

Reponses

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

Nous avons choisi de représenter ces vecteurs à travers les 3 coordonnées cartésiennes mises en attributs avec droit d'accès en private afin de renforcer l'encapsulation.

Nous avons aussi introduit et conservé les méthodes suivantes toutes en public afin qu'elles soient facilement utilisables:

- affiche
- compare
- addition/soustraction
- oppose
- mult (multiplication par un scalaire)
- prod_scal (entre deux vecteurs)
- norme et norme2
- unitaire

Elles nous sont nécessaires lors certaines définitions de surcharges d'opérateurs que nous avons mises en dehors de la classe et qui les utilisent.

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

Oui, nous l'avons jugé plus sûr et efficace lorsque nécessaire. Il faudra cependant faire attention à bien appelé ce constructeur si on définit un constructeur de copie dans une sous classe héritant de Vecteur3D, or il n'y en a pas dans notre projet.

Finalement la classe ne contenant pas de pointeurs ce constructeur ne devrait pas poser problème.

[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 ?

Si un tel constructeur est mis en place une conversion en coordonnées cartésiennes au sein du constructeur sera nécessaire pour initialiser l'instance créée de manière correcte avec donc ses attributs en cartésiens.

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 !)

On aurait un problème car ce constructeur prendrait les mêmes paramètres que celui directement en cartésiens et on aurait ainsi une double définition qui poserait problème lors de l'instanciation. On doit en garder qu'un seul ou utiliser des constructeurs nommés aussi utilisables en c++.

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

Nous avons introduit les opérateurs suivants :

<< qui correspond à la méthode affiche.

(La définition de cet opérateur doit nécessairement être en dehors de la classe et requiert dans notre programme l'utilisation de la méthode affiche)

== et != qui correspondent à la méthode compare

+ et - l'addition/la soustraction

* le produit scalaire

^ le produit vectoriel

~ le vecteur unitaire

Nous avons décidé de mettre toutes les définitions de ces opérateurs en dehors de la classe parce qu'elles créent une nouvelle donnée en retour lorsqu'elles sont utilisées que ce soit un vecteur, un bool ou un double.

Finalement on a aussi introduit les opérateurs += et -= à l'intérieur de la classe qui n'ont pas besoin de méthode dans leur définition et ne créent pas de nouvel objet.

[Question P7.1] En termes de POO, quelle est donc la nature de la méthode dessine_sur() ?

La méthode dessine_sur() est virtuelle pure. Elle est déclarée avec la syntaxe «=0» dans dessinable rendant ainsi cette classe abstraite ainsi cette méthode devra être redéfinie dans toutes les sous-classes de dessinables pour que celles-ci ne soient pas abstraites et puissent être instanciées.

On a voulu rendre dessinable abstraite car celle-ci ne représente pas un objet concret dans notre programme et est juste une superclasse nécessaire à la conception.

[Question P8.1] A quoi faut-il faire attention pour les classes contenant des pointeurs ? Quelle(s) solution(s) est/sont envisageable(s) ?

Lorsque des objets sont créés dynamiquement avec des pointeurs il est impératif de les libérer pour éviter des fuites de mémoire. La solution envisageable est la mise en place d'un destructeur qui va libérer la mémoire avec la syntaxe «delete» une fois qu'on a fini d'utiliser l'instance de l'objet, dans notre cas ce sera à la fin de la simulation.

[Question P8.2] Comment représentez-vous la classe Systeme ?

Expliquez votre conception (attributs, interface, ...).

Nous la représentons avec en attributs : une Enceinte et un vecteur de pointeur sur des Particules.

Les pointeurs sont utilisés pour obtenir la résolution dynamique des liens et permettre lors de l'appel des méthodes de chaque instance du tableau que ce soit la méthode de la sous-classe qui soit appelée et pas celle de Particule.

[Question P8.3] Comment empêchez-vous cela (copie et affectation) ?

Pour empêcher toute copie nous mettons pas de constructeur de copie ou quelque autre méthode de copie dans notre classe.

[Question P9.1] Comment (et à quel(s) endroit(s)) intégrez-vous ces deux aspects ?

[Question P9.2] Quelle est la complexité de l'algorithme de simulation lorsqu'il utilise cette méthode pour savoir si deux particules se rencontrent ?

[Question P11.1] Si vous souhaitez voir l'enceinte, c.-à-d. la faire dessiner, comment devez-vous modifier (ou pas) votre conception/votre code ?

Pour voir l'enceinte et voir la particule nous avons créé à travers nous avons créé une méthode DessineEnceinte() dans VueOpenGL qui dessine uniquement les arrêtes de l'enceinte et donne ainsi l'effet de la transparence. Nous avons aussi adapté la méthode dessine() qui prend une enceinte en paramètres pour que la taille dans les 3 dimensions de l'enceinte correspondent bien à la forme dessinée on peut alors représenter tout type d'enceinte rectangulaire.

[Question P13.1] Comment représentez-vous cette/ces nouvelle(s) particule(s) ? Où s'incrit-elle(s) dans votre conception ?

[Question P14.1] Avant de préciser les détails d'implémentation, quelle est la complexité temporelle pire cas de cette solution en fonction du nombre de particules ? [Attention : on

suppose, par hypothèse même du modèle de gaz parfait, que les particules sont équiréparties dans l'enceinte. On suppose de plus que la taille d'une case est petite par rapport à la taille de l'enceinte. Ainsi on peut raisonnablement faire l'hypothèse que le nombre de particules par case est négligeable ($O(1)$) devant le nombre total de particules (c.-à-d. toutes les particules ne se retrouvent pas en même temps dans la même case). Quel(s) inconvénient(s) présente cependant cette solution ?

[Question P14.2] Comment et où avez-vous implémenté cette nouvelle façon de calculer les collisions ?