Multi-Agent Contest 2009 Edition

Tristan Behrens, Mehdi Dastani, Jürgen Dix,
Michael Köster, Peter Novak

Overview

1. contest introduction
2. teams
3. videos
4. winner announcement

Aim

- stimulate research in the area of multi-agent systems
- identifying key problems
- collecting suitable benchmarks that can serve as milestonesfor evaluating new tools, models, and techniques

Challenge

Solve a cooperative task in a dynamically changing environment.

Brief History

2005:	First CLIMA Contest
2006:	Second CLIMA Contest
2007:	First ProMAS Contest
2008:	Second ProMAS Contest
2009:	Third CLIMA Contest

Discrete Simulation

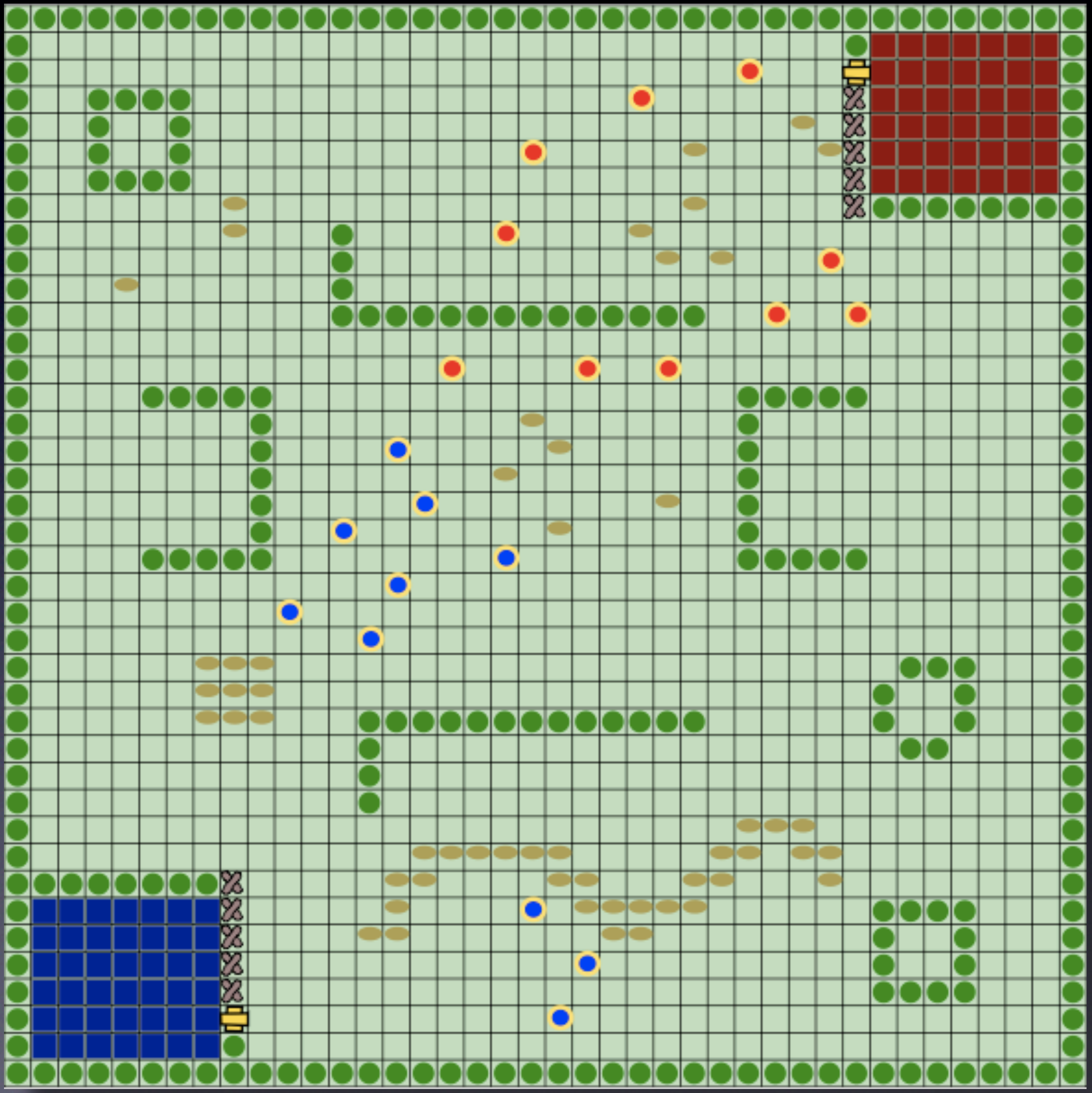
In each step do

- send perceptions to agents
- wait for agents' actions or timeout
- let agents act and move cows

