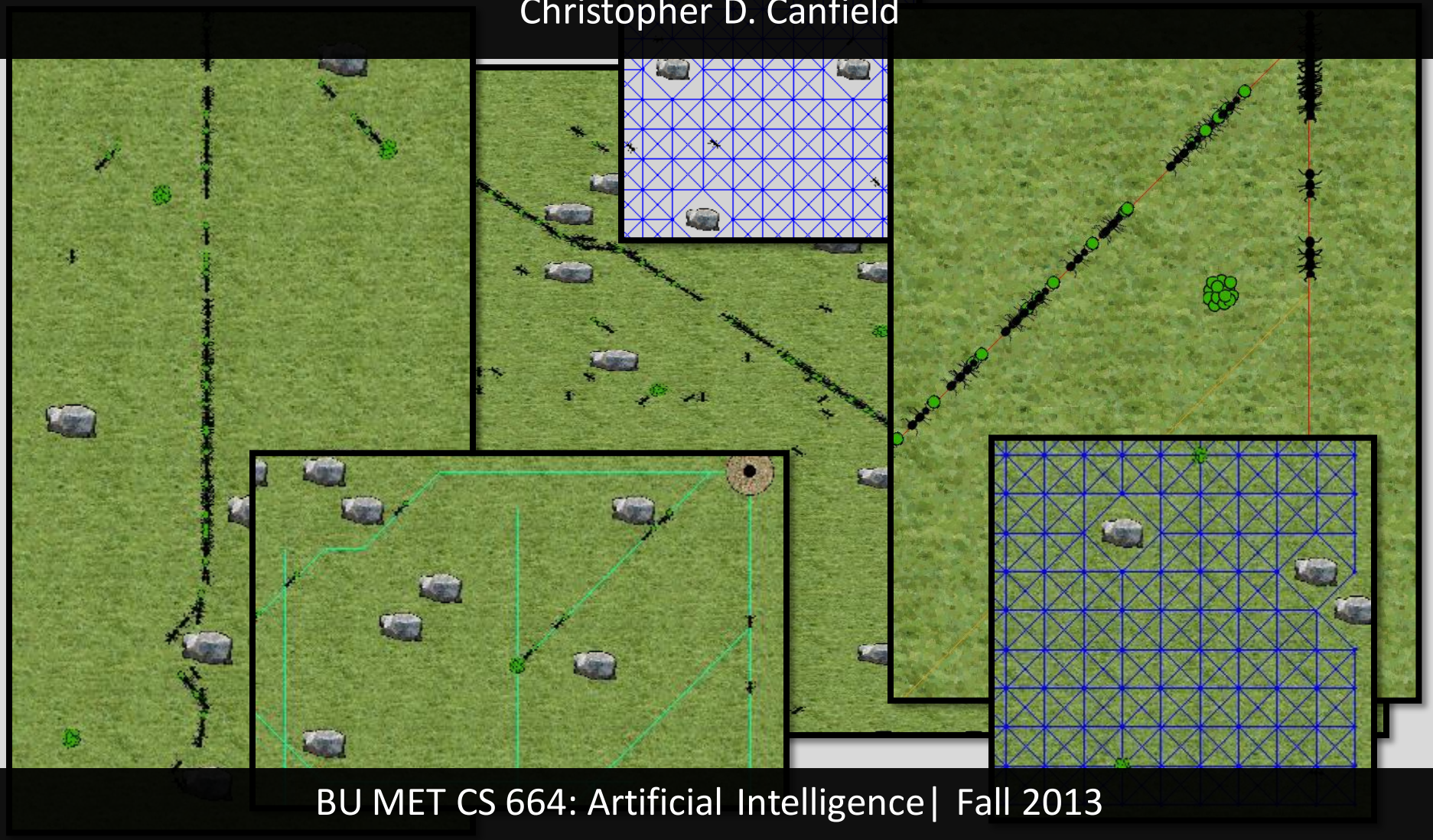
Ant Simulator:

Swarm Intelligence using Goal-based Agents & A*

Christopher D. Canfield



BU MET CS 664: Artificial Intelligence| Fall 2013

Project Overview

- Ant colony simulator: Swarm intelligence, goal-based agents, A* pathfinding
- Swarm intelligence: Many independent agents appear to coordinate actions, as if controlled by a higher-level coordinating AI
- Ants are a good example of this: they use pheromones to notify other ants about the location of food[1] and "to alert nestmates for colony defense."[2]
- *ants, termites and stingless bees mark the route between their nest and discovered food sources with a chemical (pheromone), thus indirectly leading nestmates to the food.*[1]

