



Prácticas AIN

Jason-JGOMAS

JADE Game Oriented MultiAgent System

Sesión 2

Primeros pasos con JGomas

Trabajo a Realizar

- ❖ Objetivos (en tres sesiones):

- ❖ 1. Implementar un agente **atacante** (allied) o **defensor** (axis) que imprima por pantalla su posición.
 - ❖ 2. . Implementar un agente **atacante** (allied) o **defensor** (axis) que imprima por pantalla la lista de objetos que ve.
 - ❖ 3. Implementar un agente **atacante** (allied) o **defensor** (axis) que recorra las esquinas del mapa.
 - ❖ 4. Implementar un agente **atacante** (allied) o **defensor** (axis) que localice un compañero y lo vaya siguiendo.
-

Trabajo a Realizar

- ❖ Organización.
 - ❖ Grupos de **dos** alumnos como máximo
 - ❖ Cada uno de los grupos elige equipo: allied o axis
 - ❖ Subir el código a la tarea de Poliformat
 - ❖ Entregar: código, memoria de una página con los cambios hechos
 - ❖ Documentación:
 - ❖ Programming Multi-Agent Systems in AgentSpeak using Jason
 - ❖ Documento “Jason JGOMAS - Manual.pdf”
 - ❖ Jason internal actions: <http://jason.sourceforge.net/api/jason/stdlib/package-summary.html>
 - ❖ Acciones para gestión de listas, conjuntos, strings, comunicación, etc...



Principales creencias (ver manual)

- ❖ `tasks(task_list)`
 - ❖ `fovObjects(object_list)`
 - ❖ `state(Estado_actual)`
 - ❖ `my_health(X)`
 - ❖ `my_ammo(X)`
 - ❖ `my_position(X,Y,Z)`
-

Principales planes (ver manual)

- ✧ `!init`
 - ✧ `!perform_look_action`
 - ✧ `!perform_aim_action`

 - ✧ `!get_agent_to_aim`

 - ✧ `!perform_no_ammo_action`
 - ✧ `!perform_injury_action`
 - ✧ `!performThresholdAction`

