



Prácticas AIN

Jason-JGOMAS

JADE Game Oriented MultiAgent System

Sesión 1



Índice

- ❖ Introducción
 - ❖ Especificación
 - ❖ Uso
 - ❖ Trabajo a realizar
-



Índice

- ❖ ► **Introducción**
 - ❖ Especificación
 - ❖ Uso
 - ❖ Trabajo a realizar
-



Introducción (I)

- ❖ Desarrollo de un SMA en un entorno cooperativo y competitivo
 - ❖ Objetivo: Mejorar el comportamiento individual y colectivo
 - ❖ Uso de una plataforma de agentes sobre entornos 3D para simulaciones y videojuegos
 - ❖ Básicamente...
 - ❖ Un puñado de agentes repartidos en dos bandos...
 - ❖ ... con unos objetivos que cumplir...
 - ❖ ... integrados en un entorno virtual
-



Introducción (II)

- ❖ Taxonomía de Agentes → Diferentes tipos
 - ❖ Componente Estratégico
 - ❖ Integración en Entorno
 - ❖ Entre Agentes → cooperación
 - ❖ Con el Terreno → dificultad de movimiento
 - ❖ Comunicación → paso de mensajes
-



Índice

- ❖ Introducción
 - ❖ ► **Especificación**
 - ❖ Uso
 - ❖ Trabajo a realizar
-



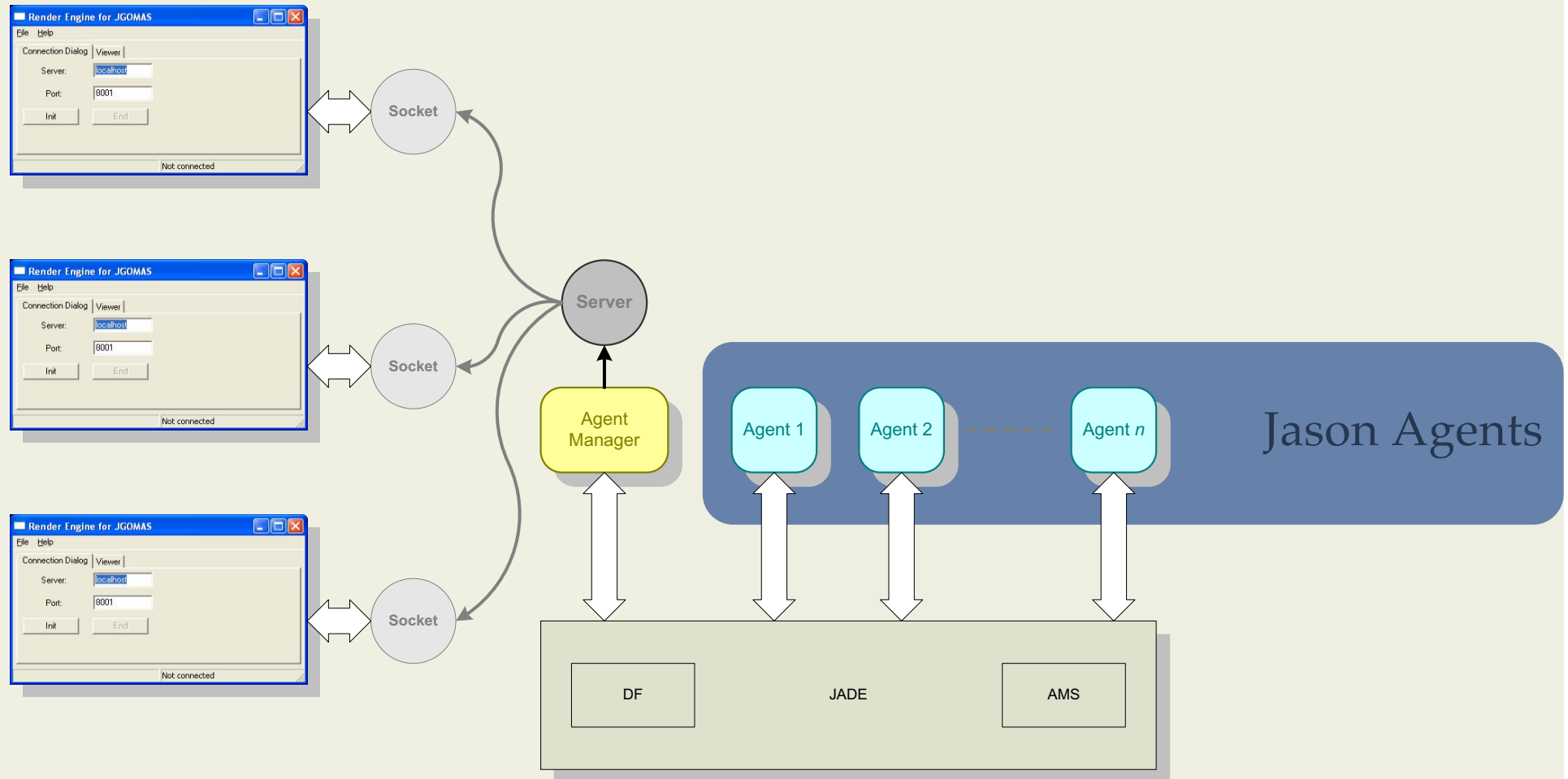
Especificación

- ❖ Arquitectura
 - ❖ Tareas
 - ❖ Bucle de Ejecución
 - ❖ Interfaz (API)
-

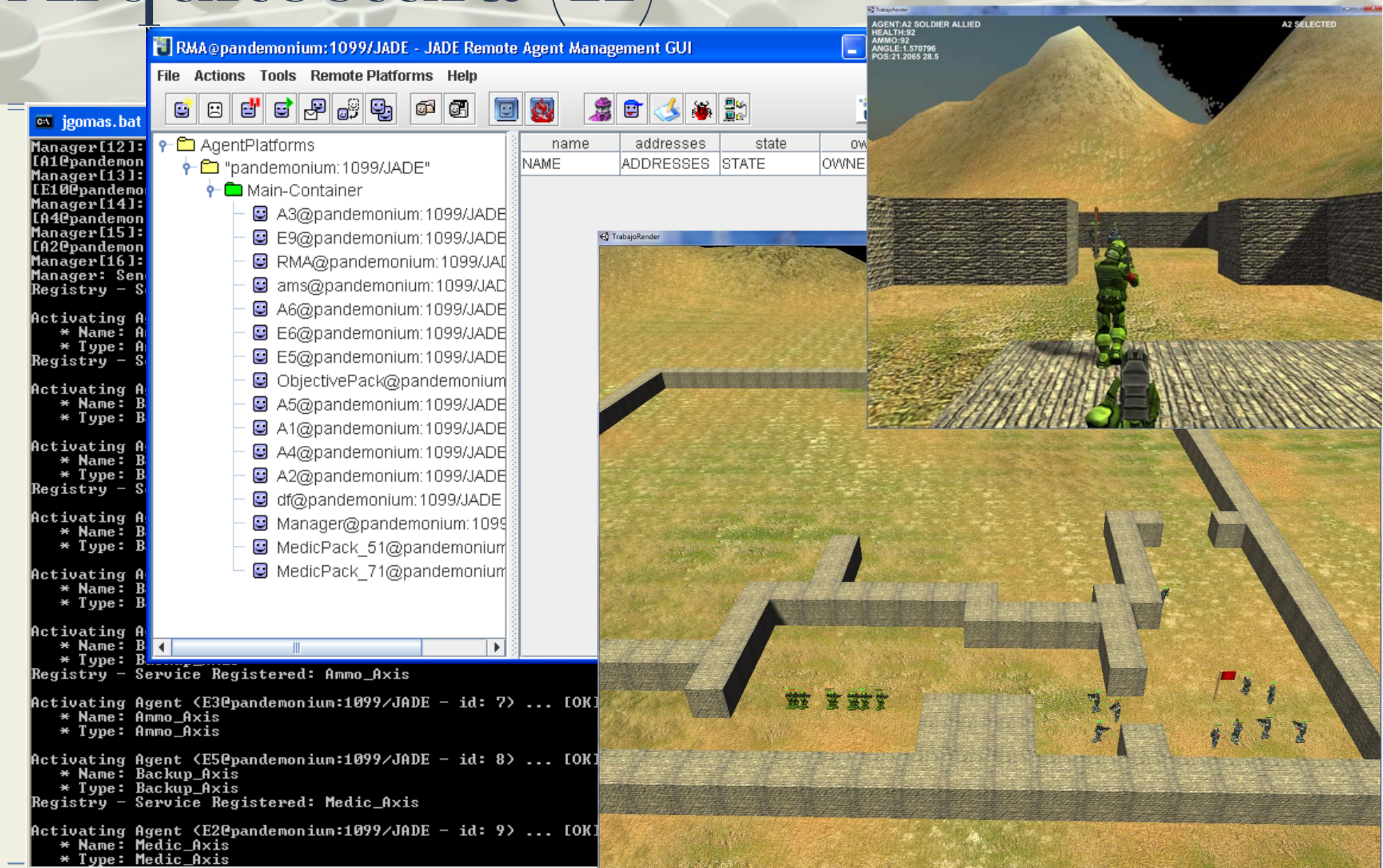
Especificación

- ❖ ► **Arquitectura**
 - ❖ Tareas
 - ❖ Bucle de Ejecución
 - ❖ Interfaz (API)
-

Arquitectura (I)



Arquitectura (II)



Arquitectura (III) Mapas

- ✧ Una partida se desarrolla en un entorno virtual (mundo virtual) de 256×256
- ✧ Posición de los agentes:

$$x [0 \dots 255], \quad z [0 \dots 255]$$

- ✧ Cada agente tiene acceso parcial al mapa donde se desarrolla la partida:
 - ✧ Sabe su posición (puede no ser la real)
 - ✧ Ve objetos cuando está a cierta distancia
-

