Autonomous Drone Delivery System

Intelligent System Engineering Project

Email: davide.dimarco5@studio.unibo.it

Let’s start – Idea & Why?

- **Motivation: Develop an automated pizza delivery system** using autonomous drones and a Multi-Agent System (MAS) architecture.
- **Efficiency:** Autonomous drones can **deliver pizzas faster** than traditional methods.
- **Cost:** Reduces labor costs.
- **Scalability:** The system can handle multiple orders simultaneously.
- **Adaptability:** The system can be extended to other applications, such as **package delivery or medical supply** transportation.

Goals and Requirements

- The goals of this project is to provide a simulation of **automated drones delivery system for a pizzeria**.
- We can imagine a pizzeria with some drones **able to delivery pizzas** to certain destinations.
- To be reached, the **main goal can be split into multiple requirements**.

Drone Battery Management

Dynamic Drone Navigation

Drone Collision Avoidance

Rescue Robot Policy

Pizza Assignment Policy

User Interface

Drone Power Engine Selection

MAS, BDI Architecture, JASON

MAS (Multi Agent System): A system consisting of one or more agents capable of:

- **Making decisions** based on perceptions and beliefs.
- **Collaborating** to achieve common or individual goals.
- **Adapting to failures** to ensure the final goal is reached.

JASON LANGUAGE

BDI Architecture

- **Beliefs:** Information about the environment and the agent's internal state (e.g., battery level, location).
- **Desires:** Goals the agent aims to achieve (e.g., deliver a pizza, return to the base).
- **Intentions:** Plans adopted to achieve desires (e.g., follow a delivery route).
- **Decision-Making Process**: Agents adapt plans and actions based on new beliefs or unexpected events.

Requirements Analysis

Functional Requirements:

- Order Assignment Policy
- Drone Delivery
- Obstacle Collision Avoidance
- Battery Management
- Robot Recovery
- User Experience

Non Functional-Requirements:

- Timing Synchronization
- Remediation to Failure
- Information flows between agents
- Logging

TOP DOWN DESIGN

Design

Phase 1

Define Entities & Environment

Key Agents:

- **Pizzeria:** Manages orders and drone assignments.
- **Drones:** Perform deliveries, manage battery levels.
- **Robot:** Recovers broken drones.

Environment:

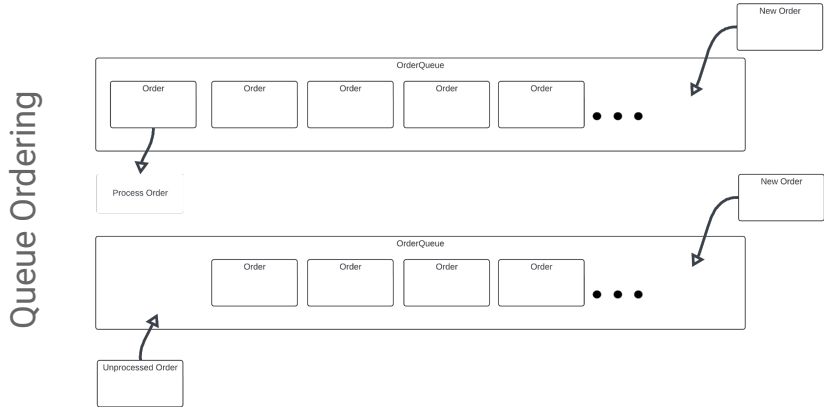
- **User Interface:** a map shows in real time the actual state of the system
- **Simulation of failure:** the environment is responsible of the random failure of the drone.
- **Battery:** Battery is related to the environment, it is not part of drone directly.
- **Obstacles:** are part of the environment and are randomically generated each time the simulation starts.

