Testing

# Test Plan

* Black box unit testing
* White box testing
* Integration tests
* State that erroneous data does not occur due to sliders
* Start that extreme data does not occur due to being a simulation

***Note: evidence for tests is found at the test id (where is test code?)***

# Test data

## misc\_tools unit tests

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test | Function | Tests | Reason | Input | Expected | Result |
| 1 | validateDirection | Returns an angle in the range 0 ≤ θ < 2π | **Typical** | 5π | 1π | **PASS** |
| 2 | **Typical** | 0 | 0 | **PASS** |
| 3 | **Typical -** Negative value | -2π | 0 | **PASS** |
| 4 | angleTo | Returns the correct angle to get from coord to target | **Typical** | coord : {x : 5, y : 5}, target : {x : 7, y : 5} | ½ π | **PASS** |
| 5 | **Typical** | coord : {x : 5, y : 5}, target : {x : 7, y : 7} | 3π/4 | **PASS** |
| 6 | **Boundary –** Coordinate on top of ant i.e. navigating to its own position | coord : {x : 5, y : 5}, target : {x : 5, y : 5} | π/2 | **PASS** |
| 7 | **Typical -** wrapped direction i.e. going off map to get to position | {x : 8, y : 8}, target : {x : 1, y : 1} | 3π/4 | **FAIL** - revised and passed with function update |
| 8 | **Typical -** wrapped direction | coord : {x : 7, y : 8}, target : {x : 3, y : 2} | ≈ 3.38657 | **PASS** |
| 9 | boundary | Returns the correct warped coordinate | **Typical** | {x : 4, y : 2} | {x : 4, y : 2} | **PASS** |
| 10 | **Boundary -** Both x and y boundaries | {x : 10, y : 10} | {x : 0, y : 0} | **PASS** |
| 11 | **Typical -** X-axis boundary only | {x : 10, y : 4} | {x : 0, y : 4} | **PASS** |
| 12 | **Typical -** Y-axis boundary only | {x : 7, y : 10} | {x : 7, y : 0} | **PASS** |
| 13 | clone | Produces an exact clone of an object | **Typical** | {a : [1, 2, 3], b : {c : 4, d : '5'}, e : {f : {g : 6}}, h : [['end', 'of', ['object']]]} | {a : [1, 2, 3], b : {c : 4, d : '5'}, e : {f : {g : 6}}, h : [['end', 'of', ['object']]]} | **PASS** |
| 14 | Produces a copy of the object i.e. not by reference | **Typical** | let b = {a : 1} | {a : 1} != b | **PASS** |
| 15 | coordToIndex | Returns the correct index for a specific coordinate | **Typical** | {x : 0, y : 0} | 0 | **PASS** |
| 16 | **Typical** | {x : 5, y : 2} | 25 | **PASS** |
| 17 | distance | Returns the correct distance | **Boundary -** Both the wrapped distance and the regular distance are the same | coord1 : {x : 0, y : 0}, coord2 : {x : 0, y : 0} | 0 | **PASS** |
| 18 | **Typical** | coord1 : {x : -2, y : 1}, coord2 : {x : 1, y : 5} | 5 | **PASS** |
| 19 | getBlock | Returns a block of cells the correct size | **Boundary -** 0 size | coord : {x : 5, y : 5}, size : {width : 0, height : 0} | A single block centred at {x : 5, y : 5} | **PASS** |
| 20 | **Typical** | coord : {x : 5, y : 5}, size : {width : 3, height : 3} | A 7 wide and 7 tall block centred at {x : 5, y : 5} | **PASS** |
| 21 | **Typical -** Non square | coord : {x : 5, y : 5}, size : {width : 1, height : 3} | A 3 wide and 7 tall block centred at {x : 5, y : 5} | **PASS** |
