JGOMAS

JGOMAS

JADE Game Oriented MultiAgent System

MAS: coordination & emergent behoviour

Sistemas Inteligentes
FI, 2006

Toni Barella
tbarella@dsic.upv.es

1

JGOMAS

Índice

- Comportamiento Emergente
- Coordinación
- API
- Trabajo a realizar

1

JGOMAS

Índice

- ► **Comportamiento Emergente**
- Coordinación
- API
- Trabajo a realizar

1

JGOMAS

Comportamiento Emergente

- Bucle de Ejecución
- Umbrales

1

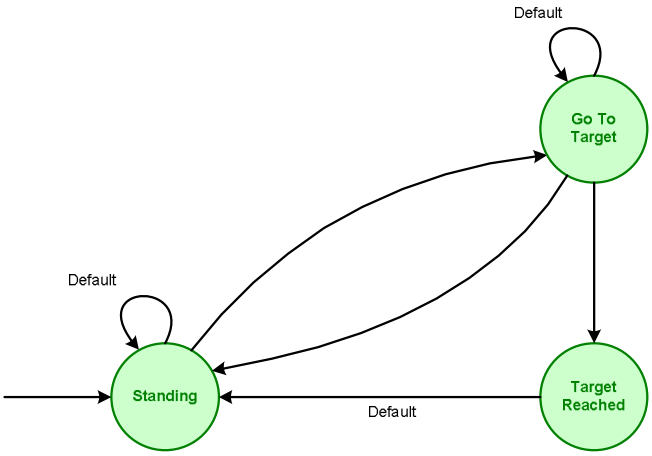
Comportamiento Emergente

- ► **Bucle de Ejecución**
- Umbrales

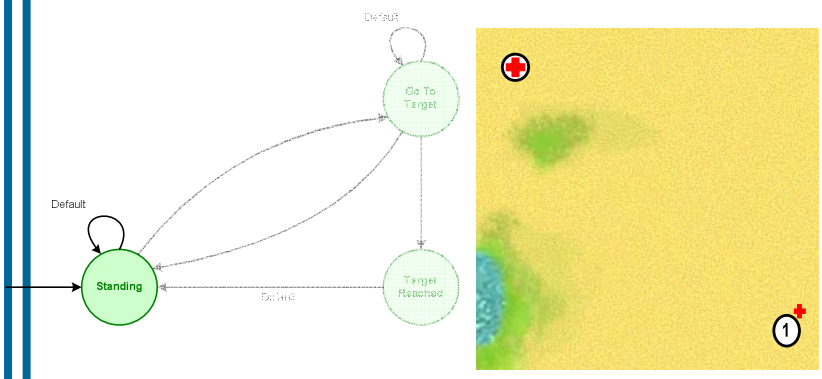
Bucle de Ejecución (I)

- 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

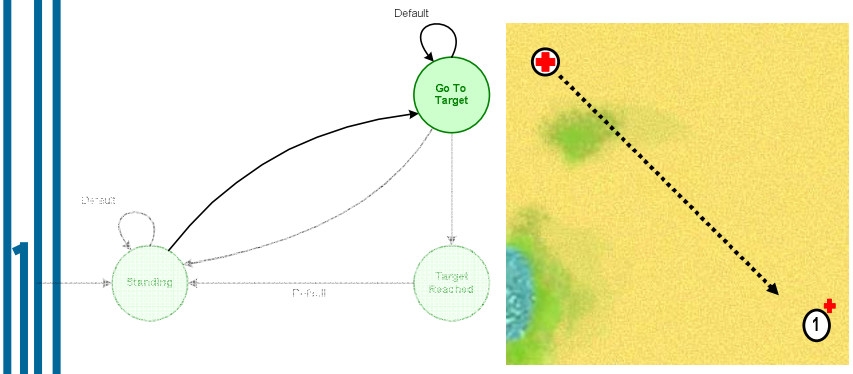
Bucle de Ejecución (II)