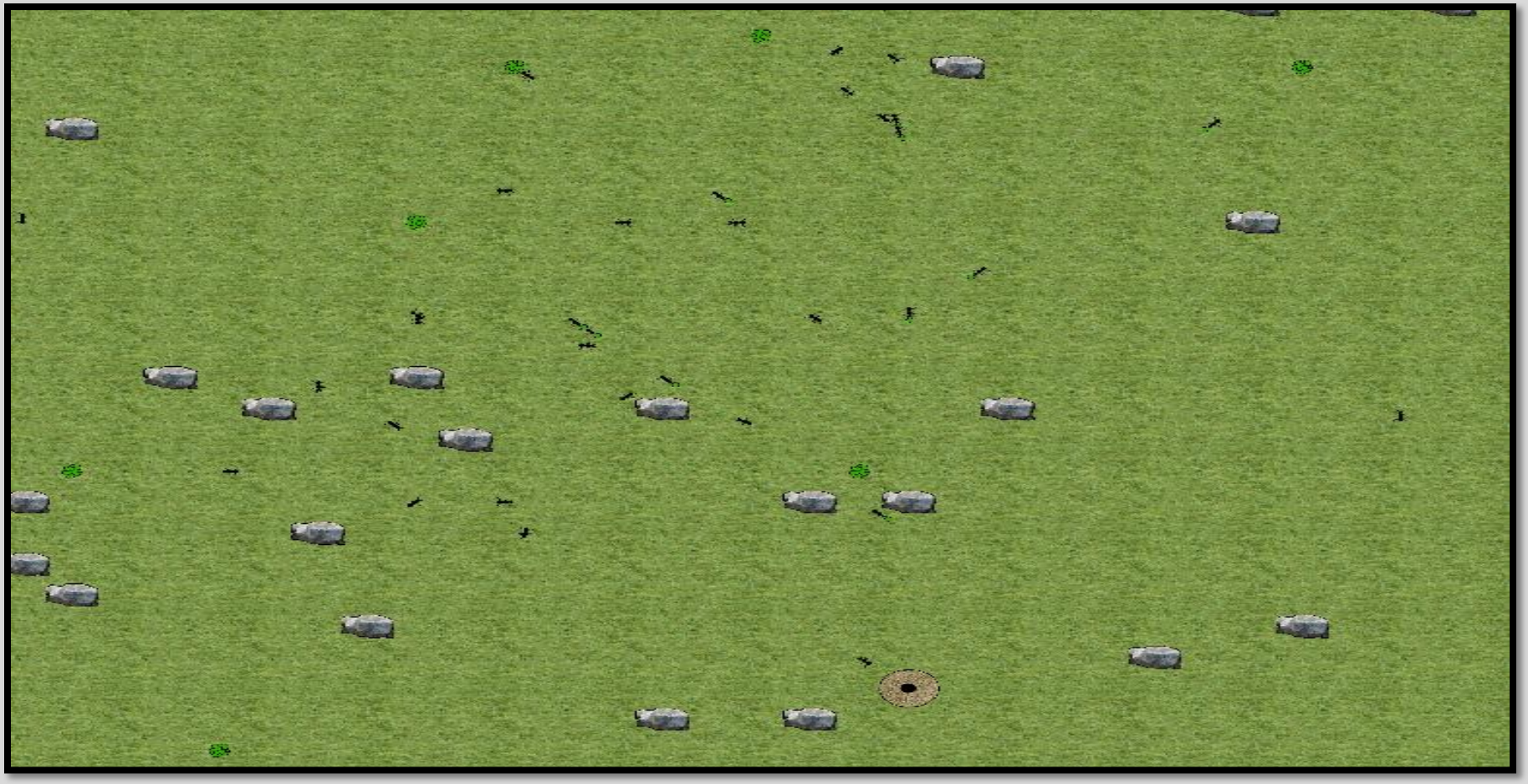
1. A. Dussutour, S. C. Nicolis, G. Shephard, M. Beekman and D. J. T. Sumpter, "The role of multiple pheromones in food recruitment by ants," *The Journal of Experimental Biology*. Aug 2009.
2. M. Mizunami, N. Yamagata and H. Nishino, "Alarm Pheromone Processing in the Ant Brain: An Evolutionary Perspective," *Frontiers in Behavioral Neuroscience*. vol. 4, article 28, June 2010.

