



JGOMAS

JADE Game Oriented MultiAgent System

Sesión 1

Sistemas Inteligentes
FI, 2006

Toni Barella
tbarella@dsic.upv.es

Índice



- Introducción
- Especificación
- Uso
- Trabajo a realizar

Índice



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

1

Introducción (I)



- 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

1

Introducción (II)



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

Índice



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

Especificación



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

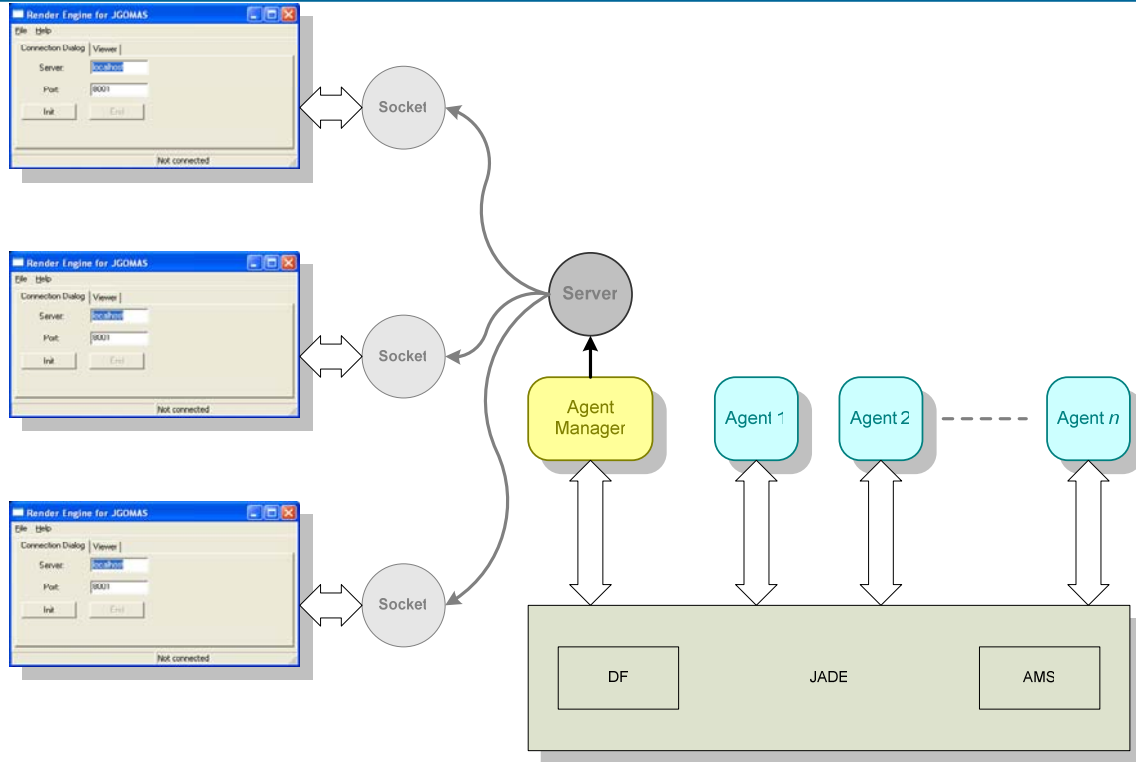
Especificación



- **▶ Arquitectura**
- Tareas
- Bucle de Ejecución
- Interfaz (API)



Arquitectura (I)



Arquitectura (II)

