SimChamp Documentation

Team: **Crazy Ants**
Members:
JAYANT THATTE (EE09B109), ALAK
SAGAR SAHU (EE09B101), TAPTI

1 Introduction

- Ants are born in a nest at a **constant birth rate**, which can be **decided by the user**.
- Ants explore the arena in a random fashion till they find food, taking one step per iteration.
- **Location of food particles is set at random**, to make the code more realistic. The user can decide the number of food particles in the arena.
- Pheromone is assumed to **evaporate at a constant rate**, decided by the user.
- When an ant finds food, it leaves food trail, which other ants pick up. All ants **follow their own trail back** to the nest, irrespective of how they reached a food particle.
- When an ant encounters multiple food trails at the same location, it **picks that food trail which has higher concentration of pheromone**.
- Food sources are modelled with **finite capacity** and **finite area**. When a food source is exhausted, ants start exploring surrounding area for more food.
- **Obstacles** have been included in the simulation. User can specify the location of obstacle at the beginning of simulation.
- Ants may **die of hunger, aging or when they go too far away** from the nest (boundary of the arena).

2 Implementation

2.1 User Configurable Part

Birth rate, life time, maximum hunger tolerance, number of food particles, rate of evaporation of pheromone.

2.2 Finding Status of Ants

An ant can have two possible states: Exploration or Following a food trail.

Initially ants explore until they find either a food source or a trail. If an ant finds a food source it returns to the nest following its exploration trail and releasig food pheromone on the way, which other ants can pick up.

2.3 Movement

Any ant can move to one of eight pixels around it. Exploring ants do this by generating a random number.

Whenever an ant encounters a food trail, its checks for all possible, non-faded, food trails passing through its current location. It then chooses that trail which has maximum pheromone. This is achieved as follows:

History of all the exploring ants is stored in a 3-d array, with one layer for each ant. When an ant encounters a trail, it can determine which all ants have left a trail passing through its current location by searching in the history matrix. Thus all the available trails are identified. Then for each trail, pheromone value is checked at three points chosen randomly along the length of the trail. This then decides the trail with maximum pheromone. The ant chooses to follow this trail.

2.4 Generating Output

Output is displayed dynamically in a 250 pixel by 250 pixel window. All dimensions are scaled in proportion. During each iteration, pheromone and food levels are computed and displayed in the window.

- Green: Exploration pheromone.
- Red: Food trail.
- Blue: Food / Obstacle
  In the output matrix (named as 'pher' in the code), each food pixel is given an initial value of (R,G,B)=(0,0,1), while each obstacle is denoted as (R,G,B)=(0,0,2). Since food levels at a pixel can only reduce, food and obstacle can never be confused by the ants.
- Normalised Scheme of Display:
  Since a lot of ants may deposit pheromones at a given pixel, typically one or more of R, G or B values at the pixel would saturate. Hence most parts of the display would become coloured and the relative pheromone levels at different pixels would not be apparent in the display. To rectify this, the pixel with highest pheromone concentration (either exploration or food pheromone) is assigned a value of unity, and all other pixels are normalised with this value taken as base.