Tecnológico de Monterrey

Evidence 2. Review 1

Members

Eugenio Guzmán Vargas | A01621084

Diego Alberto Carrillo Castro | A01645484

Braulio Fernando Antero Díaz | A01645356

Diego Núñez García | A01644371

Victor Manuel Laureano Vega | A01638979

Modeling of Multi-Agent Systems with Computer Graphics

10/11/2025

1. Team Composition

Team Members

1. Eugenio Guzmán Vargas A01621084
2. Diego Alberto Carrillo Castro A01645484
3. Victor Manuel Laureano Vega A01638979
4. Braulio Fernando Antero Díaz A01645356
5. Diego Núñez García A01644371

Individual Profiles

Eugenio Guzmán Vargas

- **Strengths:** Software architecture and design, Database design, AI integrations
- **Areas of Opportunity:** Algorithm implementation and design, graphic engines (Unity)
- **Expectations for the Block:** I hope to be able to learn about MAS, how to design and architecture them, and how to understand where this type of systems can be applied. Specifically, the applications of technology in the agricultural sector is something that deeply interests me.

Victor Manuel Laureano Vega

- **Strengths:** frontend development, database design
- **Areas of Opportunity:** Design and implementation of algorithms, backend development, usage of graphics engines
- **Expectations for the Block:** I hope to learn a little more about computational agent theory, and see if other things that interest me can be applied in this discipline, such as different programming paradigms (like logic) or concepts related to mathematics or even the humanities.

Diego Alberto Carrillo Castro

- **Strengths:** backend development, database design
- **Areas of Opportunity:** Advanced Algorithms & Data structures, Frontend development
- **Expectations for the Block:** How to create agents

Braulio Fernando Antero Díaz

- **Strengths:** Frontend development, database design
- **Areas of Opportunity:** Algorithm implementation, backend development.

- **Expectations for the Block:** I hope to use agents to solve some different problems and how can we take advantage of them.

Diego Núñez García

- **Strengths:** Frontend development, database design
- **Areas of Opportunity:** Artificial intelligence, Advanced usage of graphic engines
- **Expectations for the Block:** I hope I can get better at creating and implementing artificial intelligence into the project.

Team Goals & Commitments

What we expect to achieve and obtain as a team:

- To develop a fully functional simulation of the distribution and conditions of a real warehouse, where crops have unique values to several attributes that affect their conditions.
- To gain a deep understanding of agent and MAS architecture, their advantages and disadvantages, their use cases, and the way these can be applied to solve the proposed problem.
- To maintain clear communication and implement good coding and project management practices to ensure all work is done on time and shape.

Our commitments to achieve these goals:

- To properly inform ourselves about the project requirements, especially those that are fundamental to its development (the impact on tomatoes, the characteristics of the space where the agents will be moving, etc.).
- To apply what we learn in the MAS module to generate a model of the problem (in the graphics engine) that makes sense from a computational point of view and also from a "real-world" point of view (for example, from an agricultural perspective).
- To meet the deadlines set by the professors.
- To work as a team to ensure that the skills of all members are utilized and valued.
- To ensure that the project documentation is clear and complete at each stage of development.

2. Collaborative Work Tools

- **GitHub Repository:** https://github.com/eugeguzmanv/greenhouse-MAS
- **Communication Tool:** WhatsApp group

3. Formal Challenge Proposal

Description of the Challenge

The challenge is to develop a Multi-Agent System (MAS) to automate the monitoring of tomato plants in a simulated greenhouse. This system will be implemented in a dynamic, grid-based digital environment (Unity). The system will feature two mobile agents (R1 and R2) tasked with patrolling the greenhouse, "sensing" the state of each plant, and making autonomous decisions.

Each plant will have dynamic properties (e.g., temperature, hydration, visual anomalies) that the agents will process. This processing will involve a "black box" model (a neural network, for example) to classify a plant's health and determine if it poses a disease risk. Based on this analysis, the agent will decide whether a plant needs to be "cut" (flagged for removal). A central database will store plant data, agent positions, and agent decisions, facilitating awareness and logging.

Identification of the Agents Involved

Scout Agent (R1): A mobile agent responsible for patrolling, sensing, and analyzing a designated zone of the greenhouse.

Scout Agent (R2): A mobile agent with identical capabilities to R1, responsible for its own designated zone.