Tournament Structure

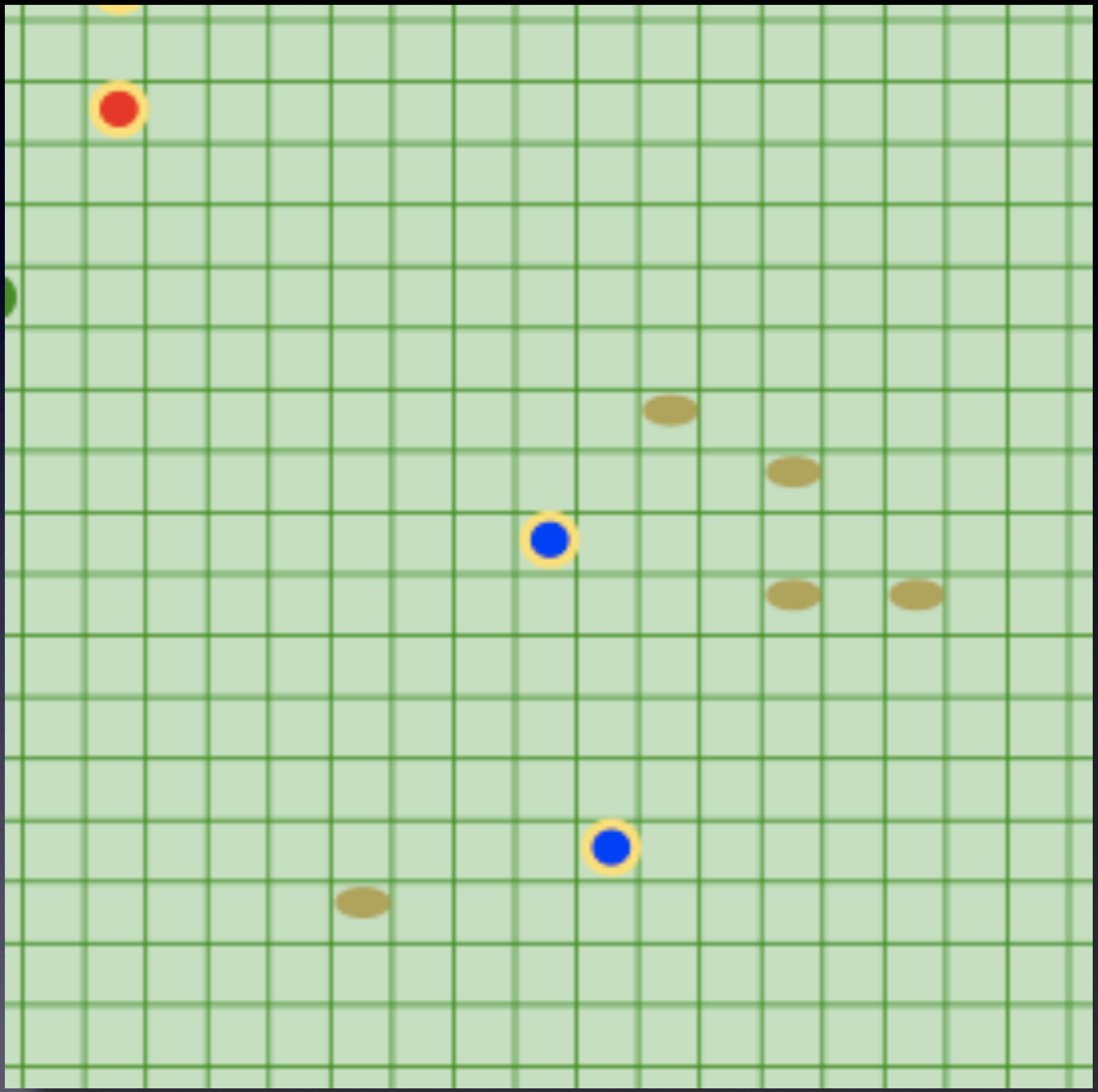
- maximum step duration around 4 seconds
- 1000 - 1400 steps per simulation
- 3 simulations = 1 match
- each team plays against all others, 1 match per pair