Simulation Overview

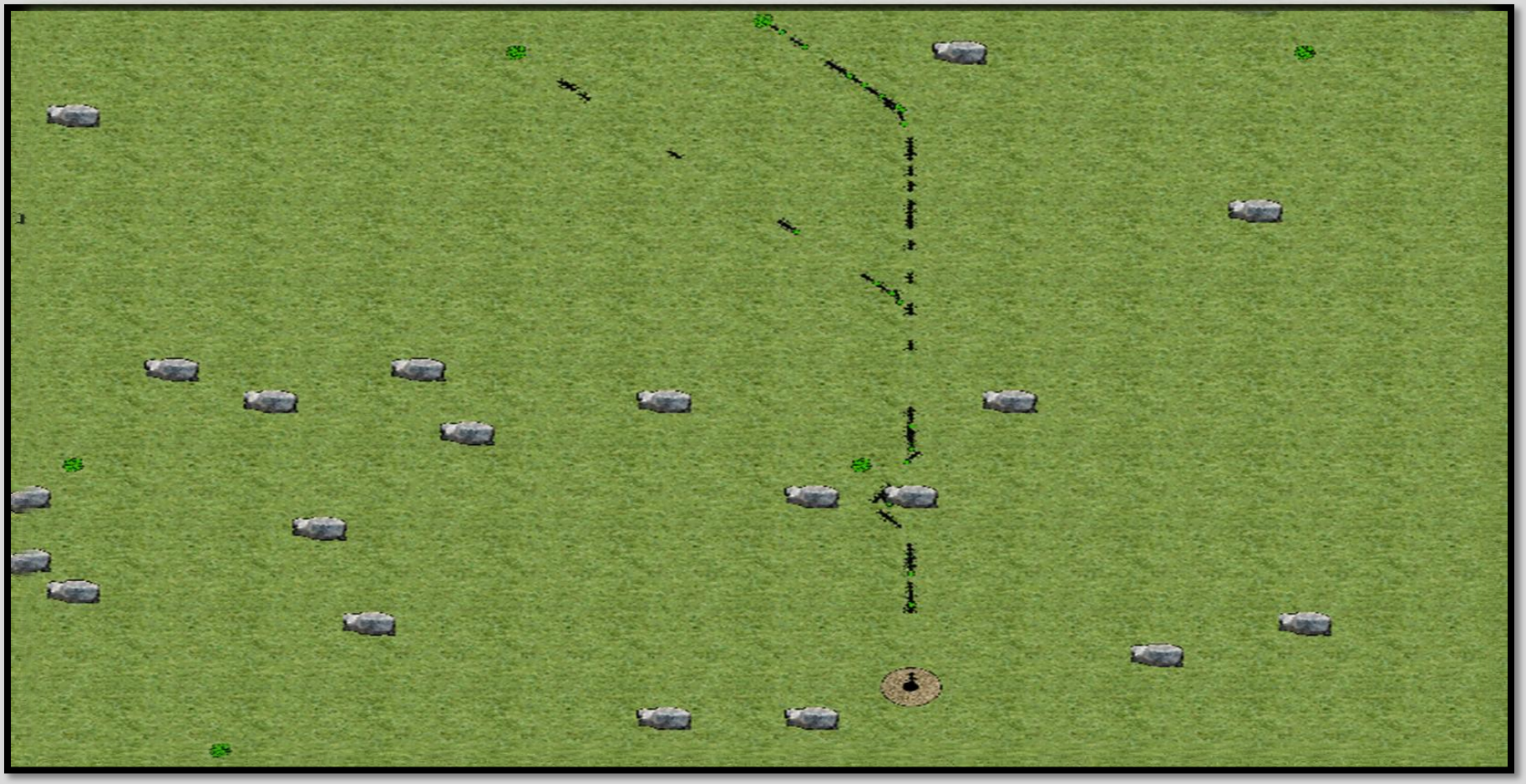
- World Creator: Random map generator
- Navigation Graph
- Ants: Goal-based agents
  - The most complex component of the simulation
- A*
- World has ants, navigation nodes & edges, pheromones, obstacles, an ant hill, and piles of green food

- Technical:
  - C++
  - SFML, Boost, POCO libraries for graphics, input processing, UUID, circular array…
  - Microsoft Unit Testing Framework

First: Did It Work?

