TECHNOLOGY AND ALGORITHMS: A TEMPORARY SOLUTION TO BEES EXTINTION.

Isabella Arango Restrepo
Universidad EAFIT
Colombia
iarangor1@eafit.edu.co

Juan David Rengifo Castro
Universidad EAFIT
Colombia
jdrengifoc@eafit.edu.co

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

ABSTRACT

The extinction of the human race is normally seen as sci-fi; however, this could be closer than expected. According to the IUCN (International Union for Conservation of Nature) “nearly one in ten wild bee species face extinction in Europe”. This is something to pay close attention, due to these little pollinators are vital for earth’s life. In fact, the bees are fundamental for plant’s reproduction. In consequence, its extinction implies an authentic catastrophe, increasing the famine, the contamination, and the water reduction.

1. INTRODUCTION

It is not a secret that the human damage to earth is a big deal. The overpopulation and our self-destructive behavior have deteriorated the life on earth in almost all the aspects, and bees are not the exception. These little pollinators are an essential part of our life despite we are conscious or not. Sadly, the bee is in danger of extinction. According to BIP (Bee Informed Partnership) between 1988 and 2015 the 42.1% of the colonies died.

2. PROBLEM

The bee extinction is a major problem, due to this little animal is essential for plants reproduction. The extinction or decrease of its population avoid pollination, something that would increase the famine, the contamination and the water reduction. For this reason, this must be solved as soon as possible.

3. RELATED WORK

Collision detection is a very common problem that is applied in the programming of videogames, searches in multidimensional spaces, analysis of images and even in real life problems. To give an effective solution to this, different data structures are used to reduce the detection time of possible collisions.

3.1 Spatial hash.

A Spatial hash is a data structure that offers a faster way to solve a collision detection problem. It is a 2D o 3D table that consists of dividing the space into several cells and locating the objects that are in it according to their spatial coordinates. Then, these are placed in a hash table in such a way as to achieve an effective location of those found in the same frame.

3.2 AABBs tree.

This data structure is very useful and effective at searching objects collisions in an undetermined three-dimensional space. It consists of boxes that are aligned and represent the coordinates of each element. Then, it is very easy to determine if they present an intersection, because it is enough to look if the following is met or not:

$$\text{maxx1} > \text{minx2} \ \&\& \ \text{minx1} < \text{maxx2} \ \&\& \ \text{maxy1} > \text{miny1} \ \&\& \ \text{miny1} < \text{maxy2},$$

where maxx1 refers the x-coordinate of the upper right corner of object1, maxy1 to the y-coordinate of the same corner, minx1 and miny1 for the lower left corner and so on for the object2.

3.3 Quadtree.

The quadtree is a type of spatial decomposition that consists of dividing the space recursively into four cells and when one reaches its maximum capacity, it is divided again into four cubes. They are too useful when detecting collisions, because it allows to evaluate only those objects that may collide, instead of reviewing the entire list of objects that are in space. For this reason, it takes less time to determine those elements that are very close.

3.4 R-Tree.

This data structure is very used in problems with multidimensional spaces, such as map locations. In addition, it is one of the most efficient methods to find objects that are within a space delimited by a rectangle, which allows searching in a specified area of the tree, instead of searching in each sector that composes the space. An advantage is that it allows to work and analyze large volumes of data without spending a lot of time.

REFERENCES

Runestone. Available from http://interactivepython.org/runestone/static/pythoned/SortSearch/TransformacionDeClaves.html (2012); accessed 24 August 2018.

Azure from the trenches. Available from: https://www.azurefromthetrenches.com/introductory-guide-to-aabb-tree-collision-detection/ (2017); accessed 24 August 2018.

Wkipedia. Available form https://en.wikipedia.org/wiki/R-tree (2018); accessed 24 August 2018.

Evantotuts+. Available from https://gamedevelopment.tutsplus.com/tutorials/quick-tip-use-quadtrees-to-detect-likely-collisions-in-2d-space--gamedev-374 (2018); accessed 24 August 2018.

Wikipedia. Available from https://en.wikipedia.org/wiki/Quadtree#Some_common_uses_of_quadtrees (2018); accessed 24 August 2018.

The IUCN Red List of Threatened Species. Available from: http://www.iucnredlist.org/news/nearly-one-in-ten-wild-bee-species-face-extinction-in-europe-while-the-status-of-more-than-half-remains-unknown-iucn-report (2018); accessed 24 August 2018.