Arquitectura (IV) Mapas

- ❖ Los mapas se almacenan en la carpeta `bin\data\maps`
 - ❖ Existe una subcarpeta por cada mapa con el nombre `map_XX`, donde `XX` es el número del mapa.
 - ❖ Ejemplo: `map_04`
 - ❖ `map_04_cost.txt`: define las paredes del mapa por medio del uso de *
 - ❖ `map_04_terrain.bmp`: aspecto artístico del mapa.
 - ❖ `map_04_cost.bmp`: Este fichero define las paredes del mapa usando una imagen en blanco y negro, donde el blanco representa la pared.
 - ❖ `map_04.txt`: contiene la definición de diferentes parámetros de configuración para el SMA y el visor gráfico
-

Arquitectura (V) Mapas

map_04.txt

- JADE_OBJECTIVE: Localización inicial de la bandera.
 - JADE_SPAWN_ALLIED: Localización de la base Aliada.
 - JADE_SPAWN_AXIS: Localización de la base del Eje.
 - JADE_COST_MAP: Tamaño y nombre del fichero de costes.
 - RENDER_ART_MAP: Tamaño y nombre del fichero de arte.
 - RENDER_COST_MAP: Tamaño y nombre del fichero de arte de los costes.
 - RENDER_HEIGHT_MAP: Tamaño y nombre del fichero de arte de las alturas.
-

Arquitectura (V) Mapas

❖ Contenido de map_04.txt

[JADE]

JADE_OBJECTIVE: 28 28

JADE_SPAWN_ALLIED: 2 28 4 30

JADE_SPAWN_AXIS: 20 28 22 30

JADE_COST_MAP: 32 32 map_04_cost.txt

[JADE]

[RENDER]

RENDER_ART_MAP: 256 256 map_04_terrain.bmp

RENDER_COST_MAP: 32 32 map_04_cost.bmp

RENDER_HEIGHT_MAP: 32 32 map_04_heightmap.bmp

[RENDER]

Especificación

- ✧ Arquitectura
 - ✧ ► **Tareas**
 - ✧ Bucle de Ejecución
 - ✧ Interfaz (API)
-

Especificación Tareas (I)

- ❖ Definición

task(Priority, TaskType, Agent, Position, Content)

- ❖ Priority: Prioridad de la tarea
 - ❖ TaskType: Tipo de tarea (siguiente transparencia)
 - ❖ Agent: agente asociado a la tarea
 - ❖ Position: Posición donde llevar a cabo la tarea
 - ❖ Content: Posible contenido adicional
-

Especificación Tareas (II)

- ✧ Algunos de los tipos de tareas son:
 - ✧ TASK_GIVE_MEDICPAKS
 - ✧ TASK_GIVE_AMMOPAKS
 - ✧ TASK_GIVE_BACKUP
 - ✧ TASK_GET_OBJECTIVE
 - ✧ TASK_GOTO_POSITION
 - ✧ ...
-

Especificación Tareas (III)

- ❖ **Priority** indica la prioridad de la tarea
 - ❖ Se lanza siempre la tarea de prioridad **más alta**
 - ❖ Es posible **redefinir la prioridad** de cada tipo de tarea
 - ❖ Las tareas las pone en ejecución el sistema, no el usuario
-