| 22 | Returns a block of cells the correct position | **Typical** | coord : {x : 1, y : 3}, size : {width : 2, height : 2} | A 5 wide and 5 tall block centred at {x : 1, y : 3} which wraps to the other side of the map | **PASS** |
| 23 | **Boundary -** Wrapped in all quadrants | coord : {x : 0, y : 0}, size : {width : 1, height : 1} | A 3 wide and 3 tall block centred at {x : 0, y : 0} which wraps to every corner of the map | **PASS** |
| 24 | getCellCoord | Returns the correct cell for a particular coordinate | **Typical** | {x : 7.2, y : 3.6} | {x : 7, y : 3} | **FAILED -** revised and passed with function update |
| 25 | getSector | Returns sector of correct radius | **Typical** | coord : {x : 25, y : 25}, radius : 15, direction : 0, angle : 2π | A circle of radius 15 centred at {x : 25, y : 25} | **PASS** |
| 26 | **Typical** | coord : {x : 25, y : 25}, radius : 6, direction : 0, angle : 2π | A circle of radius 6 centred at {x : 25, y : 25} | **PASS** |
| 27 | Returns sector at the correct angle | **Typical** | coord : {x : 25, y : 25}, radius : 15, direction : 0, angle : π | A sector of radius 15 centred at {x : 25, y : 25} with an angle of π i.e. a semi-circle | **PASS** |
| 28 | **Typical** | coord : {x : 25, y : 25}, radius : 15, direction : 0, angle : π/4 | A sector of radius 15 centred at {x : 25, y : 25} with an angle of π /4 | **PASS** |
| 29 | Returns sector in correct direction | **Typical** | coord : {x : 25, y : 25}, radius : 15, direction : π, angle : π /4 | A sector of radius 15 centred at {x : 25, y : 25} with an angle of π/4 pointing downwards | **PASS** |
| 30 | **Typical** | coord : {x : 25, y : 25}, radius : 15, direction : 6π/ 4, angle : π /4 | A sector of radius 15 centred at {x : 25, y : 25} with an angle of π/4 pointing along the centre line of the 4th and 3rd quadrants | **PASS** |
| 31 | **Typical** | coord : {x : 25, y : 25}, radius : 15, direction : 3.657, angle : π/4 | A sector of radius 15 centred at {x : 25, y : 25} with an angle of π/4 pointing along the centre line of 3.657 radians | **PASS** |
| 32 | indexToCoord | Returns the correct coordinate from a specific index | **Typical** | 25 | {x : 5, y : 2} | **PASS** |
| 33 | **Typical** | 0 | {x : 0, y : 0} | **PASS** |
| 34 | randColour | Generates random colours (run 3 times) | **Typical** | N/A | A HEX colour | **PASS** |
| 35 | randFloat | Returns a value within a specific range | **Typical** | {min : -1, max : 1} | In range -1 to 1 inclusive | **PASS** |
| 36 | **Typical** | {min : -0.1, max : 0.1} | in range -0.1 to 0.1 inclusive | **PASS** |
| 37 | **Typical** | {min : 0, max : 2} | In range 0 to 2 inclusive | **PASS** |
| 38 | randInt | Returns a value within a specific range | **Typical** | {min : 0, max : 100} | In range 0 to 100 inclusive | **PASS** |
| 39 | **Boundary -** No range of values | {min : 0, max : 0} | 0 | **PASS** |
| 40 | **Typical** | {min : -20, max : 20} | In range -20 to 20 inclusive | **PASS** |
| 41 | randProperty | Returns a random property from an object literal (run 3 times) | **Typical** | {a : 0, b : 1, c : 2} | either 'a', 'b' or 'c' (random each time) | **PASS** |
| 42 | scaleCoord | Scales coordinates correctly | **Typical** | {x : 4, y : 2} | {x : 23, y : 12} | **PASS** |