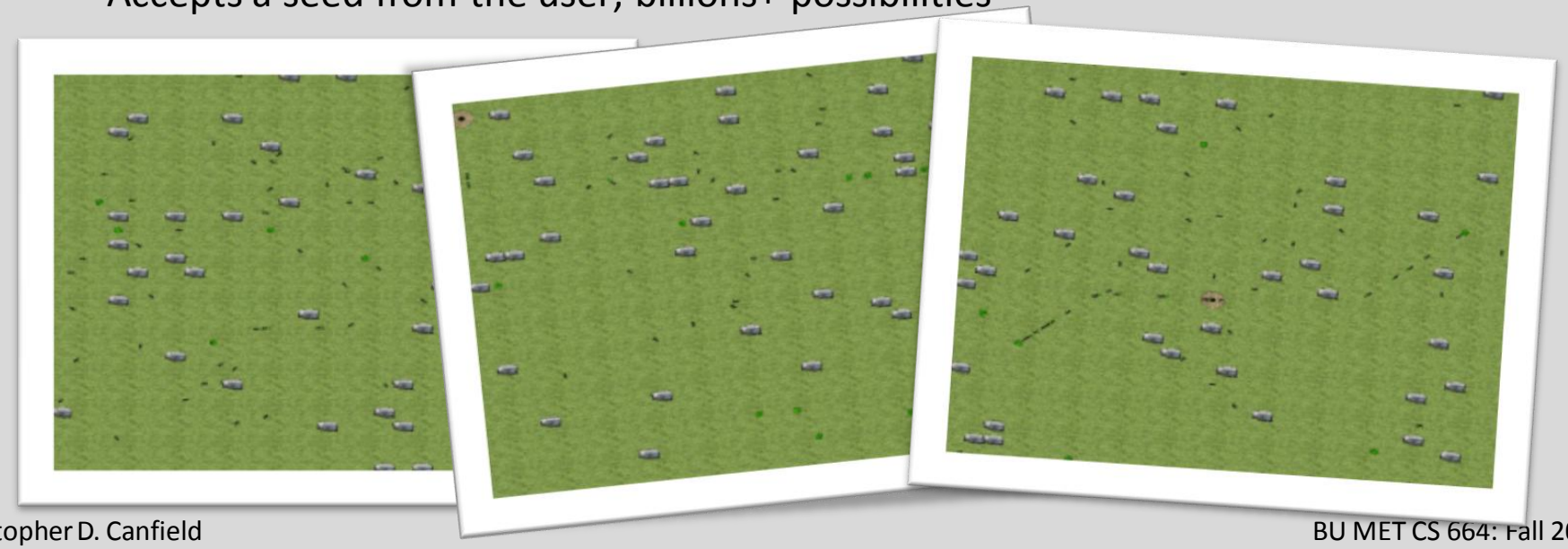


Yes!



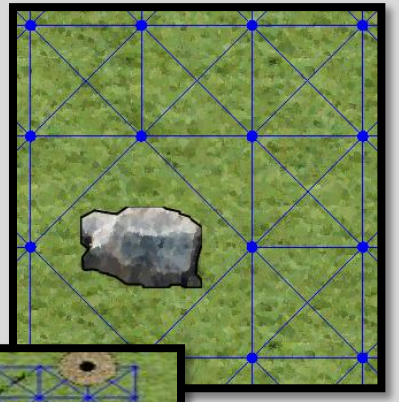
World Creator

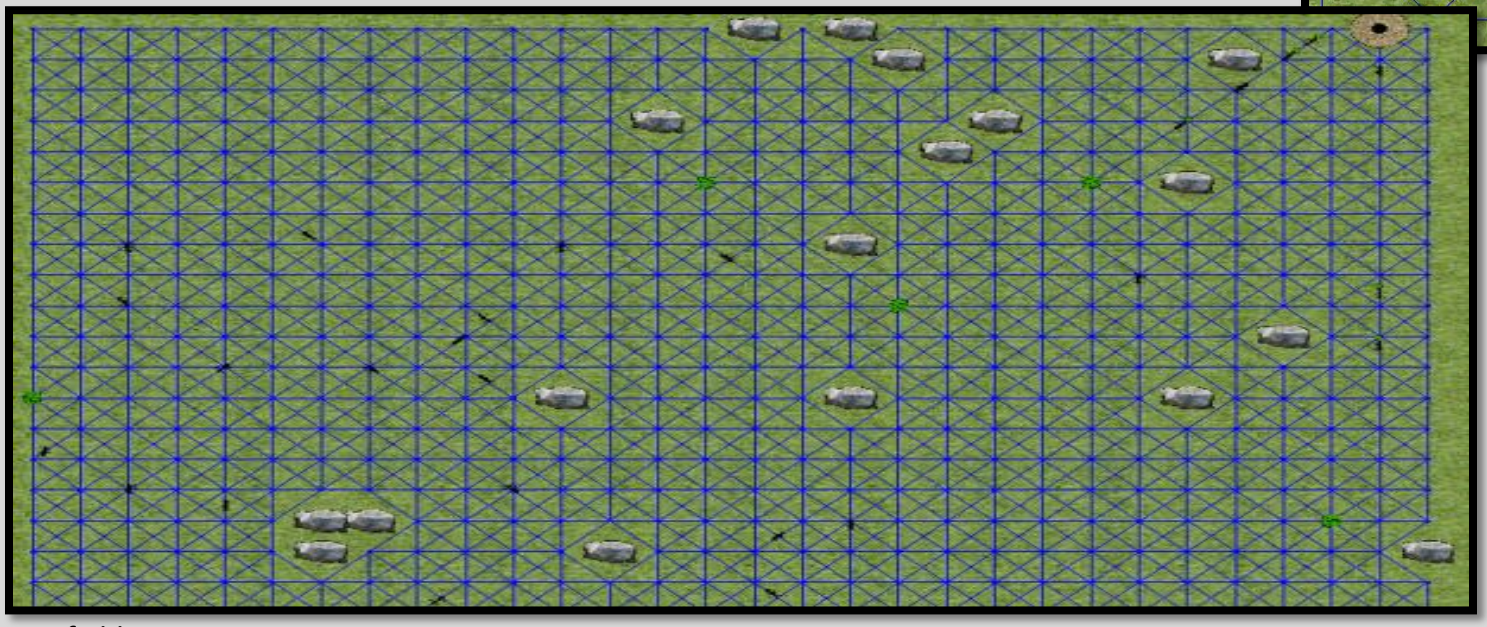
- Generates random maps with up to 900 navigation nodes and thousands of edges
- Places the ant hill, 10-50 obstacles (rocks), and 5-15 food piles throughout the map
- Places 30-60 ants in the ant hill
- Accepts a seed from the user; billions+ possibilities



