GESTION PARA EL RECONOCIMIENTO DE COLISIONES ENTRE DRONES POLINIZADORES

Laura María Giraldo Castrillón
Universidad Eafit
Colombia
lmgiraldo1@eafit.edu.co

Andrés Felipe Oquendo Usma
Universidad Eafit
Colombia
afoquendou@eafit.edu.co

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

RESUMEN
Por la posible desaparición de las abejas y por la preocupación de que no haya nadie que haga su importante labor, se ha visto la necesidad de crear drones que puedan, en alguna medida, suplantarlas. Nuestro objetivo, es encontrar una forma eficiente y efectiva de que los drones no colisionen. Que la respuesta sea lo más rápida posible para una gran cantidad de drones y que estos puedan cumplir su función sin inconvenientes. Algunos problemas similares son el spatial hashing, quadtree, octree y el árbol AABB para la detección de colisiones.

1. INTRODUCCIÓN
Los avances de la industrialización y de la tecnología han sido un factor que ha impactado especialmente el medio ambiente, actualmente una de las mayores problemáticas es la desaparición paulatina de distintas especies. El enfoque de este trabajo va centrado a como el aumento de los pesticidas y de los depredadores naturales que han reducido en gran porcentaje la población de abejas y esto puede llegar a ser catastrófico puesto que la polinización es un proceso sumamente importante no solo para poder obtener alimentos para el consumo humano sino también el de otros animales

2. PROBLEMA
¿Qué sucedería si las abejas desaparecen? Gracias a la tecnología podríamos pensar en suplantar esta especie por medio de drones, de hecho, hay unos primeros desarrollos frente a esto. El problema central de este trabajo es dar una posible solución para el reconocimiento de colisiones que deben tener estos drones.

3. TRABAJOS RELACIONADOS

3.1 Spatial Hashing
Problema: Se necesita renderizar un juego, para esto es necesario conocer las posiciones de cada elemento en el plano que estamos utilizando, usar una sola matriz se hace un proceso muy lento. ¿Cuál es una posible solución para optimizar el renderizado?

El "Spatial hashing" es una forma de indexar objetos. Funciona con espacios 2D y 3D, una tabla de hash y una función hash. Consiste en convertir el espacio 2D o 3D a una tabla de hash en donde se indexará cada objeto gracias a la función hash. Esto es especialmente útil para estudiar las colisiones de dos o más objetos. Los objetos con el mismo índice hacen referencia a una posible colisión y descarta a los demás objetos indexados en posiciones diferentes de la tabla. Esto siempre y cuando las celdas de la tabla tengan un tamaño adecuado, ni muy grandes ocupando memoria que no se utiliza, ni muy pequeñas ocupando tiempo de ejecución al tener que examinar cada objeto como si no estuviéramos utilizando este método.

Gráfico 1: Spatial Hashing

3.2 AABB Tree Collision Detection
El "AABB Tree Collision Detection" los AABB son cajas con coordenadas, las cuales tienen sus ejes alineados y pueden definirse por 2 puntos ya sea en 2D o 3D. El árbol AABB, permite organizar e indexar los AABB (serían las hojas), dentro de otros AABB más grandes (ramas), y las raíces que pueden ser las ramas o las hojas.

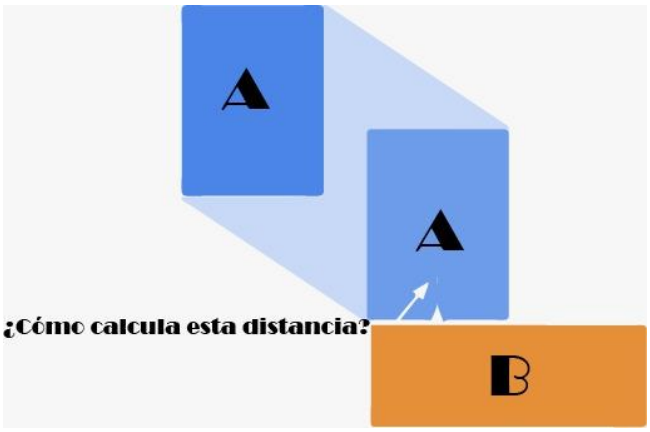

Gráfico 2: AABB Tree Collision Detection

3.3 Quad Tree
Problema: tenemos dos puntos azules en la esquina superior izquierda y uno en la derecha, necesitamos saber cuáles puntos harán colisión, es claro que la más próxima es entre los puntos de la izquierda por lo tanto podríamos descartar aquellos que se encuentran más lejos y se hacen obviamente innecesario comprobar si se chocaran o no. ¿cuál sería el algoritmo más eficiente en este caso?

Quad tree se basa en la descomposición recursiva del espacio, por lo tanto, este algoritmo se dio en solución a la necesidad de guardar los datos que se insertaban con valores idénticos lo que traducido a la gestión de colisiones es identificar cuando un objeto esta simultáneamente cerca a otro, comúnmente se usa para representar puntos, áreas, curvas, superficies y volúmenes. Se trata de una detección eficiente de la colisión en n dimensiones. Este tipo de estructura de datos va muy de la mano con el concepto conocido en toda la programación divide y conquistaras.