The screenshot shows the JGomas Viewer interface. The main window displays a 3D environment with a yellow floor and a maze-like structure. Several agents are visible, including a "MedicPack" and a "Backup_Axis". The interface includes a "JGomas Viewer" window, a "Command Console" window, and a "Remote Platforms" window. The "Command Console" shows the following output:

```
Manager[12]: Sending notification
[11@pandemonium:1099/JADE1]: Begin
Manager[13]: Sending notification
[E10@pandemonium:1099/JADE1]: Begin
Manager[14]: Sending notification
[A4@pandemonium:1099/JADE1]: Begin
Manager[15]: Sending notification
[12@pandemonium:1099/JADE1]: Begin
Manager[16]: Sending notification
Manager: Sending Objective notif
Registry - Service Registered: A
Activating Agent <A3@pandemonium
* Name: Ammo_Allied
* Type: Ammo_Allied
Registry - Service Registered: B
Activating Agent <E9@pandemonium
* Name: Backup_Axis
* T
Activat
* N
* T
Regist
Activat
* N
* T
Regist
Activat
* N
* T
Regist
Activat
* N
* T
Regist
Activat
* N
* T
Regist
Activat
* N
* T
Regist
Registry - Service Registered: Medic_Axis
Activating Agent <E2@pandemonium:1099/JADE - id: 9> ...
* Name: Medic_Axis
* Type: Medic_Axis
```

Especificación



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

1

Tareas (I)



- Definición

```
class CTask {  
    AID          m_AID;  
    int          m_iType;  
    int          m_iPriority;  
    Vector3D     m_Position;  
    ...  
}
```

- `m_AID` es el identificador del agente que *provoca* la creación de la tarea

1



Tareas (II)

- `m_iType` es el tipo de tarea
- Algunos de los tipos son:
 - TASK_GIVE_MEDICPAKS
 - TASK_GIVE_AMMOPACKS
 - TASK_GIVE_BACKUP
 - TASK_GET_OBJECTIVE
 - TASK_GOTO_POSITION
 - ...



Tareas (III)

- `m_iPriority` indica la prioridad **actual**
- Se lanza siempre la tarea de prioridad más alta
- Es posible redefinir la prioridad de cada tipo de tarea
- Las tareas las crea el sistema, no el usuario



Tareas (IV)

- Declaración:
 - `AddTask(int _tTypeOfTask, AID _owner, String _sContent)`
 - `AddTask(int _tTypeOfTask, AID _owner, String _sContent, int _iPriority)`

- Ejemplo de uso:

```
String sNewPosition = " ( " + x + " , " + y + " , " + z + " ) ";  
AddTask(TASK_GOTO_POSITION, getAID(), sNewPosition);
```



Especificación

- Arquitectura
- Tareas
- **▶ Bucle de Ejecución**
- Interfaz (API)

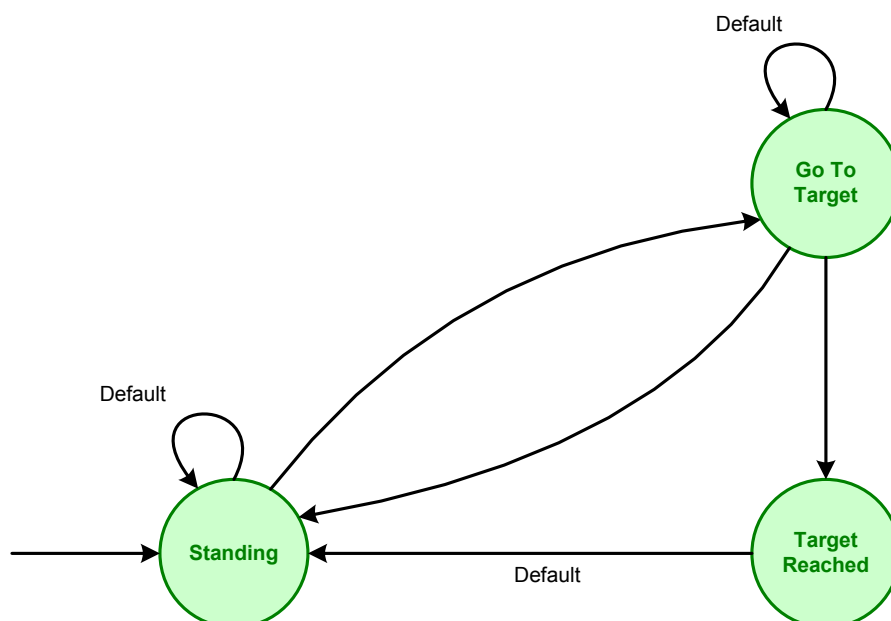


Bucle de Ejecución (I)