Navigation Graph

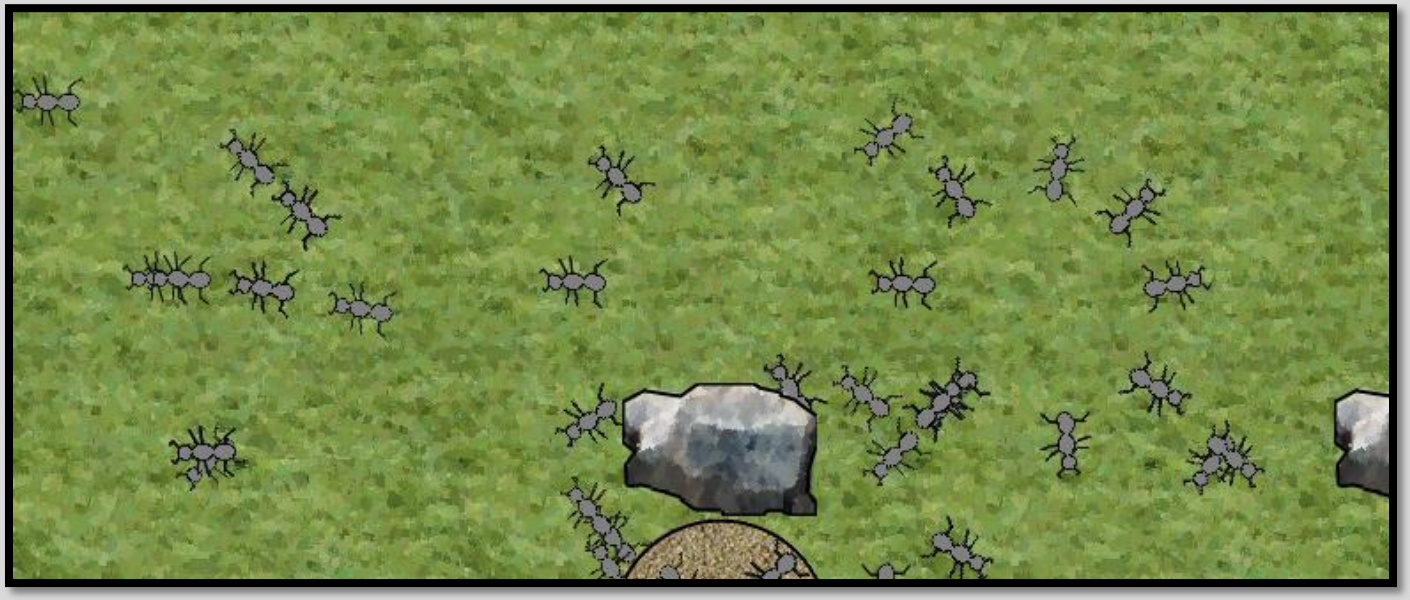
- Node
  - Used for A* start and end points
  - Static objects (rocks, ant hill, food piles) must be on a node
- Edge
  - A* g cost
  - Ants release pheromones onto edges





Ants: Overview

- Ants have goals and subgoals
- Ants need to eat, and will die if unable to eat for a long period of time



Gray ants have died

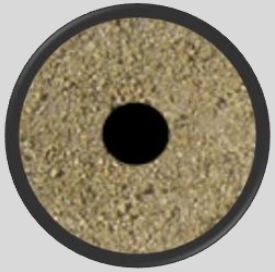
- Ants have a desire to bring as much food to the colony as possible

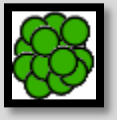
Ants: Percepts

- Ants can sense the following:
  - The current node, if any, plus connected edges (or the current edge if not on a node)
  - Pheromones, on edges
    - Pheromones are released on edges when ants pass them while carrying food
    - Pheromones degrade over time if not replenished, and eventually wear away completely. This is graphically simulated using red->orange->yellow lines
  - Food pile, if one exists on the current node
    - The ant cannot sense food piles that are not on the current node
    - Food piles have a limited amount of food, and do not regenerate

Ants: Knowledge Base

- Ants:
  - Do not know ahead of time where food is located
  - Remember the location of up to 3 food piles. Certain goals pop off the most recently found food pile from the knowledge base.
  - Know where the ant hill is located
  - Implicitly know where obstacles are located, because obstacles eliminate the navigation nodes and edges that are blocked





Ants: Goals & Subgoals

- The current goal may be set by the ant, when the previous top-level goal has completed, or by other goals and subgoals.
- Ants have 7 goals and subgoals (listed from simplest subgoals to most complex top-level goals):

1. AntMoveToNode
   - Instructs an ant to move from one node to an adjacent node
   - The simplest subgoal
   - Used by many other goals
2. AntFollowPath
   - Takes a target node, generates a path using A*, and moves the ant along the path
   - Uses AntMoveToNode to move between individual nodes

Ants: Goals & Subgoals

3. AntFindFood
   - Instructs the ant to attempt to find food by doing the following:
     - Check the locations of the three most recently found food piles, starting with the most recent location
     - If that does not work, select a series of random nodes in an attempt to find food or pheromones
     - If at any time pheromones are encountered, follow that path
   - Uses the AntMoveToNode subgoal
4. AntGoHome
   - Instructs the ant to return to the ant hill
   - Uses the AntFollowPath subgoal

