THE NEAR EXTINCTION OF BEES AND FUTURE OF THE ROBOBEES

Duvan Ramírez.
Eafit University.
Colombia.
daramirezs@hotmail.com

Santiago Santacruz Ramirez.
Eafit University.
Colombia.
ssantacrur@eafit.edu.co

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

ABSTRACT

It is well known that all around the world, several species of bees are listed as endangered species, which is a problem due to the fact that bees are one of the most important pollinators and are responsible of about the 70% of the food that is consumed worldwide because it depends on pollination [1]. On the other hand, a big part of the Terrestrial biodiversity depends largely on pollination, which implies that the extinction of bees embrace a complete change in the earth and in the way our lives are going on, affecting the things we eat, the clothes we wear and the plants and animals that surround us. Therefore, this essay has the purpose to analyze the problematic that revolves around the extinction of the bees and the facts that influence their disappearance. As well as coming up with solutions that get to break the actual problem.

The solution is to implement a data structure from a HashMap that allows us to identify each bee with a only key and a linked list for bees that collide. The algorithm has a favorable complexity which favors its performance and avoids the delay in result, however, there is a margin of error.

Keywords:

-Data structure, nodes, HashMap, data, LinkedList, especial hash.
-ACM Classification: CCS → Information systems → Data management systems → Data structures → Data access methods → Multidimensional range the search

1. INTRODUCTION:

Since the beginning, humans have been individuals that make changes over the time, we advance in terms of technology, science and economy. The first and second industrial revolution bring benefits like exportation and articles production, whoever while the enterprises were growing along with the economy of the countries, the demand of more primary products brings more contamination like water, atmospheric and land pollution. These facts affect the equilibrium of the ecosystem, putting in danger animals, some of them are already extinct and were essential for the ecosystem. Nowadays the bees are the principal species that have a very important task, this specie is in charge of pollination of the plants and in an indirect way they contribute more than 80% of the food production of the world, but the human being push this specie at the edge of extinction. We need to generate some solutions for this problem, and this document have the purpose to give a solution of a alternative that are electronic bees, using algorithms pre-programmed taking into account the cardinal position of every bee ant their spatial dimensions.[1]

2. PROBLEM

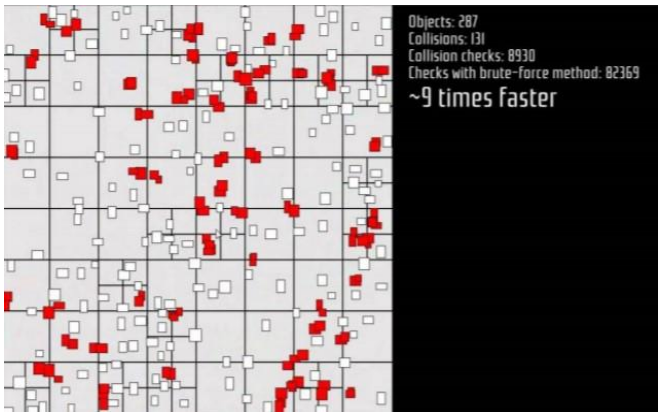
The use of insecticides (such as neonicotinoids), climate change and loss of habitat with the human invasion on wild ecosystems, have led to the deaths of millions of honeybees and put at risk of extinction to many of these [2]. With the permanent artificial bees' implementation, plants pollination can be achieved globally (or at least nationally) in case this crisis worsens. with this it is not intended to allow bees to become extinct and we supplant them but rather to look for alternatives to solve a global problem such as the disappearance of 70% of the food we consume daily

3. Related works

The alternative that we want to implement are the Robo-Bees, that could perform the same activities as the bees but there are problems with the collision they can have if they are close to each other, so we present some data structured alternatives so we can solve this type of problem.

3.1 The Quadtree

“Quadtree” is a type of data structure invented by Jon Bentley in 1970, where each original or parent node has four lower-level or subsequent child nodes, where each element is continually split into four pieces. These are used to efficiently store data of points on a two-dimensional space. These data structure can be used to detect collisions in some video games like worms, and also it could be used to detect collisions in Robo-Bees, this is a model of 2D collisions, but it has a implementation of 3D that its call Octree. figure 1.