- Cada agente ejecuta una FSM:
 - STATE_STANDING
 - STATE_GOTO_TARGET
 - STATE_TARGET_REACHED
- FSM se utiliza para realizar tareas:
 - Inicio (Lanzamiento)
 - Desarrollo (Ejecución)
 - Final (Acción y Destrucción)
- Se lanza siempre la tarea de prioridad más alta

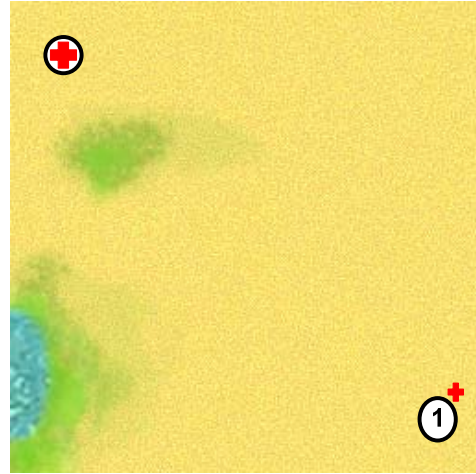
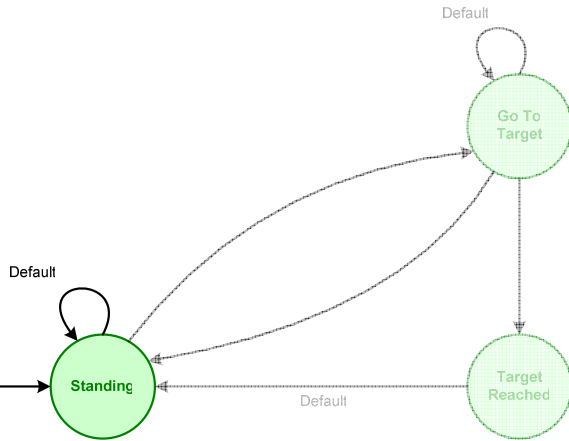


Bucle de Ejecución (II)

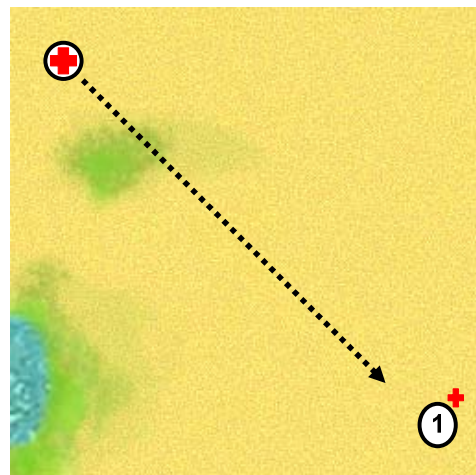
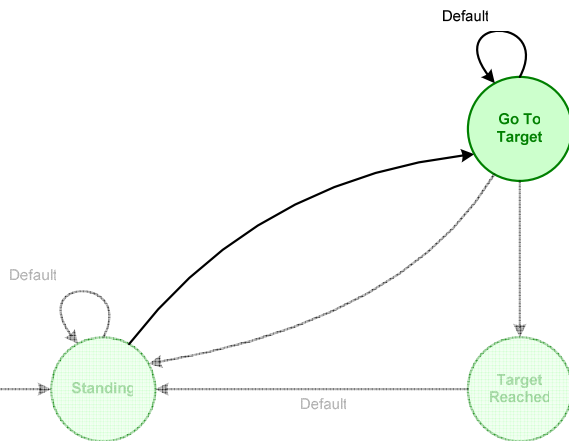




Ejemplo (I)