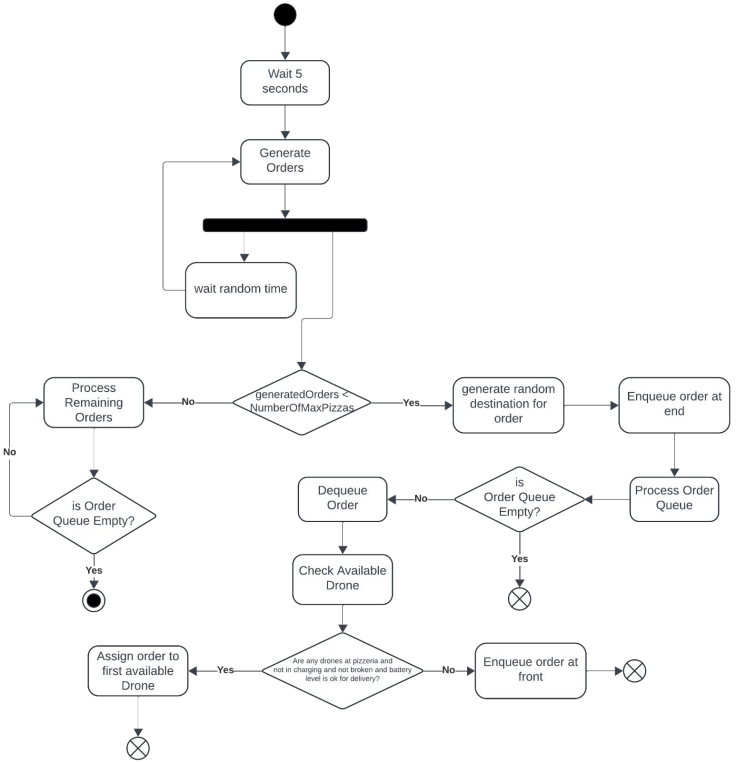
Solution: To achieve the goal of this project, a Multi-Agent System (MAS) was created. The agents involved are the Pizzeria, the Drones, and the Robot. **The Pizzeria is the core** of drone management, acting as a centralized system. **The agents operate autonomously** in relation to their **specific responsibilities**.

Pizzeria Agent Behavior

Key Concepts:

- **Order Generation** after random time
- **Enqueue Order at front or at end** depending on actual state
- **Process Order Queue** (Internal to the agents beliefs)
- **Assign order to Available Drone** depending on their state (battery level, broken Status, Actual location)

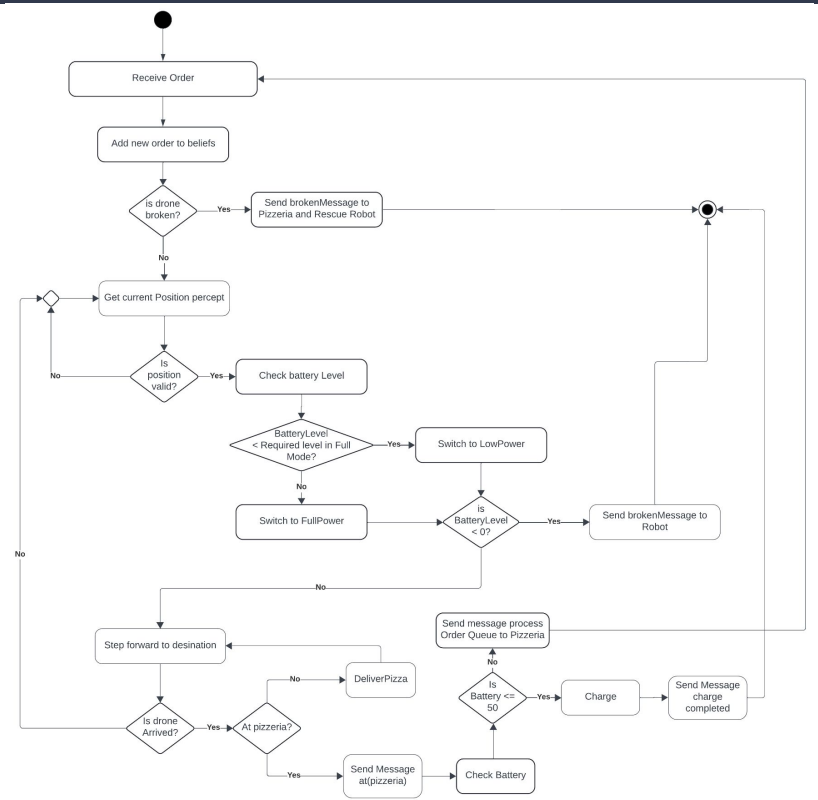




Drone Agent Behavior

Key Concepts:

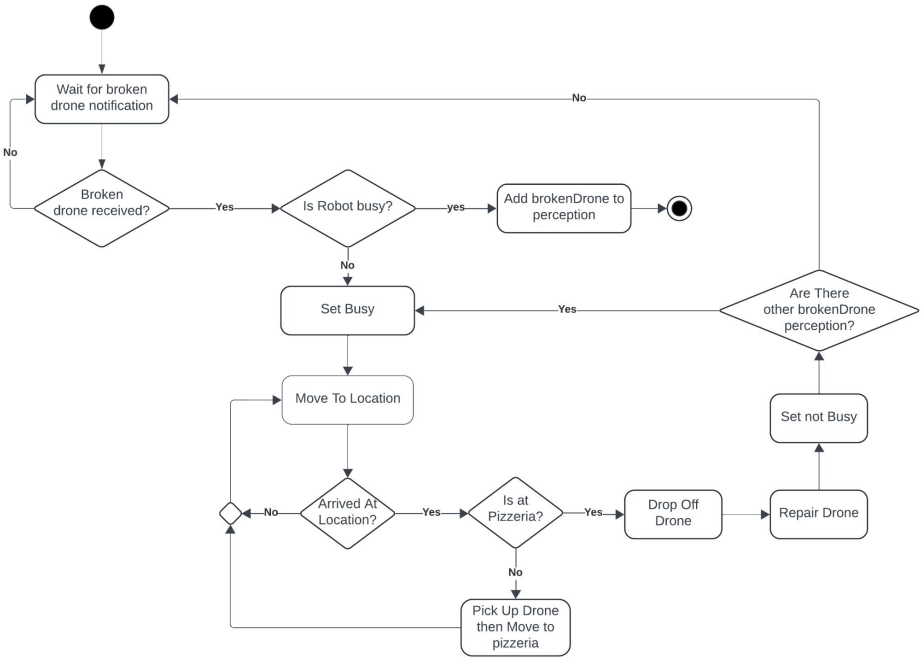
- **Receive Order** from pizzeria
- **Battery Management** (Charge, Switch power mode)
- **Deliver** pizza (Decision depending)
- **Engine Power Management** depending on distance of destination.
- **Communicate with Robot** in case of failure



Robot Agent Behavior

Key Concepts:

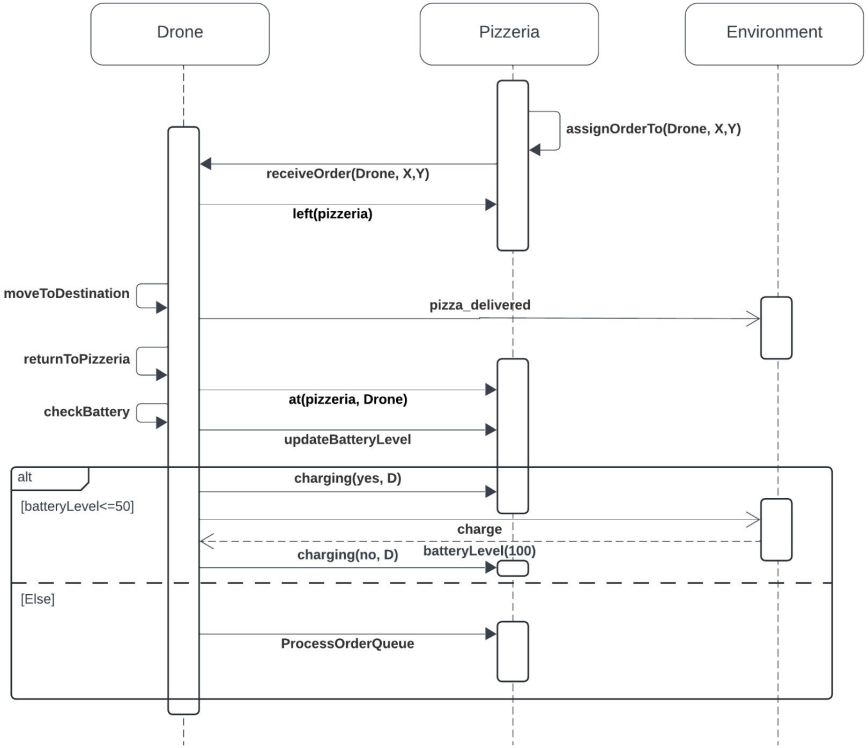
- **Wait for broken notification** by drone
- **Busy state**
- **Movement capability** (reuse of the code of drone)
- **Repair Drone** when at pizzeria