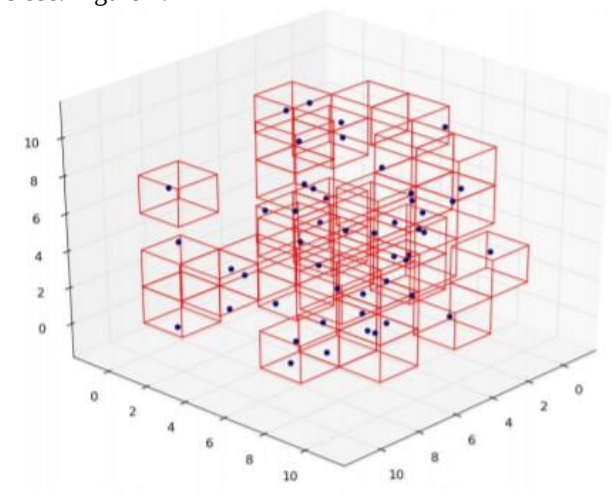


Vinh, T. SFML-Quadtree Collision Detection, 14 october 2012. Retrieved february 17,2019, from YouTube: https://bit.ly/2DTyK0i

Figure 1.

3.2 Spatial hashing

"Spatial hashing" is a 2Dor 3D extension of the hash table, and then is projected into a 1D hash table, allowing fast location of objects and if they collide. If there are many objects, they part the space so only you need to verify if they are about to collide. For the Robo-Bees problem, it could be used because if the number of Robo-Bees are too many, the spatial hashing will help to determinate only the ones that are close. Figure 2.

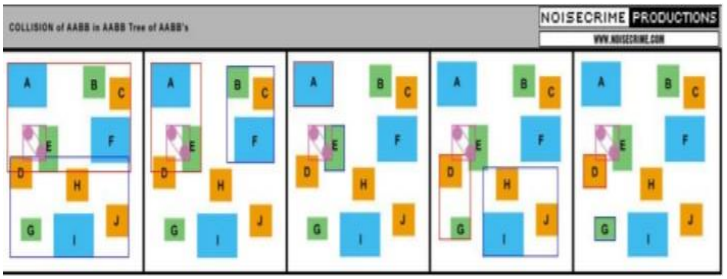


Spatial Hashing in C++, Part 1, 23, july 2016 . Retrieved february 15,2019, from My Internet Weblog: https://bit.ly/2GwK3Pr.

Figure 2.

3.3 AABB tres

AABB trees is a simply binary tree, where all the AABB are stored at the leave. The main advantage for this kind of broad-phase is that this is a border-less data structure, and it doesn't require you to explicitly specify an area which other kinds of data structures such as grids or QuadTrees require. These data structure is not as optimize as other data structure like the spatial hashing, because it will need to compare all the squares that are in the position. But it can be used for objects collision. Figure 3.

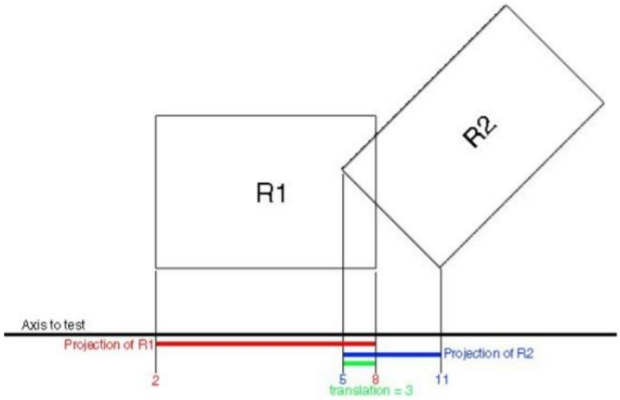


Jungle, B. AABB Tree - Where's the poly o_O?, March 22, 2005. Retrieved february 17,2019. from gamedev: https://bit.ly/2Sa0T8n

Figure 3.

