



Game Oriented Multi Agent System, based on Jade

*Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Camino de Vera s/n. 0622 Valencia (Spain)*

Antonio Barella Álvarez

email: tbarella(at)dsic(dot)upv(dot)es

Professor Vicente J. Botti Navarro

email: vbotti(at)dsic(dot)upv(dot)es

Dr. Carlos Carrascosa Casamayor

email: carrasco(at)dsic(dot)upv(dot)es

Presentación

- Visión general
- Descripción de JGOMAS



- Plataforma de agentes sobre entornos 3D para simulaciones y videojuegos
- Juego: *Capturar la Bandera*
(*Capture The Flag- CTF*)
- Básicamente ...
 - Un puñado de agentes repartidos en dos bandos ...
 - ... con unos objetivos que cumplir ...
 - ... integrados en un entorno virtual



■ Descargar JGOMAS:

- Sitio JGOMAS (UPV)

- Downloads

jgomas.jar (1.0.0)

- Descomprimir: se genera una carpeta jgomas



■ Antes de trabajar con eclipse vamos a ejecutarlo :

- Abrir carpeta: **jgomas** → **bin** → **mas**
- Desplazar **student.jar** a la carpeta **lib**
- Editar archivo: **jgomas_example.bat**

```
java
-classpath lib\jade.jar;lib\jadeTools.jar;lib\Base64.jar;lib\http.jar;
lib\iiop.jar;lib\beangenerator.jar;lib\jgomas.jar;lib\student.jar
;..
jade.Boot -gui
Manager:es.upv.dsic.gti_ia.jgomas.CManager(2 map_01 125
10)
A1:student.MyMedic(ALLIED)
E2:student.MyMedic(AXIS)
```

actualizar



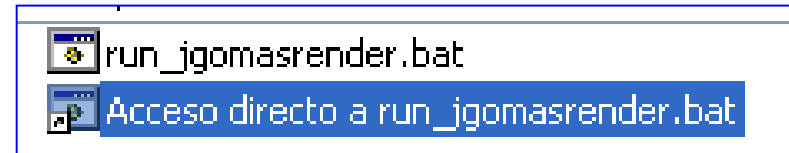
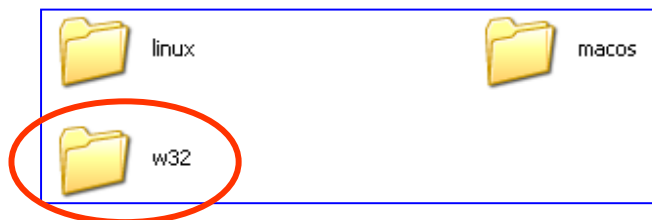
■ Antes de trabajar con eclipse vamos a ejecutarlo :

□ Subimos a la carpeta bin (contiene a mas):

■ jgomas → **bin** → mas

■ Abrir: **render**

■ Abrir: **w32**

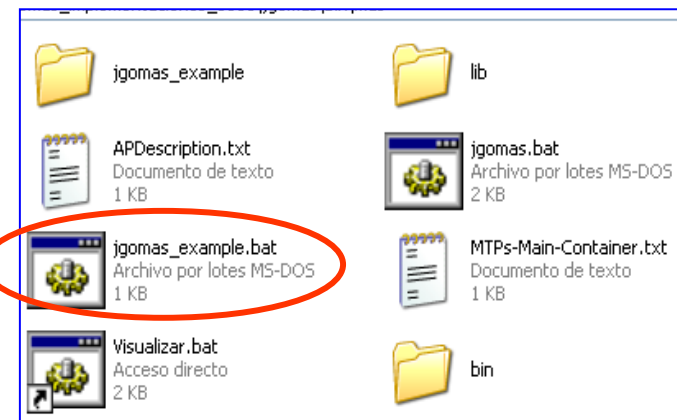


■ Crear acceso directo de: **run_jgomasrender.bat**

■ Cambiarle el nombre: **Visualiza.bat**

■ Llevarlo a la carpeta **mas**

□ Ejecutar: **jgomas_example.bat**



■ Crear proyecto en eclipse:

- Crear Workspace: JGOMAS

- Crear New Java Project

Project name: practica

- **Seleccionar:** Create Project from existing source

- **Buscar la carpeta del proyecto:** Browse...

