



Agentes Inteligentes (AIN)

Práctica 1: Programando mi luchador

- ❖ Recordatorio:

Desarrollar un agente que trate de sobrevivir con sólo información del entorno

- ❖ Programar un agente que trate de sobrevivir en un escenario hostil
- ❖ La información que dispone es únicamente a través de sus creencias
- ❖ Ganan los que logran sobrevivir después de un tiempo máximo

pyGOMAS



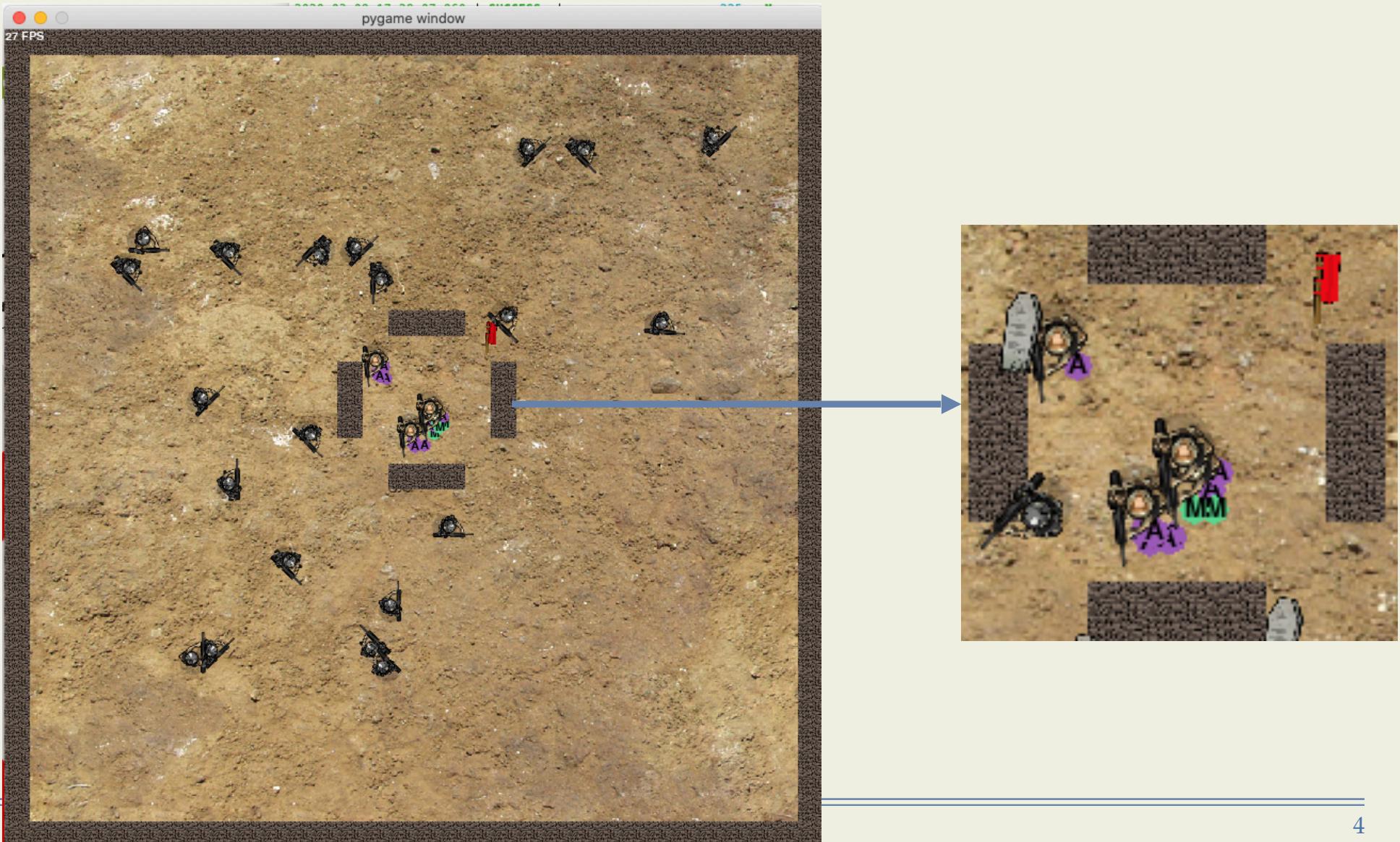
Escenario ARENA



pyGOMAS



Escenario ARENA



Creencias de los Agentes Tropa (I)