Ants: Goals & Subgoals

5. AntExplore
   - Instructs the ant to explore random paths
   - If food is encountered, its location is noted and then the ant continues
   - The ant does not follow pheromones when it is pursuing this goal
   - Uses the AntMoveToNode subgoal
   - Top-level goal: Selected 1% of the time when a new top-level goal is needed, if the ant is not hungry
6. AntForage
   - Instructs the ant to search for food
   - If food is found, it is brought back to the ant hill, where it is deposited
   - A pheromone trail is released while the ant is holding food
   - Uses the AntFindFood subgoal
   - Top-level goal: Selected 99% of the time when a new top-level goal is needed, if the ant is not hungry

Ants: Goals & Subgoals

7. AntEat
   - Instructs the ant to find food for consumption
   - First goes to the ant hill, using the AntGoHome goal; if food is available, it is eaten and the goal is completed
     - Pheromones are ignored in this state
   - If no food is available there, it uses the AntFindFood subgoal to search for food in the locations of the three most recently encountered food piles.
     - Pheromones are followed in this state
   - If no food is found in any of the last three food pile locations, and no pheromones have been encountered, the ant will randomly search locations for food and pheromones
     - Pheromones are followed in this state
   - Top-level goal: Selected if the ant is hungry

Ants: PEAS

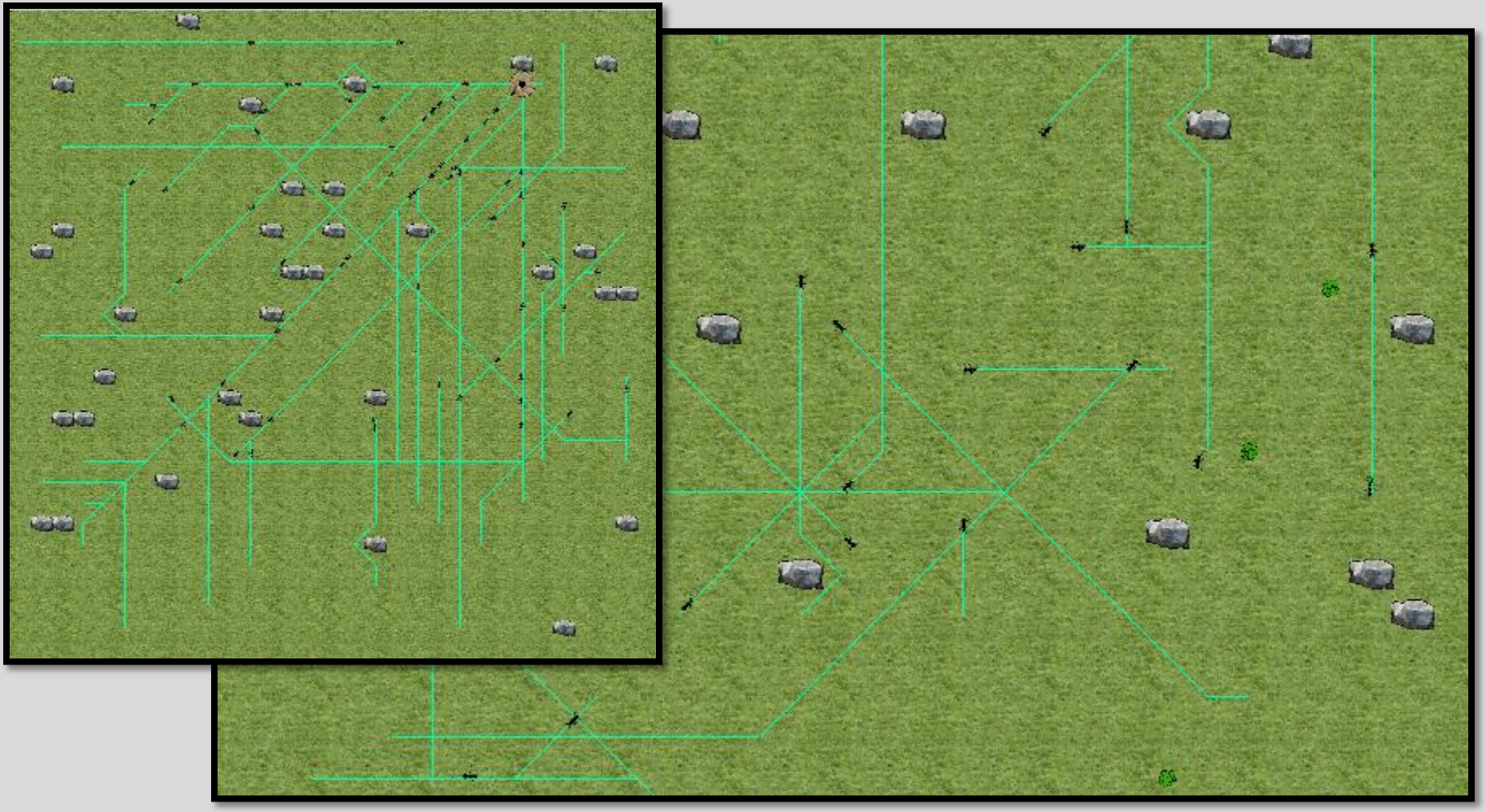
Performance	Environment	Actuators	Sensors
• Is alive • Is not hungry • Colony has food	• Partially viewable (knows nav graph, but does not know food or pheromone locations) • Partially dynamic (nav graph is static, but food disappears and pheromones appear and wear out) • Navigation graph with nodes and up to 8 edges per node • Obstructions, food piles, ant hill, other ants (no direct interaction), pheromones	• Move • Pick up & drop food • Lay pheromone trail • Eat food	• View current navigation node or edge • Smell pheromones on current edge, or on all edges connected to current node

