



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Escola Tècnica
Superior d'Enginyeria
Informàtica



etsinf

Escola Tècnica Superior d'Enginyeria Informàtica

Universitat Politècnica de València

9 de junio de 2025

PRÁCTICA 2

PYGOMAS

AGENTES INTELIGENTES AIN | GRUPO 1 - EOB

Grado en Informática Industrial y Robótica

Autores/Autoras:

Marcos Belda Martínez | mbelmar@etsinf.upv.es

Ángela Espert Cornejo | aespcor@etsinf.upv.es

Lourdes Francés Llimerá | lfralli@epsa.upv.es

Tutor:

Miguel Rebollo Pedruelo | mrebollo@upv.es

TABLA DE CONTENIDOS

| | |
|---|----------|
| 1. DESCRIPCIÓN DE LA ESTRATEGIA..... | 1 |
| 1.1. INTRODUCCIÓN..... | 1 |
| 1.2. EQUIPO IMPLEMENTADO (AXIS)..... | 1 |
| COMPORTAMIENTO INICIAL | 1 |
| 1.3. ESTRATEGIAS IMPLEMENTADAS..... | 2 |
| SOLDADOS | 2 |
| MÉDICOS Y OPERADORES DE CAMPO | 4 |
| FRANCOTIRADOR | 5 |
| 1.4. COMPORTAMIENTOS..... | 5 |
| 1.5. PROGAMA EN EJECUCIÓN..... | 6 |
| 1.6. MODIFICACIONES PARA EL FRANCOTIRADOR | 8 |
| MANAGER.PY..... | 8 |
| BDITROOP.PY | 8 |
| 1.7. ARCHIVOS DE LA ENTREGA | 9 |



1. DESCRIPCIÓN DE LA ESTRATEGIA

1.1. INTRODUCCIÓN

El presente documento tiene como objetivo describir y explicar la implementación del comportamiento de un equipo de 10 agentes en un entorno simulado, utilizando el lenguaje de programación *AgentSpeak*. El equipo en cuestión tiene como objetivo principal defender la bandera en un entorno potencialmente hostil donde otros 10 agentes del equipo enemigo buscan capturarla y eliminarlos. El código implementado detalla las acciones y reacciones del equipo en respuesta a diferentes situaciones y estímulos del entorno.

1.2. EQUIPO IMPLEMENTADO (AXIS)

El equipo implementado está formado por 10 agentes (4 *BDISoldier*, 3 *BDIFielOps*, 3 *BDIMedic* y 1 *BDISniper*) los cuales heredan funcionalidades básicas de movimiento y combate, así como de curación y reposición de munición. Además, se añaden a estas clases nuevos comportamientos y ciertas modificaciones para optimizar la respuesta del equipo ante hostilidades del entorno.

ROLES Y RESPONSABILIDADES

- **Soldados (*BDISoldier_EOB*):** Realizan patrullas alrededor de la bandera, detectan enemigos y responden a las amenazas.
- **Francotiradores (*BDISniper_EOB*):** Proporcionan apoyo de largo alcance y responden a las notificaciones de enemigos detectados.
- **Médicos (*BDIMedic_EOB*):** Atienden a los agentes que requieren curación y gestionan las propuestas de ayuda.
- **Operadores de Campo (*BDIFieldOp_EOB*):** Gestionan la munición y proporcionan apoyo a los agentes que lo necesitan.

COMPORTAMIENTO INICIAL

Todos los agentes se inicializan en su base con una salud y una munición inicial de 100.

SOLDADOS, MÉDICOS Y OPERADORES DE CAMPO

Para iniciar su comportamiento, se emplea la acción **+flag(F)**, que les indica que se desplacen hacia la bandera que tienen que defender. Para conseguir que dicha defensa sea óptima y correcta, se posicionan formando puntos de control (órbitas) alrededor de la bandera.

PRIMERA ÓRBITA

Esta órbita se encuentra muy próxima a la bandera. En esta se sitúan los *BDIMedic*, los cuales se encargan de cubrir la retaguardia y atender a los heridos en caso de ser necesario.

SEGUNDA ÓRBITA

En esta órbita, situada en el medio y un poco más alejada de la bandera, se sitúan los *BDIFieldOps*, encargados de suministrar munición al equipo.

TERCERA ÓRBITA

En esta última órbita, la más lejana de la bandera, se sitúan los *BDISoldier*. Estos se encargan de patrullar los alrededores y de informar al equipo sobre avistamiento enemigo para solicitar refuerzos.