- Seleccionar carpeta: **jpgomas** → **bin** → **mas**

- Agregar las librerías externas de jade (están en lib)



■ Ejecutar **ejemplo1** JGOMAS:

□ Crear interfaz-ejecutable:

Open Run Dialog → **New Java Application** →

□ **Name:** **example1**

□ **MainClass:** jade.Boot

□ **Arguments** (copiar de **jgomas_example.bat**):

-gui

Manager:es.upv.dsic.gti_ia.jgomas.CManager(2 map_01 125 10)

A1:student.MyMedic(ALLIED) E2:student.MyMedic(AXIS)

□ **Apply -> Run**



- Visualizar **ejemplo1 JGOMAS:**
 - GUI JADE: **visualizar**

The screenshot displays the JADE (Java Agent Development Environment) GUI. The main window is titled "example1_jgomas [Java Application]" and shows the console output of the application. The console output includes the following text:

```
example1_jgomas [Java Application] C:\Archivos de programa\Java\jre1.6.0_07\bin\javaw.exe (20/03/2009 9:25:57)
INFO: Service jade.core.event.Notification initialized
20-mar-2009 9:26:00 jade.core.AgentContainerImpl joinPlatform
INFO: -----
Agent container Main-Container@JADE-IMTP://G003790 is ready.
-----
JGOMAS v. 0.1.1 (c) GTI-IA 2005 - 2006 (DSIC / UPV)

Activating Agent (Manager@G003790:1099/JADE - id: 0)
* Name: JGomasManager
* Type: Management
Registering in DF ... [OK]
Manager (Expected Agents): 2
Manager: [A1] is Ready!
Manager: [E2] is Ready!
Manager (Accepted Agents): 2
Manager[1]: Sending notification to fight to: A1
Manager[2]: Sending notification to fight to: E2
Manager: Sending Objective notification to agents
[E2@G003790:1099/JADE]: Beginning to fight
[A1@G003790:1099/JADE]: Beginning to fight
Registry - Service Registered: Medic Axis
```

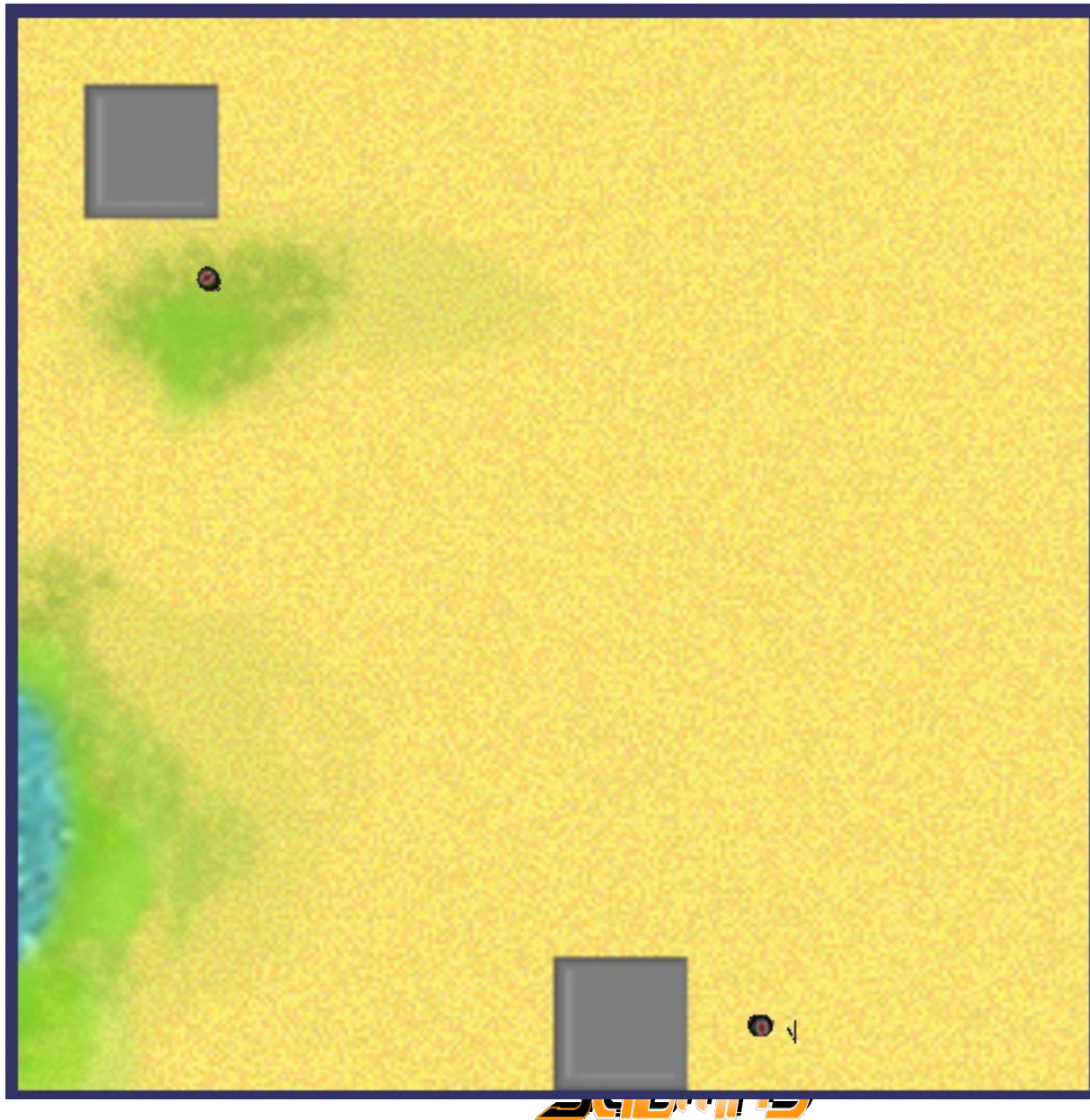
Overlaid on the bottom right is the "RMA@G003790:1099/JADE - JADE Remote Agent Management" window. This window shows a tree view of the agent platforms and a table of agents.

