

# SID Primavera 2025

## Práctica 1/4 (Laboratorio)

Sergio Álvarez  
Javier Vázquez

Febrero/Marzo 2025

### 1. Introducción

En esta práctica, se pide implementar tres agentes inteligentes con cierto grado de autonomía que puedan sobrevivir e intentar superar al equipo contrario en un entorno multi-agente. Cada agente tendrá varios objetivos: (a) deberá percibir el entorno para construir un conjunto de creencias que le permita interpretar la situación de la partida, (b) deberá cumplir con el objetivo de su equipo, ya sea capturar la bandera o evitar que el otro equipo la capture, y (c) deberá aprovechar sus acciones disponibles, dependiendo de su rol.

### 2. Resumen del escenario: pyGOMAS

pyGOMAS<sup>1</sup> es una plataforma de simulación de juegos que combina un sistema multi-agente con varios motores de renderizado para ejecutar agentes en un juego tipo “Capturar la Bandera”. Los agentes utilizan el lenguaje AgentSpeak para sus estrategias. Los agentes se ejecutan en la plataforma SPADE, que se basa en el estándar XMPP para la comunicación entre agentes y utiliza comportamientos para modelar las acciones de los agentes. La extensión SPADE-BDI permite a los agentes tener comportamientos similares a BDI, codificados en AgentSpeak.

La documentación detallada sobre el entorno, incluyendo una explicación de las creencias y acciones disponibles para los agentes, está disponible en [https://pygomas.webs.upv.es/wp-content/uploads/2024/02/pyGOMAS-Manual-v1\\_0.pdf](https://pygomas.webs.upv.es/wp-content/uploads/2024/02/pyGOMAS-Manual-v1_0.pdf).

#### 2.1. Capturar la Bandera

En este juego, en cada mapa hay una única bandera, que en el caso del equipo *Allied* han de conseguir capturar y en el caso del equipo *Axis* han de

---

<sup>1</sup><https://pygomas.webs.upv.es>

conseguí defender. Hay un límite de tiempo: si el equipo asaltante no logra capturar la bandera dentro de este período, el equipo defensor se asegura la victoria.

Hay tres roles:

- Soldado: sus armas hacen el doble de daño.
- Médico: puede crear paquetes de medicina que recuperan algo de salud a quien lo coge.
- Operador de Campo: crean paquetes de municiones que recuperan algo de munición a quien lo coge.

## 2.2. Condiciones de cada partida

En cada partida, las condiciones serán las siguientes:

- El mapa es aleatorio, pero sólo puede ser uno de los siguientes incluidos en el repositorio de pyGOMAS: *map\_01*, *map\_04*, *map\_09*, *map\_12*, *mine\_large*, *map\_02*, *map\_07*, *map\_10*, *map\_arena*, *mine\_medium*, *map\_03*, *map\_08*, *map\_11*, *mine*.
- El número de agentes por equipo será aleatorio, entre 3 y 8. Cada equipo tendrá el mismo número de agentes en cada partida.
- Los roles de los agentes serán totalmente aleatorios, pero como mínimo habrá un agente de cada uno de los tres roles.
- El tiempo máximo de ejecución (si no se captura la bandera y quedan agentes en cada equipo) será de 5 minutos.
- Al iniciarse la partida, cada agente sobre el entorno será escogido de entre las implementaciones de todos los grupos de laboratorio, de manera aleatoria. Esto significa que puede darse el caso de que más de un agente dentro de un mismo equipo haya sido implementado por el mismo grupo de laboratorio. Sin embargo, no se dará el caso de que haya dos agentes implementados por el mismo grupo de laboratorio en equipos contrarios en la misma partida.

## 3. Objetivo de la práctica

El objetivo de la práctica es aplicar las nociones vistas en teoría sobre agentes inteligentes, racionalidad, agentes lógicos, y el lenguaje AgentSpeak visto en laboratorio para implementar agentes que se comporten de manera racional en pyGOMAS.

Para ello cada grupo deberá implementar tres agentes, uno para cada rol, e implementar un cuarto agente extra para uno de esos roles. El objetivo de cada

agente es que su equipo (*Allied* o *Axis*) gane la partida. Para ello su implementación debería seguir un criterio bien definido sobre lo que es *racional* para su rol. Siempre y cuando el comportamiento se derive de una implementación coherente con lo definido como racional, no es obligatorio que los agentes tengan un rendimiento determinado. Sin embargo, habrá puntos extra por tener agentes que, de media, tengan un rendimiento superior al 80 % de grupos (ganen más partidas).

## 4. Plazos y evaluación

Durante las primeras semanas de laboratorio os introduciremos a las tecnologías que obligatoriamente tendréis que usar (AgentSpeak, SPADE, pyGOMAS).

Aunque podéis usar código Python para monitorizar, inspeccionar y manipular los agentes en tiempo de ejecución para probarlos, la entrega consistirá únicamente en los ficheros `.asl`.

La fecha de entrega del código de los agentes, junto con una documentación en `.pdf` explicando las decisiones de diseño, será el día **30 de marzo (30/03/2025)**.

La nota base de la práctica se evaluará según la rúbrica descrita en la Figura 1. Esta nota tendrá un valor máximo de 10. Aparte de esta nota base, existe la posibilidad de sumar puntos extra que permitirían tener una nota mayor de 10:

- 2 puntos extra a los grupos que implementen los agentes que se sitúen en el percentil 20 % superior en número de partidas ganadas. Para evaluar esto, se hará un número considerable de ejecuciones de partidas en condiciones aleatorias: mapa aleatorio, agentes aleatoriamente escogidos entre los grupos, con roles aleatorios. Los resultados de las ejecuciones se harán públicos en forma de trazas de pyGOMAS para que las visualicéis.

La nota de esta práctica contará un 25 % de la nota global de laboratorio. Si la nota es superior al 10, para hacer media no se reducirá a 10 sino que se mantendrá superior.

Criterio	Peso	Expectativa
Ejecución	40 %	<p>Los agentes funcionan bien y no fallan en la plataforma.</p> <p>El comportamiento de los agentes en el entorno no es errático y se puede interpretar que las acciones que ejecutan sirven un propósito.</p>
Implementación	30 %	<p>Las creencias, objetivos y planes están adecuadamente diseñados y se ajustan a la estrategia documentada.</p> <p>El agente es proactivo a la vez que reactivo.</p> <p>Se usan las características de AgentSpeak de manera adecuada, sin complicar la implementación más de lo necesario pero cumpliendo con patrones para razonamiento BDI.</p>
Diseño	30 %	<p>La documentación da una definición de racionalidad para los agentes en general, y para cada rol en particular.</p> <p>La definición de racionalidad se justifica por las características del entorno y los objetivos del juego.</p> <p>Dada esta definición de racionalidad, se diseña la estrategia a seguir por parte de cada agente, en términos de creencias, objetivos y planes e identificando los contextos que dan lugar a objetivos o planes diferentes.</p>

Figura 1: Rúbrica de evaluación de la práctica