### misc\_tool unit tests – Evidence

Test 28

Test 25

Test 31

Test 30

Test 29

Test 27

Test 26

### misc\_tool unit tests – Failed tests

#### Test **7**

Test **7** failed as angleTo did not pick the angle which would lead to the shortest path to the target coordinate. The correct angle would have pointed of the map, as the wrapped distance was shorter than the normal distance. This was because angleTo did not take wrapped directions into account in its original form:

**function** angleTo(coord, target) {

**var** dx = target.x - coord.x;

**var** dy = target.y - coord.y;

**return** Math.atan2(dy, dx) + Math.PI/2;

}

The new function takes into account wrapping around the map to get the angle of the shortest path. It uses a similar algorithm to the one in the distance function.

**function** angleTo(coord, target) {

**if** (GRID\_SIZE.width **-** Math.abs(target.x **-** coord.x) **>** Math.abs(target.x **-** coord.x)) {

        dx **=** target.x **-** coord.x;

    } **else** {

        dx **=** GRID\_SIZE.width **-** (target.x **-** coord.x);

    }

**if** (GRID\_SIZE.height **-** Math.abs(target.y **-** coord.y) **>** Math.abs(target.y **-** coord.y)) {

        dy **=** target.y **-** coord.y;

    } **else** {

        dy **=** GRID\_SIZE.height **-** (target.y **-** coord.y);

    }

**return** Math.atan2(dy, dx) **+** Math.PI **/** 2;

}

#### Test **24**

Test **24** originally failed due to the getCellCoord function using Math.round rather than Math.floor in its operations, this meant that if the fractional part of x or y was 0.5 or greater the function would be out by one cell.

## Ant – unit tests

| Test | Function | Tests | Reason | Input | Expected | Result |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Ant.scan | Adds all ants within viewing distance to ant.itemsInView.ants | **Typical** | No ants in view | this.itemsInView.ants.length = 0 i.e. cannot see any ants in view | **PASS** |
| 2 | **Typical** | Multiple ants within viewing distance | this.itemsInView.ants.length = 4 | **PASS** |
| 3 | Adds food in viewing distance to ant.itemsInView.food | **Typical** | No food in view | this.itemsInView.food.length = 0 i.e. cannot see any food in view | **PASS** |
| 4 | **Typical** | Multiple pieces of food in view | this.itemsInView.food.length = 3 | **PASS** |
| 5 | Ant.secrete | Adds pheromone of correct concentration | **Typical** | No pheromones to start | A new pheromones of concentration 0.5 | **PASS** |
| 6 | **Typical -** Adding pheromone | A pheromone of the same species | A new pheromones of concentration 0.9 i.e. 0.4 + 0.5 | **PASS** |
| 7 | **Typical -** Not adding pheromones as different species | A pheromone of a different species | A new pheromones of concentration 0.5 | **PASS** |
| 8 | **Typical -** Doesn’t exceed MAX\_PHEROMONE\_CONCENRATION | A pheromone of the same species | A new pheromones of concentration 1 | **PASS** |
| 9 | Ant.atNest | Ant is on nest or not | **Typical** | not on the nest | FALSE | **PASS** |
| 10 | **Typical** | On the nest | TRUE | **PASS** |
| 11 | Ant.seeNest | Can see nest when in range | **Typical** | Cannot see the nest | FALSE | **PASS** |
| 12 | **Typical** | Nest within view | TRUE | **PASS** |
| 13 | Ant.smell | Adds all pheromones within range to pheromonesInRange | **Typical** | No pheromones in rang | this.pheromonesInRange.length = 0 i.e. cannot smell any pheromones in view | **PASS** |
| 14 | **Typical -** Shouldn't read pheromone it’s on | 4 Pheromones in range with one on top of the ant | this.pheromonesInRange.length = 3 | **PASS** |
| 15 | Ant.takeFood | Takes correct amount of food | **Typical** | void(0) i.e. No food | 0 | **PASS** |
| 16 | **Typical** | A single piece of food of amount 3 | After three runs of Ant.takeFood food = void(0) | **FAIL -** PASSED after re writing function |
| 17 | Ant.wonder | Ant picks correct direction given some amount of pheromones | **Typical** | No pheromones in view | this.direction === this.prioritizeDirection | **PASS** |
| 18 | **Typical** | A single pheromone | Ant is pointing at the pheromone | **PASS** |
| 19 | **Typical** | Multiple pheromones | Ant is pointing slightly upwards to the right | **PASS** |