



# Prácticas AIN

---

## pyGOMAS

### Práctica 2: Comunicación y Coordinación

#### Añadiendo acciones internas

---



# Índice

---

- ❖ Recordatorio
  - ❖ Ejemplo de creación de una acción interna
    - ❖ Fichero .py
    - ❖ Fichero .asl
    - ❖ Fichero .json
-

# Recordatorio

---

- ❖ Una acción interna estará disponible desde el código JASON y comenzará con un ‘.’
  - ❖ Esta acción corresponderá a un método escrito en python, que pertenecerá a la clase de la que es nuestro agente.
-

# Ejemplo de creación de una acción interna

Fichero .py (I)

- \* Información disponible desde Python:
  - \* Atributos de AbstractAgent:
    - \* `team` : Número que identifica el equipo al que pertenece el agente
    - \* `services` : Lista con los identificadores de servicio que ofrece el agente.
  - \* Atributos de BDI Troop:
    - \* `manager` : jid del Agente Manager.
    - \* `service` : jid del Agente de Servicios
    - \* `is_objective_carried` : (true / false) indica si lleva la bandera o no
    - \* `fov_objects` : lista de objetos actualmente en el campo de visión del agente
    - \* `aimed_agent` : agente al que actualmente está apuntando (o None)
    - \* `health` : salud actual del agente
    - \* `ammo` : munición actual del agente
    - \* `is_fighting` : indica si el agente está luchando en este momento (True / False)
    - \* `is_escaping` : indica si el agente está escapando en este momento (True / False)



# Ejemplo de creación de una acción interna

Fichero .py (I)

- \* Información disponible desde Python:
  - \* Atributos de AbstractAgent:
    - \* `team` : Número que identifica el equipo al que pertenece el agente
    - \* `services` : Lista con los identificadores de servicio que ofrece el agente.
  - \* Atributos de BDI Troop:
    - \* Relativas al movimiento:
      - \* `map`
        - \* `map.can_walk(X, Z)` : indica si es pisable la posición (X, 0, Z) (True/Fals)
        - \* `map.allied_base.get_init_x()` , `map.allied_base.get_init_y()` , `map.allied_base.get_init_z()`
        - \* `map.allied_base.get_end_x()` , `map.allied_base.get_end_y()` , `map.allied_base.get_end_z()`
        - \* `map.axis_base.get_init_x()` , `map.axis_base.get_init_y()` , `map.axis_base.get_init_z()`
        - \* `map.axis_base.get_end_x()` , `map.axis_base.get_end_y()` , `map.axis_base.get_end_z()`
      - \* `velocity_value` : velocidad actual del agente
      - \* `destinations` : lista ordenada de los próximos destinos del agente.
      - \* `movement`
        - \* `movement.velocity.x`, `movement.velocity.y`, `movement.velocity.z`
        - \* `movement.heading.x`, `movement.heading.y`, `movement.heading.z`
        - \* `movement.destination.x`, `movement.destination.y`, `movement.destination.z`
        - \* `movement.position.x`, `movement.position.y`, `movement.position.z`

# Ejemplo de creación de una acción interna

Fichero .py (I)

- ❖ Información disponible desde Python:
  - ❖ Atributos de AbstractAgent:
    - ❖ `team` : Número que identifica el equipo al que pertenece el agente
    - ❖ `services` : Lista con los identificadores de servicio que ofrece el agente.
  - ❖ Atributos de BDI Troop:
    - ❖ `self.soldiers_count = 0`
    - ❖ `self.medics_count = 0`
    - ❖ `self.engineers_count = 0`
    - ❖ `self.fieldops_count = 0`
    - ❖ `self.team_count = 0`
  - ❖ `threshold = Threshold()` Limits of some variables (to trigger some events)
    - ❖ `threshold.health`
    - ❖ `threshold.ammo`
    - ❖ `threshold.aim`
    - ❖ `threshold.shot`

# Ejemplo de creación de una acción interna

Fichero .py (II)

❖ drunkenMonkey.py

```
import json
import random
from loguru import logger
from spade.behaviour import OneShotBehaviour
from spade.template import Template
from spade.message import Message
from pygomas.bditroop import BDITroop
from pygomas.bdifieldop import BDIFieldOp
from agentspeak import Actions
from agentspeak import grounded
from agentspeak.stdlib import actions as asp_action
from pygomas.ontology import DESTINATION

from pygomas.agent import LONG_RECEIVE_WAIT

class BDIDrunkenMonkey(BDIFieldOp):
    def add_custom_actions(self, actions):
        super().add_custom_actions(actions)
```

```
@actions.add(".drunkenMonkey", 0)
```

```
def _drunkenMonkey(agent, term, intention):
```

```
    randX = random.randrange(self.map.get_size_x() - 10)
    randZ = random.randrange(self.map.get_size_z() - 10)
    while (self.map.can_walk(randX, randZ) == False):
        randX = random.randrange(self.map.get_size_x() - 10)
        randZ = random.randrange(self.map.get_size_z() - 10)
```

```
    self.movement.destination.x = randX
```

```
    self.movement.destination.z = randZ
```

```
    self.bdi.set_belief(DESTINATION, tuple((randX, 0,
    randZ),))
```

```
    yield
```





# Ejemplo de creación de una acción interna

## Fichero .asl

### ❖ bdifieldop\_DM.asl

...

+enemies\_in\_fov(ID,Type,Angle,Distance,Health,Position)

<-

.drunkenMonkey;

.print("Drunken Monkey Fighting...");

.shoot(3,Position).



# Ejemplo de creación de acción interna

## Fichero .json

```
{
  "host": "gtirouter.dsic.upv.es",
  "manager": "ccc_m",
  "manager_password": "secret",
  "service": "ccc_s",
  "service_password": "secret",
  "axis": [
    {
      "rank": "BDISoldier",
      "name": "ccc_axis",
      "password": "secret",
      "amount": 8,
      "asl": "bdisoldier.asl"
    },
    {
      "rank": "BDIMedic",
      "name": "ccc_medic_axis",
      "password": "secret",
      "asl": "bdimedic.asl"
    },
    {
      "rank": "drunkenMonkey.BDIDrunkenMonkey",
      "name": "ccc_DM_fieldop_axis",
      "password": "secret",
      "asl": "bdifieldop_DM.asl"
    }
  ],
  "allied": [
    {
      "rank": "BDISoldier",
      "name": "ccc_allied",
      "password": "secret",
      "amount": 8,
      "asl": "bdisoldier.asl"
    },
    {
      "rank": "BDIMedic",
      "name": "ccc_medic_allied",
      "password": "secret",
      "asl": "bdimedic.asl"
    },
    {
      "rank": "BDIFieldOp",
      "name": "ccc_fieldop_allied",
      "password": "secret",
      "asl": "bdifieldop.asl"
    }
  ]
}
```

- ❖ Fichero JSON:  
Añadimos un nuevo tipo de agente que incorpora la acción comentada.