Agent Relationships:

<u>R1 <-> R2 (Peer-to-Peer):</u>

- Deconfliction: They will read each other's positions from the database to avoid collisions.
- Coordination: The greenhouse grid will be divided into two zones (e.g., R1 takes the left half, R2 takes the right half) to prevent redundant work and ensure full coverage.

<u>R1/R2 <-> Database (Stigmergy/Reporting):</u>

- Reporting (Write): Each agent writes its current position, the status of plants it has scanned, and any "cut" decisions to the database.
- Sensing (Read): Each agent reads the database to get its assigned patrol zone and the real-time position of the other agent.

Agent Architectures

Scout Agent (R1): Architecture Type: Hybrid

Scout Agent (R2): Architecture Type: Hybrid

Hybrid Agent Architecture (for both R1 and R2)

Reactive Components:

Inputs (Sensors): Receives raw sensor data about the agent's immediate surroundings. This includes information about obstacles directly in its path (walls or the other agent) and data from the plant on its current tile (the actual data “transmitting” and “reading” methods are not defined yet, but will probably be through OOP attributes and methods).

Outputs (Actuators): Issues the basic, low-level commands. This includes commands for movement (like "move forward" or "turn") and a signal sent to the Deliberative layer when it detects a plant that needs analysis.

Rules**:** A set of simple, high-priority rules that execute automatically.

- Collision Avoidance: The highest-priority rule, which immediately stops or turns the agent if an obstacle is detected, overriding all other commands.
- Sensing Trigger: When a plant is detected on the current tile, this rule stops the agent and sends the "analyze" signal to the deliberative layer.
- Path Following: The default behavior. It takes the next "move" command from the deliberative layer's plan and executes it.

Deliberative Components:

Beliefs**:** This is the agent's internal "memory" or "state" of the world, which it uses for planning. It includes, but could not be limited to, the following:

- Its own current location.
- The location of the other agent (read from the database).
- Its assigned patrol zone.
- A memory of which plants it has already scanned and what their status was (healthy, sick, cut, etc.), most likely stored in a database.
- The currently active plan or path it is following.

Desires: These are the high-level objectives that the agent's planner tries to fulfill.

- Scan all plants in the assigned zone.
- Avoid colliding with the other agent.
- Analyze any plant the agent is currently on.

Intentions: This is a library of scripts or pre-defined plans. The agent chooses which plan to execute based on its current Desires and Beliefs.

- Initial Patrol Plan: A plan that runs once at the beginning to map out the most efficient 'snake-like' path to cover its entire assigned zone.

- Standard Patrol Plan: The default plan that follows the generated path, feeding move-by-move commands to the Reactive layer.

- Plant Analysis Plan: A plan triggered by the Reactive layer. It runs the plant data through the data processing algorithm, makes the 'cut'/'no-cut' decision, and reports the result to the database.

- Deconfliction Plan: A plan that activates if its intended path is blocked by the other agent. It will make the agent pause for a short time before re-checking and proceeding.

4. Work Plan

The following is a breakdown of all activities and the workload distributed among the project execution timeframe. Times may vary on practice, but this outline must be followed to achieve a good time and form of delivery.

Week 2

- Unity configuration and grid design for the warehouse (Entire team)
- Plant and robot model in Blender and C# script with initial attributes (Braulio)

Week 3

- Design of the plant data structure with basic attributes (id, position, properties, etc.) [Eugenio, Victor]
- Development of C# scripts to manage basic operations such as creating, reading, updating, and deleting data related to the agents (Eugenio)
- Program the logic to divide the greenhouse grid into two zones (for the agents to traverse) [Diego N.]
- Creation of variables and data structures to store the internal state of the agents (Diego C.)

Week 4

- Implement methods to obtain immediate information about object or plant sightings (Victor, Braulio)
- Implement movement functions in C# (Diego N, Diego C)
- Program collision avoidance rules and sensing triggers (Diego N, Diego C)
- Implement the logic of the plant analysis plan so that it can be executed (“black box” algorithm) [Eugenio]

Week 5

- Implement the algorithm that allows agents to cover the area (Diego N, Victor, Braulio)
- Integrate the “black box” algorithm (which has been under development over the previous weeks) into the full system [Eugenio, Diego C]
- Perform final tests of agent functionalities integration and complete data logging (Entire team)