Cow-Herding Scenario

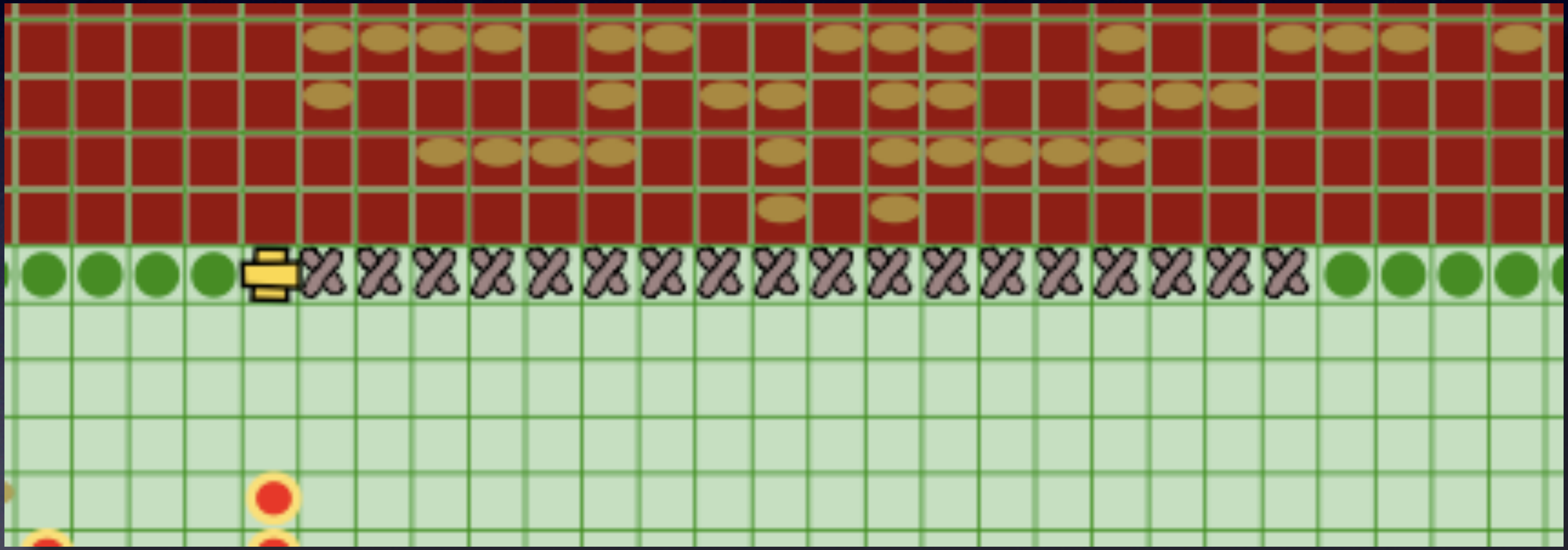


- cows
- cowboys
- obstacles
- fences
- corrals

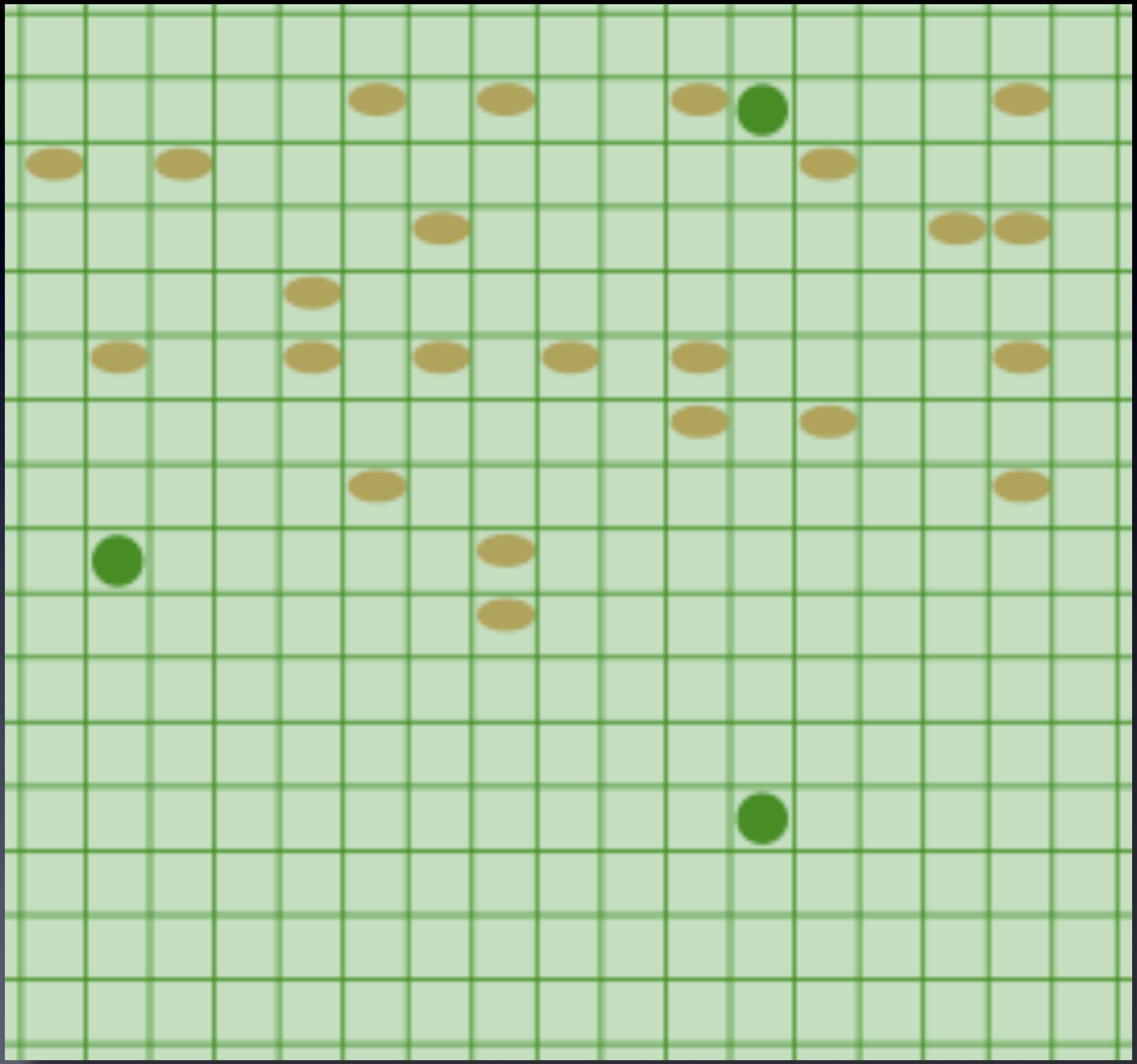
Perception Range



Fences



Cows



Teams

AF-ABLE

Team:

- Rem Collier, Mauro Dragone, David Lillis, Jennifer Treanor, Howell Jordan, Greg O'Hare - University College Dublin, Ireland

Solution Architecture:

- AFAPL2 agents running on Agent Factory platform
- Low-level behaviours (explore, open fence, etc.) written in Java
- Centralized task allocation based on costs and values

AF-ABLE

Results:

- Behaviour code complex and buggy, caused inconsistent performance
- No successful offensive behaviours - many cows conceded
- No defensive behaviours - many successfully-herded cows escaped or 'stolen'!

Lessons learned:

- Agent-oriented software must be engineered not built!
- Automated test suite for behaviours is essential
- Next time: move more logic to the agent layer?

Jadex@HAW

Second time participation in the Multi-Agent Contest.

Team:

- Gregor Balthasar - Developer - MSc student - MMLab, HAW Hamburg
- Jan Sudeikat - Supervisor - Ph.D. student - MMLab, HAW Hamburg
- Wolfgang Renz - Supervisor - Associate professor - MMLab, HAW Hamburg

Jadex@HAW

- using the Jadex BDI-agents middleware (v0.96)
- all algorithms in "Plans" implemented in Java
- using decentralized coordination contrary to the centralized coordination
- from the first participation utilized the Tropos methodology and toolsets as a guideline

Jadex@HAW

Results:

- a way more stable MAS which needed nearly no supervision
- increased the autonomy of the single agents drastically

Difficulties:

- debugging of decentralized coordination
- covering all situations that can appear during a simulation

Jason-DTU

Participants from DTU Informatics:

- Niklas Skamriis Boss & Andreas Schmidt Jensen - Developers - MSc students
- Jørgen Villadsen - Supervisor - Associate professor

Department of Informatics and Mathematical Modelling
Technical University of Denmark (DTU)

Jason-DTU

Starting point:

- New advanced AI course in spring 2009 with more than 50 students
- Included two lessons and a project on logic-based agent programming using Jason
- Coded algorithms such as A* in Java
- Used the Prometheus methodology as a guideline
- Code from last year for the integration with the contest simulator was provided by the 2008 Jason team (cf. RomanFarmers)

Jason-DTU

Main results:

- Jason framework allowed for easy implementation of agents with goals and plans
- Three kinds of cowboys: herders, a scout and a leader
- design with a single leader delegating targets leads to a less autonomous approach
- choice to heavily limit the number of cows in a single cluster is probably not optimal
- Did not implement a strategy to prohibit the opposing team from scoring points

JIAC V

JIAC V has been developed by researchers of the CC-ACT of DAI-Labor at TU Berlin, Germany

- Cracking the whip: Axel Hessler
- Valuable Contributions: Thomas Konnerth, Jan Keiser, Benjamin Hirsch, Tobias Kuester, Marcel Patzlaff, Alexander Thiele, Tuguldur Erdene-Ochir
- 3rd participation in the Contest

JIAC V

- JIAC meta-model is the frame for design and implementation
- ontology-based world model (beliefs and communication vocabulary)
- Domain dependent (cowboy) capabilities (plans, rules) are aggregated to roles, composed with standards roles (memory, communication) to form the agent and then executed by JIAC runtime
- Decentralized coordination and cooperation
- own agile MIAC methodology and our JIAC Toolipse guide the process

JIAC V

Results:

- contest is a cool scenario for teaching AO principles
- found and fixed many bugs: lifecycle and execution cycle of agents, interpretation of the world model
- tuned several core components (performance and reliability): improved performance by factor 8
- implemented many features that make the life of the AO programmer easier (easier to learn, easier to use, easier to debug, easier to deploy)

MicroJIAC

Team:

- Developer: Anand Bayarbilig, MSc Student, TU-Berlin,
- Supervisor: Erdene-Ochir Tuguldur, PhD Student, DAI-Labor, TU-Berlin

MicroJIAC

- using MicroJIAC agent framework developed by DAI-Labor
- model-based reflex agents: world model, rules
- all agents are equal and there isn't any specialised agent
- fully self-organized
- coordination/cooperation is reached by sharing perceptions/intentions

MicroJIAC

Results/Difficulties:

- the scenario is too complex for one student
- driving a single cow
- debugging of self-organization is complicated
- the agents were constructed for maps with many fences

RomanFarmers

Participants:

- Jomi F. Hübner (Federal University of Santa Catarina, Brazil)
- Rafael H. Bordini (Federal University of Rio Grande do Sul, Brazil)
- Gustavo Pacianotto Gouveia, Ricardo Hahn Pereira, Jaime S. Sichman (University of Sao Paulo, Brazil)
- Gauthier Picard (Ecole des Mines de Saint-Etienne, France)
- Michele Piunti (Universita di Bologna, Italy)

RomanFarmers

Design based on three paradigms and abstraction levels

- Organisation Oriented Programming (MOISE): define groups and shared plans and goals to herd, explore, pass fences, ...
- Agent Oriented Programming (Jason): define how goals are achieved by the agents
- Object Oriented Programming (Java): define algorithms to find paths, cluster of cows, ...

RomanFarmers

Results:

- improvements on Jason and MOISE
- found only 1 bug in Jason

Main difficulties:

- debugging (several agents, tools, languages, decentralisation, ...)
- tuning of parameters (clusters max size, #cows per cowboy, ...)

smaperteam

Team:

- Chenguang Zhou: Designer and developer, M.Sc. student at RMIT

First time to participate Multi Agent Contest.

Uses JACK Intelligent Agents framework.

smaperteam

Difficulties:

- debugging is difficult
- herder agents are still centralized, they communicate with a coordinator which does path finding for them by A*
- system is still not stable, needs more testing

unknown

Team:

- Developer: Slawomir Deren, M.Sc. Student, TU-Clausthal
- Supervisor: Peter Novak

Background:

- language: Jazzyk with 3 modules (i.a. ruby-modul for parsing of information)
- MAS: Open Agent Architecture for exchange of messages

unknown

Implementation:

- two subteams, each subteam consists of 1 leader and 4 herders; leader searchs for the cows and coordinates the herders agents
- a fence can be opened only by 2 certain agents
- agents communicate in each simulation step and share the information (switch)
- leader computes the move (direction) of cows using A*

