Appendix

Study of Ants' Collective Behaviours and Mechanisms to Reflect on Possible Implementations of Software / Robotic Simulations

Alexandre Jean-Pierre Vanini
Computer Science, IT University of Copenhagen, Rued Langgaards Vej 7, 2300
Research Project (K-CS and K-SD) Autumn 2020, KIREPRO1PE
alva@itu.dk

A Abstract model of the software simulation

The following piece of code is an abstract implementation of the possible software simulation.

```
World {
  var is_day // for visual only
  var temperature, day, season, time, speed, is_paused
  var global_population_increase_rate, global_population,
      noise_operation_magnitude
  var operating_cost // -> might be infered by above variable.
  var zoom //?? might be useful to scale every pixel

  Nest nests[]
  Terrain terrain
  ...
}

Terrain {
  var width, height
  Pixel map[width][height]
  ...
}

Pixel {
  var height
  var type // Nest, food, ...
  var label[^{
    colony : '',
    magnitude : int // Every time an ant of the colony walks on a pixel with its
        same chemical label, increase magnitude by a pre-defined offset.
  }] // Chemical left by ant, if exists.
  ...
}

Task {
  var id, description
  ...
  operating_cost(this): cost // Is a task a high operating cost or a low
      operating cost?
}

Nest {
  var origin // pos(x,y) of the very first pixel of the nest
  var surface, health, population_increase_rate

  Resource resources[{
    var food_supply,
    var water,
    var ?
  }]

  var nb_eggs

  Colony colony
  ...
}

Resource {
  var type
  var ?
  var quality // influence in task allocation
  ...
}

// Colony of ants
Colony {
```

```
  var label // refers to the chemical attributes
  var population_size // nb of ants
  Queen queens[] // One or many queens
  Worker workers[]
  ...
}

Ant {
    var type, health, size, speed, has_task, current_task, age, max_age
    var strenght // hit
    var walk_randomness // To what extend the ant walks straight
    var x, y // its position in the terrain
    ...
}

Queen extends ant {
  ...
}

Worker extends ant {
  ...
}

... extends ant {
  ...
}

// Some way of implementing a brain control
Brain {
    Behaviour behaviours
    /* An ant does not have rules, it only follows strict behaviours */
}

Behaviour {
  ...
}

Rule {
  var starving_cost? // one could imagine to have a dedicated class filled with
      such
  var food_distribution? // how is food distributes on the map
    ...
}

function war_cost() {
  /* Based on energy, nest.population_size and such, fitness function that tells
       an ant if the war is     worth fighting for */
}
```

B Master thesis draft time plan

MASTER THESIS: DRAFT TIME PLAN

PATH 1: Implementation of the full simulation, software only
PATH 2: Implementation of low-level behaviours in simulation + robot implementation

MONTH	WEEK	PATH 1	PATH 2	
1	1	preparation for ant brain implementation (creation of low-level simulation, make sure there's as much tool as needed in the simulation to implement everything in the following weeks) -> Implementation of abstract model		
	2	implementation: ant movement mechanisms		SIMULATION
	3	implementation: ant task allocation		
	4	implementation: ant task allocation + chemical badge		
2	5	implementation: tasks		
	6	implementation: operating cost of a task		
	7	Improvement of week 1 to 6	design and mount of the robot	
	8	Improvement of week 1 to 6	adaptation: movement to resources	
3	9	Improvement of week 1 to 6	adaptation: task allocation + badge	
	10	Addon: ant resources management	adaptation: task allocation + badge	
	11	Addon: Notion of time	adaptation: task allocation + task	
	12	Addon: Notion of time	adaptation: tasks	
4	13	Addon: Queens control over colony	adaptation: tasks	
	14	Addon: Environmental condition	adaptation: operating cost of a task	
	15	Focus on paper		
	16	Focus on paper + hand in		