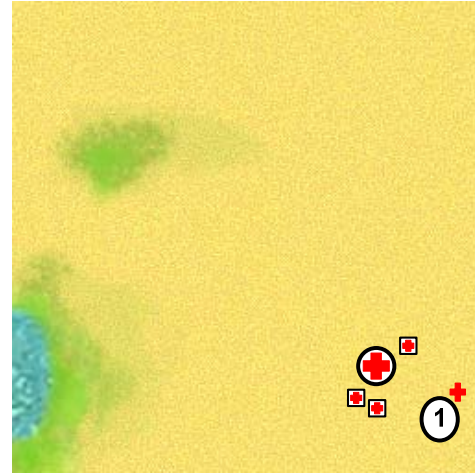
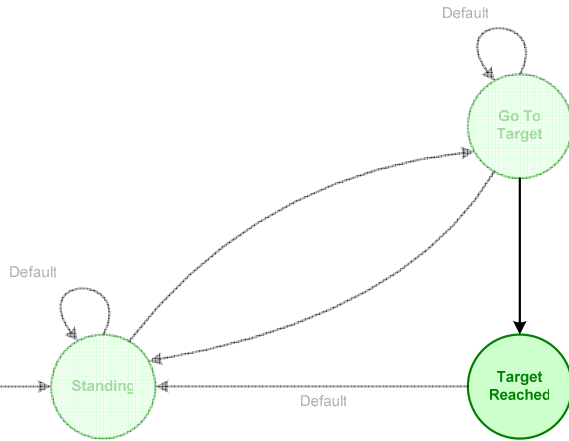


Ejemplo (II)

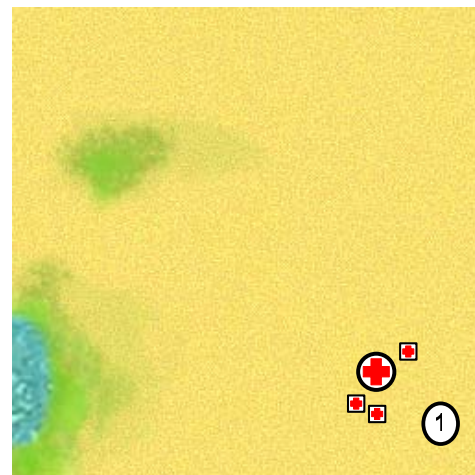
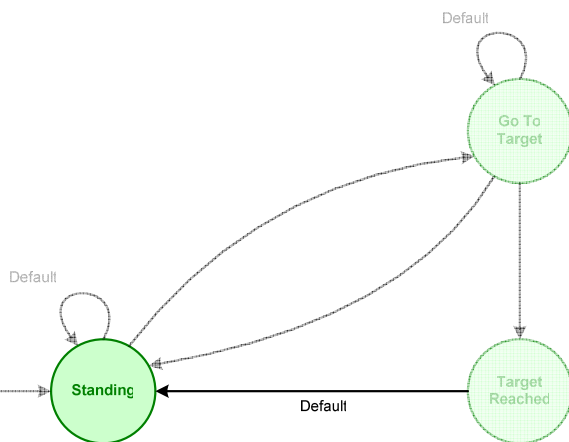




Ejemplo (III)



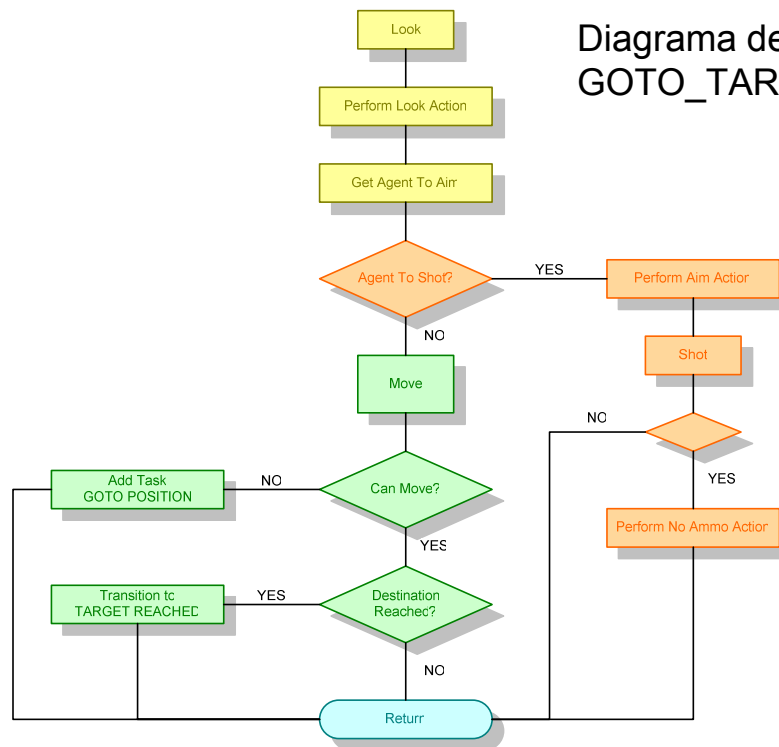
Ejemplo (IV)