 - ✧ `!setup_priorities`
 - ✧ `!update_targets`
-

Código de un agente

❖ Inicio

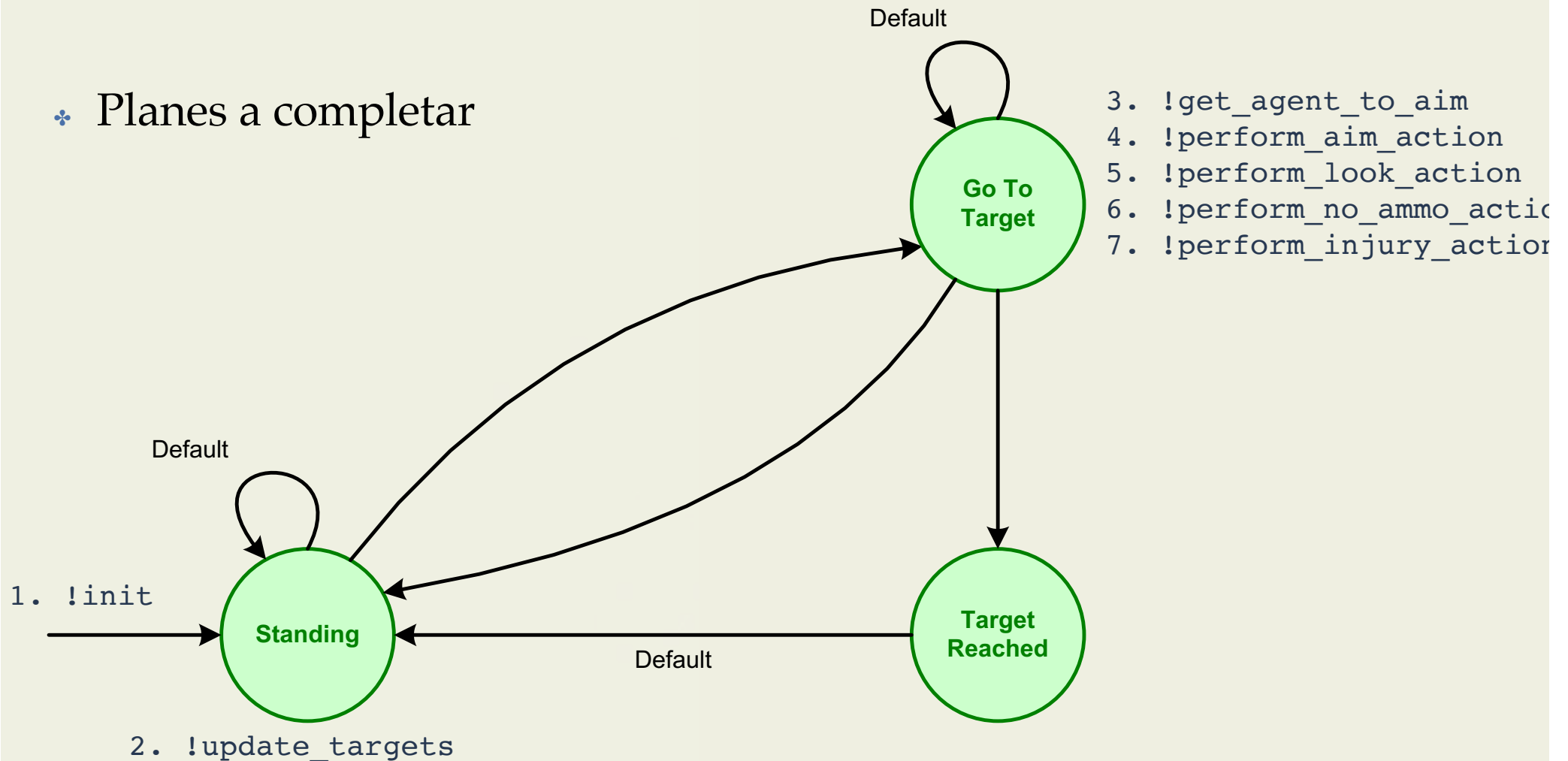
```
debug(3).  
// Name of the manager  
manager("Manager").  
// Team of troop.  
team("ALLIED").  
// Type of troop.  
type("CLASS_SOLDIER").  
{ include("jgommas.asl") }
```

❖ Planes (solo principales)

```
+!get_agent_to_aim    // veámoslo sobre el código  
+!perform_aim_action  
+!perform_look_action  
+!perform_no_ammo_action  
+!perform_injury_action  
+!update_targets  
+!init
```


Trabajo a Realizar

✧ Planes a completar



Trabajo a Realizar

- ❖ Objetivo 1:

- ❖ 1. Implementar un agente **atacante** (allied) o **defensor** (axis) que imprima por pantalla su posición.

IDEA

- ❖ Consultar creencia de posición
 - ❖ Pensar donde sería el plan adecuado para imprimir
-

Trabajo a Realizar

- ❖ Objetivo 2:

- ❖ 2. . Implementar un agente **atacante** (allied) o **defensor** (axis) que imprima por pantalla la lista de objetos que ve.

IDEA

- ❖ Consultar la creencia donde esté la lista de objetos
 - ❖ Pensar en que plan sería adecuado imprimirla
-

Trabajo a Realizar

- ❖ Objetivo 3:

- ❖ Implementar un agente **atacante** (allied) o **defensor** (axis) que recorra las esquinas del mapa.

IDEA

- ❖ Incorporar tareas del tipo “TASK_GOTO_POSITION”
 - ❖ Todas a la vez
 - ❖ De forma progresiva
 - ❖ Mejora: antes de incorporar la posición comprobad que ésta es válida
-

Trabajo a Realizar

- * ¿Cómo añadir tareas?