Recordatorio

- **class(X)**: X es la clase a la que pertenece el agente:
 - NONE = 0, SOLDIER = 1, MEDIC = 2, ENGINEER = 3, FIELOPS = 4
- **enemies_in_fov(ID, TYPE, ANGLE, DIST, HEALTH, [X,Y,Z])**: El Ag. Tropa 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 Ag. Tropa ha visto un compañero de equipo...
- **packs_in_fov(ID, TYPE, ANGLE, DIST, HEALTH, [X,Y,Z])**: El Ag. Tropa ha visto un pack ...
 - Tipos de Pack: 1000 (None), 1001 (MEDICPACK), 1002 (AMMOPACK), 1003 (FLAG).

Creencias de los Agentes Tropa (II)

Recordatorio

- **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].
- **name(X):** X es el nombre del agente.
- **myMedics([id ...]):** Lista de médicos del equipo activos.
- **myFieldops([id ...]):** Lista de FieldOps del equipo activos.
- **myBackups([id ...]):** Lista de Soldados del equipo activos.
- **position([X,Y,Z]):** [X, Y, Z} es la posición actual del agente.
- **team(X):** el Ag. Tropa pertenece al equipo X.

Creencias de los Agentes Tropa (III)

Recordatorio

- **threshold_health(X)**: X es la salud mínima antes de lanzar una acción especial como respuesta.
- **threshold_ammo(X)**: X es la munición mínima antes de lanzar una acción especial como respuesta.
- **threshold_shots(X)**: Límite máximo de disparos simultáneos.
- **velocity([X,Y,Z])**: [X, Y, Z] es la velocidad actual del Ag. Tropa.
- **destination([X,Y,Z])**: Objetivo del Ag. Tropa: [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]).

Acciones de los Agentes Tropa

Recordatorio

- Movimiento:
 - `.goto([X,Y,Z])`: Establecer [X,Y,Z] como destino del ag. Pone al ag. tropa en marcha hacia dicho lugar, usando un algoritmo JPS para desplazarse por el terreno.
 - `.stop`: Detener el mov. del ag. tropa.
 - `.turn(R)`: Modificar la orientación del ag. tropa una cantidad (pos. o neg.) R de radianes. Útil para alterar el campo de visión.
 - `.look_at([X,Y,Z])`: Orientar el ag. tropa 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 C. Ej.: patrullar alrededor de la bandera.

Acciones de los Agentes Tropa

Recordatorio

- ❖ Envío de mensajes al Service Agent:
 - ❖ `.register_service("servicio_a")`: Enviar mens. al Service Ag. para registrar un servicio especificado.
 - ❖ `.get_medics()`: Enviar mens. al Service Ag. para obtener los médicos de su equipo.
 - ❖ `.get_fieldops()`: Enviar un mensaje al Service Ag. solicitando los fieldops de campo de su equipo.
 - ❖ `.get_backup()`: Enviar un mensaje al Service Ag. solicitando los backup de su equipo.
 - ❖ `.get_service("servicio_a")`: Enviar un mensaje al Service Ag. solicitando otro servicio (distinto de los mencionados anteriormente) que los agentes tropa de su equipo ofrezcan.

Acciones de los Agentes Tropa

Recordatorio

- **.shoot(N,[X,Y,Z]):** Disparar N disparos a [X,Y,Z].
- **.cure:** Crear paquetes de medicina. Solo los médicos pueden realizar esta acción.
- **.reload:** Crear paquetes de munición. Solo los operadores de campo pueden realizar esta acción.



