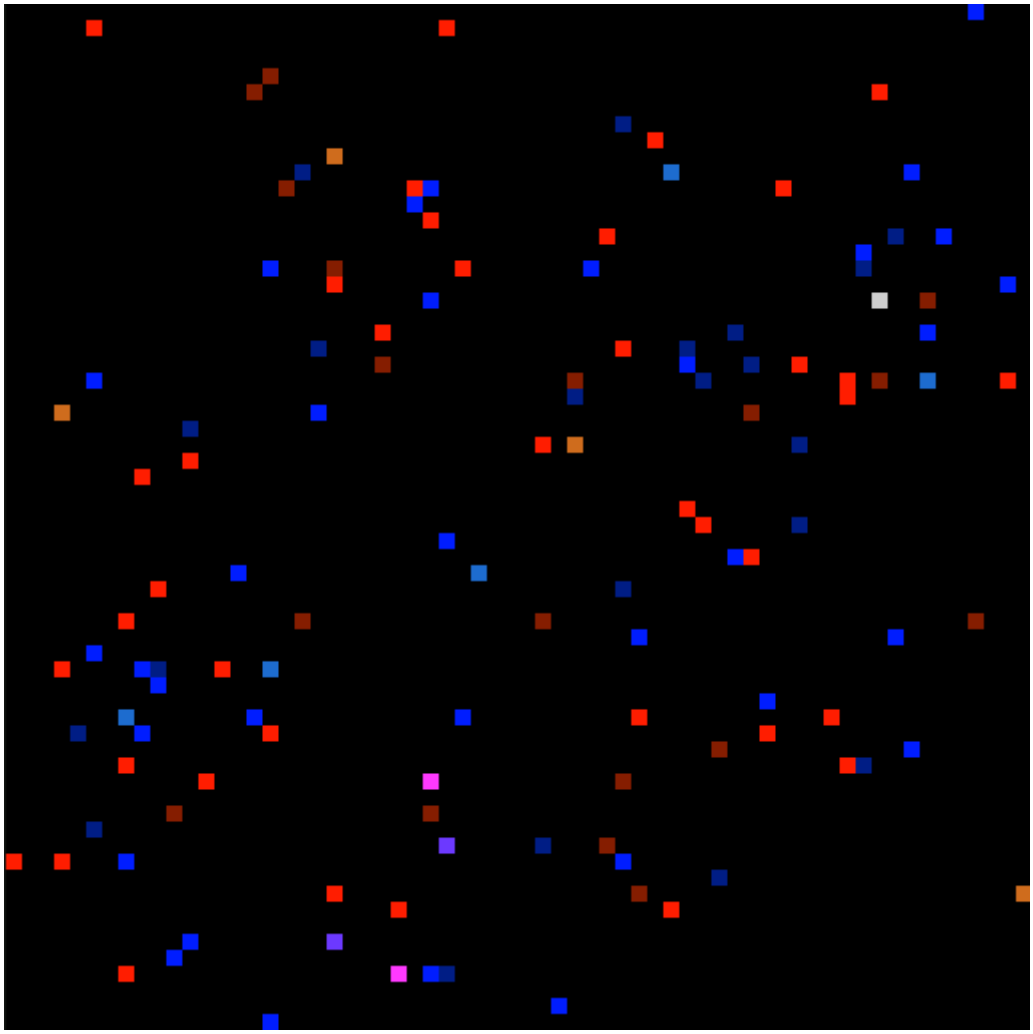
Dots

Simulator



Made by **E_net4**

(enet4mikeenet@gmail.com)

2011

Introduction

Dots is a space-time simulator inhabited by primitive beings called “Dots”. These Dots have characteristics of their own, along with probabilistic and deterministic behaviour, in a way that making experiments with them is possible (and amusing).

The World

The world is made of a bi-dimensional grid of a certain positive width and height. The Dots, whose position is snapped to this grid, reappear in the opposite boundary when they surpass a boundary. In other words, the world has a limited area, but has no real borders (making it similar to a planet).