Bucle de Ejecución (III)

Diagrama de flujo del estado GOTO_TARGET



Especificación

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



Interfaz (API)

- CBasicTroop
 - Métodos finales
 - Métodos sobrecargables
 - Atributos

1



API: Métodos finales

- `boolean CheckStaticPosition()`
- `boolean CheckStaticPosition(double _x, double _z)`
- `void AddTask (int _tTypeOfTask, AID _Owner, String _sContent)`
- `void AddTask (int _tTypeOfTask, AID _Owner, String _sContent, int _iPriority)`

1

API: Métodos sobrecargables



- `void PerformLookAction()`
- `boolean GetAgentToAim()`
- `void PerformNoAmmoAction()`
- `void PerformTargetReached(CTask _CurrentTask)`
- `boolean GeneratePath()`

API: Atributos



- `CTask` `m_CurrentTask`
- `Vector3D` `m_AStarPath []`
- `int` `m_iAStarPathIndex`
- `ArrayList` `m_FOVObjects`
- `CSight` `m_AimedAgent`
- `CMobile` `m_Movement`
- `CTerrainMap` `m_Map`



- Introducción
- Especificación
- **▶ Uso**
- Trabajo a realizar



- Línea de comandos:

```
java -classpath
lib\jade.jar;lib\jadeTools.jar;lib\Base64.jar;lib\http.jar;
lib\iiop.jar;lib\beangenerator.jar;. jade.Boot -gui
Manager:es.upv.dsic.gti_ia.jgommas.CManager(2 map_04 125 10)
A1:es.upv.dsic.gti_ia.jgommas.CMedic(ALLIED)
E1:es.upv.dsic.gti_ia.jgommas.CMedic(AXIS)
```

- Agente:

nombre:clase(parámetros)

Manager:es.upv...CManager(4 map_04 125 10)

↑
Nombre
Agente

↑
Clase

↑
Parámetros:
4: nº de agentes
map_04: nombre del mapa
125: refresco en ms.
10: duración (minutos)

A1:es.upv..CMedic(ALLIED)

↑
Nombre
Agente

↑
Clase

↑
Parámetros:
BANDO

Uso: Render Engine



- Línea de comandos:
 - 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**

Trabajo a Realizar (I)



- Objetivo:
 - Implementar un agente aliado que vaya mostrando por pantalla su posición y su distancia hasta el destino.

1

Trabajo a Realizar (II)



- Objetivo:
 - Implementar un agente defensor que vaya cambiando su posición aleatoriamente.
 - Implementar un agente defensor que localice un compañero y lo vaya siguiendo.

1

Trabajo a Realizar (III)



- Notación Húngara: Identificadores
 - CString `m_sFirstName`;
Prefijo Calificador
 - Prefijo:
 - » Siempre en minúsculas
 - » Indica el tipo del dato
 - Calificador:
 - » Resto del nombre, indica el uso de la var.
 - » Empieza por mayúscula, para delimitarlo del prefijo
 - » Si contienen más de una palabra, la inicial de cada una va en mayúscula. Ej.: `int nEstoEsUnIdMuyLargo`;

Trabajo a Realizar (IV)



- Notación Húngara: Prefijos comunes
 - **s** ó **str**: *String*
 - **c**: Carácter
 - **i** ó **n**: Entero
 - **f**: *Float*
 - **d**: Doble
 - **b**: *Boolean*
 - **l**: *Long*
 - **C**: Clase
 - **m_**: Identificador de un miembro de una clase
 - **v**: *Void*, sin tipo



JGOMAS

JADE Game Oriented MultiAgent System

Sesión 1

Sistemas Inteligentes
FI, 2006

Toni Barella
tbarella@dsic.upv.es