

# Recouvrement

Le but de ce TD est de vous faire manipuler les différents types de tests de recouvrement (Overlap testing) utilisés en jeu vidéo. Il est basé sur le TD Bounding Volumes fait en cours et sous DirectX.

## Instructions :

1. Ajoutez dans votre scène un deuxième vaisseau de sorte que celui-ci soit visible. Il sera statique mais on pourra choisir d'utiliser et visualiser sa bounding sphere ou sa bounding box avec la touche B. Le but est de pouvoir déplacer le premier vaisseau jusqu'à ce qu'il rentre en collision avec le deuxième. Il faudra alors donner un indice visuel de la détection de collision (changer la couleur du bounding volume par exemple)
2. Recouvrement Sphere-Sphere : Implémentez la détection de collision Sphere-Sphere et testez-la avec les bounding spheres des vaisseaux.
3. Recouvrement Sphere-Box : Implémentez la détection de collision Sphere-Box et testez-la.
4. Recouvrement AABB-AABB : Implémentez la détection de collision AABB-AABB et testez-la.
5. Recouvrement OBB-OBB : Implémentez la détection de collision OBB-OBB et testez-la.

## Informations :

Vous pouvez utiliser le code de l'implémentation de l'algorithme d'Arvo (Gomez) disponible sur Gamasutra : [http://www.gamasutra.com/view/feature/3383/simple\\_intersection\\_tests\\_for\\_games.php?page=4](http://www.gamasutra.com/view/feature/3383/simple_intersection_tests_for_games.php?page=4).

Il vous est conseillé de créer une classe Overlap (utilisant le pattern Singleton) qui prendra en paramètre de ses fonctions des objets que vous avez implémentés au TD précédent.  
Par exemple : `bool OverlapSphereSphere(BoundingSphere* _pBs1, BoundingSphere* _pBs2)`

## Documentation :

### **D3DXVec3Dot Function**

Determines the dot product of two 3D vectors.

#### **Syntax :**

```

FLOAT D3DXVec3Dot (
    __in const D3DXVECTOR3 *pV1,
    __in const D3DXVECTOR3 *pV2
);

```

#### **Parameters**

*pV1* [in]

[\*\*D3DXVECTOR3\*\*](#)

Pointer to a source [\*\*D3DXVECTOR3\*\*](#) structure.

*pV2* [in]

[\*\*D3DXVECTOR3\*\*](#)

Pointer to a source [\*\*D3DXVECTOR3\*\*](#) structure.

#### **Return Value**

[\*\*FLOAT\*\*](#)

The dot-product.

### **D3DXVec3LengthSq Function**

Returns the square of the length of a 3D vector.

#### **Syntax :**

```

FLOAT D3DXVec3LengthSq (
    __in const D3DXVECTOR3 *pV
);

```

#### **Parameters**

*pV* [in]

[\*\*D3DXVECTOR3\*\*](#)

Pointer to the source [\*\*D3DXVECTOR3\*\*](#) structure.

#### **Return Value**

[\*\*FLOAT\*\*](#)



The vector's squared length.

Source : <http://msdn.microsoft.com/en-us/library>