

---

## Modelling Ant Navigation

---

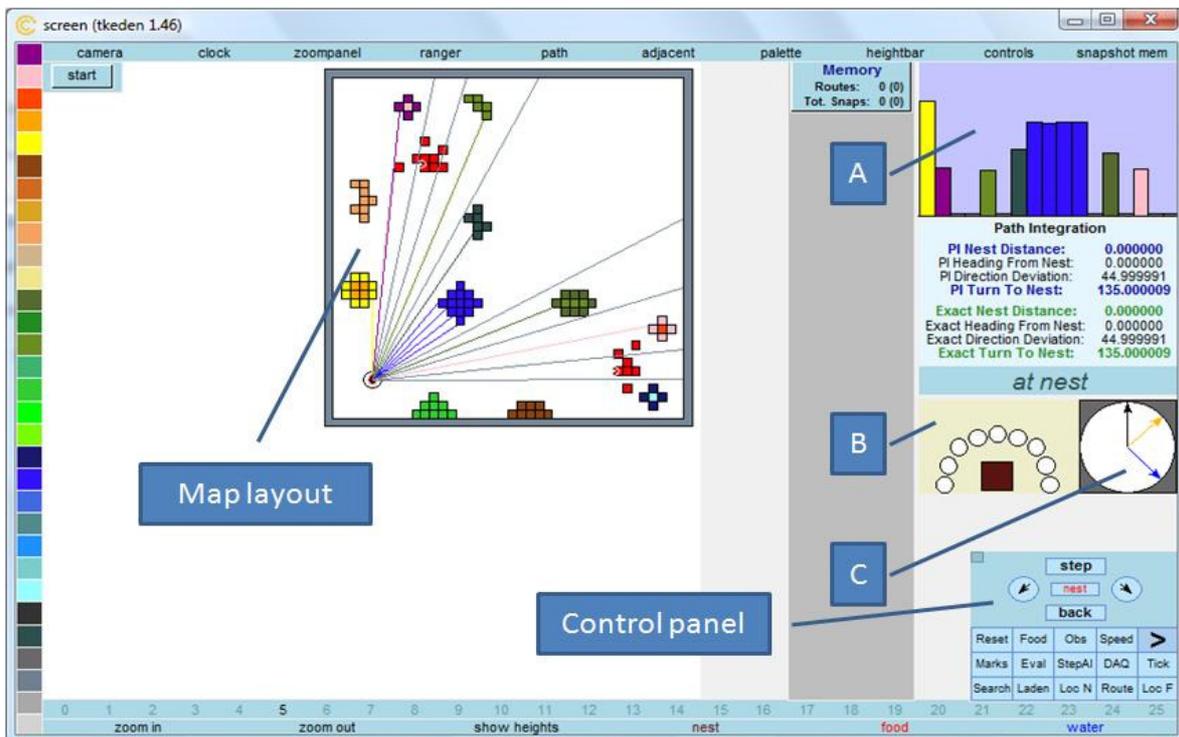
Different species of ant use different navigational strategies. The cataglyphis desert ants that are the subject of David Attenborough's video take snapshots that enable them to make a speedy return to the nest when they discover food. Other kinds of cataglyphis ant forage randomly and use similar snapshot techniques to find their way back to a food source from the nest. Modelling the behaviour of such ants was the theme of a final year computer science project by Daniel Keer in 2005. His project was based on research into ant navigation being carried out by biologists at Oxford. Daniel's model is built using unusual principles and tools that have been developed here at Warwick over many years by students working under the direction of Meurig Beynon and Steve Russ. We call our approach to computing "Empirical Modelling" or EM, for short.

---

To set up the Ant Navigation model, you should first download antnavigationKeer2010 from the EM Projects Archive, and invoke tkeden with the file `run.e`. (The model was developed with tkeden-1.46, and it may be appropriate to use this tkeden variant.) This will open a display labelled 'screen' that includes the following control panel (see pictures below). It will also be convenient to open a browser with this file in it, and the *input window* for tkeden-1.46.