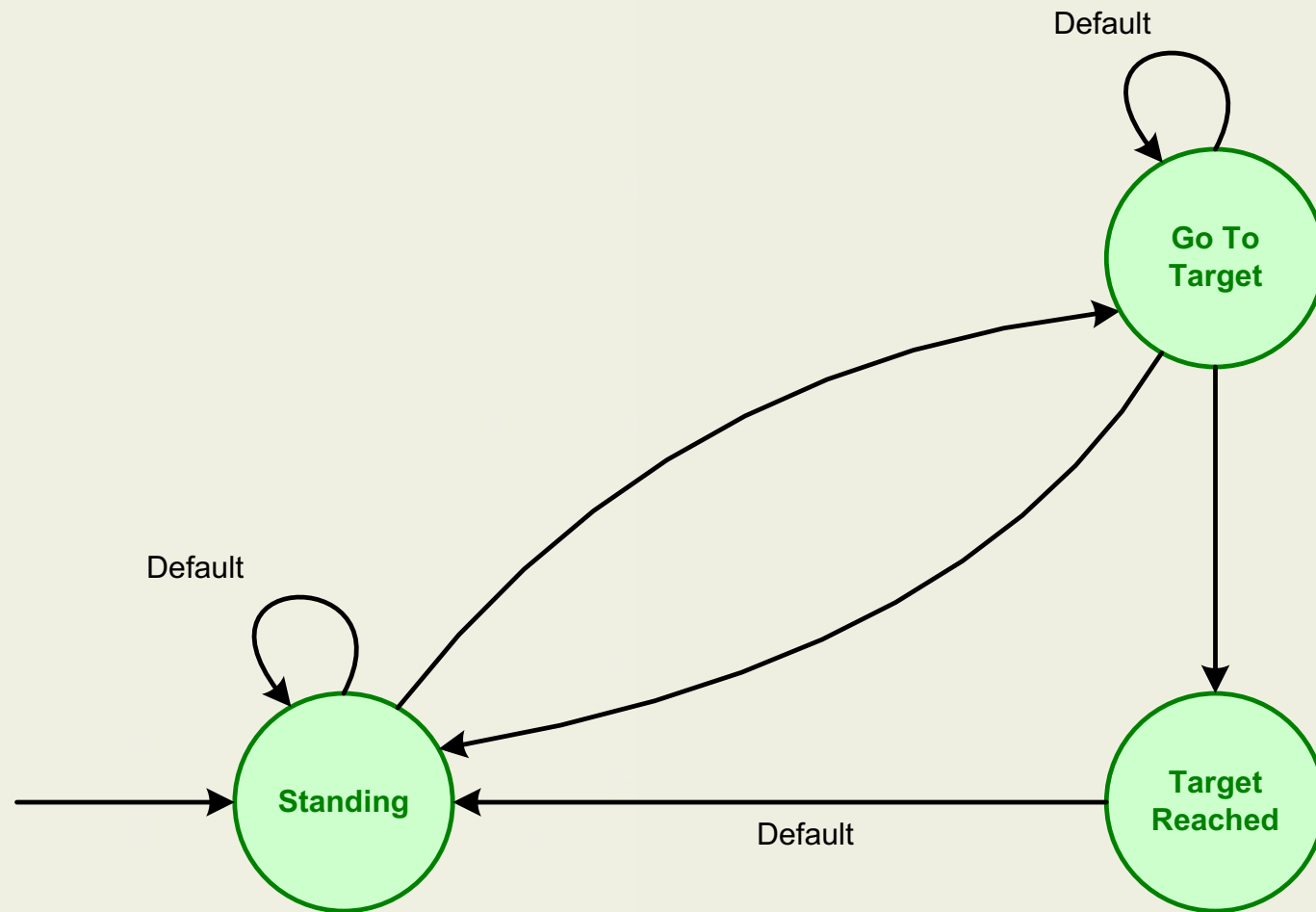
Especificación

- ❖ Arquitectura
 - ❖ Tareas
 - ❖ ► **Bucle de Ejecución**
 - ❖ Interfaz (API)
-

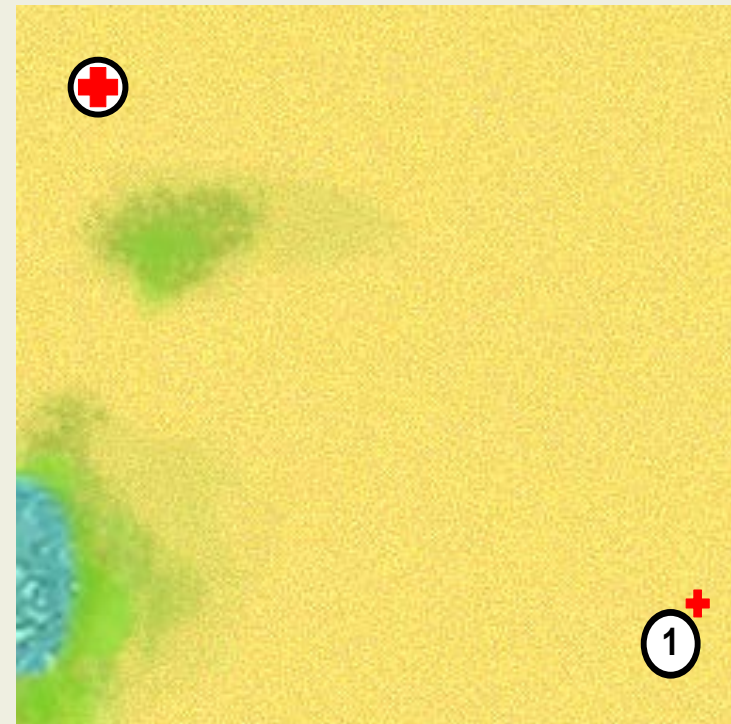
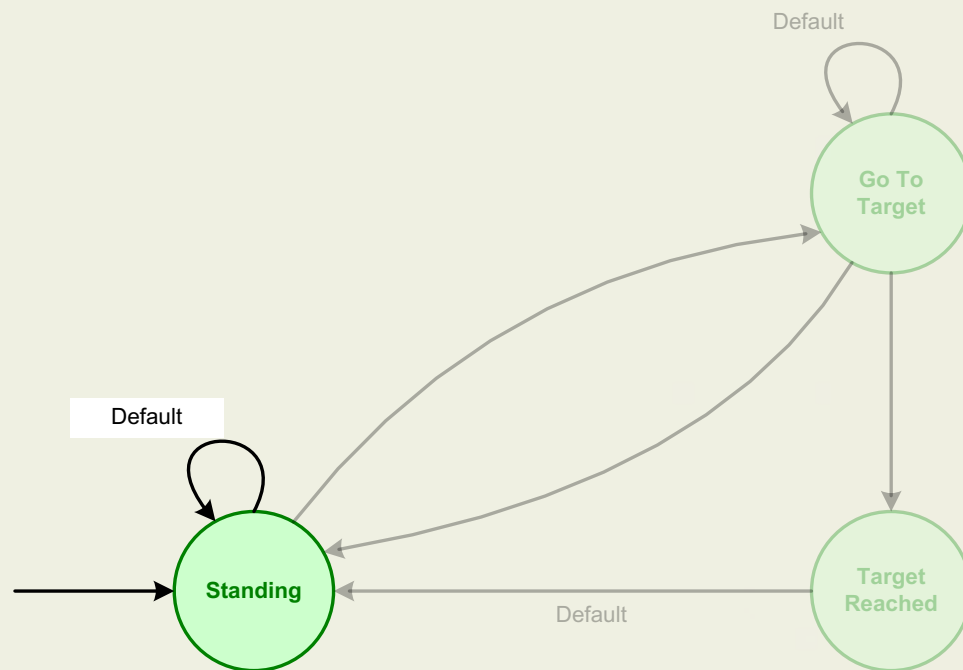
Bucle de Ejecución (I)

