

Detección de colisiones de abejas robóticas en java

Omar Alexis Becerra
Universidad EAFIT
Colombia
oabecerras@eafit.edu.co

Juan José Tamayo
Universidad EAFIT
Colombia
jjtamayoa@eafit.edu.co

Mauricio Toro
Universidad EAFIT
Colombia
mtorobe@eafit.edu.co

RESUMEN

El principal objetivo de construir un algoritmo que no tenga una complejidad muy alta y de tiempo de ejecución corto capaz de detectar las colisiones entre las abejas y dar una solución óptima al problema de la creciente desaparición de abejas la cual es un componente importante en la agricultura con lo cual las abejas robótica podrían en un caso reemplazarlas en el proceso de la polinización de las flores el cual es el proceso de fecundar los óvulos de la flor, haciendo posible la producción de semillas y frutos que es muy importante para supervivencia de la vida humana.

Un problema muy parecido es el problema de manejo de colisiones de los videojuegos donde se tiene que detectar la colisión de un objeto con otro en el mayor tiempo posible para que el juego no pierda fluidez y dar una experiencia al usuario excelente.

PALABRAS CLAVE

Colisiones, Nodo, Rango, Distancia, Límite.

1.INTRODUCCIÓN

Las abejas son una de las especies que se encuentra en peligro de extinción en el mundo, debido a las acciones humanas cotidianas. El uso de químicos y pesticidas en la producción agrícola, la contaminación y calentamiento global son algunos de los factores que han puesto en riesgo la vida de estos insectos.

Debido a esto Científicos de la Universidad Politécnica de Varsovia han creado la primera abeja robótica diseñada para polinizar artificialmente las flores capaces de encontrar una flor, recoger su polen, y transferirlo cuidadosamente de la flor masculina a la femenina para fertilizar.

El problema radica en que si esto se llegara hacer de manera masiva habría colisiones entre las mismas

abejas robóticas en este documento se va a dar posibles soluciones al problema basado en algoritmo de computación donde se manejan las posiciones de las abejas a través de coordenadas geodésicas y así prevenir que estas mismas colisionan entre sí y que su funcionamiento sea rápido en la vida real

2. PROBLEMA

La importancia de este problema radica en que las abejas son las encargadas del proceso de polinización, una de las herramientas de la producción alimentaria más rudimentaria del mundo. Tres cuartas partes de los cultivos de alimentos en el mundo dependen de la polinización por insectos y otros animales.

3. TRABAJOS RELACIONADOS

3.1 algoritmo V-COLLIDE

es una biblioteca de detección de colisiones para entornos grandes. Está diseñado para operar en grandes cantidades de objetos poligonales. No hace suposiciones sobre la estructura de entrada y funciona en modelos arbitrarios, también conocidos como *sopas de polígono*

3.2 algoritmo DEFORMCD

es una biblioteca de detección rápida de colisiones diseñada para acelerar el cálculo para deformar objetos cuyos vértices están vibrando, se usa una solución de readaptación AABB para la detección de colisiones

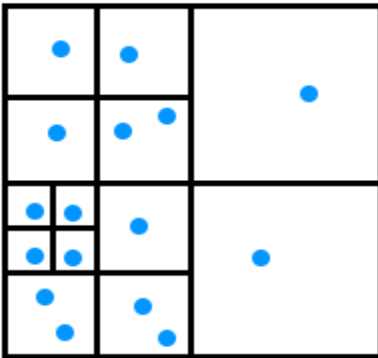
3.3 algoritmo I-COLLIDE:

Funciona para modelos que son poliedros convexos. Pero explota las características especiales del politopos convexos para determinar muy rápidamente el estado del contacto. También aprovecha la coherencia temporal, por lo que los tiempos de consulta de colisión son extremadamente rápidos cuando los modelos se mueven solo una cantidad relativamente pequeña entre fotogramas.

3.4 Algoritmo de distancia de Gilbert-Johnson-Keerthi

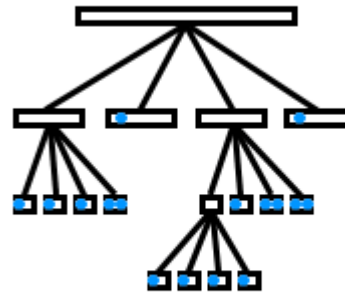
Los algoritmos GJK a menudo se utilizan de forma incremental en sistemas de simulación y videojuegos. En este modo, el símplex final de una solución anterior se usa como conjetura inicial en la siguiente iteración, o "marco". Si las posiciones en el nuevo cuadro son similares a las del cuadro anterior, el algoritmo convergerá en una o dos iteraciones. Esto produce sistemas de detección de colisiones que operan en tiempo casi constante.

4 QuadTree



Se relacionan(chocan) únicamente los puntos (en éste caso las abejas) que se encuentran en el mismo cuadrado y cada vez se va dividiendo en cuadrados más pequeños.

4.1 Operaciones



Se analizan los puntos (las abejas) que se relacionan(chocan) de acuerdo al cuadrado en el que estén ubicados y la distancia entre los puntos.

4.2 Criterios de diseño

Diseñamos la estructura de datos de ésta manera, con un solo ciclo, para poder disminuir la complejidad ya que al hacerlo con dos ciclos analizando dos veces la lista de datos, el algoritmo tendría una complejidad alta, que sería $O(n^2)$ y la idea es implementar un algoritmo con una complejidad menor, creando nodos para las posiciones de las abejas y con un solo bucle comparar las posiciones de las abejas y poder saber si éstas se chocan.

REFERENCIAS

Referenciar las fuentes usando el formato para referencias de la ACM. Léase en <http://bit.ly/2pZnE5g> Veán un ejemplo:

- <http://gamma.cs.unc.edu/V-COLLIDE/>
- https://en.wikipedia.org/wiki/Gilbert%E2%80%93Johnson%E2%80%93Keerthi_distance_algorithm
- <https://www.cs.ucy.ac.cy/~chrysis/specs/ACM-refguide.pdf>
- <http://amca.mx/memorias/amca2003/Articulos/M25Marmolejo46.pdf>
- http://82.98.165.8/seis/is/is48/IS48_23.pdf
- <https://es.wikipedia.org/wiki/Quadtree>
- <https://www.genbetadev.com/programacion-de-videojuegos/teoria-de-colisiones-2d-quadtree>
- <http://jimkang.com/quadtreevis/>