A*

- A* method takes a heuristic function pointer
- Implemented Manhattan and Straight-line heuristic functions
- The simulation uses Straight-line, because the nav graph allows 8 directions of movement

- Despite extensive testing, initial implementation had flaws:
  - Paths could contain a loop (clearly not the optimal path)
  - Rarely, paths weren't fully connected (big problem!)
  - These issues were uncovered when I ported this to the Wumpus World
- Problem: I wasn't properly accounting for g cost (cumulative distance from start). Fixing that resolved the issues.

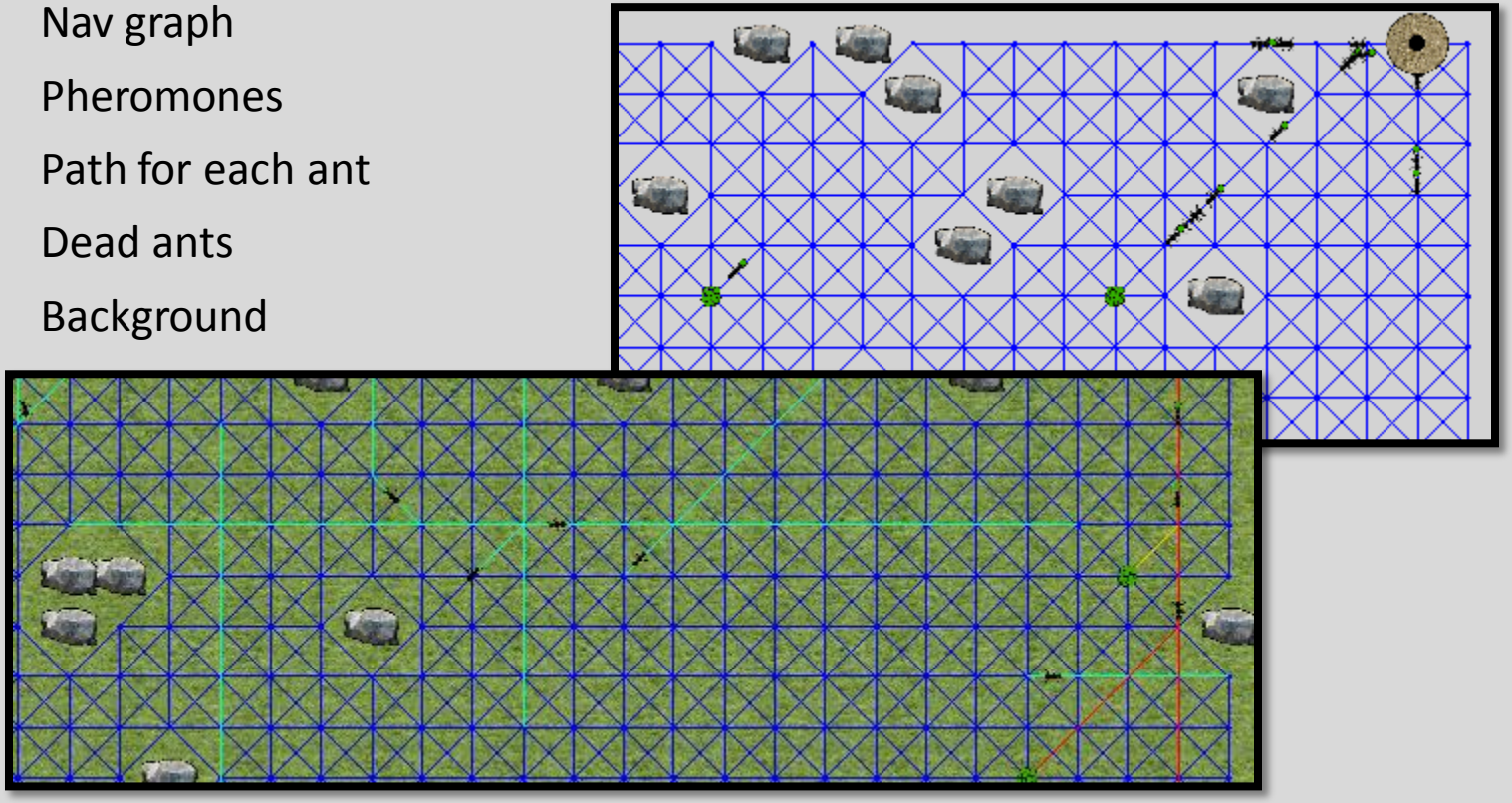
A*



Light-blue lines are paths calculated by A*

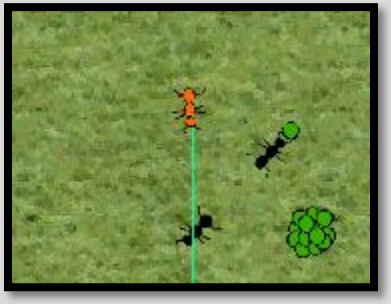
The Simulator

- The user sets the random map seed, and can set the following to display or not:
  - Nav graph
  - Pheromones
  - Path for each ant
  - Dead ants
  - Background



The Simulator

- An ant can be clicked to receive additional information, including
  - Current goal & all current subgoals
  - Hunger
  - Whether the ant is alive
  - The ant's path



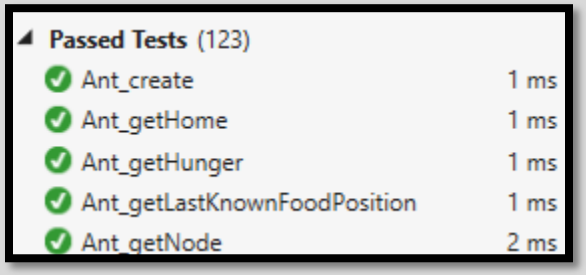
```
Ant 28c7e650-5936-45e9-ab0e-2febb0eb0e3a selected
  Hunger: 61% | Is Alive
  Current Goal: AntForage | Subgoal: AntFindFood | Subgoal: AntMoveToNode
```

- Colony-wide information is printed to the console every 10 seconds, including
  - Total number of ants (alive + dead)
  - Total number of living ants
  - Amount of food in ant hill

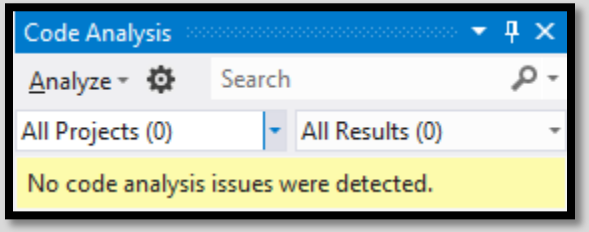
```
Simulator: Ants: 1124 | Live Ants: 1124 | Food in Anthill: 743