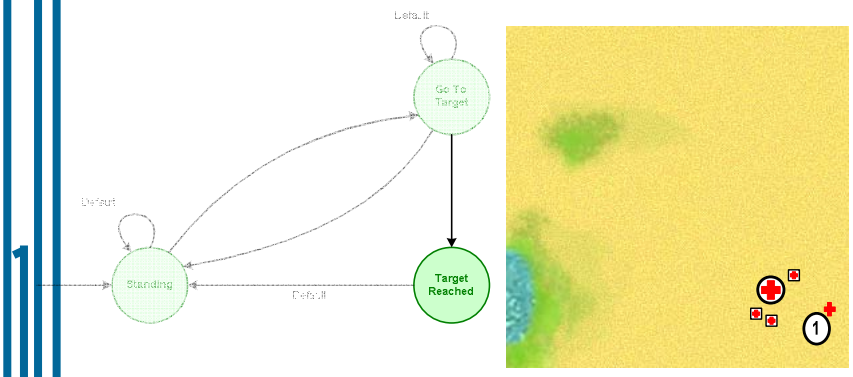
Ejecución: Caso 1 (I)



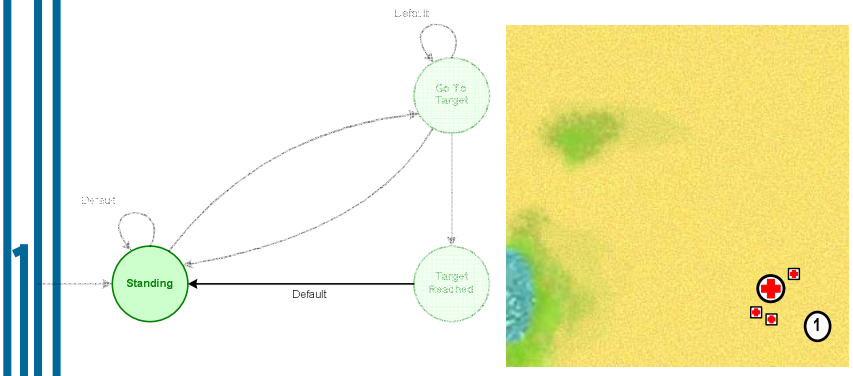
Ejecución: Caso 1 (II)



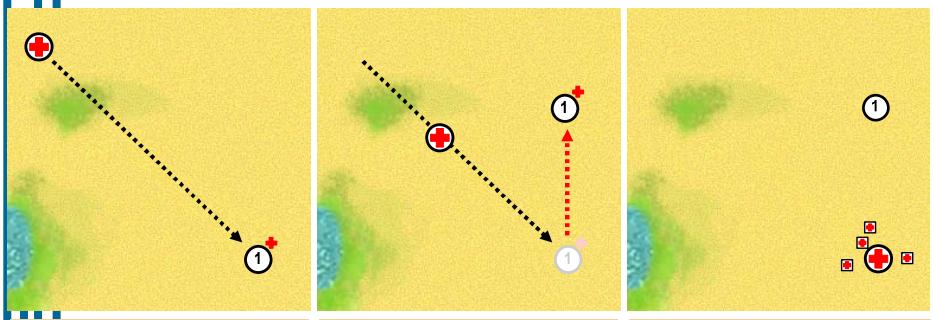
Ejecución: Caso 1 (III)



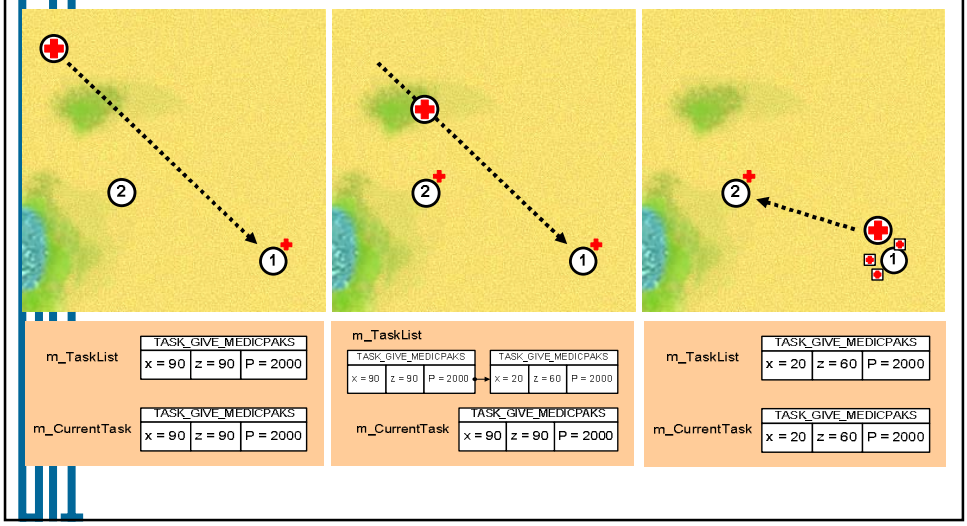
Ejecución: Caso 1 (IV)