Interaction – Order Assignment and Delivery

Key Concepts:

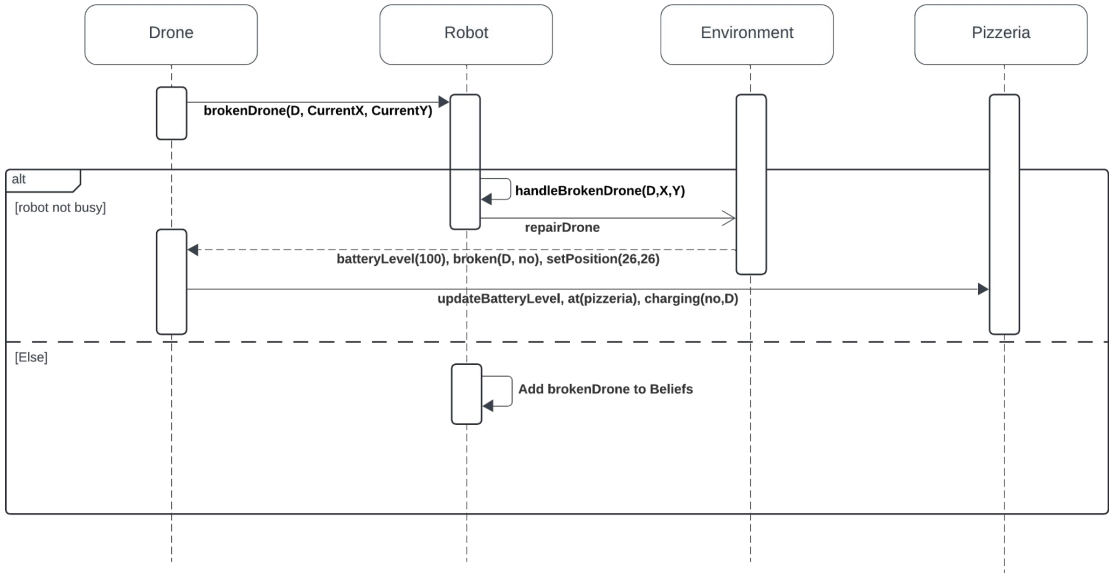
- Pizzeria **assigns** Order.
- Drone **receive** the order then left the pizzeria communicating the state.
- Drone **moves** to destination and execute the **action of delivery** then come back to pizzeria check battery and inform the pizzeria about its battery level.
- If **battery is ok**, the **drone continues to process** order queue.



Interaction – Broken Drone Handling

Key Concepts:

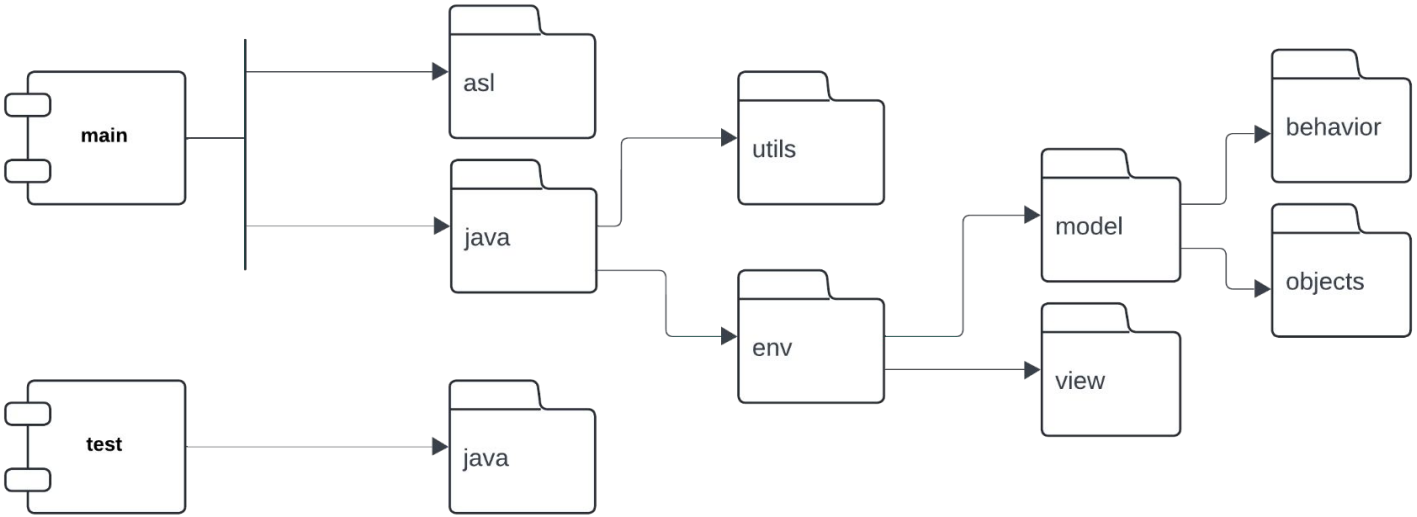
- **Drone send message to Robot** when broken
- **Robot repair the drone** with an action
- The **environment set the drone as repaired**
- **Drone updates** Pizzeria Beliefs



