Evidence 2 - Review 3

Modeling of Multi-Agent Systems with Computer Graphics

Grupo 101

Ana Elena Velasco García	**A01639866**
Baltazar Servín Riveroll	**A01643496**
Emilio Pardo Gutiérrez	**A01644781**
Jozef David Hernández Campos	**A01644644**
Maria José Medina Calderón	**A01639205**

Iván Axel Dounce Nava

Jesús Israel Hernández Hernández

Oscar Guadalupe Hernández Calderon

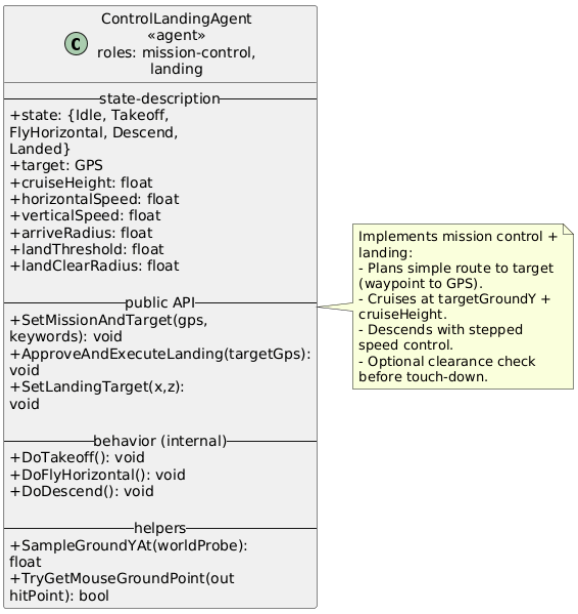
Fecha de entrega:

5 sept 2025

Index

1. Agent UML diagrams

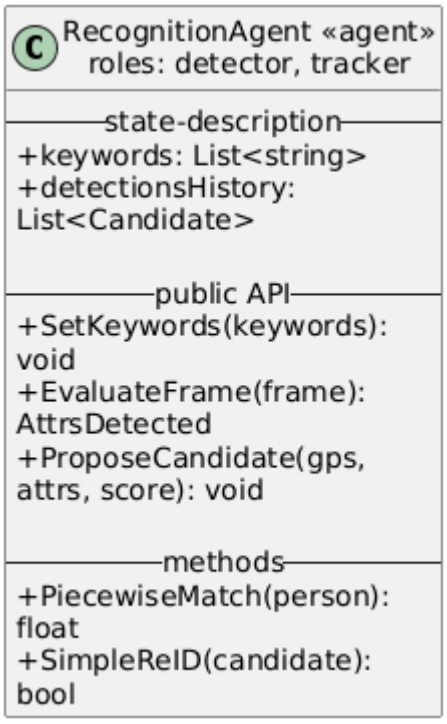
Figure 1.1 - Landing-Decision Agent Class Diagram



Description:

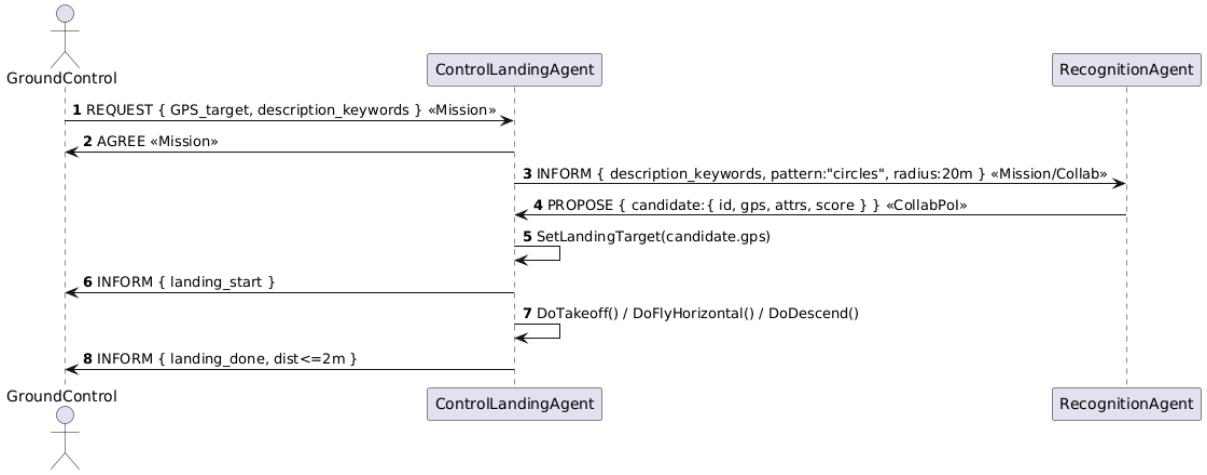
The Landing-Decision Agent manages the overall logic of the MAV. It sets the mission and is the source of information. It also ensures that the MAV can perform a safe landing near the person of interest.

