Common Unity issues:

* Public variables with values in scripts can be overridden by Unity Editor values – Andy often didn’t set the correct value.
* Not really a great way round this if there is a mix of public and private variables
* Can make public variables not editable in the editor
* Additionally putting everything into AntScales would help since these values can't be changed in the editor
* Ideally don’t touch the editor values for anything that will be changed permanently

Andy AntManager Errors

* IsOldNestOccupied not called from antmovement (very misleading function name)
* quorumAssessNoise was 2 instead of 0.1
* LineOfSight didn’t have distances adjusted for the different arena size
* Had a few extra code blocks, not sure the purpose of them
* Had the more complex older version for calculating LGUT
* Senses collider value was too small (taken from greg prefab value, but it was overridden in a script)
* Several other incorrect parameters (such as averageantenna = 2f which is 20x the diameter of the ants). Again lots of problems from not adjusted parameters or parameters taken from somewhere but they had been overridden somewhere else.
* When estimating the tandem leader position the angle was calculated incorrectly
* Senses Collider was tagged as ‘Ant’, so ants with it activated were avoid collisions from very far away

## Tandem Running Bugs

Meeting 11/07

* determine if simulations are deterministic across tick rates
* creates matrices of
* determine if simulations are valid – watch simulations & perform martin’s tests (speed accuracy/speed cohesion tradeoffs)
* SPACE booklet containing the science and a rundown of the software would be useful
* Could create a ‘code approval system’. Series of tests (determinism, expected emergent features) that future SPACE developers have to pass to push their code.

# Meeting 18/07

* Better to produce data in SPACE then analyse in an external program like python
* Tests for determinism with plots of executiontime/simtime vs tick rate and error vs tickrate. Could also automatically suggest the maximum effective tickrate for your computer.
* Tests for speedacc tradeoffs - 20 repeats/0-20 quorum (select maybe half)/colony 200/arenas were equidistant,larger,largest?.
* Accuracy is measure of ants assigned to the best nest / total colony size. Martin's varied between 80-100% from qthresh 0 -20. Other measure is NestChoiceAcc = number of ants in best > num ants in worst. (true/false). For standard .3/.7 nests this is essentially always true. see below
* Quite possibly that the high correct nest acceptance rate (essentially 100% for all quorums) is due to the difference in nest option values (.3-.7). would be interesting to do the same tests with closer values (.45-.55) to see how the speed/acc tradeoff graphs change.
* graph of ants that have assessed and accepted nest vs actual quorum - this is affected by recruiters and assessors spending time in the nest. (assess time contributes a lot).
* Quorum Assessment - percieved quorum should be a float instead, test the stdev maybe around 0.4-0.5. It's fine that quorum is only met once.
* oldnest quality should be so low it can never be accepted, like -10 or something
* Could have more general case tests for running simulations - maybe give warnings if something like arenasize->antsize is wildly off.
* Best to do all the scales stuff now - set everything to reasonable values, move everything to AntScales, mark the parameters taken from literature that should be locked in place (and other relationships defined by multiple parameters such as tandem run speed & failure rate).
* Tandem run success rate should be pretty high so current method is fine.

# Determinism Tests

# Speed-Accuracy Tradeoff Tests

# Scales

## Lengths – 1 Unity unit = 1cm or 10mm

### Ant Size

* Ant size – was previously very unclear about size of the ant, since it was represented by an upright capsule. I’ve rotated it, so the capsule now represents this volume of the ant shown below. Instead of antprefab being 0.1 scale it’s now 1, with everything else set to match
* The antennae are not included since these are simulated as raycasts – only the main body segments are included. For now I’ll assume body length to be in the region of ~2mm

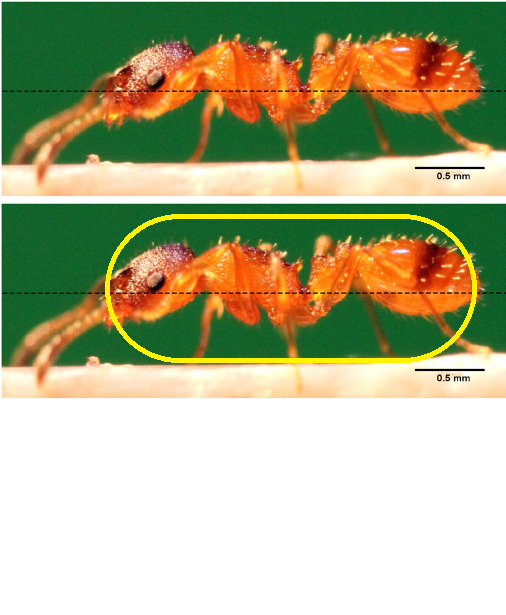


Figure 1 - Trail laying during tandem-running recruitment in the ant Temnothorax albipennis

* Antennae position function added, returns the front of the ‘capsule’. Should be used for all raycasts so that distances don’t need to take the ant length/radius into account (as before) 