Design

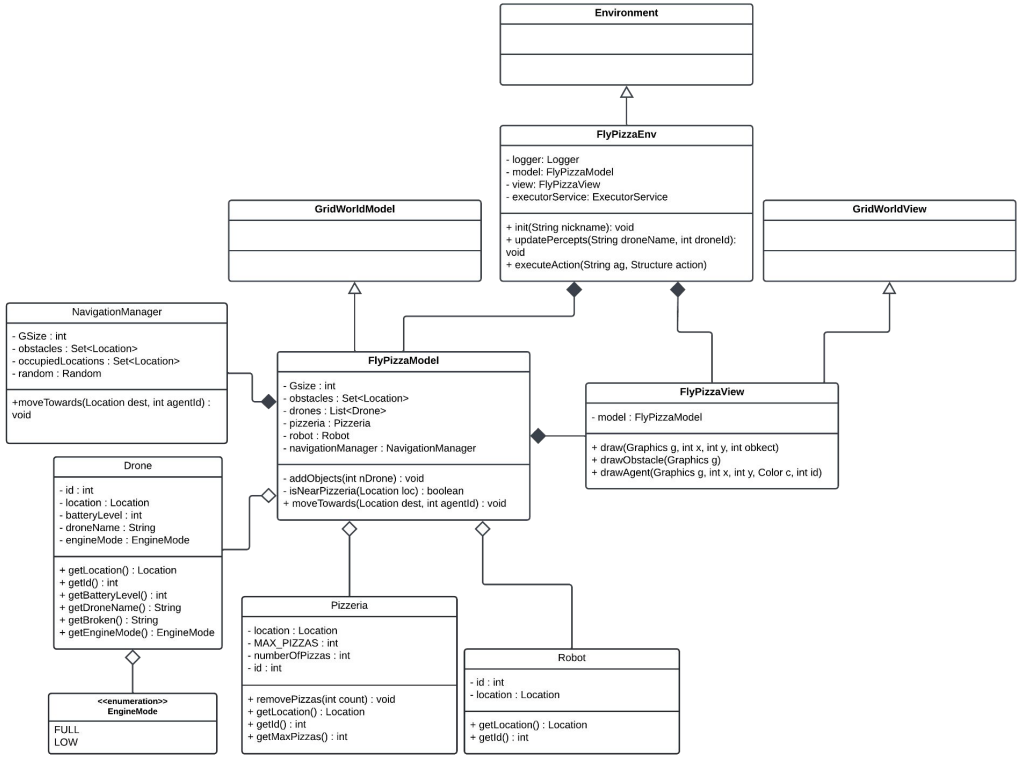
Phase 2

Package Diagram



Class Diagram (Model)

- Inherited **Jason Framework** classes.
- Separate **responsibilities** between classes reflecting the agents.
- Easy to **maintain and extend**.



