**CE313 assignment**

**2016-17**

This assignment asks you to consider how you would implement - define and design and program - a system of many agents in a simple world.

The agents and the world are broadly but not fully described, so you may need to refine the description. That is an important part of the design.

The assignment is intended to allow a range of achievement, from a simple but coherent non-running sketch of a basic agent, to an elaborate and sophisticated system.

That makes it essentially impossible to give an exact definition of how to achieve any particular mark. However, it is expected that

* it will be fairly hard for any serious attempt not to secure a passing mark of 40%
* a strikingly good but not astonishing answer could get 90% (though a mark of 100% is very unlikely)

The assignment comes in stages. Each stage should either be answered separately, or, if an answer is for several stages, how each stage is answered should be made clear.

The answer to each stage should come in two parts:

* A discussion of your design, in words and diagrams and pictures and perhaps pseudo-code, for the system you would implement.
* The code for the system you implement, together with evidence that it runs (as far as it does), either as a trace, or as instructions for how to run the code.

The code should be Java or Python (or Prolong or Lisp). Other languages may be possible but must be agreed first. If code is to be run, the instructions for doing that need not be detailed, but they must be clear. The instructions should say which system/ IDE/ language implementation they are expected to run on, and it should be one routinely available in CSEE. If specialized systems are required, that must be agreed, and demonstrations may be required. Specialized systems such as Unity or Net Logo are unlikely to be agreed.

It must be clear from the code, + comments, + perhaps further description, how the code implements the design.

It is perfectly possible not to answer all stages of the assignment, and do pretty well. A good answer to all stages, or an excellent answer to some stages, could both do well.

An answer that is only design - good design - without implementation, can certainly pass adequately, but implementation as code is needed to do well.

**Stage 1**

The world is a grid of squares, each with a single tree in each square. There are several kinds of tree, though many fewer than there are squares.

There are a number of robots in the world. Each robot is in a single square, and there can be only one robot in a square - otherwise they crash into each other.

Time goes by in ticks. On each tick, each robot can do nothing, or perform a single action.

A robot can move ahead, or rotate one quarter turn left or right. A robot can always see the tree in the square it is in, and they can look and see the tree in any adjacent square. Robots can recognize the different sorts of tree.

Create a system that simulates and presents what happens in this world if the robots move pointlessly – randomly, or according to a simple rule about how to wander.

What happens if two robots try to move on to the same square at once?

*[Warning: it is easy to deal with this as just a simulation of a pinball table, with the agents as the balls rolling round an environment, as if driven by a sort of simple physics implemented by the simulation. The agents then seem not to be sensing or acting on the environment.*

*To avoid that*

* *Separate the environment and the agents.*
* *Each agent will have its position in the environment, and the simulation will know that position - but the agent will not, just as you do not know where you are when you are lost.*
* *Give each agent its own “inner state" which is affected by what they encounter and controls what it does, and is different from the environment.*

**Stage 2**

Robots have a health level. This goes down one unit per tick. If the health level reaches zero, they die.

A robot can eat a fruit from the tree of the square they are in. The level goes up, or down, or is unaffected, when they eat the fruit of certain kinds of tree - but they do not initially know what the effect of each sort of fruit on them is. After eating each sort of fruit, the robot can notice the change in its health level.

At this stage, the robots’ goal is to live as long as possible.

At each tick, a robot has to work out what to do. How does it do that at this stage?

**Stage 3**

Now the robots also want to map as much of the world as possible.

In order to map the world, they have to know what sort of tree is on which square. They know something if there is a record of it in their memory. They have only finite (though perhaps large) memories.

There is no time signal, so there is no absolute time - there is only "30 ticks ago".

There is no GPS, and squares are not named, so there is no absolute time - there is only "30 ticks ago".

What do they remember? How do they remember it?

*[If you do decide to assume that there is a time signal, or that there is GPS, be explicit about that. It is a considerable simplification.]*

**Stage 4**

At this point, a robot will sometimes be able to work out where it could get to by making some short sequence of moves starting from the current square, and what it could eat if it went there, and how good that would be. (Of course, if it is at an unfamiliar square, it might be puzzled, and could only go back or move at random.)

So

* suggest a simple design for how the robot might work out what it could do ...
* ... and then how it could decide what it should do ...

**Stage 5**

* ... and implement some of that suggestion

*[This is expressed in terms of "some of" because a simple implementation of a simple suggestion of how a robot might work out what to do is not hard, but a complete implementation of an advanced suggestion would be startlingly difficult.*

*In the past, the range of solutions for similar assignments has been very great. Given the width of that range, it seems right to accept a range of answers about what might be done.]*

*2016 12 14*