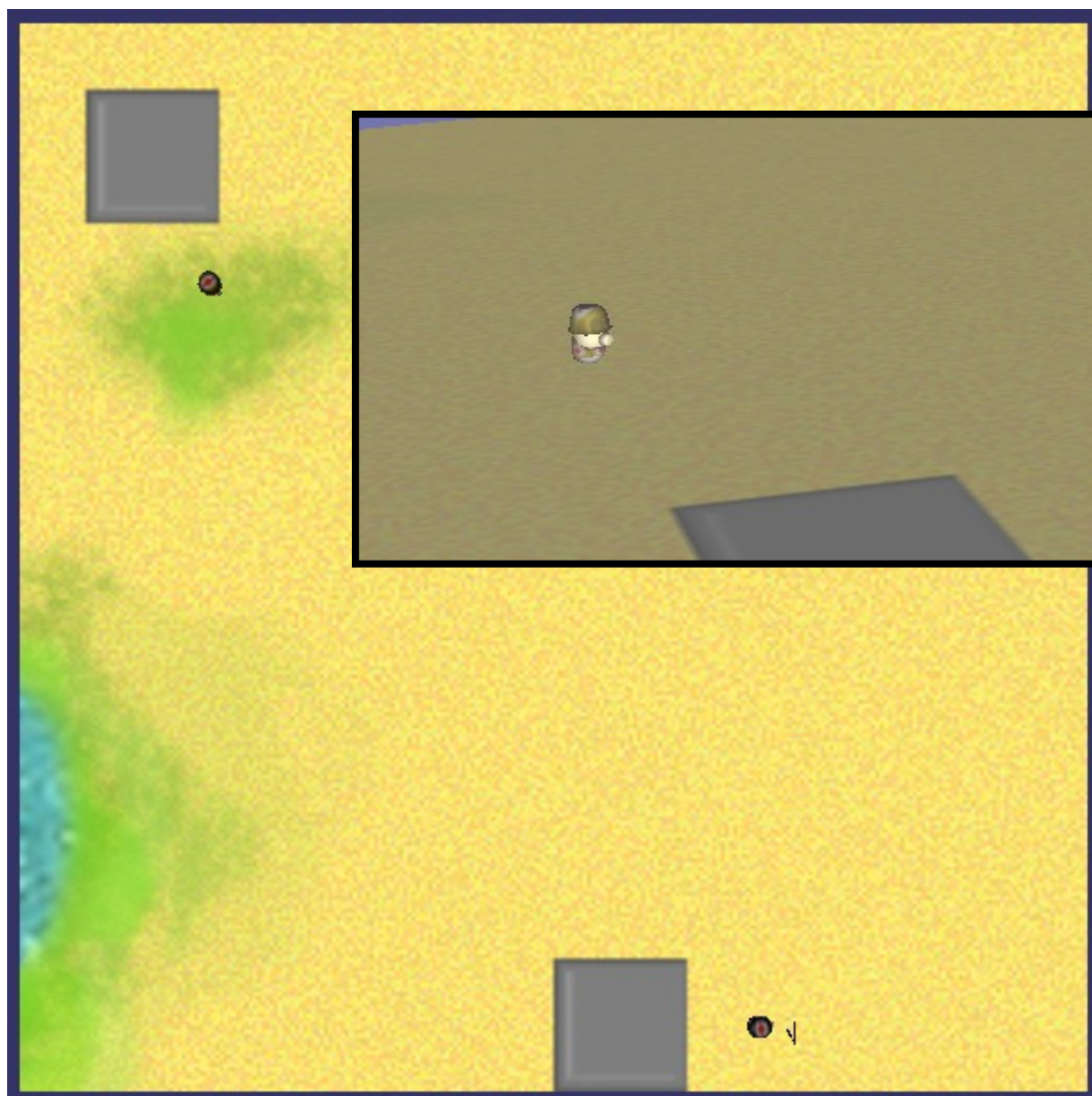
The tree view shows the following structure:

