

Rapport de Projet - Modélisation Géométrique

Introduction

Le but de ce projet est de modéliser l'évolution d'un banc de poisson dynamiquement à l'aide d'un système de particules. Initialement, chaque poisson a une position et une vitesse propres. La mise à jour du déplacement des poissons se fait à l'aide de la formule : $\text{position}(t + \Delta t) = \text{position}(t) + \text{vitesse}(t) \times \Delta t$.

Plusieurs comportements de poissons ont été implémentés :

- Ils peuvent se déplacer en ligne droite en rebondissant sur les bords.
- Ils peuvent subir une propagation du comportement d'un poisson leader en se faisant contaminer lorsqu'ils sont à proximité, c'est l'effet Trafalgar.
- À l'instar des forces physiques sur les particules, ils peuvent subir des forces (répulsion, alignement, attraction) par rapport à leurs voisins qui influent sur leur mouvement individuel. Cette règle comportementale de Aoki est plus réaliste et, selon les paramètres choisis, peut mener à des comportements plus divers et complexes.

Intérêts du modèle

Ce genre de modèle peut avoir un intérêt dans le monde des jeux vidéos où on peut vouloir modéliser ce genre de comportement, par exemple dans le jeu Subnautica, des bancs de poissons parcourent le monde et doivent avoir un comportement naturel. De la même manière, dans le jeu Lethal Company, il existe un type d'ennemi qui est un essaim d'abeille, cet essaim a différents comportements suivant s'il attaque ou s'il est au repos au niveau de sa ruche.



Figure 1 : Subnautica, banc de poissons. Lethal Company, essaim d'abeilles

Aussi, les systèmes de particules peuvent avoir un intérêt dans la simulation physique d'objets réels ou de particules élémentaires, en prenant des forces ayant un sens physique. Par exemple, modéliser des fluides ou des interactions élémentaires entre électrons/protons etc...

Limites du modèle

L'implémentation de ce genre d'algorithmes est coûteuse à calculer ; cela pose surtout

problème dans les situations où l'on veut une simulation en temps réel, par ex. dans les jeux vidéos. Les modèles de simulation de fluides, avec des comportements complexes et un nombre conséquent de particules, sont aussi difficiles en temps réel. Le comportement de Aoki nous a aussi montré la difficulté à trouver des bons paramètres pour que les interactions entre particules paraissent naturelles.

Fonctionnement de notre algorithme

Nous avons adopté une approche orientée objet en python. Un poisson (`fishes.py`) a des attributs (vitesse, position, comportement) et évolue dans un aquarium (`aquarium.py`), qui constitue l'environnement et dont la responsabilité est de faire la mise à jour régulière des positions en fonction des comportements (`behaviors.py`).

Un aquarium peut être 2D, 3D ou avoir des caractéristiques différentes si besoin (`AquariumKdTree2D`) en héritant de l'objet `Aquarium` qui est abstrait. Les aquariums ont principalement un tableau des poissons et une fonction `update_all()` qui va appeler la fonction `update_state()` sur chaque poisson, qui ensuite appelle la fonction `behave()` du comportement du poisson.

En effet, les comportements des poissons héritent de la classe abstraite `Behavior` et contiennent les fonctions `behave(fish)` modifiant les attributs d'un poisson. Ces comportements sont bien documentés dans le notebook et dans le fichier `behavior.py`.

Lors de la simulation, on peut donc appeler la fonction `update_all()` de l'aquarium pour faire avancer la simulation d'une itération. On utilise les animations `matplotlib` pour cela.

Pour plus de réalisme, les comportements "`StraightVariableReboundBehavior`" et "`TrafalgarBehavior`" ajoutent une direction aléatoire toutes les `delay_rand_variation` itérations dans un cône maximal (défini par `max_angle_rand_variation`) autour de la vitesse de base de poisson.

Pour lancer les différentes simulations, un notebook jupyter est fourni. Il permet de lancer tous les exemples pour les 3 premières parties en 2D ou en 3D.

Conclusion

Les comportements proposés n'étaient pas toujours convaincants pour modéliser des bancs de poissons. Cependant, les modèles sont de plus en plus réalistes de la partie 1 à la partie 3. Et les règles comportementales de Aoki sont difficiles à calibrer pour obtenir un mouvement réaliste. En revanche, la partie 2 peut être intéressante pour modéliser des épidémies de populations animales et la partie 3 se rapproche du comportement des particules physiques.

Pour obtenir un comportement plus réaliste, nous pourrions ajouter un champ de vision aux poissons pour choisir les voisins qui influencent leur comportement.