Tecnológico de Monterrey

Guadalajara campus

Evidence 2. Review 1

José Yael Varela García - A01645324

Faisal Alali- A01830963

Sebastian Certuche - A016449942

Sebastián Alejandro Soria Piñuela

Modeling of multi-agent systems with computers graphics

Group 301

November 06, 2025

Description of the challenge

The challenge consists of developing a multi-agent system in Unity capable of autonomously identifying and classifying tomatoes as good or bad based on their visible characteristics (such as color, spots, or surface damage). Each agent will navigate through an environment, detecting and avoiding obstacles while approaching tomatoes, analyzing them, and deciding their condition. Once classified, the agents will pick up each tomato and place it in the corresponding box, coordinating their movements to prevent collisions, bottlenecks, and overlaps. The simulation highlights how multiple agents can combine perception, decision-making, navigation, and collaboration to perform a sorting task efficiently in a dynamic, obstacle-filled scenario.

Although this is a simulated environment, the project is inspired by real agricultural challenges, where 20–40% of global production is lost each year due to diseases, pests, and labor scarcity and inspections are often done too late. A system that can reliably distinguish healthy from damaged fruits represents a first step toward automated, early anomaly management in crops. In a real-world context, similar multi-agent systems could help reduce waste, optimize the use of inputs such as water, nutrients, and agrochemicals; and support farmers by easing dependence on manual visual inspection, improving productivity and protecting farmers' income.

Identification of agents

Agent 1 - Scanning Robot

-REACTIVE ARCHITECTURE-

This agent moves through the map through a pre-processed patrol path when in an idle state. It detects the presence of tomatoes and anomalies in them, sending the gathered information [Agent 2] for it to analyze.

Reaction Layers

Layer 0 – Avoid obstacles (lowest priority)

```
IF FrontalObstacleDistance() < NEAR_DISTANCE THEN
        Stop()
        Turn(45)
```

```
ELSE
    ContinueCurrentPath()
```

Always active for the agent to be able to operate fully and not get stuck.

Layer 1 – Communication with other agents

```
IF LostConnection(Agent_2) THEN
    SetConnected(false)
    RetryConnection()
ELSE
    SetConnected(true)
```

Keeps connectivity alive but doesn't control motion.

Layer 2 – Scan environment

```
IF CameraDetects(TomatoCluster) THEN
    Stop()
    Data = GetTomatoData(Tomato)
    SendDataTo(Agent_2, Data, Coordinates)
ELSE IF CameraDetects(Anomaly) AND IsConnected() THEN
    FlagForInspection(Tomato)
    SendAlertTo(Agent_2)
```

Temporarily overrides patrol when found tomatoes to examine them.

Layer 3 – Patrolling and recharging (highest priority)

```
IF IdleState() THEN
    Follow(PatrolPath[i])
IF DetectedTomato() THEN
    MoveTo(TomatoPosition)
IF LowBattery() THEN
    ReturnTo(ChargingStation)
```

Executes only when no other activity is active (idle state). It will eventually take the agent towards more tomatoes.

Agent 2 – Defective Tomato Cutter Agent

-HYBRID ARCHITECTURE-

This agent receives data from [Agent 1], analyzes it based on its desires, and acts given its Intentions. Then, it'll act through a reactive control layer. After determining the state of a tomato, if it detects abnormalities, remove the objectives safely.

Beliefs, Desires and Intentions

Beliefs
- bool: isTomatoDefective
- Coordinate: posTomato
- float: distanceToTomato
- List<Coordinate>: mapOfDetectedTomatoes
- float: batteryLevel

Desires/GOALS
- RemoveDefectiveTomatoes
- MaintainCleanField
- AvoidDamageToHealthyTomatoes
- ReturnToChargingStationIfLowBattery

Intentions
- CURRENT_PLAN: MoveForwardTo(posTomato)
- CURRENT_PLAN: CutTomato()
- CURRENT_PLAN: UpdateStatus(str:message)
- CURRENT_PLAN: ReturnToBaseIfLowBattery()

Reaction Layers

Layer 0 – Avoid obstacles (lowest priority)

```
IF FrontalObstacleDistance() < NEAR_DISTANCE THEN
    Stop()
    Turn(45)
```

Ensures safe movement by preventing collisions with nearby objects.

Layer 1 – Communication with other agents

```
IF LostConnection(Agent_1) || LostConnection(Agent_3) THEN
    SetConnected(false)
    RetryConnection()
ELSE
    SetConnected(true)
```

Maintains stable communication links with Agent_1 and Agent_3 . If connection is lost, it retries automatically to reestablish it.

Layer 2 – Navigation

```
info OBJECTIVE_TOMATO = NULL

IF IsConnected() THEN
    OBJECTIVE_TOMATO = GetInfoFrom(Agent_1)

IF OBJECTIVE_TOMATO THEN
    MoveToward(OBJECTIVE_TOMATO.position)
```

Guides the agent toward defective tomatoes identified by the scanning robot. Reliant with communication with Agent_1.

Layer 3 - Removal (highest priority)

```
IF TomatoDetected() AND isTomatoDefective() THEN
    CutTomato(Tomato)
    SendReportTo(Agent_3)
```

Makes the job of detecting and removing defective tomatoes. When a faulty tomato is confirmed, the robot prioritizes this action above all others and reports the operation to Agent_3.

Agent 3 – Harvester and Transport Agent

-DELIVERATIVE ARCHITECTURE-

This agent focuses on collecting tomatoes and transporting them to a transport area.
It ensures full coverage of the map, being able to collect tomatoes from anywhere.

Beliefs

- List<Tomato>: ripeTomatoesDetected
- Coordinate: posTransportArea
- Coordinate: posMe
- float: loadCapacity
- bool: isStorageFull

Desires/GOALS

- HarvestRipeTomatoes
- DeliverLoadToTransportArea
- AvoidMixingWithDefectiveTomatoes
- OptimizePathForCollection

Intentions

- CURRENT_PLAN: MoveForwardTo(posTomato)
- CURRENT_PLAN: PickTomato()
- CURRENT_PLAN: MoveForwardTo(posTransportArea)
- CURRENT_PLAN: DropTomatoes()
- CURRENT_PLAN: ReplanRouteIfObstacleDetected()

Workplan

Activity	**Date**	**Estimated effort interval time**	**Responsible**
Research crop anomalies to learn when they have become infected.	Nov 12-14	4 hrs (Low Effort)	All members
Define systems agents and roles	Nov 15-17	3 hrs (Medium Effort)	Danilo
Design the hybrid architecture	Nov 15-17	4 hrs (High Effort)	Sebastián Certuche
Create initial Unity environment to set up the 3D crop field with basic navigation and obstacles	Nov 18-22	6 hrs (High Effort)	Sebastián Soria and Yael Varela
Program exploration and detection within the crop field.	Nov 23-29	5 hrs (High Effort)	Yael Vaela
Program to check if Crops are bad or good	Nov 23 - 25	3 hrs (Low effort)	Sebastián Soria and Faisal
Program throwing bad crops away and storing good ones	Nov 25 - 29	3 hrs (Medium Effort)	Faisal
Implement agent behaviors	Nov 23-29	6 hrs (High Effort)	Sebas Certuche
Add interaction flow between agents	Nov 30-Dec 2	4 hrs (Medium Effort)	Danilo
Integrate and test system	Dec 3 - 5	6 hrs (Medium Effort)	All members