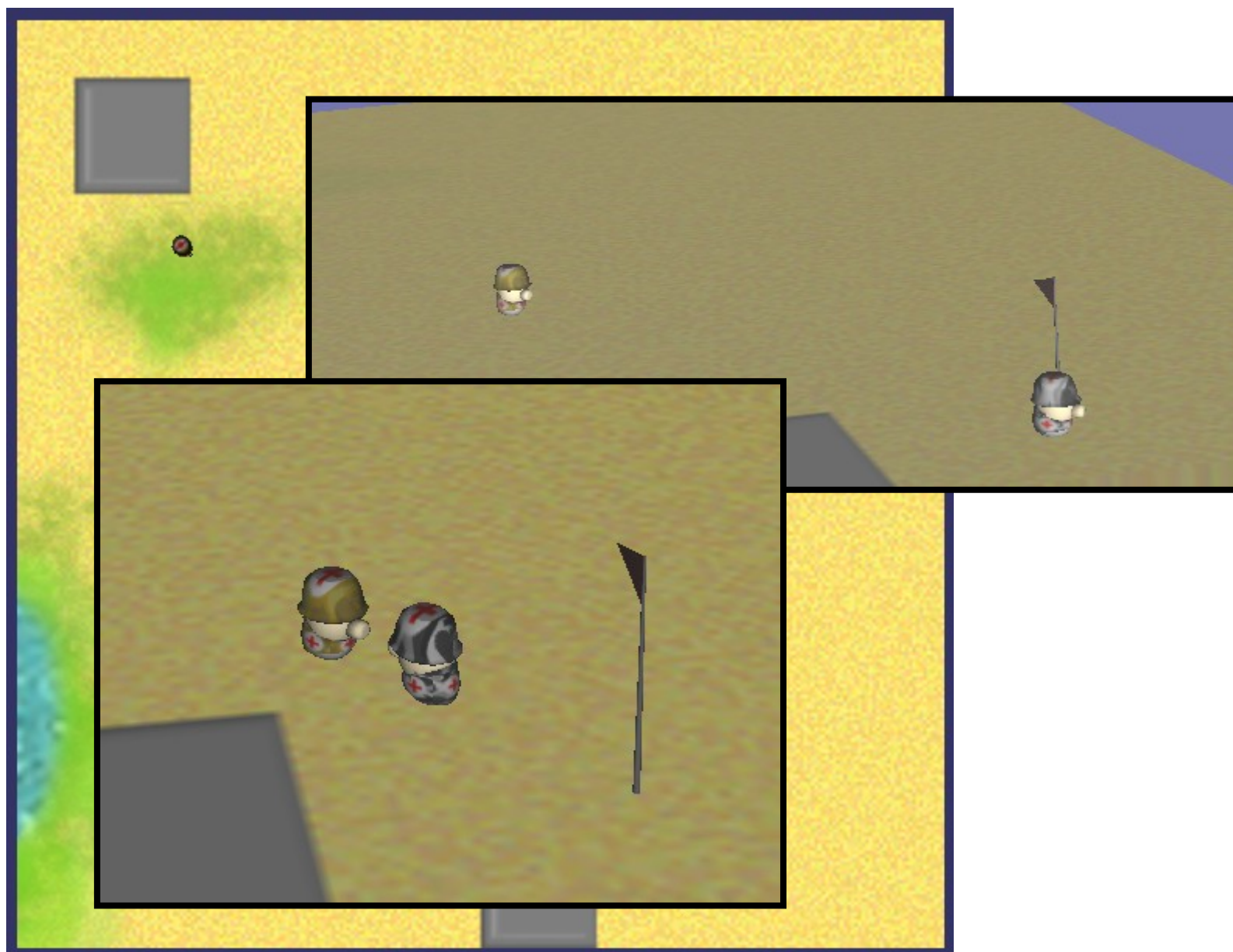
- AgentPlatforms
 - "G003790:1099/JADE"
 - Main-Container
 - A1@G003790:1099/JADE
 - RMA@G003790:1099/JADE
 - ObjectivePack@G003790:1099/JADE
 - E2@G003790:1099/JADE
 - ams@G003790:1099/JADE
 - df@G003790:1099/JADE