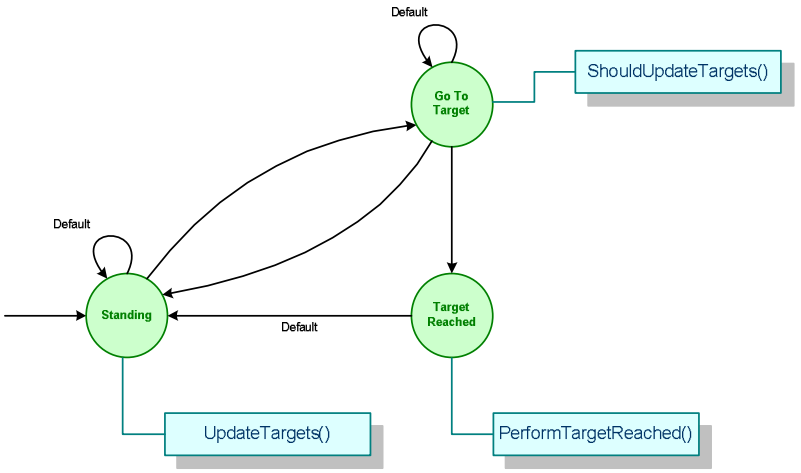
Ejecución: Caso 2



Ejecución: Caso 3



Bucle de Ejecución (III)



Comportamiento Emergente

- Bucle de Ejecución
- ► **Umbrales**

Umbrales (I)

- Cada agente dispone de una variable umbral: **`m_Threshold`**
- Permite definir límites de:
  - Salud
  - Munición
- Una vez excedidos, se lanza:
  - **`CallForMedic()`**, o
  - **`CallForAmmo()`**

Umbrales (II)

- Además, también se ejecuta el método:
  - **`PerformThresholdAction()`**
- Métodos de acceso a los umbrales:
  - Cota inferior de salud:
    - **`void SetHealth ( int _iHealth )`**
    - **`int GetHealth ( )`**
  - Cota inferior de munición:
    - **`void SetAmmo ( int _iAmmo )`**
    - **`int GetAmmo ( )`**

Índice

- Comportamiento Emergente
- ► **Coordinación**
- API
- Trabajo a realizar

JGOMAS

Coordinación (I)

- JGOMAS dispone de mecanismos que permiten la coordinación entre agentes:
  - Sin comunicación (implícita):
    - Sensorización del entorno
  - Con comunicación (explicita):
    - Mediante paso de mensajes

JGOMAS

Coordinación (II)

- El usuario puede evaluar lo que sucede en el juego:
  - Evaluación cualitativa: supervisión visual
  - Evaluación cuantitativa: fichero de estadísticas

```
Winner Team: ALLIED
Duration: [3m:51s]

Statistics for ALLIED TEAM
     - GENERAL:
          * Alive:              6
          * Avrg. Health:       94.3

     - SHOTS:
          * EnemyHit:           121
          * TeamHit:            0
          * FailedHit:          2
          * TOTAL:              123
```

JGOMAS

Índice

- Comportamiento Emergente
- Coordinación
- ► **API**
- Trabajo a realizar

JGOMAS

API

- CTroop
  - Métodos finales
  - Métodos sobrecargables
  - Atributos
- CMedic
- CFieldOps
- CSoldier

JGOMAS API

CTroop: métodos finales

- `final int GetHealth ()`
- `final int GetAmmo ()`
- `final int GetStamina ()`
- `final void UseStamina ()`
- `final int GetPower ()`
- `final void UsePower ()`
- `final void AddServiceType (String _sServiceType)`
- `final boolean CheckStaticPosition ()`
- `final boolean CheckStaticPosition (double _x, double _z)`
- `final void AddTask (int _tTypeOfTask, AID _Owner, String _sContent)`
- `final void AddTask (int _tTypeOfTask, AID _Owner, String _sContent, int _iPriority)`
- `final void Look ()`
- `final boolean Shot (int _iShotNum)`
- `final void PerformAimAction ()`
- `final boolean HaveAgentToShot ()`

