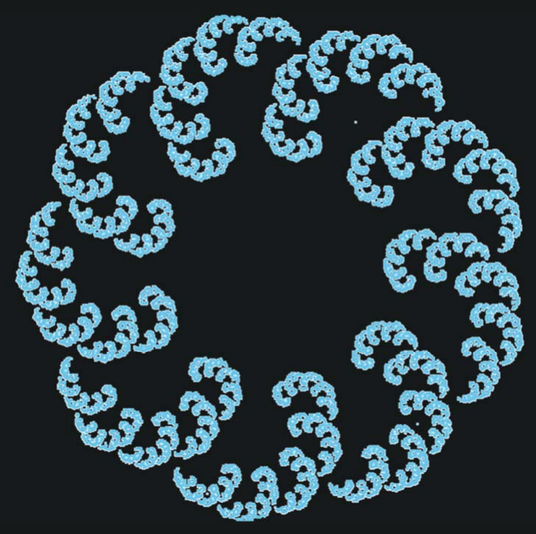
The game of chaos

FRACTALS

Beautiful, recursive patterns that consist of a group of sub-elements, all similar to the whole structure.



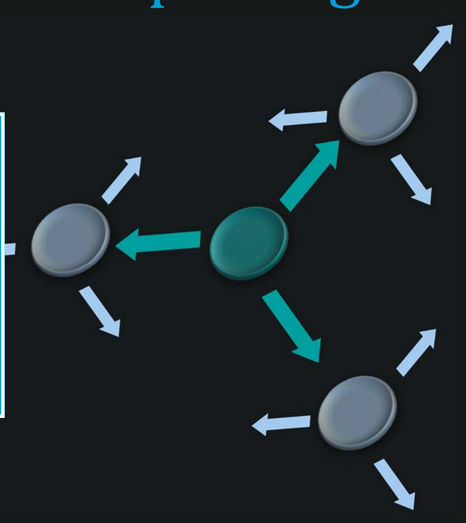
Applications

Apart from the aesthetical ones, fractals can have a lot of applications, such as being more effective at compressing images of landscapes, flora and fauna. That is, if we can find a method to generate them *efficiently*.

THE "STANDARD" APPROACH

Each point generates *n* more.

Each point on the canvas is used to generate 3 other points. Each of them generate *3 other* points, used...



These instructions will create a fractal pattern, but the computer cannot just generate all the points at once and has to remember where it is and where it needs to go next. In order to do that, it stores a lot of data about the current step and takes a lot of time switching back-and-forth between points.

This produces a limitation in the depth achievable by the algorithm. Furthermore, the program does not draw a permanently symmetrical shape (i.e. firstly, it draws all the points on the left etc.). Thus, if not allowed to finish, the result will look "incomplete".

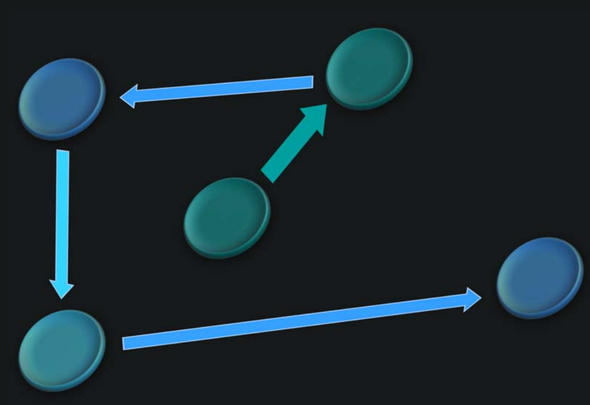
THE *ITERATIVE* APPROACH

Each point has only one successor

Game of chaos

A term introduced by Michael Barnsley, it is an iterative approach that, instead of generating all 3 points when it comes to choose, just picks one of them, on the spot, at random.

This approach relies on the fact that a large (theoretically infinite) number of points guarantees that all the possible points will eventually be reached. By picking only one point, we do not load the stack and decrease the run time. We also get a structure that, at any time, approximates the pure, absolute version of the fractal in terms of its fractal dimension. This approximation is improved with each iteration. By adjusting settings regarding the number of possible choices from each point and the actual process by which the next point is calculated (a function of phase-magnitude on a circular coordinate system), we can obtain a wide range of fractals.



THE IMPLEMENTATION AS A PROGRAM

In order to showcase the capabilities and the behaviour of this algorithm a C++ program has been developed. This application represents a set of "hardwired" instructions for generating well-known fractals, as well as an in-app feature that explains in detail the algorithm used and a menu for generating custom (and random) fractals, with the colors, shapes and sizes of the user's choice. All the menus were created using the *graphics.h* library, with the program being written from scratch to interpret each user input keystroke. Some images presenting the interface and the generated fractals can be found on the right.