FRANCOTIRADOR

Para iniciar su comportamiento, se emplea la acción ***+!start***, la cual permite registrar el servicio de “sniper” y localizar la posición de la base para que este agente se desplace a la misma.

1.3. ESTRATEGIAS IMPLEMENTADAS

SOLDADOS

PATRULLA

Esta estrategia se ejecuta cuando el agente recibe la posición de la bandera (***+flag(F)***). El agente crea puntos de control alrededor de la bandera y patrulla entre estos puntos de manera cíclica.

IMPLEMENTACIÓN

- El agente crea puntos de control alrededor de la bandera (***+flag(F)***) a una distancia preestablecida según la órbita que le corresponda (30, 20 y 10 de distancia según si es *soldier*, *fieldop* o *medic* respectivamente).
- El agente se mueve hacia el primer punto de control (***+patroll_point(0)***).
- Una vez alcanzado un punto de control, el agente se mueve hacia el siguiente punto (***+patroll_point(P+1)***).
- Si el agente alcanza el último punto de control, vuelve al primer punto (***+patroll_point(0)***).

DETECCIÓN DE ENEMIGOS Y RESPUESTA

Esta estrategia se ejecuta cuando el agente detecta un enemigo (***+enemies_in_fov(ID,Type,Angle,Distance,Health,Position)***).

IMPLEMENTACIÓN

- Dispara a la posición del enemigo (***+shoot(3, Position)***).
- El agente desactiva la patrulla (***-patrolling***).
- El agente notifica a los demás soldados y al francotirador (***+myBackups(B)*** y ***+sniper(S)***).
- El agente se dirige a la posición del enemigo (***+goto(Position)***).

Si el soldado ya está en combate, es decir, ya ha pedido refuerzos, este se limita a disparar a la posición del enemigo sin solicitar refuerzos nuevamente.

Una vez este llega a la posición del enemigo, si sigue detectando al enemigo, continúa ejecutando esta estrategia, pero, en caso de que ya no haya enemigos visibles, reanuda el patrullaje.

NOTIFICACIÓN AL FRANCOTIRADOR Y A LOS SOLDADOS

Para notificar al francotirador y a los soldados, se emplean dos funciones: *?enemies_in_fov(____,Pos)* y *.send(X, tell, enemyDetected(Pos))*, donde “X” se sustituye por “S” en el caso de las notificaciones al francotirador y, por “B” en el caso de las notificaciones a los soldados.

GESTIÓN DE SOLICITUD DE REFUERZOS

Esta estrategia se ejecuta cuando se solicitan refuerzos por parte de cualquier aliado (*+enemyDetected(Position)*).

IMPLEMENTACIÓN

- El agente deja de patrullar y se dirige a la posición que se le ha proporcionado, es decir, la posición del enemigo.
- Una vez ha llegado a esta posición, como detectará enemigo, este disparará al enemigo.
- En caso de llegar a la posición del enemigo y no encontrarlo, reanudará el patrullaje.

En el caso de que el soldado ya esté combatiendo, ignorará la nueva solicitud de ayuda.

GESTIÓN DE SALUD

Esta estrategia se ejecuta cuando la salud del agente está por debajo de un umbral (*+health(H)*).

IMPLEMENTACIÓN

- El agente evalúa si necesita atención médica (*.checkHealth(H, Res)*)
- El agente pide ayuda si su salud es menor de 50 (*+healthNeeded*).
- El agente notifica a los médicos (*+myMedics(M)*) y les envía su posición mediante un mensaje *.send(M, tell, cureProposal(Pos))*.

GESTIÓN DE MUNICIÓN

Esta estrategia se ejecuta cuando la munición del agente está por debajo de un umbral (*+ammo(A)*).

IMPLEMENTACIÓN

- El agente evalúa si necesita munición (*.checkAmmo(A, Res)*)
- El agente pide ayuda si su munición es menor de 50 (*+ammoNeeded*).
- El agente notifica a los operadores de campo (*+myFieldOps(F)*) y les envía su posición mediante un mensaje *.send(F, tell, ammoProposal(Pos))*.

GESTIÓN DE PAQUETES DE SUMINISTROS

Estas estrategias se ejecutan cuando el agente detecta un paquete de suministros del tipo que requiere (***+packs_in_fov(ID,Type,Angle,Distance,Health,Position)***). Estos pueden ser del tipo 1002 (botiquín) y del tipo 1001 (munición).

IMPLEMENTACIÓN

- El agente se dirige a la posición del paquete (***+goto(Position)***).
- Cuando el agente alcanza la posición del paquete, lo recoge y reanuda su patrullaje.

