The hackathon dry run

Bernhard Ager

October 26, 2015

Why are we here?

- Most of all: To have fun

Why are we here?

- Most of all: To have fun
- Laurent has been asked to organize the ETH hackathon
- ... and tasked me to handle the details
  - look for suitable games
  - implement a sample solution
  - modify game parameters ...
- We've finally decided to go with the Ants game (Google AI challenge 2011, `http://ants.aichallenge.org/`)

Why are we here?

- Most of all: To have fun
- Laurent has been asked to organize the ETH hackathon
- ... and tasked me to handle the details
  - look for suitable games
  - implement a sample solution
  - modify game parameters ...
- We've finally decided to go with the Ants game (Google AI challenge 2011, `http://ants.aichallenge.org/`)
- You guys ...

License: CC 2.0,
Petra Karstedt, 2005/2006.

  - please provide feedback at the end :-)

Agenda

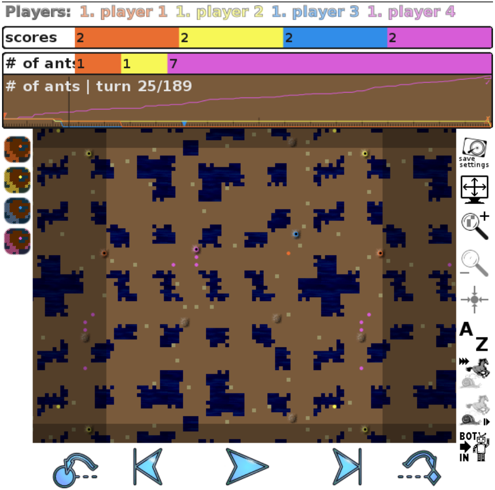
3.00pm	I present the challenge as well as some coding tips
3.30pm	You start coding, either in G71.2 or in your office
6.00pm	Food
8.00pm	The challenge ends. We meet again in G71.2 and watch the bots compete each other, live.
8.30pm	The winning team is elected and we all go to bQm to celebrate! First round is on Laurent. (Thanks!)

All times are rough estimates :-)

The Ants Game

- Player bots control hive of ants
- Goal of game: Raze (destroy) enemies' ant hills

Live demo



Players: 1. player 1 1. player 2 1. player 3 1. player 4

scores 2 2 2 2

# of ants 1 1 7

# of ants | turn 25/189

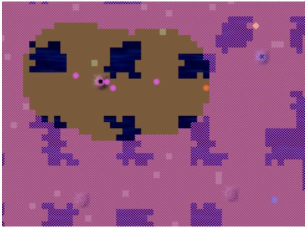
Scoring

- Points
  - Each bot starts with 1 point per hill
  - Every razed enemy hill: +2 to razer
  - Loosing a hill: -1
  - If there is only 1 survivor: Every hill not razed by end of game is torn down: +2 to survivor, -1 to owner
- What counts is the ranking between players!
- Same number of points results in same rank: Example 12, 5, 5, 1 points results in ranks 1, 2, 2, 3

Bot control

- Start of round: Bot gets updated about *visible* environment: Water, food, ants, hills
- Visibility is determined by `viewradius2` =77 (square of euklidean distance)

Fog of war



- For any number of ants, bot can issue an order: move to { n, e, s, w }
- attacking, food pickup, and food delivery performed *automatically* when within the resp. distances
- Ants die when ordered to go to the same field

How does the server evaluate?

1. Move (execute bots' orders)
2. Attack
3. Raze hills
4. Spawn ants
5. Gather food
6. Spawn food

⇒ Ants need to survive attack phase (a) to raze hills
(b) to gather food

Attack rules

- Battle mode: focus = “least distracted ants survive”
- Idea: Ants need to divide attention between all enemy ants in attack distance (`attackradius2` =5)
- An ant kills an enemy if it is less distracted than the enemy is

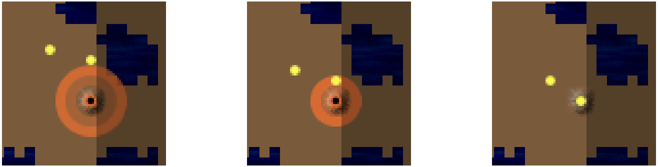
```
 .......
 ..545..
 .52125.         ......c        ......1        ......c
 .41a14.         c.a.b.c  -->  1 2 4 1   -->  c.!.!.c
 .52125.         ......c        ......1        ......c
 ..545..
 .......

Attacked       Situation     Distraction      Result
 Squares
```

- equally distracted: both ants die, e.g., in one-on-one
- If you want to know all the details, visit `http://ants.aichallenge.org/specification_battle.php`

Razing hills

- Place your own ant on an enemy hill.
- If it survives attack phase: Done.

⇒ Hill cannot be razed as long as there is food in the hive.