- ❖ Conservadoras:
 - ❖ Quedarse quieto
 - ❖ Ir a un punto más seguro: por ejemplo una esquina
 - ❖ Escapar si te disparan o se ve a un amigo
- ❖ Agresivas:
 - ❖ Ir al centro a por munición y medicina
 - ❖ Perseguir a un amigo disparando

pyGOMAS



Ejemplo: ir a una esquina

```
+flag(F): team(200)
```

```
<-
```

```
+amiesquina;
```

```
.goto([20, 0, 20]);
```

```
+miposicion([20, 0, 20]).
```

```
+target_reached(T): amiesquina
```

```
<-
```

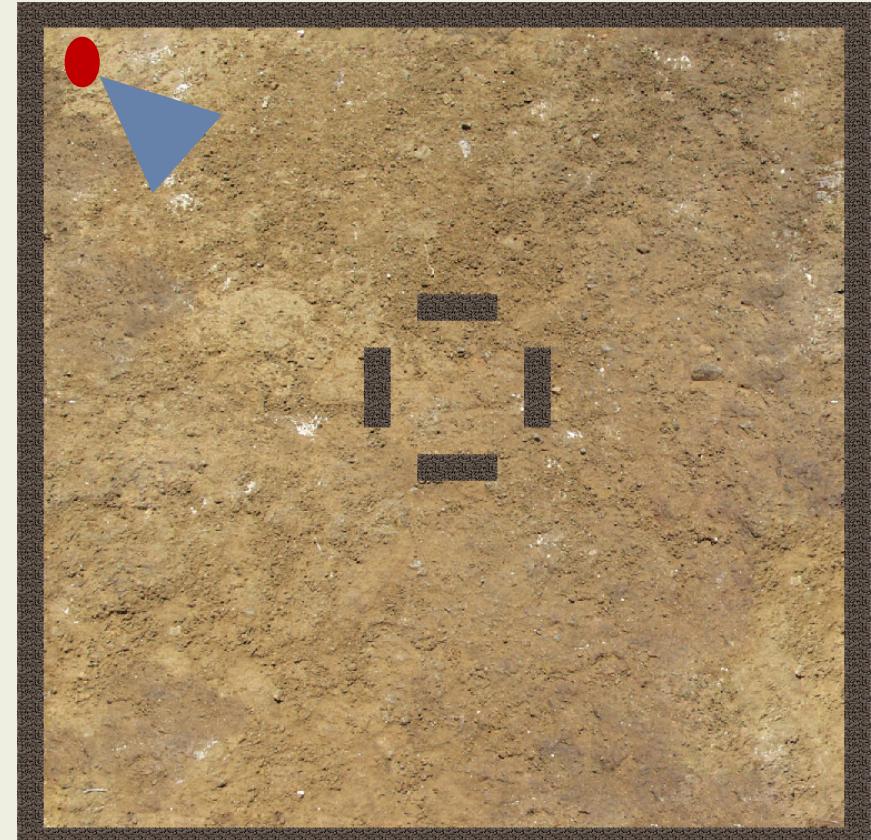
```
-amiesquina;
```

```
.print("Me quedo quieto en pos: ", T);
```

```
?flag(F);
```

```
.look_at(F);
```

```
-target_reached(T).
```



