

# **Drones polinizadores: Una propuesta tecnológica para mantener el equilibrio del ecosistema**

Isaias Labrador Sánchez  
Universidad Eafit  
Colombia  
ilabradors@eafit.edu.co

Manuel García Jiménez  
Universidad Eafit  
Colombia  
megarciaj@eafit.edu.co

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

## **RESUMEN**

### **1. INTRODUCCIÓN**

En el mundo se está disminuyendo el número de abejas. Esto debido a cosas como pesticidas y cambio climático. Para balancear la decaída que esto genera en la polinización de plantas y flores, se ha planteado crear drones que lleven el polen de una flor al estigma de otra. Para que estas puedan hacer su trabajo de manera automática es necesario que sepan evitar chocarse entre sí.

### **2. PROBLEMA**

Para que muchos drones puedan volar en un área común, deben ser capaces de que si necesitan pasar por el mismo punto, lo hagan en tiempos distintos de manera que no se choquen entre sí. Para hacer que los drones puedan hacer esto, debe crearse un algoritmo para que dos o más drones sepan cuando van hacia un punto en común.

### **3. TRABAJOS RELACIONADOS**

#### **3.1 Hashing espacial:**

Al crear videojuegos es importante tener información posicional sobre los personajes y las cosas con las que interactúan para así evitar que estos se “atravesen” entre sí al pasar por la misma posición. Lograr tener esta información es posible a través de la creación de una tabla hash la cual usa el punto en 2 o 3 dimensiones (el cual es referencia a donde se encuentra el objeto en cuestión) como llave de hash. Esto nos facilita saber que tan cerca están los objetos apenas recuperemos la llave.

#### **3.2 Arbol AABB**

Caundo queremos conocer la área general la cual ocupa un objeto con el propósito de saber cuando este choca con otro, un buen algoritmo a usar es el algoritmo AABB's. Este consiste básicamente de rectángulos los cuales acotan el área real de los objetos a tener en cuenta, usando su punto mínimo y máximo (con los componentes x y y mínimos y máximos respectivamente). Si alguno de los puntos de un AABB está dentro del AABB de otro objeto, las 2 formas están en colisión.

#### **3.3 Quad Trees**

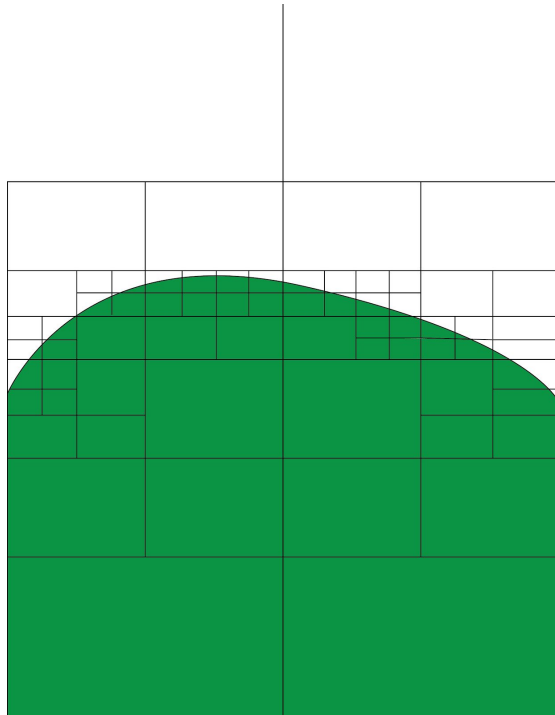
Al momento de delimitar zonas, espacios o superficies cambiantes dentro de un programa es importante que, de manera rápida y precisa, se redefina la forma o los límites de los mismos, dando la capacidad al programa de redefinir parámetros en cualquier tipo de situación. Por ejemplo cuando se define una superficie plana (2D) que hace de campo y esta es sometida al impacto de una bomba; Aquí entra el Quadtree, que básicamente define todo el plano donde está definida la superficie en cuadrados de el mismo tamaño que se van dividiendo en partes iguales de 4 SOLO en los cuadros que toque dicha superficie, definiendo cuadrados cada vez más pequeños que definen de manera precisa la superficie evitando tener que verificar zonas en “blanco”[3].

#### **3.4 Dynamic AABB tree**

Para añadir físicas y delimitar el espacio que ocupa un objeto u objetos 3D alineados en un mismo plano se toma como primera opción implementar el método de los árboles DINÁMICOS AABB(Axis Aligned Bounding Box) [4]. el cual utiliza cubos

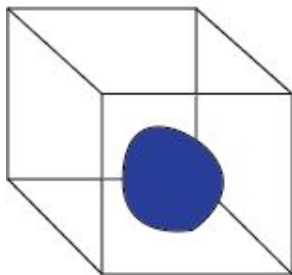
ubicados en el plano 3D que delimitan el área sobre la cual está el objeto en cuestión (Cada objeto tiene su propia “caja”) y a partir de las coordenadas de sus componente X, Y y Z minimos y maximos se determina si el objeto choca con otro o no, haciendo más sencilla la estructura de datos que compara las posiciones de cada objeto.

#### QUAD TREES : GRAPH



\*De esta manera se van haciendo cuadros cada vez más pequeños, partiendo en partes iguales los cuadrados que toquen la superficie, en este caso, de la montaña.

#### Dynamic AABB Trees:



\*La intención principal de los árboles dinámicos AABB es delimitar el espacio de un objeto para poder decir cuando el mismo entra en colisión

#### REFERENCIAS

1. MacDonald. Spatial Hashing, general and gameplay programming. Visitado 24 Feb, 2018 de Gamedev:  
<https://www.gamedev.net/articles/programming/general-and-gameplay-programming/spatial-hashing-r2697/>
2. James, Introductory guide to AABB tree collision detection. Visitado 24 Feb, 2018 de Azure from the trenches:  
<http://www.azurefromthetrenches.com/introductory-guide-to-aabb-tree-collision-detection/>
3. 2014. Quadtree Explanation. MrHeyheyhey27. Visitado 24 Feb. 2018 de: <https://en.wikipedia.org/wiki/Quadtree>
4. Gaul, R. (2013). Dynamic AABB Tree. randygaul. Visitado 24 Feb. 2018: <http://www.randygaul.net/2013/08/06/dynamic-aabb-tree/>