- ❖ Cada agente ejecuta una FSM (4 estados):
 - ❖ STANDING
 - ❖ GO_TO_TARGET
 - ❖ TARGET_REACHED
 - ❖ QUIT
 - ❖ FSM se utiliza para realizar tareas:
 - ❖ Inicio (Lanzamiento)
 - ❖ Desarrollo (Ejecución)
 - ❖ Final (Acción y Destrucción)
 - ❖ Se lanza siempre la tarea de prioridad más alta
-

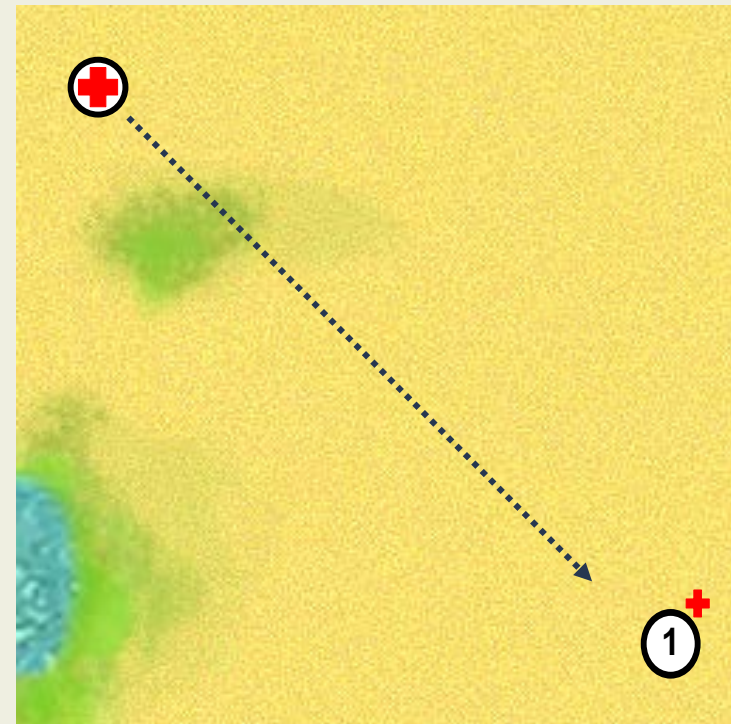
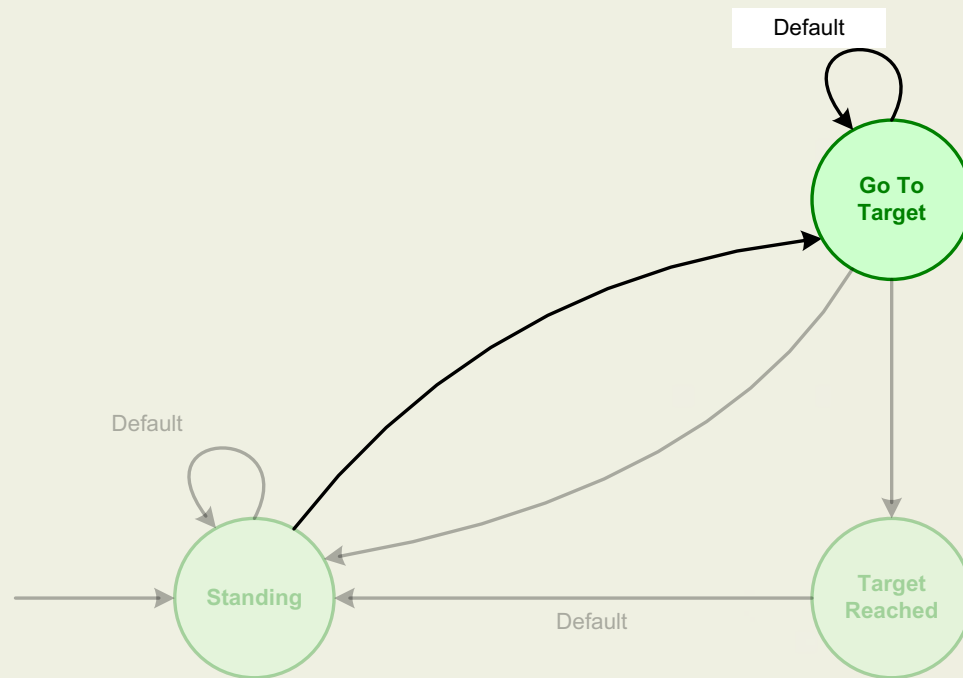
Bucle de Ejecución (II)



