Lab: Agentes (II)

Sistemas Inteligentes Distribuidos

Sergio Alvarez
Javier Vázquez

Objetivos de la sesión

- Entender cómo funciona el entorno pyGOMAS
- Ver las percepciones y acciones disponibles en el entorno
- Crear un agente sencillo que capture la bandera
- Diseñar el comportamiento del agente con objetivos

pyGOMAS

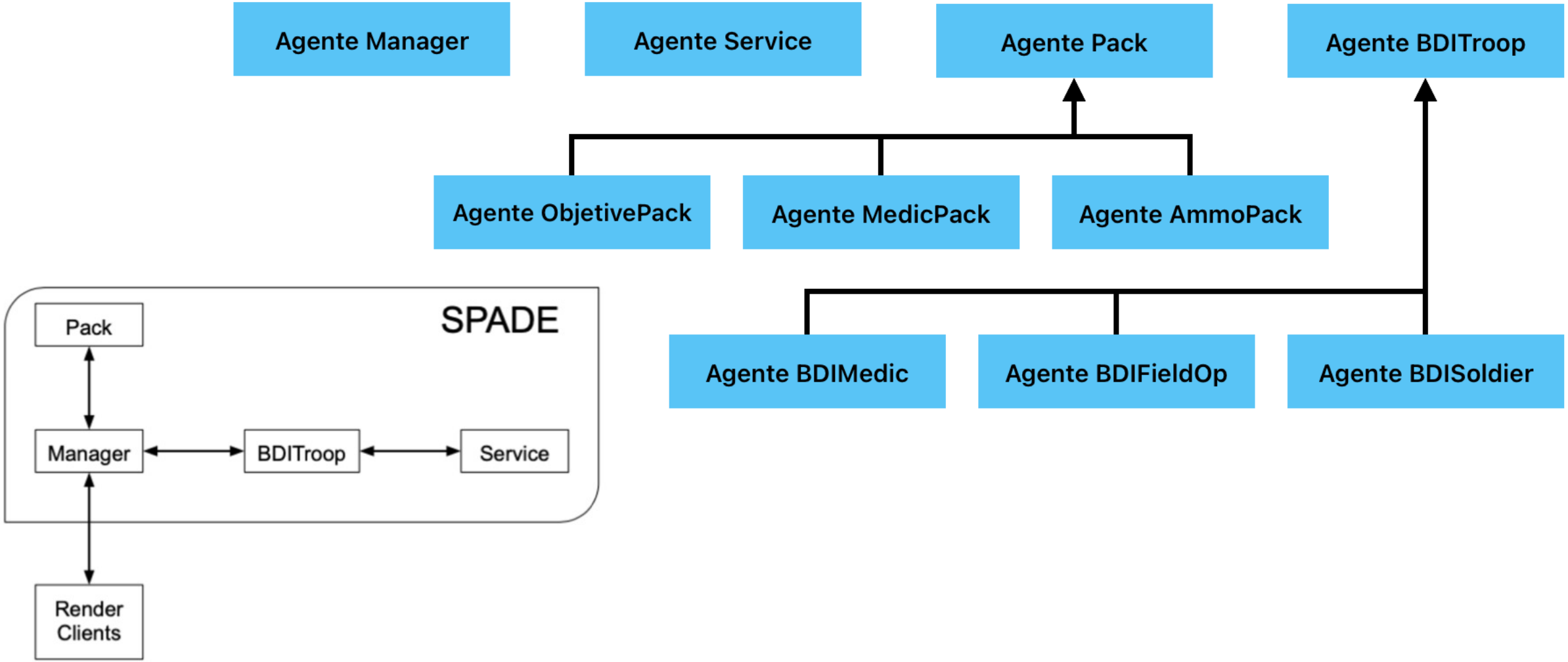
- Juego basado en “Captura La Bandera” usando agentes híbridos SPADE
- La partida se puede observar a través de visores gráficos que se pueden conectar a un servidor HTTP que se lanza en la aplicación
- 2 equipos (Aliados y Eje) que se enfrentan en un terreno limitado durante un tiempo limitado
- Cada equipo tiene su base, donde se sitúan inicialmente
- Objetivo del juego:
  - Aliados: capturar la bandera y llevarla a su base.
  - Eje: impedir la captura (eliminando todos los aliados o si se agota el tiempo).

pyGOMAS

El agente tiene dos vertores, uno de posicion y el otro donde mira.