- * Se utiliza un plan: **add_task**

!add_task(task(TaskPriority, TaskType, Agent, Position, Content))

Ejemplos (dos opciones):

- * **!add_task(task(1000, "TASK_GET_OBJECTIVE", M, pos(ObjectiveX, ObjectiveY, ObjectiveZ), ""));**

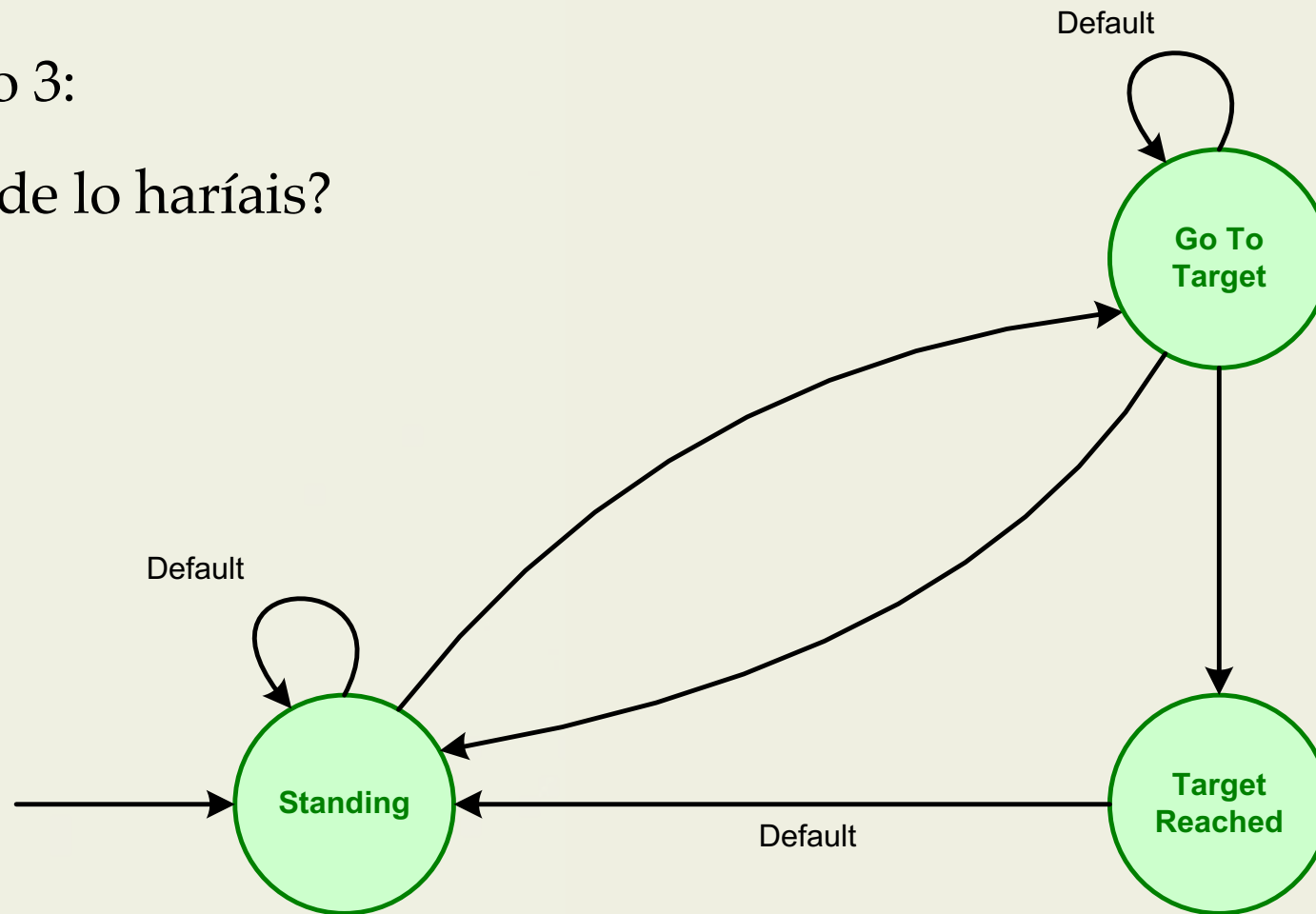
- * **!add_task(task("TASK_GET_OBJECTIVE", M, pos(ObjectiveX, ObjectiveY, ObjectiveZ), ""));**