MÉDICOS Y OPERADORES DE CAMPO

PATRULLA

Esta estrategia es la misma empleada para el soldado, cambiando la distancia que hay entre los puntos de control y la bandera.

DETECCIÓN DE ENEMIGOS Y RESPUESTA

Esta estrategia se ejecuta cuando el agente detecta un enemigo (***+enemies_in_fov(ID,Type,Angle,Distance,Health,Position)***).

IMPLEMENTACIÓN

- Dispara a la posición del enemigo (***+shoot(3, Position)***).
- El agente desactiva la patrulla (***-patrolling***).
- El agente notifica a los demás soldados y al francotirador (***+myBackups(B)*** y ***+sniper(S)***).
- El agente se retira hacia la bandera (***+retreating***) en caso de que no esté aproximándose a ayudar a algún agente que requiere asistencia médica o munición (***?flag(F)*** y ***+goto(F)***).

NOTIFICACIÓN AL FRANCOTIRADOR Y A LOS SOLDADOS

La notificación al francotirador y a los soldados se realiza igual que en el caso de los soldados.

GESTIÓN DE SOLICITUD DE ASISTENCIA MÉDICA

Esta estrategia se ejecuta cuando se solicita asistencia médica por parte de cualquier aliado (***+cureProposal(Pos)[source(A)]***).

IMPLEMENTACIÓN

- El agente deja de patrullar y se dirige a la posición que se le ha proporcionado, es decir, la posición del aliado que necesita asistencia médica.
- Una vez ha llegado a esta posición, suelta un botiquín y reanuda el patrullaje.

En el caso de que el médico ya esté yendo a una solicitud de ayuda, ignorará la nueva solicitud.

GESTIÓN DE SOLICITUD DE MUNICIÓN

Esta estrategia se ejecuta cuando se solicita munición por parte de cualquier aliado (***+ammoProposal(Pos)[source(A)]***).

IMPLEMENTACIÓN

- El agente deja de patrullar y se dirige a la posición que se le ha proporcionado, es decir, la posición del aliado que necesita munición.
- Una vez ha llegado a esta posición, suelta un paquete de munición y reanuda el patrullaje.

En el caso de que el operador de campo ya esté yendo a una solicitud, ignorará la nueva solicitud.

GESTIÓN DE SALUD, MUNICIÓN Y PAQUETES DE SUMINISTROS

Esta gestión se realiza igual que en el caso de los soldados.

FRANCOTIRADOR

ESTADO DE ESPERA

Inicialmente, el francotirador se coloca en la base. Una vez en posición, este espera a que sus aliados soliciten sus servicios.

DETECCIÓN DE ENEMIGOS Y RESPUESTA

Cuando se han detectado enemigos, el francotirador recibe una solicitud de ayuda con la posición en la que se encuentra el enemigo al que debe disparar. Tras esto, mira a la posición obtenida y dispara cinco veces.

GESTIÓN DE SALUD, MUNICIÓN Y PAQUETES DE SUMINISTROS

Esta gestión se realiza igual que en el caso de los soldados, médicos y operadores de campo. La única diferencia radica en que el francotirador, una vez recogido el paquete de suministros, vuelve a la base para ponerse en estado de espera nuevamente.

1.4. COMPORTAMIENTOS

Para el control de la munición y la salud de los agentes se ha hecho uso de un archivo *Python* de nombre “*EOB_behavior.py*”. Este archivo cuenta con una clase por cada tipo de agente: ***BDIMedicBehavior(BDIMedic)***, ***BDIFieldOpBehavior(BDIFieldOp)***, ***BDISoldierBehavior(BDISoldier)*** y ***BDISniperBehavior(BDISoldier)***. Aun así, todos cuentan con las mismas acciones: ***.checkHealth*** y ***.checkAmmo***.

PARÁMETROS INICIALES

Cada clase de agente inicializa dos parámetros clave en su constructor (***__init__***):

- ***last_health***: Representa la salud más reciente registrada del agente. Inicialmente se establece en 100.

- **last_ammo:** Representa la cantidad de munición más reciente registrada del agente. Inicialmente se establece en 100.

Además, se definen dos indicadores booleanos:

- **health_needed:** Indica si el agente necesita atención médica (1) o no (0).
- **ammo_needed:** Indica si el agente necesita reabastecimiento de munición (1) o no (0).

ACCIÓN DE CONTROL DE SALUD

La acción **.checkHealth** permite detectar si el agente ha sufrido daño desde la última verificación. Esta función toma un parámetro entero (**x**), que representa el valor actual de salud del agente. Si el valor actual es menor que **last_health**, se considera que el agente necesita atención médica, por lo que **health_needed** se establece en 1. En caso contrario, se asigna 0. Finalmente, se actualiza **last_health** con el nuevo valor recibido.