- Llevan un arma de fuego
  - Probabilidad de fallar un disparo por azar: 0.1
  - Hay fuego amigo
- Inicio de la partida con el máximo de munición (100)
- Posibles roles (extensible):
  - **Soldado**: sus armas hacen el doble de daño, reciben llamadas de refuerzo
  - **Médico**: crean paquetes de medicina que recuperan algo de salud a quien lo coge, reciben llamadas de asistencia médica
  - **Operador de Campo**: crean paquetes de municiones que recuperan algo de munición a quien lo coge, reciben llamadas de recargo de municiones
- Paquetes (**salud** o **munición**) pueden ser cogidos por el enemigo

pyGOMAS


Agente Manager
Agente Service
Agente Pack
Agente BDITroop
Agente ObjetivePack
Agente MedicPack
Agente AmmoPack
Agente BDIMedic
Agente BDIFieldOp
Agente BDISoldier
SPADE
Pack
Manager
BDITroop
Service
Render Clients

Agente Manager

- **Monitoriza** y **gestiona** la partida
- Funciones principales: **coordinación** y **sincronización** de los otros agentes y la aportación de **información** de lo que está en el **campo de visión** de estos
- Tareas:
  - **Iniciar la partida**: mensaje de inicio a los agentes tropa con el mapa del terreno y ubicación inicial de la bandera
  - Crear el **Agente de Servicios**
  - **Recibir información de cada agente tropa**: posición, velocidad, orientación, salud y munición
  - **Servir a los clientes de visualización**: les envía información de la bandera, de los agentes tropa y los paquetes

Agente Service

- **Función**: conocer e informar los servicios que dan las tropas
- Al crear un Agente Tropa, se **registra el servicio** que ofrece aquí
- Al morir un Agente Tropa, se informa a este agente para que sepa que **ya no puede ofrecer el servicio** que antes brindaba
- Sabiendo quiénes son los Agentes Tropa activos, puede responder a **peticiones de solicitud de servicios**
- Se pueden registrar y posteriormente solicitar servicios nuevos