JGOMAS API

CTroop: mét. sobrecargables

- `void CallForMedic ()`
- `void CallForAmmo ()`
- `void CallForBackup ()`
- `void UpdateTargets ()`
- `boolean ShouldUpdateTargets ()`
- `void ObjectivePackTaken ()`
- `void SetUpPriorities ()`
- `void PerformNoAmmoAction ()`
- `void PerformTargetReached (CTask _CurrentTask)`
- `void GenerateEscapePosition ()`
- `boolean GeneratePath ()`
- `void CreateControlPoints ()`
- `void PerformThresholdAction ()`
- `void PerformInjuryAction ()`
- `boolean GetAgentToAim ()`
- `void PerformLookAction ()`

CTroop: atributos (1)

- `int` `m_eTeam`
- `int` `m_eClass`

- `AID` `m_Manager`
- `Hashtable` `m_TaskList`
- `CTask` `m_CurrentTask`
- `int` `m_TaskPriority []`

- `ArrayList` `m_FOVObjects`
- `CSight` `m_AimedAgent`
- `boolean` `m_bObjectiveCarried`

- `Vector3D` `m_ControlPoints []`
- `int` `m_iControlPointsIndex`
- `Vector3D` `m_AStarPath []`
- `int` `m_iAStarPathIndex`

CTroop: atributos (2)

- `CThreshold` `m_Threshold`
- `CMobile` `m_Movement`
- `CTerrainMap` `m_Map`

- `String` `m_sMedicService`
- `String` `m_sAmmoService`
- `String` `m_sBackupService`

- `int` `m_iSoldiersCount`
- `int` `m_iMedicsCount`
- `int` `m_iFieldOpsCount`
- `int` `m_iTeamCount`

JGOMAS API

CMedic

- Métodos finales
  - `final int CreateMedicPack ()`
- Métodos sobrecargables
  - `void SetUpPriorities ()`
  - `boolean checkMedicAction (String _sContent)`

JGOMAS API

CFieldOps

- Métodos finales
  - `final int CreateAmmoPack ()`
- Métodos sobrecargables
  - `void SetUpPriorities ()`
  - `boolean checkAmmoAction (String _sContent)`

JGOMAS API

CSoldier

- Métodos finales
  - `no hay`
- Métodos sobrecargables
  - `void SetUpPriorities ()`
  - `boolean checkBackupAction (String _sContent)`

JGOMAS

Índice

- Comportamiento Emergente
- Coordinación
- API
- ► **Trabajo a realizar**

JGOMAS

Trabajo a Realizar (I)

- Objetivo:
  - Implementar una coordinación del equipo atacante sin comunicación:
    - cada agente ha de seguir al agente de su propio equipo que se encuentre más alejado de él.

1

JGOMAS

Trabajo a Realizar (II)

- Objetivo:
  - Implementar una coordinación del equipo defensor con comunicación:
    - cada agente ha de patrullar pasando por los puntos de patrulla del resto de agentes.

1

JGOMAS

Trabajo a Realizar (III)

- Objetivo:
  - Diseñar e implementar un equipo de 10 agentes con la distribución de tipos que deseéis (médicos, soldados y fieldops) para jugar a **capturar la bandera** en un mapa cualquiera como **atacante y como defensor**, de manera que **ganen en cualquier situación a los equipos suministrados**.

JGOMAS

Trabajo a Realizar (IV)

- Reglas Básicas:
  - No se puede consultar/solicitar información del sistema sobre el bando contrario que no sea suministrada por el entorno.
  - No puede existir comunicación entre agentes que no sea usando FIPA ACL y de acuerdo a la especificación proporcionada.

JGOMAS

Trabajo a Realizar (V)

- Entrega:
  - Ficheros *.java desarrollados. El código, comentado y documentado debe seguir unas mínimas normas de estilo:
    - tabulado, comentado, y usando notación húngara.
  - Fichero <login_alumno>.jar
  - Pequeña memoria, indicando las principales ideas de mejora aplicadas al equipo, así como unas breves conclusiones sobre los resultados obtenidos.

JGOMAS

JGOMAS

JADE Game Oriented MultiAgent System

MAS: coordination & emergent behoviour

Sistemas Inteligentes
FI, 2006

Toni Barella
tbarella@dsic.upv.es