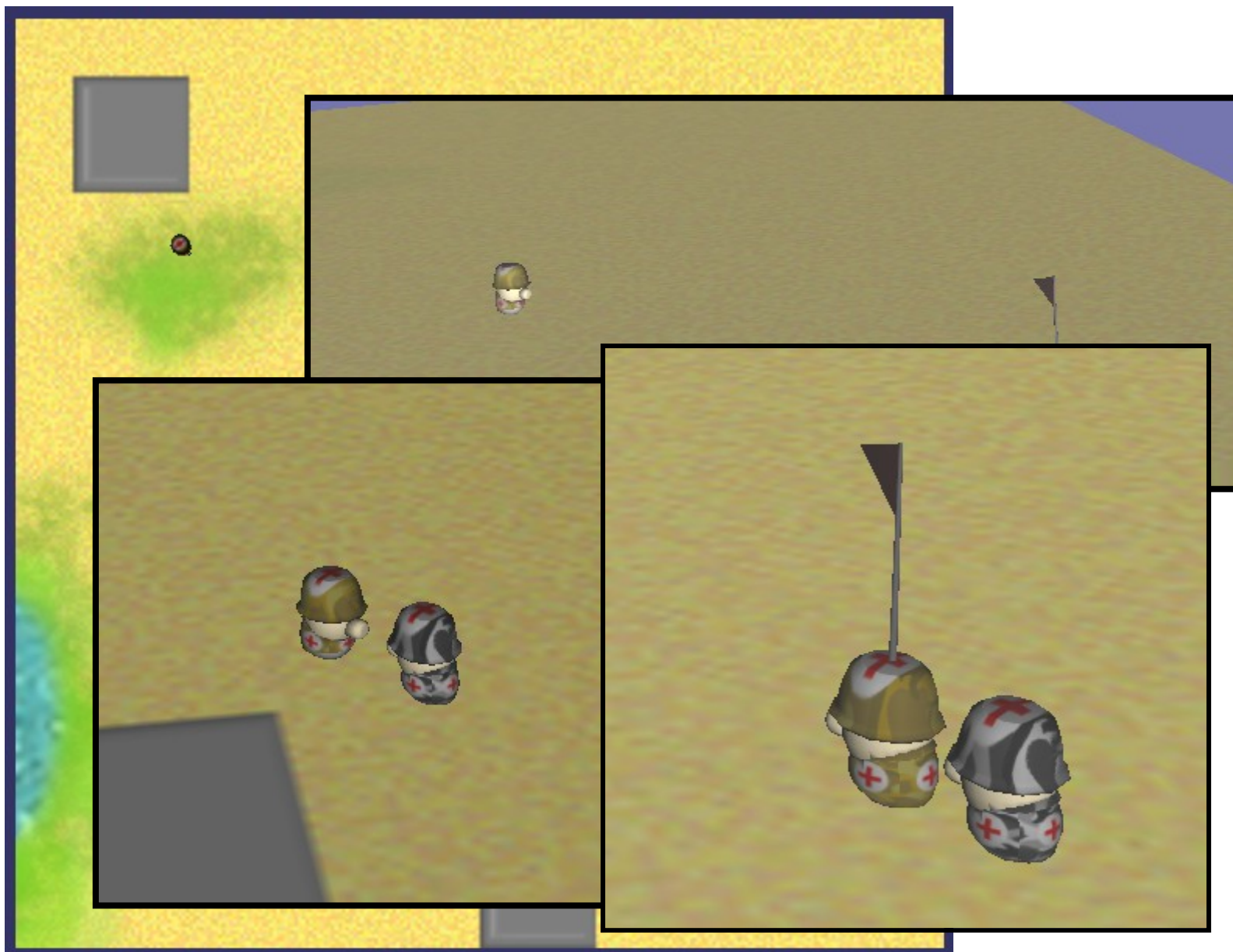
The table on the right shows the following data:

name	ad
NAME	AD











■ Ejecutar **ejemplo2** de JGOMAS, **jgommas.bat**:

☐ Crear interfaz-ejecutable:

Open Run Dialog → **New Java Application** →

☐ **Name:** **example2**

☐ **MainClass:** jade.Boot

☐ **Arguments** (copiar de **jgommas.bat**):

☐ **Apply -> Run**



Manager:es.upv.dsic.gti_ia.jgomas.CManager(20 map_01 125 10)

A1:es.upv.dsic.gti_ia.jgomas.CMedic(ALLIED)

E1:es.upv.dsic.gti_ia.jgomas.CMedic(AXIS)

A2:es.upv.dsic.gti_ia.jgomas.CMedic(ALLIED)

E2:es.upv.dsic.gti_ia.jgomas.CMedic(AXIS)

A3:es.upv.dsic.gti_ia.jgomas.CFieldOps(ALLIED)

E3:es.upv.dsic.gti_ia.jgomas.CFieldOps(AXIS)

A4:es.upv.dsic.gti_ia.jgomas.CFieldOps(ALLIED)

E4:es.upv.dsic.gti_ia.jgomas.CFieldOps(AXIS)

A5:es.upv.dsic.gti_ia.jgomas.CSoldier(ALLIED)

E5:es.upv.dsic.gti_ia.jgomas.CSoldier(AXIS)

A6:es.upv.dsic.gti_ia.jgomas.CSoldier(ALLIED)

E6:es.upv.dsic.gti_ia.jgomas.CSoldier(AXIS)

A7:es.upv.dsic.gti_ia.jgomas.CMedic(ALLIED)

E7:es.upv.dsic.gti_ia.jgomas.CMedic(AXIS)

A8:es.upv.dsic.gti_ia.jgomas.CMedic(ALLIED)

E8:es.upv.dsic.gti_ia.jgomas.CMedic(AXIS)

A9:es.upv.dsic.gti_ia.jgomas.CFieldOps(ALLIED)

E9:es.upv.dsic.gti_ia.jgomas.CFieldOps(AXIS)

A10:es.upv.dsic.gti_ia.jgomas.CFieldOps(ALLIED)

E10:es.upv.dsic.gti_ia.jgomas.CFieldOps(AXIS)