Ant Spawning

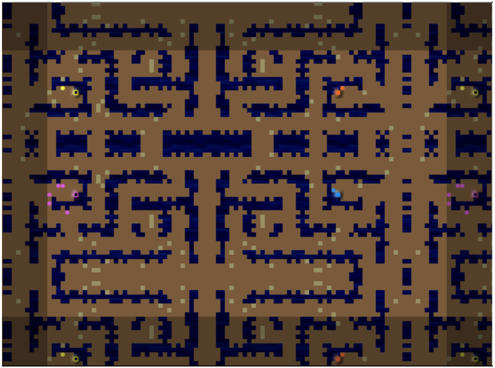
- If there is enough food in the hive, one ant per hill per turn spawns
- Hill must not be razed
- Spawning can be controlled by bot: put an ant on your hill to stop new ants from spawning

Food harvesting

- Harvesting controlled by `spawnradius2` =1 ⇒ place ant next to food to collect it.
- Changed from original game: *Food has to be carried back to hill.*
- Every harvested food item is stored in hive. One food = one new ant.
- Ants are very strong: Can carry up to 100 food.
  - Actually, this rule prevents food from blocking an ant's path :-)

Food Spawning

- At start, some food is guaranteed to be visible by each ant
- Food is placed symmetrical for fairness
- If two competing players are next to the same food, food disappears.
- Details complicated. You can quite safely assume: exact spawn locations hard to predict.



Endbot conditions

- Player has no live ants left
- The bot crashed
- Timeout (1s)
  - After a timeout or crash, ants remain on map and participate in attacks and food gathering, but cannot move any more.

Further information

Detailed description of all of the above as well as bot input format, bot output format, replay format(s) and map format available at `http://ants.aichallenge.org/specification.php`

Programming hints

Where to start?

- *Important:* You have to use *my* version of the game engine `http://people.ee.ethz.ch/~bager/tools-dryrun.tar.gz` (because rules about food have changed)
  - Also contains sample bots to get more ideas and to test against
- Starter kits in 26 programming languages available on the AI challenge web site: `http://ants.aichallenge.org/starter_packages.php`
  - Implementation of communication protocol
  - Some basic state tracking, e.g., map building
  - Easy access to interesting objects: ant locations, hill locations, food locations
  - Only reports *visible* items though

Which of my ants carries food?

Unfortunately, you have to track that yourself!

- Food is picked up when ant is next to food
- Remember locations of ants with food
- When ant moves, update location of ant

Getting this right is tricky ...

Some advice

- Run time limitation per round is very strict
  - Avoid single long calculations
  - Regulary check how much time is remaining
  - Stop calculating new moves when time budget is low, e.g., `if self.ants.time_remaining() < 100: break`
  - Think carefully how you want to trade-off programming efficiency with run-time efficiency
- Don't over-engineer! 5 hours is very little time to come up with a useful bot.

Useful algorithms

- Depth first search (DFS)
- Breadth first search (BFS)
- A* search
- Minmax search (with alpha-beta pruning)

These algorithms are useful, but sometimes alternatives are better!

Debugging

- Timeout, randomness, and communication channels make interactive debugging a pain ⇒ use log files or stderr
  - You could increase timeout value (`--turntime`) and fix random generator seed values (`--player_seed` and `--engine_seed`)
  - Still: stdin/stdout are used for communication with game engine ... may be hard to overcome
- Use `-e` to make playgame.py report stderr
- `-R` logs game for replay into directory given by `-l`
- Check play.sh for a live visualization
- When replay logging is enabled, open 0.replay.html in your browser for an HTML5 visualization

Profiling

Python problem: Profiling with profile or cProfile does not work with starter kit: Instead, use

```python
def __init__(self):
    # [...]
    self.turn = 0
def do_setup(self, ants):
    # [...]
    self.profile = profile.Profile()
    self.profile.enable()
def do_turn(self, ants):
    self.turn += 1
    if self.turn == 100:
        self.profile.disable()
        self.profile.dump_stats("/tmp/mybot.stats")
    # [...]
```

Hackathon rules

- This is a dry-run for a real event ⇒ *Don't talk to people outside the group about anything related to this event.*
- Copy&paste from competition bots strictly forbidden. Violations lead to disqualification.
- It's perfectly ok to get ideas, though.
- Feel free to discuss with other teams and/or run against their bots.

Competition evaluation

- Your bot has to be able to run on Ubuntu 14.04.3 LTS
- 2 GB of RAM for each bot (plenty!)
- Intel(R) Core(TM) i7-2600K CPU @ 3.40GHz
  (4 hyper-threaded cores = 8 virtual cores)
- Multiple runs on different maps
- Lowest average rank wins the competition
- I'll install extra software (libraries etc.) if reasonable effort is needed to do so and at reasonable prior notice.
- Bot submission:
  - Source code only (no binaries!)
  - Build and run instructions in a text file ("README")
  - Bot executable should contain team name
  - via email to `bernhard.ager@tik.ee.ethz.ch`
- Play fair!

Organizational

- 4 teams of two (and three)
- Join NSG slack channel #hackathon-dry-run for discussion and coordination
- Food ... pizza?
- In case of trouble, ask me. I'll give support as much as time allows. Excluded from this offer: low-level programming and debugging help.

Summary

- Bots controls ants and try to raze other player's hills
- Take *my* version of the "tools" and choose a starter kit in your preferred language
- You have very little time: be smart about how to use it

Have Fun!