

Moteur Physique

1.Introduction

Bienvenue dans ce manuel à destination des futurs utilisateurs de ce moteur physique. Dans celui-ci nous observerons les différents composants du moteur en question et la manière dont ils fonctionnent.

2.Les différents objets

-Les vectors 2 et 3 :

Il s'agit de points dans l'espace (x,y pour le vector 2 et x,y,z pour le vector 3)qui servent pour effectuer différents calculs et situer les objets dans un espace physique, en 2 dimensions pour les vecteurs 2 et 3 dimensions pour les vecteurs 3.

-Rigidbody :

Le rigidbody est un point dans l'espace qui contient une position, une vitesse et une force. C'est lui qui recevra les informations de force et d'accélération et les appliquera après calcul de la vitesse au gameObject.

-Collider :

Le collider est en quelques sorte une boîte entourant le gameObject et qui contient plusieurs formules qui peuvent déterminer si l'objet rentre en contact avec le collider d'un autre objet.

-AABB :

Un peu comme le collider, cet objet détermine si il rentre en contact avec un autre AABB, la différence étant que l'AABB est bien plus grand que le collider et si l'AABB d'un objet n'en touche pas un autre alors on n'a pas besoin de calculer les collisions des colliders.

-World :

Le world doit contenir les formules à appliquer lorsque deux objets ou plus se collisionnent, dans ce moteur il ne les contient toutefois pas par manque de compréhension de ma part.

-QuadTree :

Le quadTree s'occupe de séparer le world en quatre parties pour faciliter les calculs de collision et définit une limite d'objets par parties. Si la limite est dépassée la partie en question se sépare à nouveau en quatre jusqu'à posséder le nombre maximum d'objets permis ou moins.

-GameObject :

Le rôle du GameObject est de contenir un rigidbody, un AABB,un collider et d'afficher sur la console une représentation graphique de l'objet(rond ou carré). Également de changer la couleur de celui-ci ou de changer sa position.

3.Utilisation

Pour utiliser un GameObject rien de plus simple :

-Commencer par le créer dans le Main avec une taille et une position :

```
GameObject rect1(Vector2(size.x,size.y), Vector2(position.x,position.y));
```

-Vous pouvez changer sa couleur via la fonction :

```
rect1.SetColor(color);
```

-Définissez à présent sa vitesse(si vous ne le faites pas elle sera égale à zéro) :

```
rect1.Setspeed(Vector2(speed.x,speed.y));
```

-Dans la boucle principale créez la formule qui calcule le déplacement de votre ou vos gameObject en inscrivant la position lors du lancement :

```
for (float t = 0.0f; ; t += 1.0f / 60.0f)
{
    rect1.SetPos(Vector2(position.x,position.y));
}
```

-Pour finir afficher les grâce à la commande Draw à cet endroit :

```
window.clear();
rect1.draw(&window);
window.display();
```

4.Conclusion

À présent vous avez les cartes en main pour commencer à utiliser ce moteur physique.

Tester, explorer, créer et surtout amusez-vous, c'est le plus important.

Ce moteur est entièrement gratuit et open-source et vous pouvez trouver le code source à cette adresse en libre accès : <https://github.com/feltos/Physics Project>.

N'hésitez pas à donner vos avis et retours sur mon mail ainsi que sur mon twitter personnel, je me ferais une joie de les lire et de recevoir de précieux conseils et remarques.

Mail : guillaumevolgyesi@gmail.com

Twitter : <https://twitter.com/guillaumevolgy1>