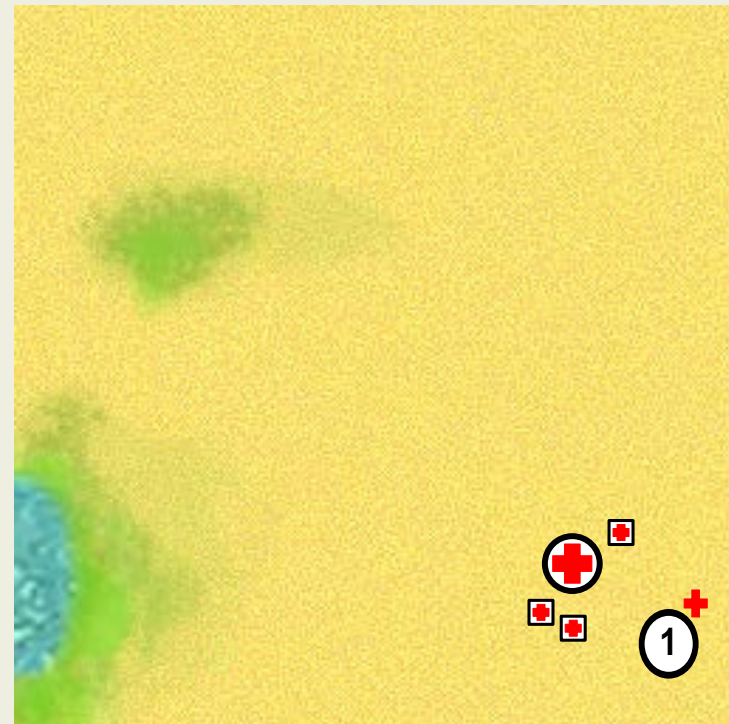
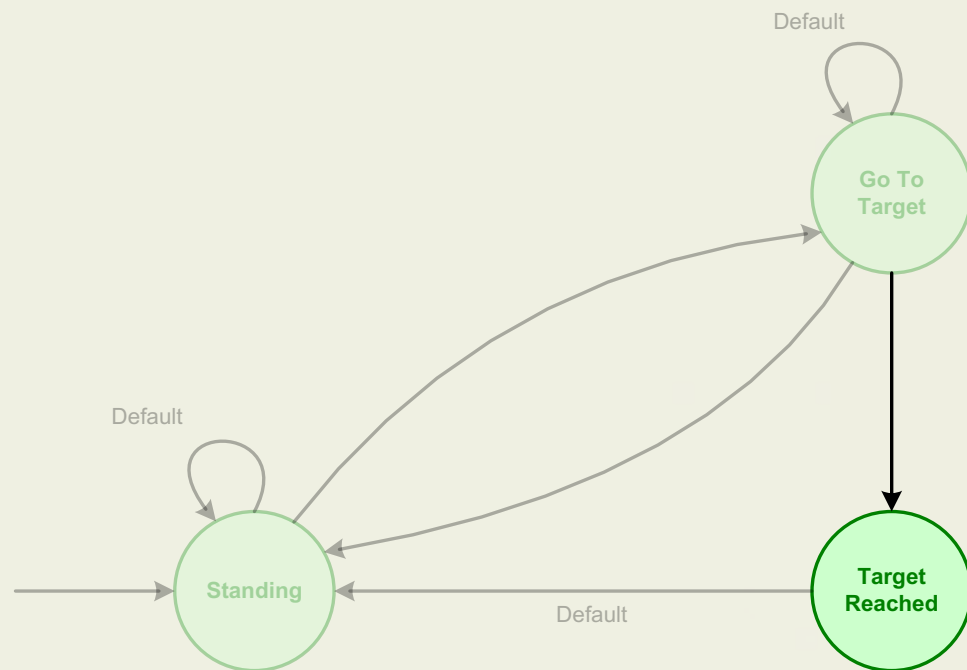
Ejemplo (I)