To begin exploring the model, you should first organise the screen so that the screen display and the browser are easy to access, then press the 'Search' button on the control panel. The result is shown in this screenshot:



## Modelling observables, agency and dependencies

*The first step in understanding the model is to identify its key features and relate these to the account of desert ant navigation in the video. In the screenshot, the square region within the grey border is a map of the terrain in which the ant is foraging for food. The ant itself is represented on the map by a circle from which there are many lines emerging. Each line corresponds to a sight ray. What the ant can see in the direction of a particular line is determined by the objects scattered around the terrain. Each object is represented by a group of coloured squares, and there is a height associated with each square. The red squares represent food lying flat on the terrain that is not visible to the ant except when it is in the immediate vicinity. Notice that the sight rays 'pass over' the red squares, whereas they 'hit' the other coloured squares.*

The key idea in EM is to make a model in which there are counterparts for the sort of things that an ant can respond to (its "observables"), the ways in which it can change the state of its environment (its "agency") and the ways in which its observables are linked in change ("dependencies"). A key concept in EM is that understanding is gained through interacting with the model in an exploratory way. Your first exercise is to use exploratory interaction with the model to find out the significance of the three figures A, B and C that appear in the right hand column of the display. For this purpose, you can move the ant around manually by clicking on the "step", "back", "nest" buttons and the clockwise and anti-clockwise turning arrows in the top half of the control panel. You should be able to interpret the three figures by seeing how they depend on the position and orientation of the ant. You can confirm these interpretations by interacting with the objects in the terrain. For instance, you can modify objects in the terrain by selecting a colour from the strip on the left-hand side of the display and clicking on squares in the map to add and remove squares. You should be able to figure out how to specify the height at each square, and can also inspect the heights associated with squares by using the 'zoom in' button. You can restore the state of the model at any point by pressing the "Search" or "Reset" buttons on the control panel.

---

## Searching and route finding

*Experiments carried out with real ants suggest that the navigational behaviour of the ant is based on the following strategy. The ant initially searches for food in the terrain by randomly walking around the environment. During this search, it retains some reasonably accurate notion of where the nest is, as determined by a distance and a direction. This estimated nest location is thought to be determined as a side-effect of the physical movement of the ant's legs, so that the estimated location is revised step-by-step (a form of 'path integration'). When the ant finds food, it heads back directly to the estimated location of the nest, stopping at regular intervals to turn right around and record a snapshot of its current view. The estimated position of the nest is generally sufficiently accurate to enable the ant to locate the real nest by scanning the immediate neighbourhood. Once the ant has a sequence of snapshots in its memory, it can match these snapshots with what it sees so as to retrace the path from the nest to the food source in a direct fashion.*

To demonstrate the initial search for food, you should first press the "Search" button on the control panel, then press the "start" button at the top-left corner of the display. In the map layout, you should see the ant wandering around the terrain until it eventually finds some food (i.e. encounters "a red square"). Whilst the search is going on, the estimated position of the nest, as computed by the 'path integration' technique, will be displayed. When food is found, the ant turns round, then moves in the direction of the nest stopping to turn and take snapshots regularly. These snapshots are recorded as they occur in the column labelled "Memory" in the centre of the display. When the ant has returned to the nest, there is then a stack of snapshots which, when read from bottom to top, characterise the route that the ant will trace to return to the food source.

To demonstrate route-finding, you should press the "Route" button on the control panel. This will set up a situation in which the ant has two pre-constructed sequences of snapshots in memory, and will follow the first of these sequences to return to a food source. To initiate the route following procedure, you will need to press the "Start" button. Once the ant establishes its 'route following' mode of behaviour, it should successfully complete the journey to the food source, and return to the nest as before.

As you observe the searching and route following procedures, you can check out other information that appears in the right-hand column of the display. The data under the heading "Path Integration" gives details of where the ant thinks the nest is located at each stage. The text immediately below this panel, which initially reads *at nest* specifies the different modes in which the ant operates. This can be especially helpful when you interact in an experimental manner whilst the ant is navigating, since it enables you to see how such interactions affect the ant's 'state-of-mind'.