Figure 1.2 - Recognition Agent Class Diagram



Description: The Recognition Agent perceives and interprets the environment through the device camera through recognition parameters obtained from processing the target description. After continuously scanning the environment, the detected candidates are proposed to the Landing-Decision Agent.

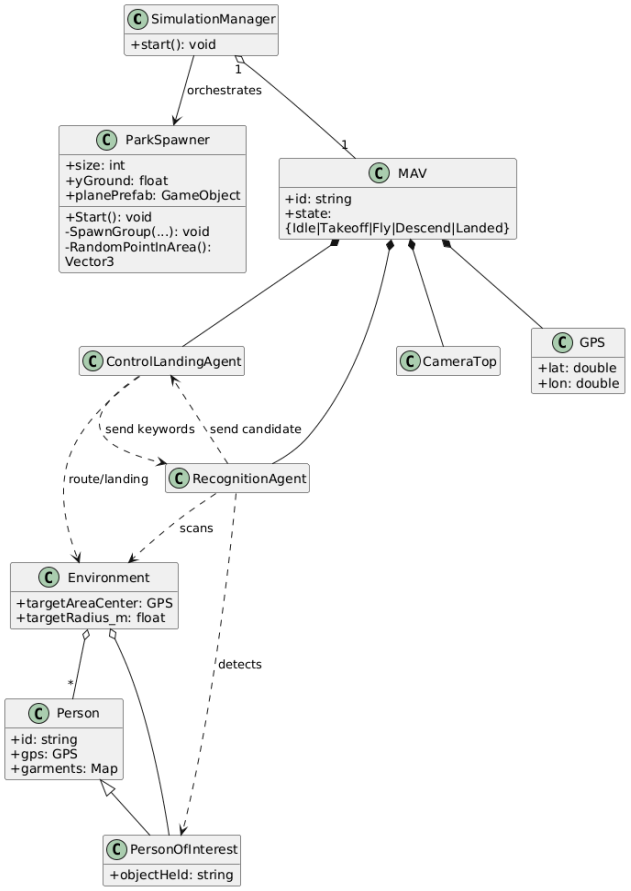
Figure 1.3 - Agent Interaction Protocol Diagram



Description: The Agent Interaction Protocol diagram models how the two agents exchange communicative acts. It shows the sequence from start to finish.