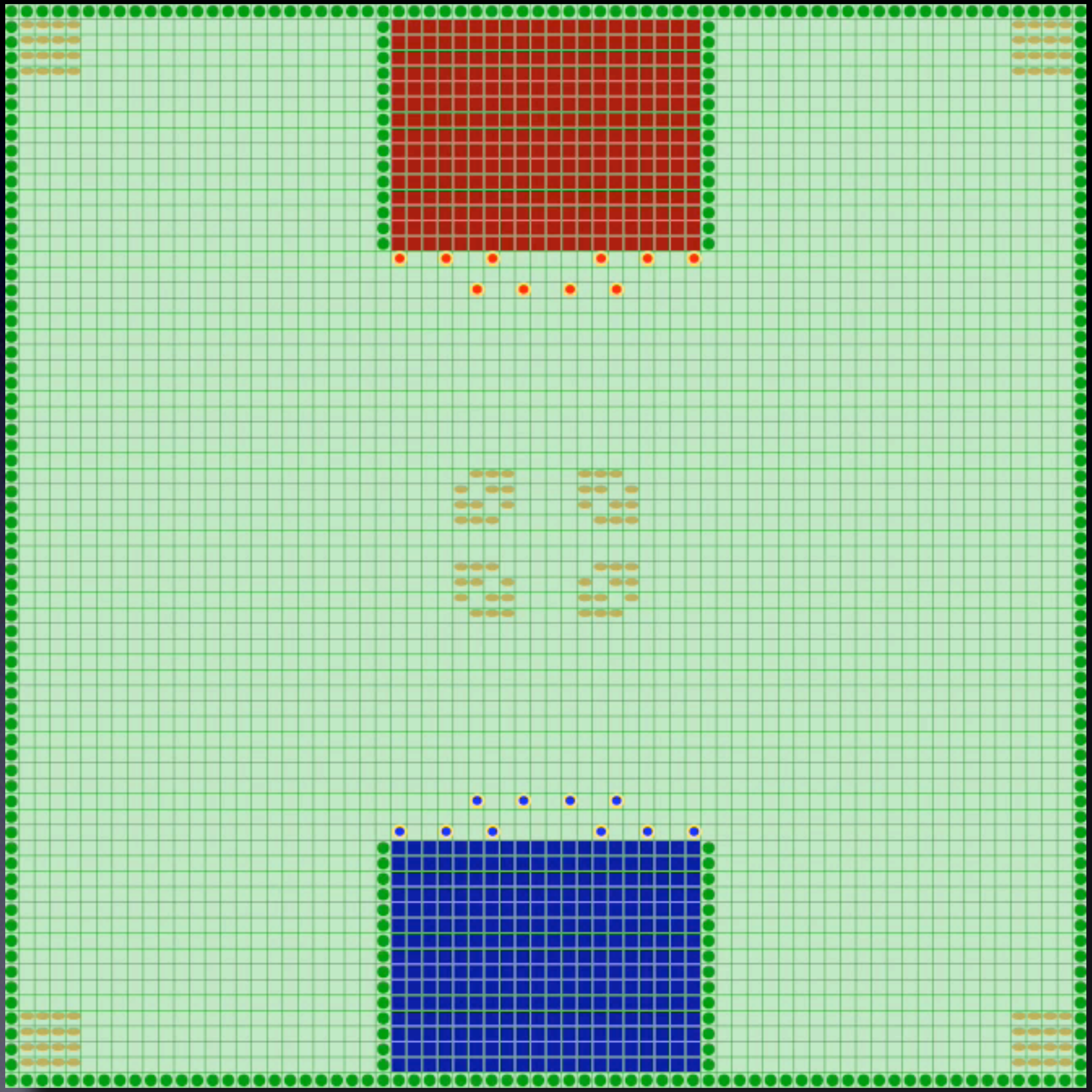
unknown

Difficulties/Results:

- the agents were able to drive only one group of cows
- general performance of agents inefficient
- the amount of leaders to small for searching
- opening of fences by short line of sight