The Dot

Each Dot has the following attributes:

- Age: The number of steps (time measure) in which the Dot has lived in the world.
- Position: X and Y integer coordinates identify the Dot's position.
- Type: **Alpha** or **Beta**. (Red or Blue, as seen in the simulator)
- Status: The Dot's current status:
  - STATUS_NORMAL
  - STATUS_DEAD
  - STATUS_HUNGRY
  - STATUS_EATING
  - STATUS_LOOKING
  - STATUS_GENERATING

There is a Markov process linked to each Dot, which is used to describe the probabilistic transitions between statuses.

The transitions' probabilities depend on the following functions:

$$d(age) = \frac{age}{\text{death\_chance\_maj}^2}$$

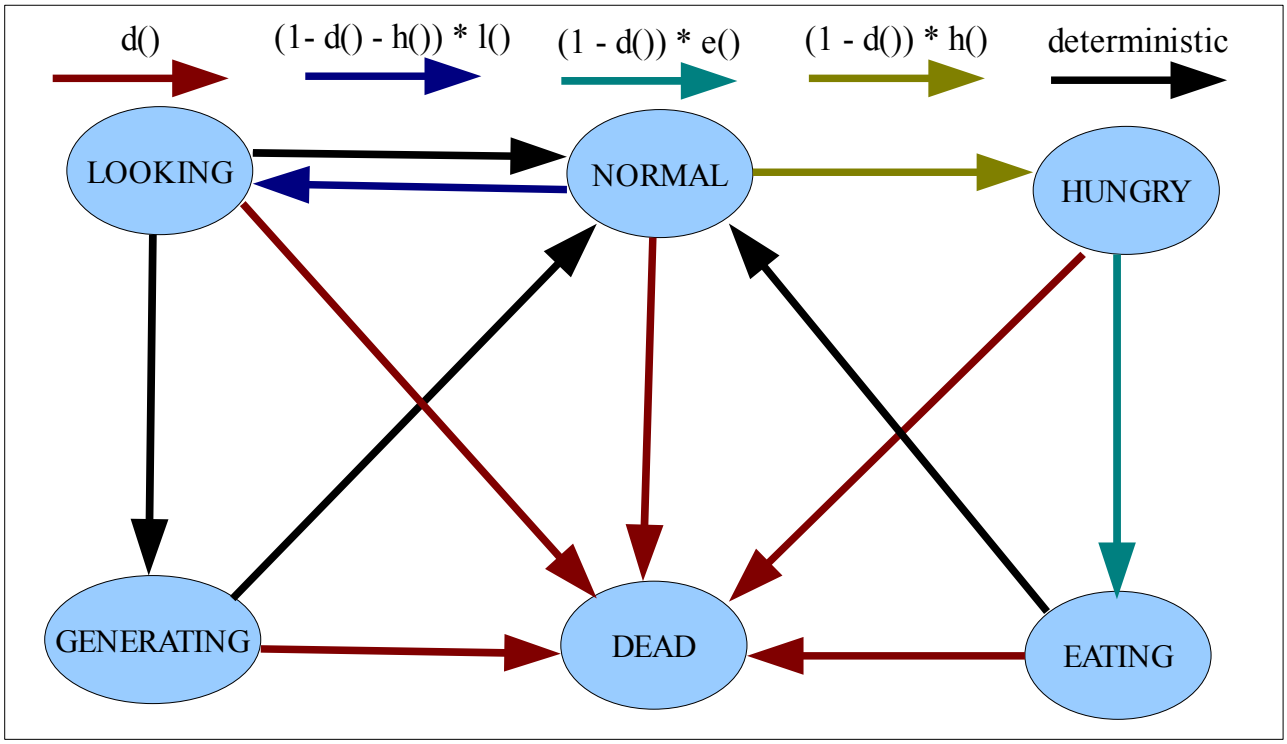
$$h() = \text{hunger\_chance}$$

$$e(\text{Dot}) = \frac{1}{1 + \text{pop\_density(Dot)}}$$

$$\text{pop\_density(Dot)} = (\text{d} \neq \text{Dot}) \sum \frac{\text{dot\_density}}{dist(\text{Dot}, d)^2}$$

$$l(idade) = \frac{\text{looking\_chance\_p}}{\sqrt{2\pi\,\text{looking\_chance\_var}}} \times e^{\frac{(\text{looking\_chance\_mean} - idade)^2}{2 \times \text{looking\_chance\_var}}}$$

This is the status diagram of a Dot:



For each Dot, a “die roll” is performed, making a status change accordingly. A status transition matrix is used, but only the line corresponding to the current status is used, therefore being the only one needed to save and calculate in the Dot.