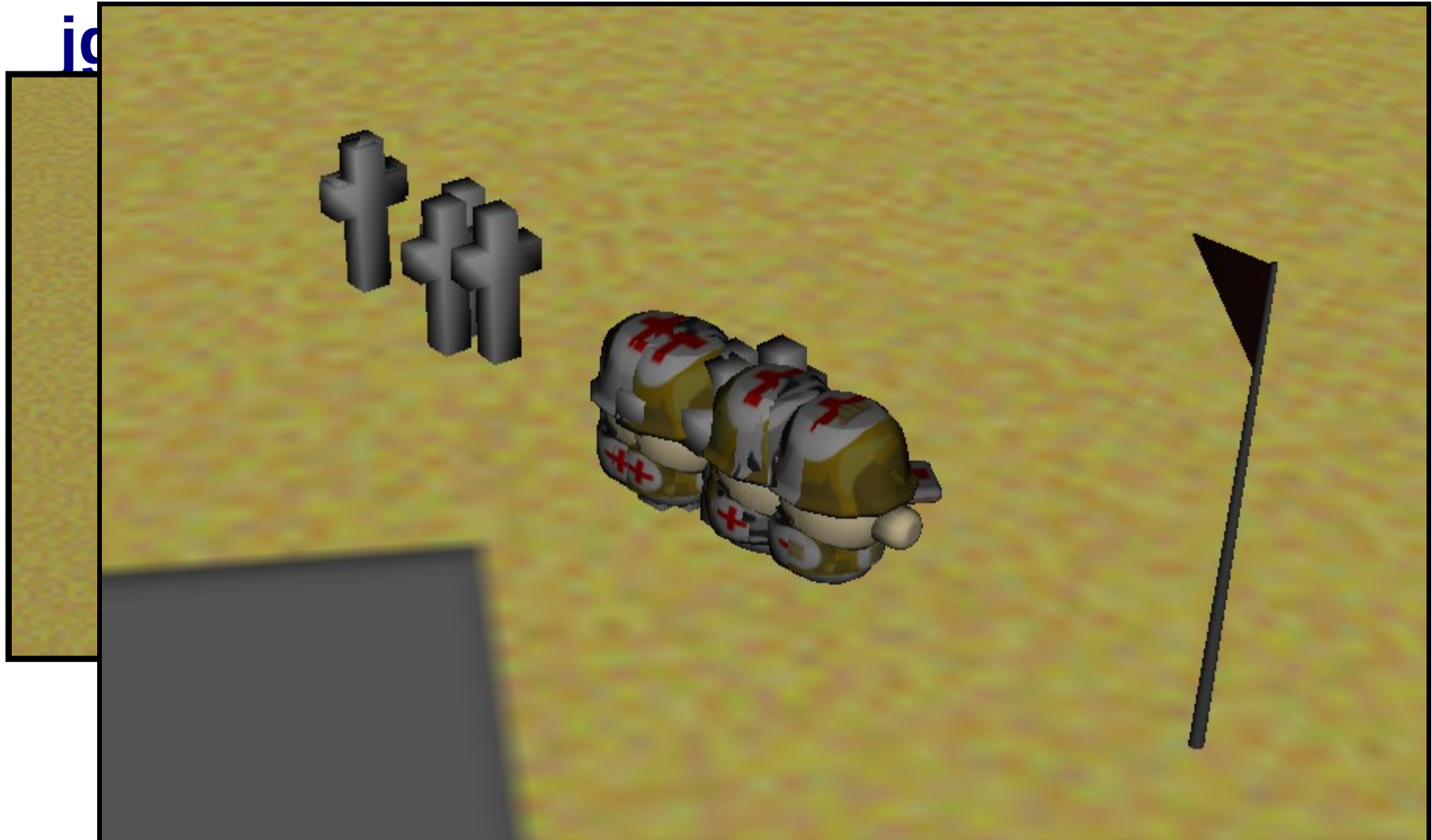


- Ejecutar **ejemplo2** de JGOMAS, `igomas.bat`:

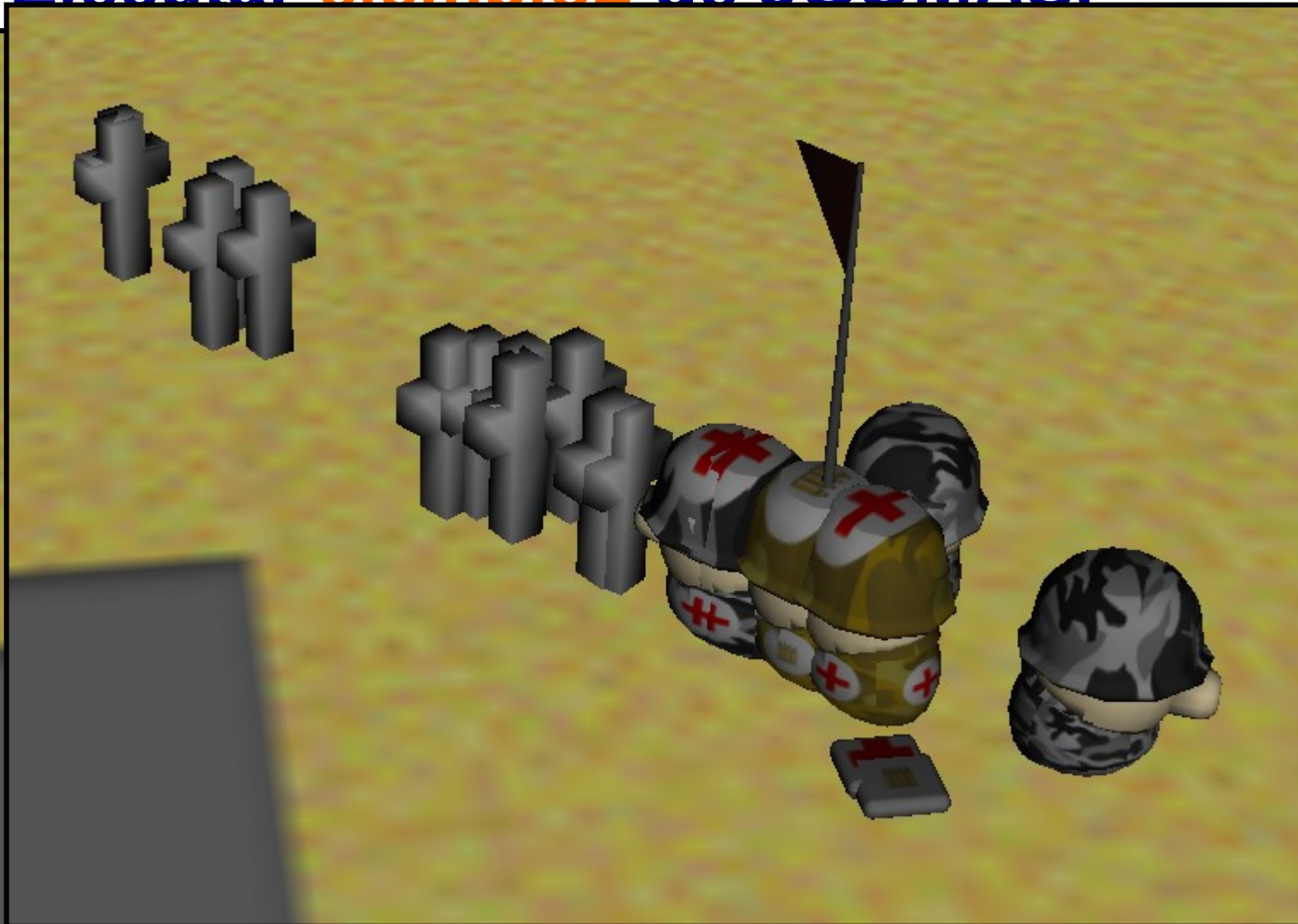


Application →

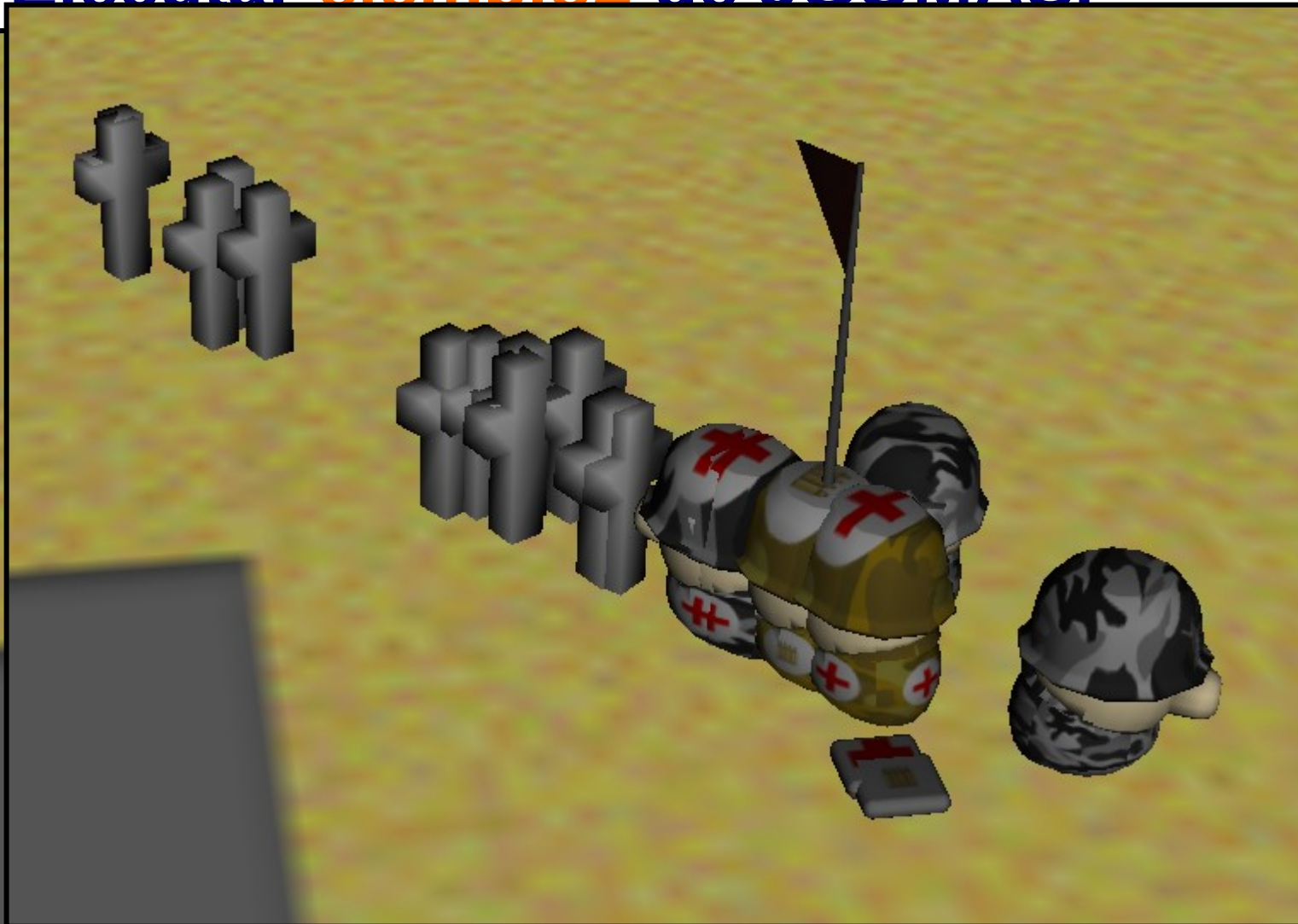
- Ejecutar **ejemplo2** de JGOMAS,



■ Ejecutar ejemplo2 de JGOMAS.



■ Ejecutar ejemplo2 de JGOMAS.



■ Ejecutar ejemplo2 de JGOMAS.



AGENT: AIO
POSITION: 142.1, 0, 185.037
HEADING: -0.584003, 0, -0.811752
HEALTH: 0 - Ammo: 31



Descripción de JGOMAS

- Taxonomía de Agentes
- Componente Estratégico
- Integración en Entorno
 - Entre Agentes -> cooperación
 - *Con el Terreno -> dificultad de movimiento*
- Comunicación Estándar



Descripción

- Objetivo del juego
- Estructura de la plataforma
- Ejecución



Objetivo (I)

- Disponemos de un número finito de agentes
- Cada agente pertenece a un bando:
 - **Aliados (Allied)**
 - **Eje (Axis)**
- Los agentes **Aliados** deben ir a la base del **Eje**, capturar la bandera, y llevarla a su base
- Los agentes del **Eje** deben defender la bandera y, en caso de ser capturada, devolverla a su base
- Hay un tiempo máximo para que los **aliados** lleven la bandera a su base

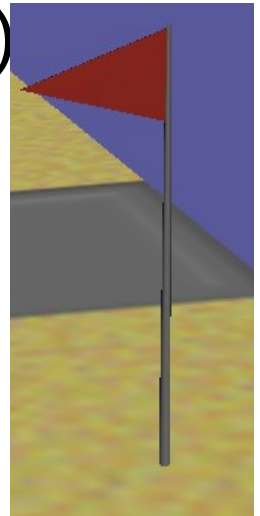


Objetivo (II)

- Comportamiento emergente como resultado de la actuación en grupo de los agentes