Ejemplo: ir a una esquina

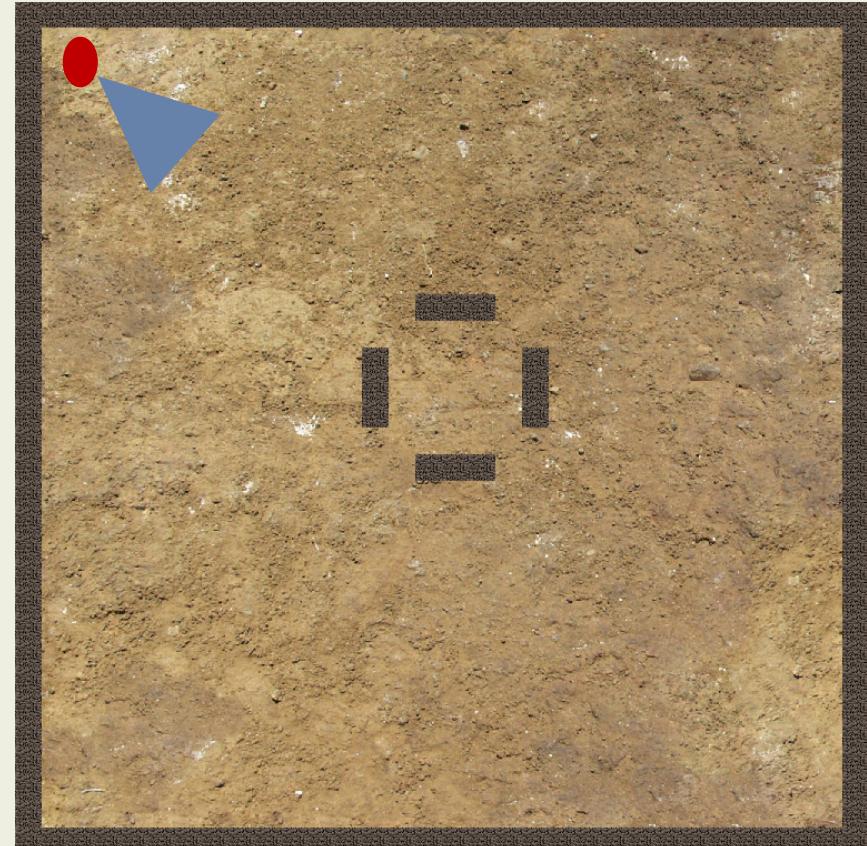
A destacar:

- La pared exterior tiene un grosor de 10 puntos en el mapa.

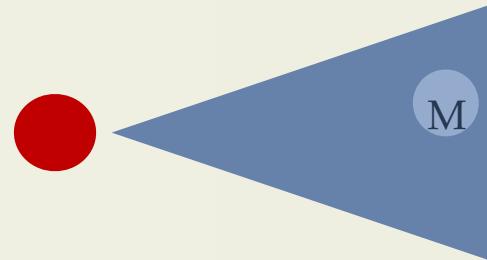
Ej: `.goto[3, 0, 3]` no funciona

- El ejemplo acude siempre a la esquina superior izquierda, pero un agente nace en cualquier sitio.

• Mejora: *Ir a la esquina más cercana*



Ejemplo: ir a por paquetes



```
+packs_in_fov(ID,Type,Angle,Distance,Health,Position): Type < 1003
```

```
<-
```

```
.goto(Position);  
+aporpaquete.
```



pyGOMAS



Ejemplo: girar todo el rato

```
+flag(F): team(200)
<-
  +mirando([[0,0,0],[250,0,0],[250,0,250],[0,0,250]]);
  +estado(0);
  !agirar.

+friends_in_fov(ID,Type,Angle,Distance,Health,Position)
<-
  .print("Disparo");
  .shoot(3,Position).
```

```
+!agirar: estado(E) & E<4
```

```
<-
  ?mirando(L);
  .nth(E, L, P);
  .look_at(P);
  .wait(1000);
  -estado(_);
  +estado(E+1);
  !agirar.
```

```
+!agirar: estado(E) & E=4
```

```
<-
  -estado(_);
  +estado(0);
  !agirar.
```

pyGOMAS



Ejemplo: girar todo el rato



RECORDATORIO

• Entrega el 27 de abril (tarea en Poliformat):

- Ficheros de código de vuestro agente: .asl, y en su caso, .py
- Fichero de configuración .json para lanzar el agente
- Documento con la descripción de la estrategia

- Se puede hacer por parejas

Enviad vuestras dudas por:

- Correo: vjulian@upv.es
carrasco@dsic.upv.es
- o por Poliformat