---

## Experimenting with the ant model

*Making a useful model of ant navigation is very different from using the computer to implement the calculations in a well-established mathematical model (e.g. as in computing the orbit of a spaceship, or programming a maze-solving robot). It's not clear what it means to 'know' how an ant navigates - we can only guess at what an ant might 'see', 'know' and 'remember', in whatever sense we can use these terms. Real ants don't have infallible navigational mechanisms - they succeed only up to a point. No matter how clever the ant's navigation mechanism may be, success will always rely on some*

assumptions about the stability of the environment (the so-called 'frame problem' in AI). One of the virtues of an EM approach is that we can make a model that (a) reflects the openness of potential interaction in a real environment, and (b) is being constructed even as we are inventing a plausible navigation mechanism - so that we don't need to have a good theory of navigation in advance.

**Illustrating (a):** You can repeat the searching and route following procedures in such a way that you intervene as an "all-powerful agent". By using the manual controls, you can make it more or less easy for the ant to find food during its searching for instance. Note how the length and complexity of the search trail will influence how well the ant's path integration technique works. You can also introduce obstacles during the search process. When the ant is route following, you will find it surprisingly difficult to distract it from its snapshot matching behaviour by using the manual controls to turn it in different directions (in part because the snapshot matching process is itself of a trial-and-error in nature). You can however do more unhelpful things, like modify obstacles in the terrain whilst the route-finding is in progress. If the ant's route following procedure fails, this will be evident from the mode indicator in the right-hand column. Failure can mean that the ant cannot match the next snapshot to the current view after a reasonable number of turning and moving steps, something that can happen because the trial-and-error process enters a loop.

**Illustrating (b):** Perhaps the best way to appreciate Daniel's achievement in developing the route-finding mechanism is to explore what was involved for yourself. Press the "Route" button twice to set up the route following task, then see how well you fare in matching up the snapshots successively by manually manipulating the ant. This task is surprisingly difficult! (To make it a little easier, you can add on a new component to the display in which the target snapshot (as selected by user mouseclick) can be displayed in a way that aids comparison with the current view. To do this, press the bold ">" button on the control panel to bring up an alternative button menu, press the "CSS" button, then press the bold ">" button again to restore the original control panel menu. ) If you want a real challenge, try to do the matching without looking at the map - e.g. by holding a piece of paper between the right and left hand sides of the display and allowing your partner to monitor your progress as you match the snapshots. To assess how well you perform compared with Daniel's route following AI, you can check out the "StepAI" button on the control panel, which plays through the automated snapshot matching step-by-step.

The above exercises are just a taste of what is really behind the Ant Navigation model. To study the model in more detail, you have to interact via the EDEN input and observe output given in the terminal window. Through the input window, we can intervene much more dramatically than through the interface e.g. to relocate and reorient the ant by typing and accepting new definitions such as:

```
%eden
bug1_x = 34.0;
bug1_dir = bug1_dir + 1.0;
```

for instance. You can also study the output that is generated during the route following activity to monitor the snapshot matching process more closely.

---

### Some references and sources for further study

- The *Amazing Cataglyphis Ant* video can be found at the url: <http://www.youtube.com/watch?v=w9KDM4C1kVg>
  - The original ant navigation model can be downloaded as antnavigationKeer2010 from the EM archive at <http://empublic.dcs.warwick.ac.uk/projects/>
  - A paper about the AN model can be found in the EM publications archive at: <http://www2.warwick.ac.uk/fac/sci/dcs/research/em/publications/papers/110>
  - Relevant scientific papers devoted to ant navigation include:
    - T. Collett, P. Graham and V. Durier, Route Learning by Insects in *Current Opinion in Neurobiology* 13, pp. 718-725. Elsevier, 2003
    - R. Wehner, Desert Ant Navigation: How Miniature Brains Solve Complex Tasks in *J Comp Physiol A* 189, 579-88. Springer-Verlag. 2003
-