avant de préciser les détails d'implémentation, quelle est la complexité temporelle pire cas de cette solution en fonction du nombre de grains ? [On supposera ici que les grains sont "assez bien répartis" dans les cases. On supposera de plus que la taille d'une case est petite par rapport à la taille du système. Ainsi on peut faire l'hypothèse que le nombre de grains par case est négligeable (O(1)) devant le nombre total de grains (c.-à-d. tous les grains ne se retrouvent pas en même temps dans la même case). Quel(s) inconvénient(s) présente cependant cette solution ?

Chaque grain va interagir avec un "petit" nombre de grains, ceux de sa case et ceux des voisines, $\mathcal{O}(1)$. Ceci pour tout les grains alors n fois. Alors l'algo a une complexité $\mathcal{O}(n)$ avec n le nombre de grains dans le système.

Question P12.2 Comment et où avez-vous implémenté cette nouvelle façon de calculer les collisions ?

On a fait une Superclasse Système qui a comme filles SystèmeP9, SystèmeP12, SystèmeP13. Avec les deux derniers les système avec des cases. Les nouvelles collision sont dans les méthodes évolue, de SystèmeP12, SystèmeP13 qui va parcourir les cases selon le nouveau algorithme.

Question P13.1 Quelle est la complexité temporelle pire cas de cette solution en fonction du nombre de grains ? [Cette question est difficile et nécessite la lecture de la documentation sur les map] Quel(s) avantage(s) par rapport à la solution précédente ?

$\mathcal{O}(n)$ avec n le nombre de grains dans le système. L'avantage est que il parcourt uniquement les cases essentielles. Tandis que P12 parcourt toutes les cases du systeme.