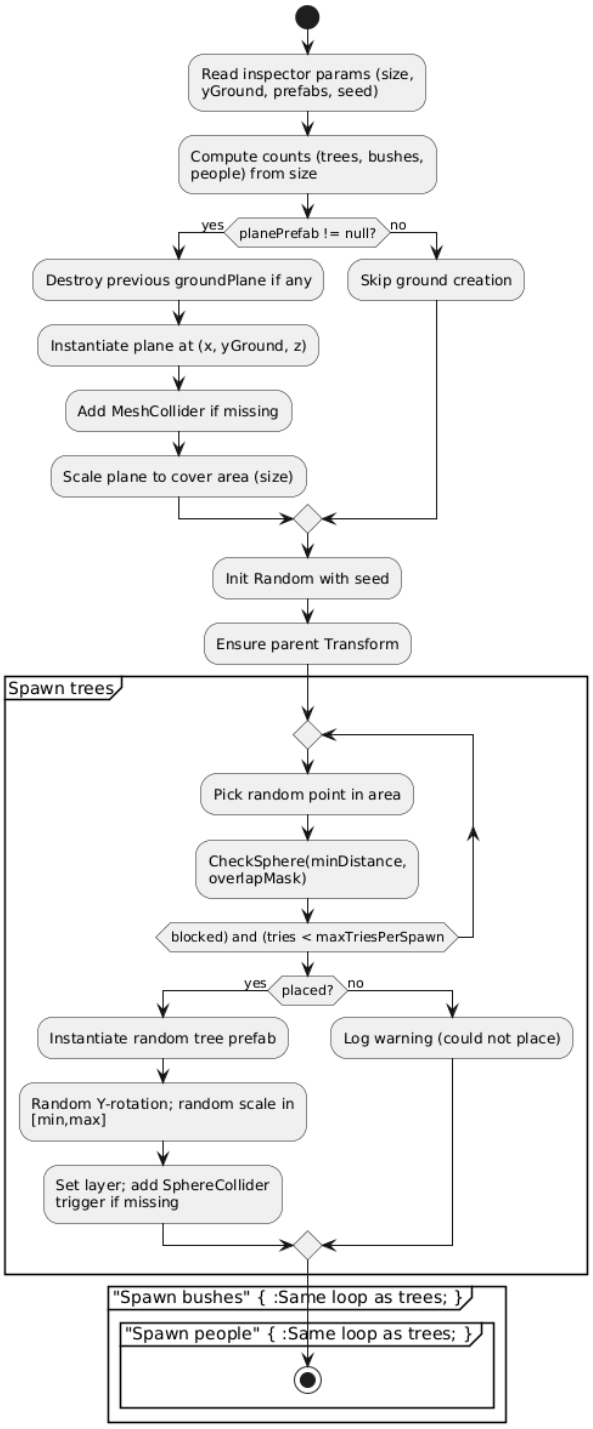
2. Simulation UML diagrams

Figure 2.1 - Simulation UML class diagram



Description: The Simulation UML Class Diagram illustrates an overview of the simulation. It includes the MAV and its two main agents along with the GPS and camera, and provides an overview of the environment containing the target.

Figure 2.2 - World Generation Activity Diagram



Description: The activity diagram provides the runtime flow of the park generation: create or the ground plane, initialize randomness with a fixed seed, and spawn groups (trees, bushes, people) by sampling random positions, rejecting overlaps, instantiating prefabs, randomizing rotation/scale, and adding colliders if needed.

3. Agents code

Link to Github DroneLandingController.cs

4. Graphical simulation code

Link to Github ParkSpawner.cs

5. Work plan

Pending activities

Phase 1

Completion Date	**Activity**	**Description**	**Responsible individuals**	**Estimated effort interval**	**Actual estimated effort interval**
20 August	UML diagrams design	Create clear and standardized UML diagrams to represent the system's	Marijo, Ana	2h	2h

		architecture and functionality.			
20 August	Challenge goals and documentation	Accurately represent the system structure and processes.	Ana, Marijo, Baltazar, Emilio Jozef	1h	1h
20 August	Formal Proposal	Design a comprehensive model that can guide software development	Ana, Marijo, Baltazar, Emilio Jozef	3h	3h
20 August	Work plan	The work plan is divided into phases with tasks distributed among team members.	Ana, Marijo, Baltazar, Emilio Jozef	1 h	1 h

Phase 2

Completion Date	**Activity**	**Description**	**Responsible individuals**	**Estimated effort interval**	**Actual estimated effort interval**
26 August	Agent UML	Diagrams	Emilio, Ana	3h	3h

	diagrams design	(Decision, Recognition, Landing, Agents)			
26 August	Interaction protocol UML diagram	Sequence of communicative acts (mission, search, candidate, land)	Marijo	3h	3h
26 August	Simulation UML Class diagram	Implementation-level UML (MAV, agents, sensors, environment)	Ana, Emilio, Jozef	2h	2h
26 August	MAV descriptions	Concise text: mission intake, GPS nav, recognition, landing, collaboration.	Baltazar, Emilio Jozef	2 h	2 h
26 August	Diagram description	Short section highlighting changes from R1 to	Ana	1h	1h

		R2 and characteristics.			
26 August	Final PDF Assembly & Repo Tag	Complete document, consistent formatting, tag “Review 2”	Ana, Emilio	2h	2h

Phase 3

Completion Date	**Activity**	**Description**	**Responsible individuals**	**Estimated effort interval**	**Actual estimated effort interval**
5 September	3D Park Environment	In Blender create the trees, bushes, terrain, grass	Baltazar	5h	5h
10 September	3D People Models	Create the target person (vest + helmet) + distractors	Marijo	5 h	-
5 September	Drone Movement	Create the Physics in Unity – basic flight, landing mechanics.	Ana, Jozef	8h	8h

10 September	Detection Logic	Recognize orange vest + helmet, differentiate people.	Baltazar, Emilio	8 h	-
10 September	Interaction Protocol Design	Define agent communication & sequence diagrams	Marijo, Ana	8h	-
10 September	Integration & First Simulation Test	Put together environment + drone + detection	Jozef, Emilio	5h	-

6. Acquired learning

During this stage, the team gained experience in translating UML diagrams into actual code. We learned how to structure Unity scripts following object-oriented principles and how to connect agent logic with graphical simulation elements. The team also discovered the importance of iterative testing. Even small changes in recognition parameters or drone physics required several simulation runs and fine-tuning, which showed us the gap between theoretical models and their behavior in practice. Overall, we deepened our understanding of how theoretical UML designs evolve into a working multi-agent simulation.