Techniques de l’informatique – programmation de jeux vidéo

420-V32-SF Programmation de jeux vidéo III

Notions additionnelles de C++ - Modificateurs const

Partie 1 : Généralités

1. Pourquoi devriez-vous toujours privilégier le mode de passage de paramètres « par référence [constante] »?

Car le passage en paramètre par référence constante évite la copie de code des headers, ce qui favorise de bonnes performances.

1. Donnez la signification (le rôle) des trois modificateurs const dans la signature de la méthode suivante :

const CollisionInfo& GetCollisionInfo(const Projectile& projectile) const;

1. La valeur retournée n’implique aucune copie, mais uniquement une référence.

2. Empêche la modification du paramètre. En bref, celui-ci ne pourra être utilisé à la gauche du symbole d’assignation (=).

3. Rend la méthode constante. En bref, si une méthode ne modifie aucune donnée (un get, par exemple), elle devrait avoir un const à la fin, bref être constante.

**Partie 2 : Exercices de codification**

Récupérez la solution ExercicesConversionProgramme réalisée plus tôt cette session.

1. Modifiez le code de la méthode GetDistanceToOrigin en lui ajoutant le(s) modificateur(s) const pertinent(s).
2. Remplacez l’instruction return (float)sqrt(x\*x+y\*y); dans cette même méthode par l’instruction suivante : return (float)sqrt(GetX()\*GetX()+GetY()\*GetY()); Que constatez-vous?

Il semble y avoir une incompatibilité quant à l’utilisation de fonctions diverses dans une méthode constante.

1. Que pouvez-vous déduire au sujet des modificateurs const?

Les méthodes constantes ne peuvent pas user de méthodes normales à l’intérieur de celles-ci, car les méthodes diverses en question ont des chances d’effectuer des modifications.

1. Donnez un autre exemple d’instruction que vous ne pourriez pas retrouver à l’intérieur de la méthode GetDistanceToOrigin.

Comme mentionné à la question précédente, il est impossible d’effectuer des modifications dans des méthodes constantes. Pour cette même raison, nous ne pouvons tout simplement pas modifier des variables existantes avant l’appel de la fonction constante.

1. Modifiez le code de l’exercice « CollisionSpheres » de manière à ajouter tous les modificateurs const nécessaires pour la classe CollisionSphere .