Dichos objetivos disparan el plan que crea la tarea, el segundo le asigna la prioridad definida por el agente

Trabajo a Realizar

✧ Objetivo 3:

✧ ¿Dónde lo haríais?



Trabajo a Realizar

Objetivo 3:

- ❖ ¿Dónde lo haríais?
- ❖ Implementación dentro del plan *update_targets*
 - ❖ Se pueden añadir todas las ordenes a la vez (teniendo en cuenta las prioridades y que las ordenes sean distintas)
 - ❖ O una a una (implementando una máquina de estados)
 - ❖ *Opción más adecuada*

Trabajo a Realizar

Objetivo 3:

```
+!update_targets <-
```

```
// fsm
```

```
// consultar el valor de una creencia que simula un contador (X)
```

```
// si  $X = 0$  añadir tarea para ir a la primera esquina
```

```
// si  $X = 1$  añadir tarea para ir a la segunda esquina
```

```
// .... hasta completar todas las esquinas
```

Trabajo a Realizar (VI)

- ❖ Objetivo 4:
 - ❖ Implementar un agente **atacante** (allied) o **defensor** (axis) que localice un compañero y lo vaya siguiendo.
 - ❖ **Implementa** el plan asociado a +!perform_look_action
 - ❖ Analiza la creencia fovObjects(ObjList)
 - ❖ Comprueba si el agente ve a un compañero
 - ❖ Incorpora una tarea del tipo “TASK_GOTO_POSITION” cada vez que lo ve.
 - ❖ Antes de incorporar la posición comprobad que es válida
-

Trabajo a Realizar (VII)

❖ Objetivo 4:

- ❖ Para implementar el plan de `#!/perform_look_action`, podéis partir del plan ya existente:

`#!/get_agent_to_aim`

Su funcionamiento es muy similar para “ver” qué hay y realizar alguna acción.

Es decir

1º paso → copiar y pegar el código de *get_agent_to_aim* a *perform_look_action*

2º paso → modificar *perform_look_action*. En este caso la acción a realizar consiste en tratar de ir a donde hemos localizado a nuestro compañero.

Trabajo a Realizar (VIII)

- ✧ Prueba de comportamiento emergente:
 - ✧ Ejecutar una partida con 1 agente que recorra las esquinas y 15 agentes que sigan al primer agente amigo que vean
-

Ampliaciones (comportamiento individual)

- ❖ ALLIED

- ❖ Realizar un ataque ordenado manteniendo distancias entre miembros del mismo equipo
 - ❖ Para ello:
 - ❖ Controlar donde están mis compañeros
 - ❖ Ir moviéndome a posiciones alejadas de compañeros un cierto umbral
 - ❖ Jugando con el umbral se pueden conseguir distintas organizaciones de ataque
 - ❖ Importante: SIN COMUNICACIÓN
-

Ampliaciones (comportamiento individual)

- ❖ AXIS

- ❖ Realizar un control de que la bandera está en su sitio, si ya no se ve, moverse hacia el enemigo

- ❖ Para ello:

- ❖ Controlar continuamente que la bandera está en mi radio de visión (crear un agente con `patrollingRadius(8)`. o menor)

- ❖ Cuando deje de visualizar la bandera es porque el enemigo la ha cogido

- ❖ Si se da esa situación, el agente debe tomar la iniciativa e ir a por el enemigo

- ❖ Importante: SIN COMUNICACIÓN