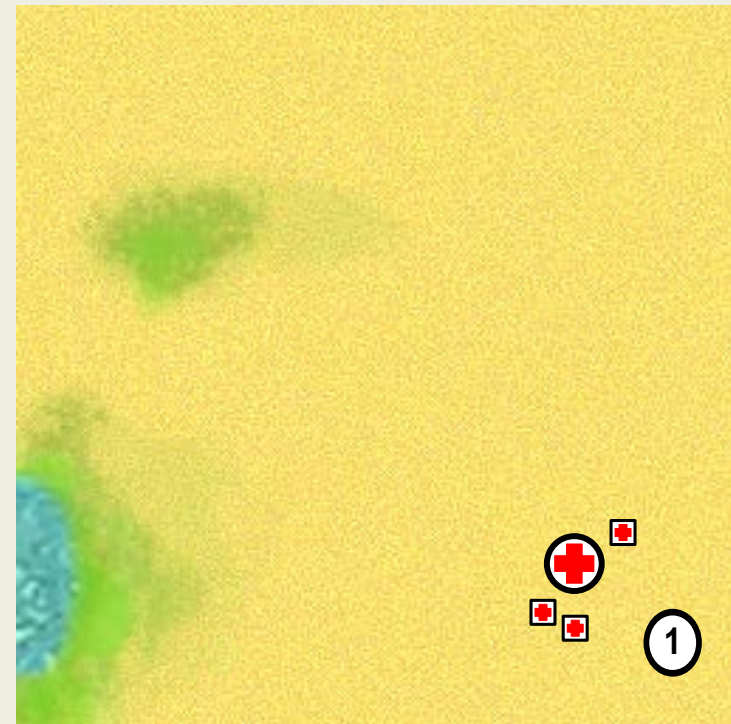
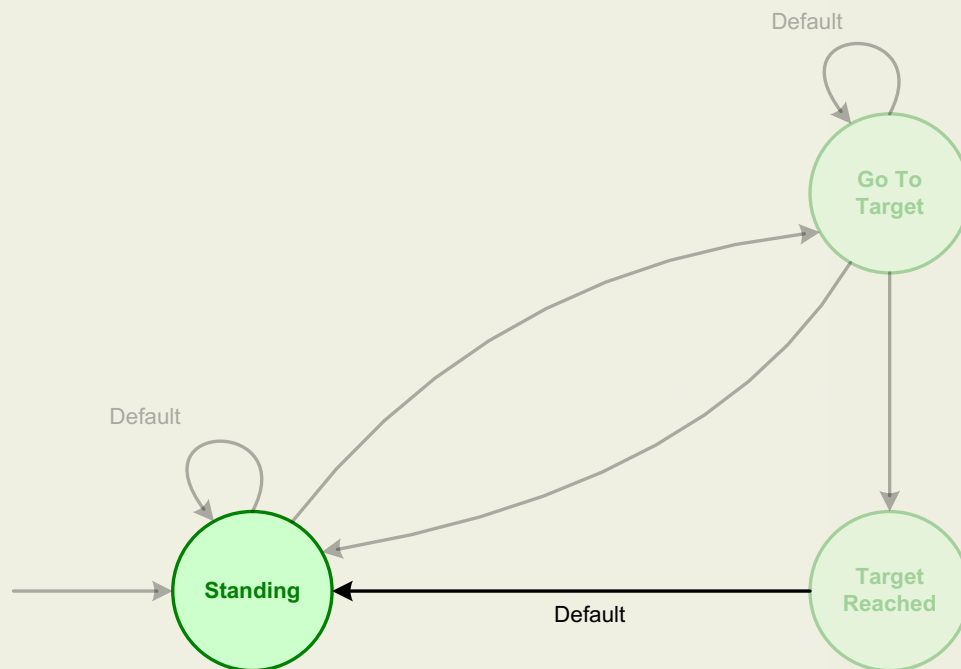
Ejemplo (II)



Ejemplo (III)



Ejemplo (IV)



Especificación

- ✧ Arquitectura
 - ✧ Tareas
 - ✧ Bucle de Ejecución
 - ✧ ► **Interfaz (API)**
-



Interfaz

jgommas.asl → Fichero con el comportamiento no modificable del agente

jasonAgent_TEAM_TYPE.asl → 6 ficheros para distintos agentes

jasonAgent_ALLIED.asl

jasonAgent_ALLIED_MEDIC.asl

jasonAgent_ALLIED_FIELDOPS.asl

jasonAgent_AXIS.asl

jasonAgent_AXIS_MEDIC.asl

jasonAgent_AXIS_FIELDOPS.asl



Índice

- ❖ Introducción
 - ❖ Especificación
 - ❖ ► **Uso**
 - ❖ Trabajo a realizar
-

Uso: MAS

- * Ejecución en dos pasos:

1º paso → Lanzar el manager

- * Línea de comandos:

```
java -classpath  
"lib\jade.jar;lib\jadeTools.jar;lib\Base64.jar;lib\http.jar;li  
b\iiop.jar;lib\beangenerator.jar;lib\jgomas.jar;student.jar;li  
b\jason.jar;lib\JasonJGomas.jar;classes;." jade.Boot -gui  
"Manager:es.upv.dsic.gti_ia.jgomas.Cmanager(4, map_04, 125,  
10) "
```

Manager:es.up...CManager(4,map_04,125,10)

↑
Nombre
Agente

↑
Clase

↑
Parámetros:

4: n° de agentes
map_04: nombre del mapa
125: refresco en ms.
10: duración (minutos)

Uso: MAS

2º Paso → Lanzar los agentes

✧ Línea de comandos:

```
java -classpath  
"lib\jade.jar;lib\jadeTools.jar;lib\Base64.jar;lib\http.jar;lib\iiop.  
jar;lib\beangenerator.jar;lib\jgomas.jar;student.jar;lib\jason.jar;li  
b\JasonJGomas.jar;classes;" jade.Boot -container -host localhost  
"T1:es.upv.dsic.gti_ia.JasonJGomas.BasicTroopJasonArch(jasonAgent_AXI  
S.asl);T2:es.upv.dsic.gti_ia.JasonJGomas.BasicTroopJasonArch(jasonAge  
nt_AXIS_MEDIC.asl);A1:es.upv.dsic.gti_ia.JasonJGomas.BasicTroopJasonA  
rch(jasonAgent_ALLIED_FIELDOPS.asl);A2:es.upv.dsic.gti_ia.JasonJGomas  
.BasicTroopJasonArch(jasonAgent_ALLIED.asl)"
```

T1:es.up... BasicTroopJasonArch(jasonAgent_AXIS.asl)

↑
Nombre
Agente

↑
Clase

↑
Parámetros:
Fichero asl con el código

Uso: Render Engine

- ❖ Se dispone de 3 aplicaciones para la visualización de la partida:
 - ❖ En Unity3D: válido para Windows, Linux y Mac
 - ❖ En Python: válido para todos los sistemas (Linux, Windows, Mac)
 - ❖ En OpenGL: válido para S.O. Windows
-

Uso: Render Engine

- ✧ En Unity:
 - ✧ Descargarse la versión del operativo correspondiente
 - ✧ Lanzar el ejecutable
 - ✧ Seleccionar la resolución y calidad gráfica, pulsar Play!
 - ✧ En la parte superior izda. indicar la dirección IP y pulsar *connect*

Se puede controlar a un agente desde el render

Uso: Render Engine

- ✧ En Python:
 - ✧ La visualización es en modo texto.
 - ✧ Disponible en el directorio `jgomas/bin/render/Python`
 - ✧ Uso:

```
bash$ python textviewer.py
```

Uso: Render Engine

- ✧ En OpenGL:
 - ✧ La visualización es en 3D.
 - ✧ Disponible en el directorio `jgomas/bin/render/w32`
 - ✧ Lanzamos el fichero batch

```
c:\> run_jgomasrender.bat
```

- ✧ Cuyo contenido es

```
set OSG_FILE_PATH=../../data  
JGOMAS_Render.exe --server <hostname> --port <integer>
```



Índice

- ❖ Introducción
 - ❖ Especificación
 - ❖ Uso
 - ❖ ► **Trabajo a realizar**
-



Prueba

Realizar distintas ejecuciones con distintos números de agentes, clases y mapas.

- Equipo de 2 contra 2 del tipo soldado
 - Equipos de 8 aliados y 4 del eje: 1 médico y 1 fieldops en cada bando
 - Equipos de 4 aliados y 8 del eje: 1 médico y 1 fieldops en cada bando
-



Prueba

- Probad diferentes mapas map_01, map02, map_04
 - **Añadir una copia del mapa map_01 (por ejemplo map_111), donde se modifiquen las bases con el objeto de que se encuentren cercanas y los soldados se vean nada más nacen.**
 - Modificar la duración de la partida
 - Probar diferentes visualizadores en función del sistema operativo
 - La creencia debug(X) permite cambiar la verbosidad del agente (su valor es entre 1 y 3), probad a cambiar la verbosidad, modificando la creencia en alguno de los ficheros de los agentes
-

Trabajo a Realizar (ahora y en casa)

1. Leerse el documento “JGomas Manual Jason.pdf”
2. Mirar con detalle el código de los ficheros “.asl”

Sobretudo:

- ❖ jgommas.asl (no se puede modificar)
 - ❖ jasonAgent_ALLIED.asl
 - ❖ jasonAgent_AXIS.asl
-