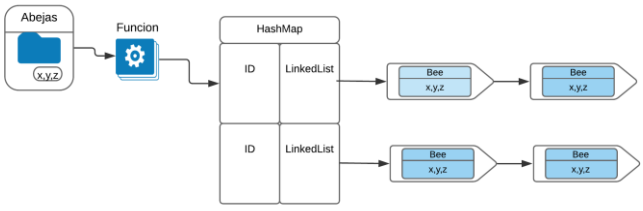
3.4 Separate axis theorem (SAT)

The "SAT" is a method to determine if two convex shapes are intersecting. The algorithm can also be used to find the minimum penetration vector which is useful for physics simulation and several other applications. SAT is a fast-generic algorithm that can remove the need to have collision detection code for each shape type pair thereby reducing code and maintenance.



Translation direction in separating axis theorem, december 2013. Retrieved february 17,2019.from stackoverflow: https://bit.ly/2SGKAVr

.4.Spatial Hash



4.1 Data structure operations

Assign Bee:

This method assigns a key to a new bee, so it can be comparing to the keys in the HashMap.

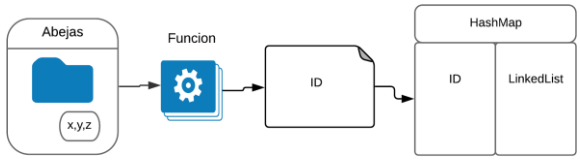


Figure 4.

Compare Alone Bees:

In this method a bee can collide with another that is in a different quadrant, because it is sorted, the bee can move through the quadrants and compare if there is a bee in an adjacent quadrant and if the distance is less than a hundred meters.

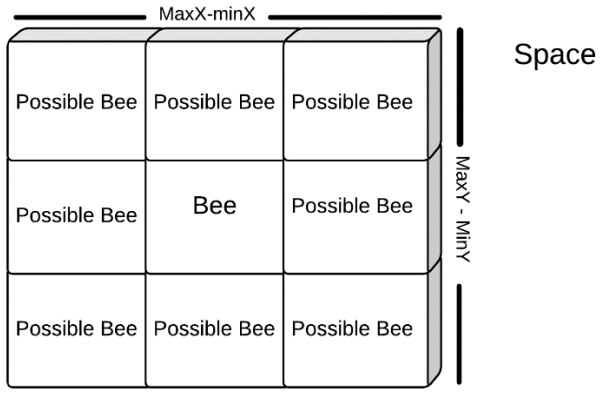


Figure 5.

Compare the size of the key:

This method compares the size of the key and if its size is greater than one, we add all the bees of that key to a LinkedList.

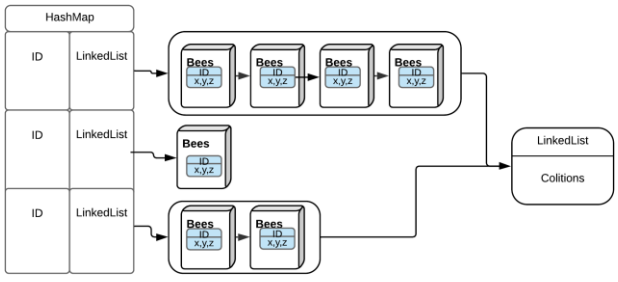


Figure 6.

4.2 Criteria of the data structure

One of the main problems when we were discussing about the complications of this project was reading and identification of the position of the bees in a low time complexity. We decided to opt for the HashMap since it is a dynamic data structure that allows us to have each bee identified, having access to them in O (1). In addition, the decision to implement a LinkedList as a value for the HashMap is because it allows us to add and remove elements in O (1). The use of a LinkedList is also an advantage because it does not consume as much memory as an array would.