Videos

JIAC V vs SmaperTeam

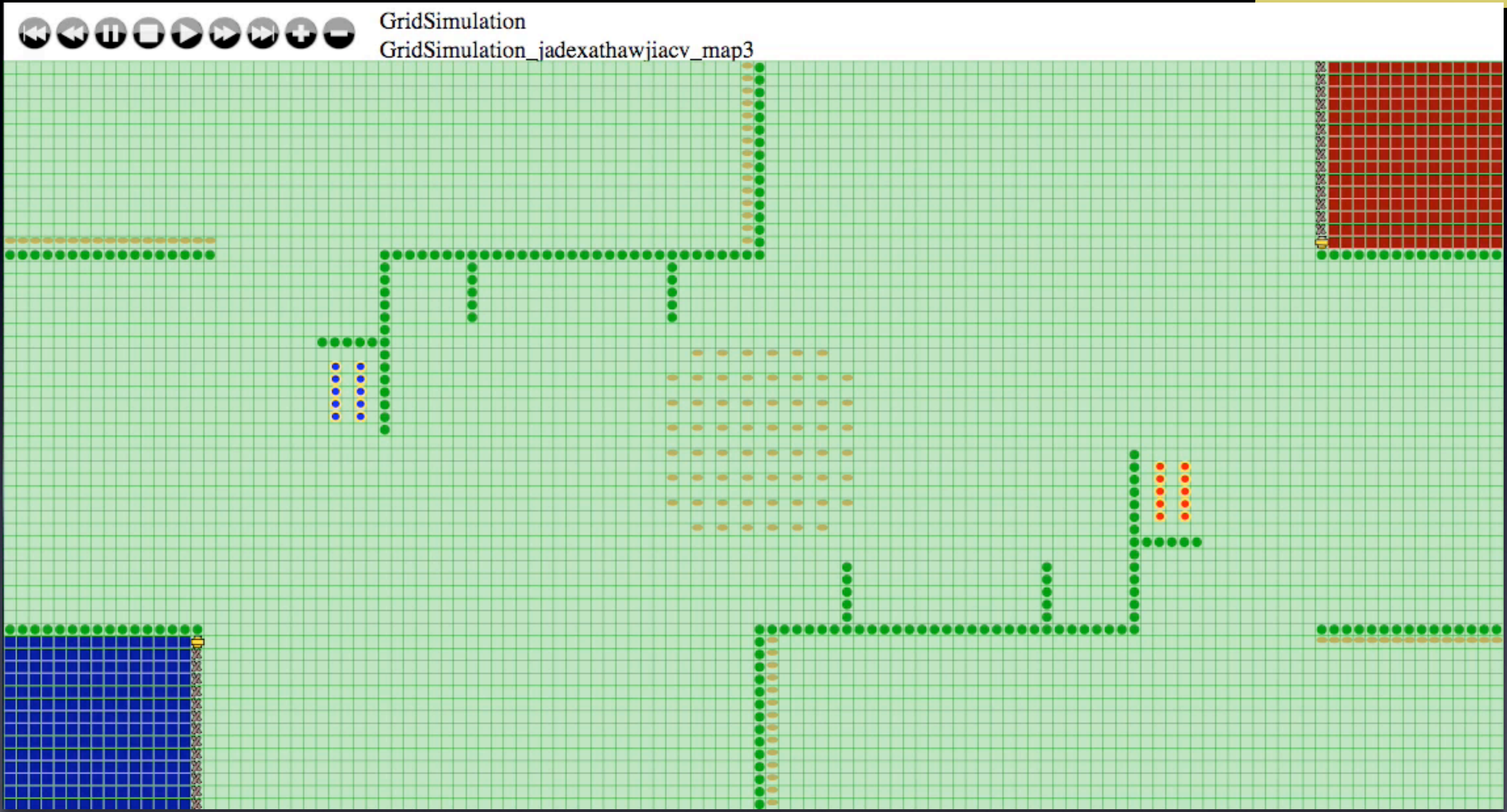


You need a strong cow driving algorithm first.

although there is a line of 6 agents trying to drive the cows out of the corral, jiacv always drive them back and getting new cows in the mean time.

Jadex@HAW vs JIAC

winning against "almighty" jiacv, stealing cows from opponent's corral together with very good cow driving algorithm will win



GridSimulation
GridSimulation_jadexathawjiacv_map3

More videos on our homepage!

Results

Results

4. Jason DTU (433 cows, 30 points)
5. smaperteam (194 cows, 23 points)
6. Micro JIAC (363 cows, 21 points)
7. AFABLE (468 cows, 20 points)
8. unknown (12 cows, 1 point)

Top Three

1. JIAC V (1627 cows, 60 points)
2. Jadex@HAW (1345 cows, 57 points)
3. Roman Farmers (840 cows, 37 points)

THANKS!

www.multiagentcontest.org