Agente Pack

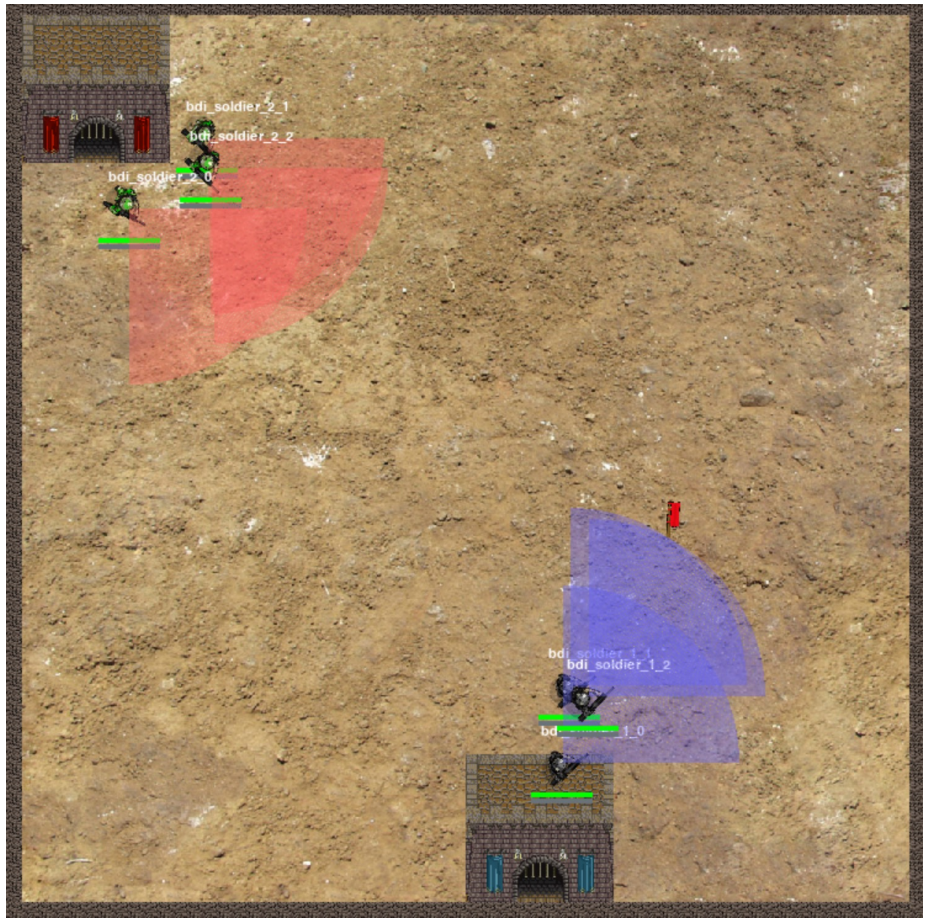
- Los **Agentes Pack** son los paquetes que se crean durante la partida, y pueden ser de tres tipos:
  - **ObjectivePack** (id=1003): se crea al inicio del juego por el Agente Manager y representa la bandera
  - **MedicPack** (id=1001): creado por los **médicos**
    - Función: **incrementar la salud** de quien lo reciba en 20 (0..100)
    - Se autodestruye pasados 25 segs. si no es cogido por nadie
  - **AmmoPack** (id=1002): creado por los **operadores de campo**
    - Función: **incrementar munición** de quien lo reciba en 20 (0..100)
    - Se autodestruye pasados 25 segs. si no es cogido por nadie

Agentes BDI

- Son agentes híbridos: goto zona de patrulla
  - **Capa Reactiva**: acciones de traslación, generar puntos de control y disparar entre otras
  - **Capa Deliberativa**: código ASL con información de las capas reactivas
  - Peticiones al **Agente Service**: médicos, operadores de campo y soldados de su equipo disponibles
- Tipos de agentes tropa:
  - **BDIMedic**: servicio médico, crear paquetes de medicina
  - **BDIFieldOp**: servicio de recargar municiones, crear paquetes de municiones
  - **BDISoldier**: sus disparos hacen el doble de daño, servicio de refuerzo (ir a la posición de un compañero, para reforzar el ataque)

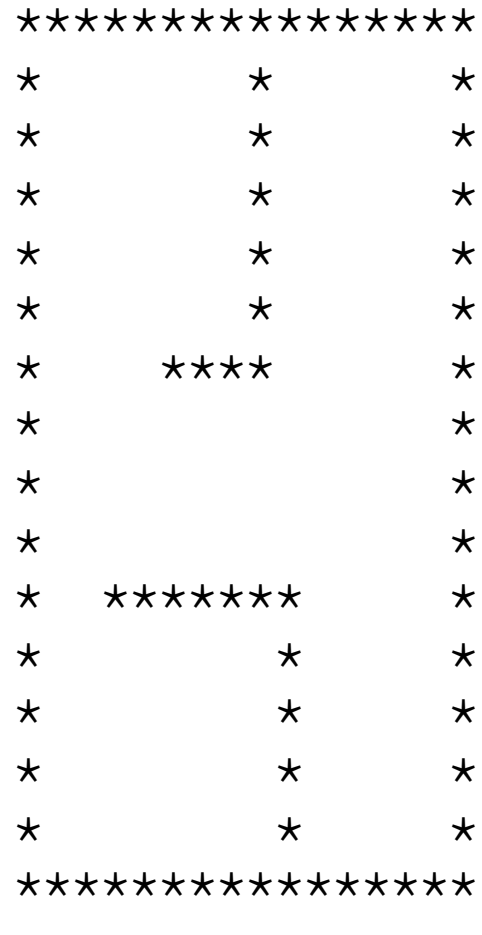
Visualización

- Campo de visión de cada soldado
- Cursores: movimiento de la cámara
- Z, X: Zoom (in/out)
- F: Mostrar/ocultar conos de visión
- Hay visores en línea de comandos y en Unity (mirad documentación)



