V-World Workshop 3: (more sophistication)
Exercise 4: Bumble 1.2

Note: ***this workshop is pretty much as Donald Nute designed it. It is useful in that it extends the capabilities of Bumble, effectively by making available & making use of more sensory data. You might consider that the new suggested predicate "move_toward_x/2" and his suggested table of directions is a little crude & you might decide that it is at this point that search should take over. However, the approach taken in this workshop could be seen as an incremental development of the agent's intelligence & capabilities.***

Notice that while a 5 x 5 piece of its world is visible to it, Bumble 1.1 only uses information about the smaller 3 x 3 piece of the world immediately adjacent to it. So it does not use all the information available to it. It seems likely that Bumble could survive longer if it used some of this information it is ignoring. Produce a new version of Bumble, Bumble 1.2 that displays the following behaviours.

1. If Bumble is hungry and there is no tree next to Bumble, but there is a tree two squares away from Bumble, and Bumble can move to an empty space next to the tree, then Bumble moves next to the tree.

2. If Bumble is hurt and there is no cross next to Bumble, but there is a cross two squares away from Bumble, and Bumble can move to an empty space next to the cross, then Bumble moves next to the cross.

3. If none of the above applies and Bumble is within two moves of something interesting or desirable (a key, bug spray, etc.,) then Bumble moves toward the desirable object.

A good way to implement these changes is to define a predicate

move_toward(Perceptions,Object,Direction)

that will pick an appropriate Direction given the agent's Perceptions and the Object you wish the agent to move toward. Then you can use move_toward/3 to define behaviours for Bumble where you need them. You will probably also need a table of directions that would move Bumble toward a square that is two moves away. For example, this table might begin with

move_toward_x(nwnw,nw).
move_toward_x(nwnw,n).
move_toward_x(nwnw,w).

Other than these changes, Bumble 1.2 should behave just like Bumble 1.1.

Save your code in a file called <yourname>- bumble_1.2.agt and test it in the worlds test1--test7.

4.1 Using the same evaluation criteria you used before, is Bumble 1.2 more or less rational than Bumble 1.1? Explain.

4.2 Bumble 1.2 should survive to 5000 most of the time in test1--test3, but it probably doesn't behave much better than Bumble 1.1 in later worlds. You may see some improvement; for example, it will find the key and at least pass through the first door more often in test7. But it still needs a good bit more intelligence to survive much longer than Bumble 1.1 in test4--test7. What additional abilities do you think Bumble needs to significantly improve its performance?

There is some code available as a possible implementation of the modifications and behaviours specified in this workshop. I have not included it here since you might find it instructive to do the work yourselves. However, if you would like to see the code and its rationale, a possible set of answers is available here.