Finally, the Dots follow this list of special rules, which are status dependent:

- STATUS_NORMAL
  - When it comes to transition, the Dot can die (DEAD), get hungry(HUNGRY) or enter a search process of a Dot of opposite type(LOOKING). It performs a *random walk* on the world in each step, using one of these 4 directions, uniformly chosen: up, down, left or right.
- STATUS_DEAD
  - It's the only status in which no other transition is possible. In the simulation program, the Dot is eliminated after entering this status, in order to free memory and reduce computational load. The Dot stands still.
- STATUS_HUNGRY
  - The only possible transition for going back to the normal status is through STATUS_EATING. Its probability depends on the population density around the Dot, without counting the Dot itself. It performs *random walk*.
- STATUS_EATING
  - As soon as the Dot leaves STATUS_HUNGRY, it must stay in this status, standing still, for *eating_time* (constant simulation variable) steps. After so, it changes to STATUS_NORMAL.
- STATUS_LOOKING
  - In this status, the Dot takes the nearest opposite Dot (Dot of opposite type) and performs a step, using the same 4 directions, in a way that it gets closer to the other Dot. As soon as they are close enough, both Dots enter STATUS_GENERATIONG. In addition, the opposite Dot must be in either STATUS_NORMAL or STATUS_LOOKING. If no other Dot fulfills these criteria, the Dot returns to STATUS_NORMAL.
- STATUS_GENERATING
  - This the status which allows more Dots to be generated. The two Dots must spend *generating_time* (constant variable) steps in this status, so as to create a new Dot, of random type, in the same position. The Dot returns to STATUS_NORMAL if the Dot's “partner” dies in this status. They stand still until the transition to STATUS_NORMAL is made.

The Simulation

A few simulation constants were already mentioned. Now, I will be introducing all of them. The config.txt file contains, in this order, the following variables:

- **rand_seed**: Random seed, used to get pseudo-random numbers.
- **grid_width**: The width of the world in the Simulation.
- **grid_height**: The world's height.
- **init_dots**: The initial number of Dots placed, with random type and position.
- **hunger_chance**: Used to calculate the chance of a Dot to go from STATUS_NORMAL to STATUS_HUNGRY.
- **dot_density**: This constant affects the population density of Dots.
- **death_chance_maj**: A *"majorant"*(maximum) taken for the Dot's age.
- **looking_chance_mean**: The mean, used for specifying the Gaussian function describing the transition STATUS_NORMAL → STATUS_LOOKING.
- **looking_chance_var**: The variance, used in the same Gaussian function.
- **looking_chance_p**: A coefficient used in the same function.
- **eating_time**: Number of steps that the Dot stays in STATUS_EATING.
- **generating_time**: Number of steps the Dot stays in STATUS_GENERATING.

With these variables, one can simulate distinct situations, only by editing the text file. You will find some instructions regarding the use of Dot.exe in the readme.txt file.

Conclusion

This program was made in C++ using Microsoft Visual Studio 2008. The initial version of Dots was delivered as a project for the subject of “Métodos Probabilísticos para Engenharia Informática” at “Universidade de Aveiro”. The OpenGL technology and the OpenGL Utilities were also used for a visual presentation of the simulations.

Now, this version can be published according to the GPL license. You might not learn as much as I did from this project, but I hope that you learn and enjoy Dots anyway.

If you want to contact me for suggesting, requesting advice, or even some nice feedback, feel free to use my e-mail address: enet4mikeenet@gmail.com