Vincent BEUGNET
Adlane LADJAL

Rapport TP4 – TP5 de C++

TP4

Exercice 1

Pour tester le mobile, avec sa méthode avance et ses deux vecteurs Position et Vitesse, nous vérifions que pour une position donnée (ici (1, 1, 1)) et une vitesse donnée (ici (0, 2, 3)), nous obtenons bien, le bon changement de position après un temps donnée (ici 5) de secondes.

Notre position d'arrivée doit donc être de (1, 1, 1) + 5(0, 2, 3) = (1, 11, 16).

Nous vérifions le calcul dans le test. Nous vérifions aussi que la vitesse est toujours la même.

Exercice 2

Lors du test de la simulation, nous testons dans cet ordre :

- ajoutCorps() : Nous ajoutons 2 corps, et vérifions que la taille de notre vecteur est bien de 2, avec les bonnes valeurs.

- simuler() : Nous lançons la simulation, et nous vérifions les bons résulutats, avec un dt égal à 2.

- afficheCorps() : Nous testons que l'affichage des corps s'est bien déroulé.

- oteCorps() : Nous retirons un corps, et vérifions que la taille est bien de 1.

Exercice 3

Il y a bien un problème ici car si un Mobile est créé de manière automatique, il sera détruit deux fois : une fois dans le destructeur de Simulation, et une autre fois en quittant la fonction.

Pour tester le destructeur, nous créons deux simulations, une dynamique et une automatique, et nous vérifions que nous arrivons bien à finir la fonction sans erreur.

Vincent BEUGNET
Adlane LADJAL

Exercice 4

Pour tester la gravité, nous avons défini une hauteur h (par exemple 15), et un petit temps dt (0,001). Nous avons lancer la méthode avance dans une boucle qui s'exécute tant que la hauteur est plus grande que 0. Nous testons ensuite la formule donné par l'énoncé, en faisant attention aux incertitudes de calculs.

Nous avons dû modifier cette fonction de test lors du tp5…

Exercice 5

Pour tester le fait que Simulation arrive à gérer sans problème un Mobile et un MobilePesant, nous avons simplement créer deux mobiles, un de chaque type. Nous les avons ajouter au corps de la simulation, et lancer une simulation.

Pour tester le constructeur par copie de Simulation, nous avons donc créés deux simulations. Une initialisée avec un Mobile et un MobilePesant dans son corps, et une autre étant la copie de cette dernière.

Nous vérifions ensuite que les deux simulations contiennent bien tous les deux une copie des deux mobiles avec lesquelles nous avons initialisé la première simulation.

Vincent BEUGNET
Adlane LADJAL

TP5

Exercice 1

Dans Vecteur3D, nous testons les opérateurs que nous avons surchargés pour cette classe, à savoir :

+=

==

=

* (double * vecteur)

<<

Au cours du développement nous avons implémenté d'autres opérateurs, comme la division par un double, ou encore le produit vectoriel (^).

Exercice 2

Dans la fonction de test de la Terre, nous avons créé un pointeur de type Terre. Nous avons en premier lieu appliqué une gravité sur un point situé à une distance RT du centre de la Terre, soit à sa surface. Nous vérifions que la gravité est bien à peu près de la bonne valeur soit autour de 9,80.

Pour tester le design pattern Singleton, nous avons créé un autre pointeur de type Terre. Nous vérifions que ce pointeur pointe bien à la même adresse que le pointeur de départ.

Exercice 3

3) Nous testons un mobile quelconque, à une hauteur de 200 000. Ici nous vérifions que lorsque que le temps donnée par la formule de l'énoncé s'est écoulé, le mobile est bien revenu sur sa position de départ à quelques mètres près

Vincent BEUGNET
Adlane LADJAL

4) Pour tester le satellite SPOT, nous avons créé ce satellite, avec la position indiqué dans l'énoncé. Nous l'avons ajouté à la simulation. Puis nous lançons une boucle, sur le temps de la simulation, étant de 1h 41 et 24 secondes.

Nous vérifions que le satellite est bien retourné à peu de chose près sur sa position de départ.

5) Pour tester nos fonctions sur la lune, nous procédons de manière analogue que précédemment.

En revanche la lune met environ 29 jours pour faire une révolution. Dans notre test, la Lune est plus rapide de deux jours. Cela est sûrement dû aux incertitudes qui se propagent, vu qu'un nombre importants d'itérations sont effectués.

Cela dit nous avons fait en sorte que le test réussisse, pour un temps de révolution de 27 jours.