4.3 Analysis Complexity

Method	Complexity
PutBeesOnHash	**O(1)**
getCuadrant	**O(1)**
getSides	**O(1)**
CuadrantCollisions	**O(n)**
AbejasSolitas	**O(n)**
Vecinos	**O(n)**
writeFile	**O(n)**
arrayListDeDatos	**O(n)**
muestraContenido	**O(n)**

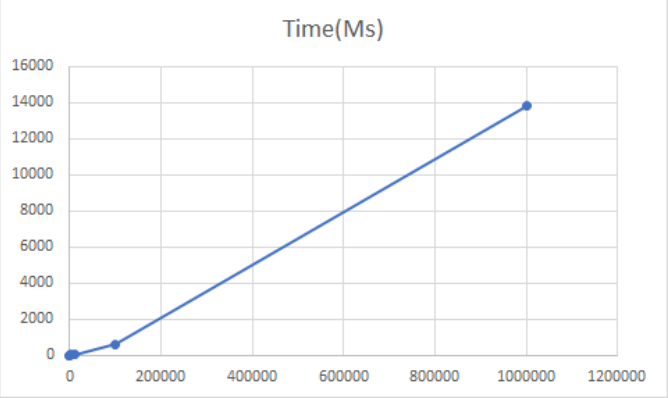
4.4 Process Time

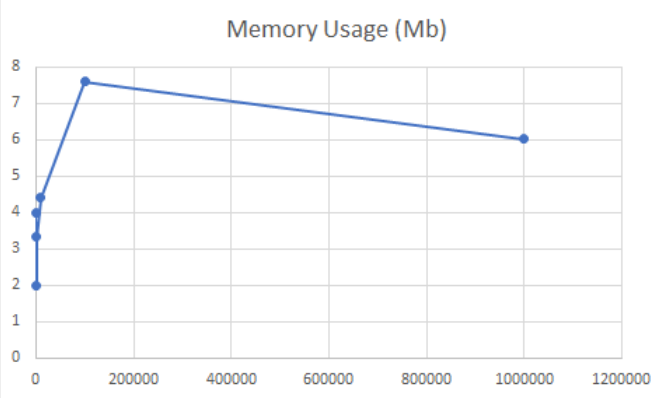
Bees	Time(Ms)
10	1
100	3
1000	9
10000	43
100000	600,666
1000000	13826

4.5 Memory

Bees	Memory Usage (Mb)
10	4
100	2
1000	3,329
10000	4,3961
100000	7,6048
1000000	6,016

4.6 Analysis of results.





In the program, while the data increases, the execution time increases. Fortunately for the one hundred data taken from each data set, the problem does not reach his worst case.

6. CONCLUSIONS

In conclusion, this data structure using HashMap of LinkedList, is efficient due to the easy access and its ability to process the data, with an accuracy of 98% of collisions.

The relatively low analysis on the bee position is due to the importance of the methods used for the identification of the collision of the bees, which have a low complexity, in this case: (O (n) and O (1)).
Among the things that we could improve, one of them is the presentation of the code. On the other hand, the most complicated task was the implementation of an algorithm that helped us to locate the bee in a position of the hash. Also, the search of bees that were alone, but collided with another bee that was in an adjacent quadrant.

5.1 Future Works

For future works we pretend to get less complexity and try to implement an algorithm that help the comparison between adjacent quadrants with a more efficient way.

Greetings

Thanks to Kevin Alexander Herrera Garces for the guidance that he gave us in the design of the data structure.

To EAFIT because this institution gives us the opportunity to be here and do projects that will help us in our life as future professionals.

References

1. Packham, C. The importnace of bees. BBC, (N.D), in association with BBKA: https://bbc.in/W7QCwA

2.Why are bees in danger?,(N.D),organicsoilassociation: https://bit.ly/2JyN3eZ

3. Spatial Hashing in C++, Part 1, 23, july 2016 . Retrieved february 15,2019, from My Internet Weblog: https://bit.ly/2GwK3Pr.

4. Translation direction in separating axis theorem, december 2013. Retrieved february 17,2019.from stackoverflow: https://bit.ly/2SGKAVr

5. Vinh, T. SFML-Quadtree Collision Detection, 14 october 2012. Retrieved february 17,2019, from YouTube: https://bit.ly/2DTyK0i