Mapas y pathfinding

- Dos ficheros .txt:
  - Dimensiones, ubicación de la bandera y bases
  - Mapa de obstáculos (asteriscos)
  - Podéis crear nuevos mapas o modificar los existentes en el directorio `pygomas/pygomas/maps`
- Movimiento:
  - Componente reactivo concurrente
  - Cola de destinos: puntos del mapa a visitar para llegar a un destino
    - Algoritmo A* (Jump Point Search)
  - Las acciones `.goto` reinician la cola



mine_medium/mine_medium_cost.txt

Creencias de cada agente

- **Las creencias se reciben de manera asíncrona, por lo que NO supongáis que están disponibles desde el inicio del agente o inmediatamente después del suceso original**
- `enemies_in_fov(ID, TYPE, ANGLE, DIST, HEALTH, [X,Y,Z])`: el agente ha visto un enemigo con identificador ID, del tipo TYPE, a un ángulo ANGLE, a una distancia DIST, con una salud HEALTH, y en la posición [X, Y, Z] .
- `friends_in_fov(ID, TYPE, ANGLE, DIST, HEALTH, [X,Y,Z])`: el agente ha visto un compañero de equipo
- `packs_in_fov(ID, TYPE, ANGLE, DIST, HEALTH, [X,Y,Z])`: el agente ha visto un pack
  - Tipos de Pack: 1000 (None), 1001 (MEDICPACK), 1002 (AMMOPACK), 1003 (FLAG)

Creencias de cada agente

- `name(X)`: X es el nombre del agente
- `class(X)`: X es la clase a la que pertenece el agente:
  - NONE = 0, SOLDIER = 1, MEDIC = 2, ENGINEER = 3, FIELDOPS = 4
- `team(X)`: el agente pertenece al equipo X: ALLIED = 100, AXIS = 200
- `flag([X,Y,Z])`: [X, Y, Z] es la posición de la bandera
- `heading([X, Y, Z])`: el Ag. Tropa está orientado hacia [X, Y, Z]
- `health(X)`: X es la salud actual del agente
- `ammo(X)`: X es la munición actual del agente
- `base([X,Y,Z])`: la base del equipo del agente está en [X, Y, Z]
- `position([X,Y,Z])`: [X, Y, Z] es la posición actual del agente

Creencias de cada agente

- `threshold_shots(X)`: límite máximo de disparos simultáneos (por defecto, 1)
- `velocity([X,Y,Z])`: [X, Y, Z] es la velocidad actual del agente
- `destination([X,Y,Z])`: objetivo del agente en coordenadas [X,Y,Z]
- `pack_taken(TYPE, N)`: si el agente ha cogido un pack de tipo TYPE (medic o fieldops) y la cantidad a aumentar de vida/munición
- `flag_taken`: si el agente ha cogido la bandera
- `target_reached([X, Y, Z])`: Se añade cuando el agente llega a su destino ([X, Y, Z])
- `myMedics([id ...])`: lista de médicos del equipo activos
- `myFieldops([id ...])`: lista de field ops del equipo activos
- `myBackups([id ...])`: lista de soldados del equipo activos

Acciones disponibles: movimiento

- `.goto([X,Y,Z])`: establecer [X,Y,Z] como destino del agente, y pone al agente en marcha hacia dicho lugar, usando un algoritmo JPS para desplazarse por el terreno.
- `.stop`: detener el movimiento del agente
- `.turn(R)`: modificar la orientación del agente una cantidad (positiva o negativa) R de radianes, útil para alterar el campo de visión
- `.look_at([X,Y,Z])`: orientar el agente hacia [X,Y,Z]
- `.create_control_points([X,Y,Z],D,N,C)`: crear un grupo de N puntos aleatorios de control a una distancia D dada de una ubicación `[X,Y,Z]` en el mapa
  - la lista de puntos se almacena en la variable C, y sirve para otras tareas, e.g. patrullar alrededor de la bandera, recordar dónde había un enemigo, etc.