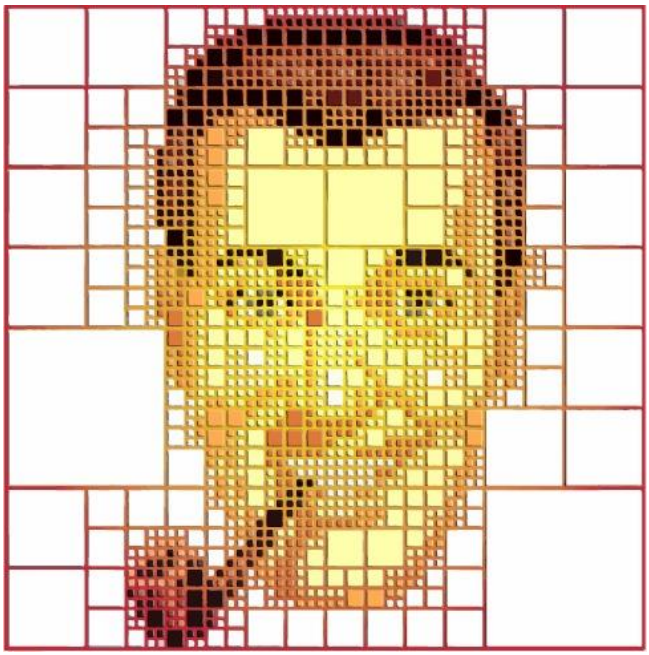

Gráfico 3: Quadtree

3.4 Octree

Problema: Almacenar objetos que se encuentran en un espacio tridimensional. Se dan las respectivas coordenadas en x, y, z y se pide decir en donde se encuentra cada objeto.

Octree funciona dividiendo el espacio en ocho y tomando como base un nodo o raíz que pueden ser infinitos puntos en el plano, se utiliza comúnmente para la detección de colisiones en espacios tridimensionales. El octree es una generalización del quad tree en la tercera dimensión. Por lo tanto funciona perfectamente a este problema ya que al dividir el espacio en subregiones es más rápido a la hora de dar respuesta de la ubicación de lo que estamos buscando.

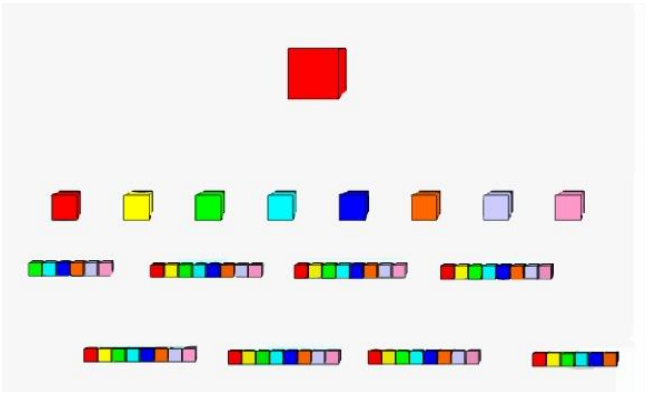

Gráfico 4: Octree

REFERENCIAS

1. Adrigm. Teoría de colisiones 2D: QuadTree, 2013. Retrieved August 26, 2018, from genbeta: https://www.genbeta.com/desarrollo/teoria-de-colisiones-2d-quadtree
2. James. Introductory Guide to AABB Tree Collision Detection, 2017. Retrieved August 26, 2018, from azure from the trenches: https://www.azurefromthetrenches.com/introductory-guide-to-aabb-tree-collision-detection/
3. James, M. Quadtrees and Octrees, 2018. Retrieved August 26, 2018, from i-programmer: https://www.i-programmer.info/programming/theory/1679-quadtrees-and-octrees.html?start=1
4. MacDonald, T. Spatial hashing, 2009. Retrieved August 26, 2018, from gamedev: https://www.gamedev.net/articles/programming/general-and-gameplay-programming/spatial-hashing-r2697/
5. Rebato, C. Por qué las abejas están muriendo y por qué debería importarte, 2015. Retrieved August 26, 2018, from gizmodo: https://es.gizmodo.com/por-que-las-abejas-estan-muriendo-y-por-que-deberia-imp-1717190711
6. Shebata, O. Redesign Your Display List With Spatial Hashes, 2016. Retrieved August 26, 2018, from game development: https://gamedevelopment.tutsplus.com/tutorials/redesign-your-display-list-with-spatial-hashes--cms-27586
7. Wikipedia. Quadtree, (N.D). Retrieved August 26, 2018, from wikipedia: https://en.wikipedia.org/wiki/Quadtree

8. Wikipedia. Octree, (N.D). Retrieved August 26, 2018, from wikipedia: https://es.wikipedia.org/wiki/Octree