- Mejorar la inteligencia de los agentes en función de:
 - ☐ Especialización
 - ☐ Estrategia por objetivo
 - ☐ Generación de caminos
 - ☐ etc.

Estructura de la plataforma (I)

- A nivel de aplicación:
 - Sistema MultiAgente sobre JADE
 - Render Engine (Visualizador Gráfico)
- A nivel de Agentes:
 - Internos:
 - **Manager**: coordina todo el juego
 - **Pack**: paquetes de medicina, munición y objetivo



■ Externos:

- BasicTroop: agentes de usuario (roles)

- Hay definidos **tres tipos de roles:**



- **Soldier**: acude a dar apoyo
- **Medic**: acude a curar
- **FieldOps**: acude a dar munición



- Un agente asume un **único rol** durante toda la partida

- Cada rol tiene unas características y ofrece unos determinados servicios



- A nivel de Código:

- Implementación:

- Comportamientos
 - Métodos
 - Variables

- Acceso:

- Privados
 - Finales
 - Sobrecarga

Ejecución

```
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;  
jade.Boot -gui  
Manager:es.upv.dsic.gti_ia.jgomas.CManager (2 map_01 125 10)  
A1:student.MyMedic(ALLIED)  
E2:student.MyMedic(AXIS)
```

- Agente:
nombre:clase(parámetros)