Acciones disponibles: directorio

- `.register_service("servicio_a")`: enviar mensaje al agente Service para registrar un servicio especificado
- `.get_medics`: enviar mensaje al agente Service, solicitando los médicos de su equipo.
- `.get_fieldops`: enviar un mensaje al agente Service, solicitando los operadores de campo de su equipo
- `.get_backups`: enviar un mensaje al agente Service, solicitando los soldados de su equipo
- `.get_service("servicio_a")`: enviar un mensaje al agente Service, solicitando otro servicio (distinto de los tres anteriores) a los agentes de su equipo que lo ofrezcan.

Acciones disponibles

- `.shoot(N,[X,Y,Z])`: disparar N disparos a [X,Y,Z]
- `.cure`: crear packs de medicina, sólo disponible para los médicos
- `.reload`: crear packs de munición, sólo disponible para los operadores de campo
- `.wait(M)`: bloquear plan durante M milisegundos

Acciones disponibles: listas

- `.length(L, N)`: escribe en la variable N la longitud de la lista L
- `.nth(P, L, X)`: escribe en la variable X el valor de la posición P de la lista L, siendo P un valor entre 0 y (N – 1)
- `.sort(L, L2)`: ordena los valores de L en orden ascendente y escribe el resultado en la variable L2
- **¡Vigilad!** Dentro de un mismo plan no reutilicéis las variables u os dará error, e.g. `.nth(0, L, A); .nth(1, L, A)`
- Si queréis ver más acciones disponibles desde agentspeak para listas o strings, visitad el código de la *stdlib* en:
  - https://github.com/niklasf/python-agentspeak/blob/master/agentspeak/stdlib.py

Trazas de la partida

- Después de cada ejecución se genera un fichero `pygomas_stats.txt` que contiene estadísticas de la partida
- Además, al empezar una partida, es posible conectar un proceso de monitorización:
  - `pygomas dump --log partida.log`
- Es posible visualizar la partida en diferido a partir de este fichero de log:
  - `pygomas replay --log partida.log`

Ejercicio: agente soldado sencillo

- Cread un fichero `.asl` para un agente ALLIED con:
  - La declaración de un objetivo `capture_flag`
  - Un plan que gestione el evento de objetivo `capture_flag`, añadiendo dos objetivos `take_flag` y `bring_flag_home`
  - Dos planes para gestionar el evento de objetivo `take_flag`, uno para el caso en el que tengamos la creencia `flag(F)` y otro en el que no *(¿por qué?)*
  - Dos planes para gestionar el evento de objetivo `bring_flag_home`, uno para el caso en el que tengamos la creencia `flag_taken` y otro en el que no
  - *Creencias/acciones que os pueden servir:* `flag(F), base(B), .wait(M), .goto(P)`
  - **¿Cómo lo haríais diferente?**
- Para probarlo con un mapa más pequeño y con sólo un agente, podéis probar:
  - `pygomas manager -j cmanager-<nombre_equipo>@sidfib.mooo.com -sj cservice-<nombre_equipo>@sidfib.mooo.com -np 1 -m mine`

Ejercicio: enemigo

- Cread otro agente, en este caso AXIS, que tenga como objetivo principal evitar que la bandera sea capturada, con el siguiente comportamiento:
  - Moverse hacia la bandera
  - Esperar a que aparezca un ALLIED
  - Eliminar a cualquier ALLIED que aparezca

Práctica 1

- Entrega: 17/03/2024
- Documentación + código de tres agentes: soldier, medic, field ops
- Objetivo de cada agente: ganar la partida, independientemente del rol asignado
  - En cada partida, cada agente puede ser utilizado como ALLIED o como AXIS, incluso en equipos diferentes en una misma partida
  - Sin hacer suposiciones sobre el mapa (tenéis 14 mapas en el repo de pygomas para probar)
- Se valorará:
  - Documentación: diseño de la estrategia (racionalidad, guía por objetivos)
  - Traducción de la estrategia a AgentSpeak (creencias, objetivos, planes)
  - Puntos extra (+20%): top 20% de grupos en % de victorias, mezclando agentes de los diferentes grupos