Simulator: Ants: 1208 | Live Ants: 1208 | Food in Anthill: 977
Simulator: Ants: 1309 | Live Ants: 1288 | Food in Anthill: 928
```

The Simulator: Verification & Testing

- Heavily test-driven: 123 unit tests in Visual Studio + 23 related JUnit tests
  - Includes 26 unit tests related to search, plus another 23 JUnit tests for the A* implementation used in my Wumpus Explorer project (different language, algorithm ported from Ant Simulator)
  - Microsoft Unit Testing Framework



- Visual Studio static code analysis



The Simulator: Verification & Testing

- 3 test applications:
  - GUI Tester: Tests gui implementation & related aspects, including various user input events, observer pattern implementation, drawing various items to screen, etc
  - A* Tester: Text-based output of paths generated by A*. Separate from the unit tests.

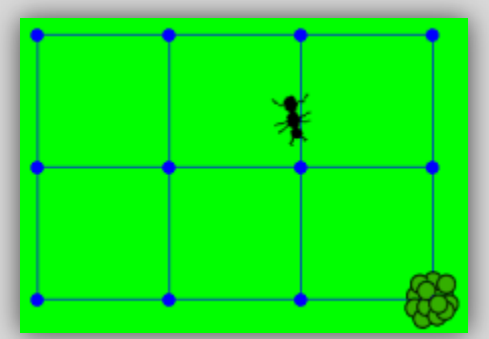
```
popped from frontier: 0,0
+reached end node; returning
 (1,2) (1,1) (1,0) (0,0)
```

```
start node: 2,2
end node: 0,1

popped from frontier: 2,2
|-searching edge: current node: 1,2
|---current node cost: 2.41421
|---found: 0
|---adding to frontier: 1,2
|-searching edge: current node: 2,1
|---current node cost: 3
|---found: 0
|---adding to frontier: 2,1
|-searching edge: current node: 2,3
|---current node cost: 3.82843
|---found: 0
|---adding to frontier: 2,3
```

  - Ant Goal Tester: Tests all ant goals using 7 test agents

```
Ant goal tester type:
  1: AntEat
  2: AntExplore
  3: AntFindFood
  4: AntForage
  5: AntGoHome
  6: AntMoveToNode
  7: AntFollowPath
  8: Pheromone Tests
  9: Run all tests
Type: _
```



Demo