Test

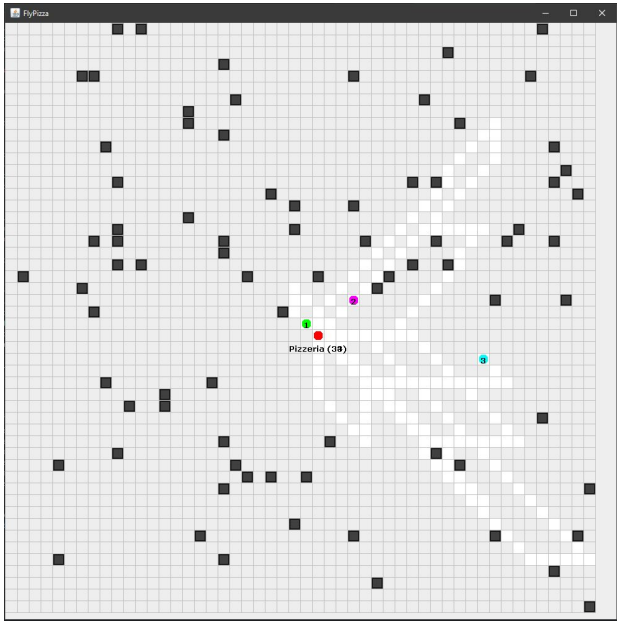
Test Results	378 ms
Test class NavigationManagerTest	378 ms
testMoveIfOccupied	321 ms
testValidateCoords	42 ms
testMoveAroundObstacle	15 ms

Navigation Manager Test

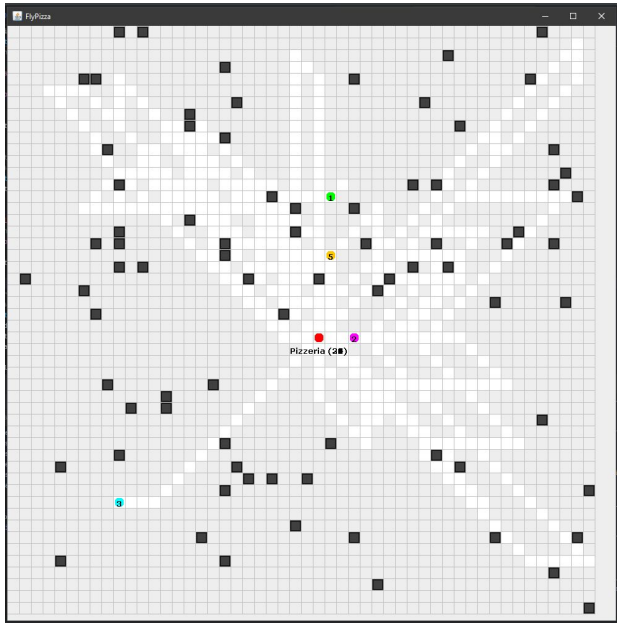
Test Results	385 ms
Test class ModelTest	385 ms
testSetEngineMode	319 ms
testInitializeDrones	40 ms
testRepairDrone	18 ms
testObstaclesAdded	3 ms
testMoveDrone	3 ms
testDecreaseBatteryLevel	2 ms

Model Test

Usage Examples



Normal scenario



Robot Deployed Scenario

FlyPizza

FlyPizza

File Home Share View

Pin to Quick access | Copy | Paste | Cut | Copy path | Paste shortcut | Move to | Copy to | Delete | Rename | New folder | New item | Easy access | Properties | Open | Edit | History | Select all | Select none | Invert selection

Clipboard | Organise | New | Open | Select

FlyPizza

Search FlyPizza

Quick access
Desktop
Downloads
Documents
Pictures
Program Files (
Corso Apprend
Documenti
System32
UTILS

OneDrive - Perso

This PC
3D Objects
Desktop
Documents
Downloads
Music

14 items

C:\Windows\System32\cmd.exe - gradlew runflypizzamas

```
Microsoft Windows [Version 10.0.19045.5073]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DAVIDE\Desktop\FlyPizza>gradlew runflypizzamas
<=========----> 75% EXECUTING [1s]
> :flyPizza:runFlyPizzaMas
```

Search

22

Next...

Agent responsible for receiving and prioritizing orders

Assigning each drone a **specific zone** on the map

Implement a **caller agent** to simulate a customer providing order details.

Enable drones to **carry multiple pizzas** in a single delivery

The end