Este mecanismo permite a los agentes reaccionar ante heridas de combate y solicitar asistencia médica de forma autónoma. Además, permite que, al modificarse la salud porque esté recuperándose la salud del agente, no se active la solicitud de ayuda médica.

ACCIÓN DE CONTROL DE MUNICIÓN

La acción **.checkAmmo** evalúa si el agente ha utilizado munición desde la última vez que se comprobó su estado. La función recibe como entrada un número entero (**x**) con el valor actual de munición del agente. Si este valor es menor que **last_ammo**, se infiere que el agente ha gastado munición, por lo que se establece **ammo_needed** en 1. En caso contrario, el valor será 0. Luego se actualiza **last_ammo** con el valor actual.

Este mecanismo permite a los agentes reaccionar ante necesidades de reabastecimiento durante el combate y solicitar asistencia de los operadores de campo de forma autónoma. Además, permite que, al modificarse la munición porque esté reabasteciéndose la munición del agente, no se active la solicitud de ayuda.

1.5. PROGAMA EN EJECUCIÓN

A continuación, se aportan algunas imágenes del programa en ejecución.

INICIALIZACIÓN DEL PROGRAMA

En la Figura 1 se puede apreciar el programa recién iniciado, después de que los agentes hayan obtenido sus puntos de control y el francotirador se haya puesto en estado de espera.

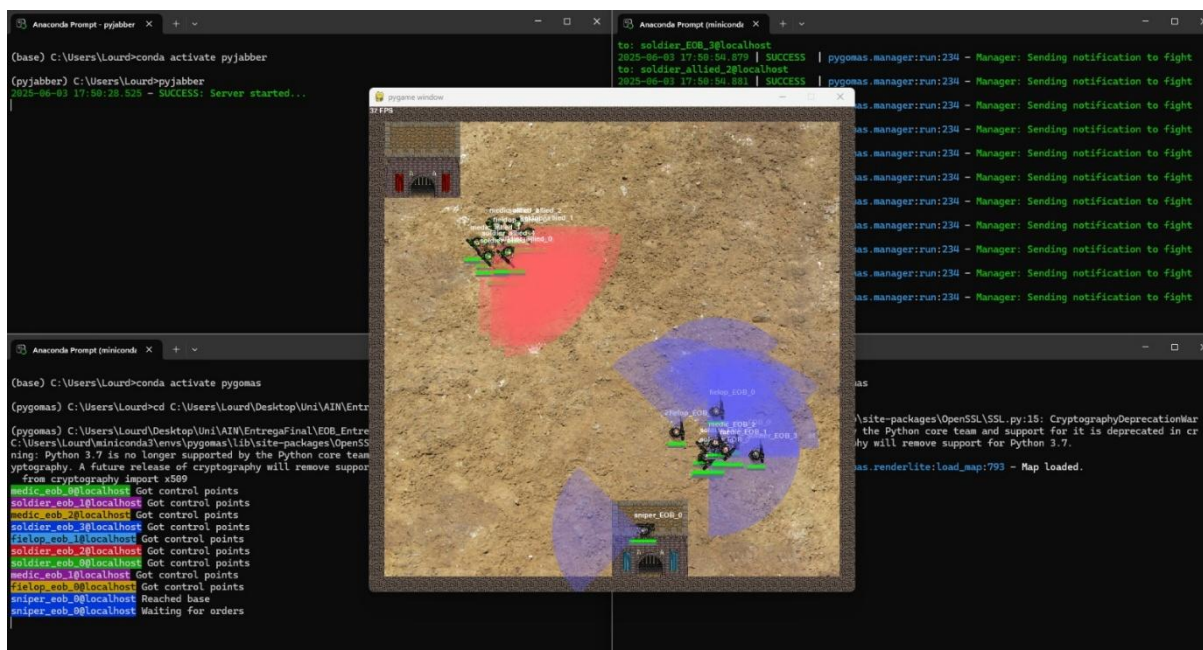


Figura 1: Programa tras su inicialización.

ESTRATEGIAS IMPLEMENTADAS

La Figura 2 muestra la estrategia de retirada y solicitud de refuerzos por parte del *fieldop_eob_0*. También muestra la actuación del *sniper*, el cual dispara a la posición del enemigo, y la actuación de los *soldier*, los cuales notifican que han recibido la solicitud de refuerzos y comienzan a aproximarse al enemigo.

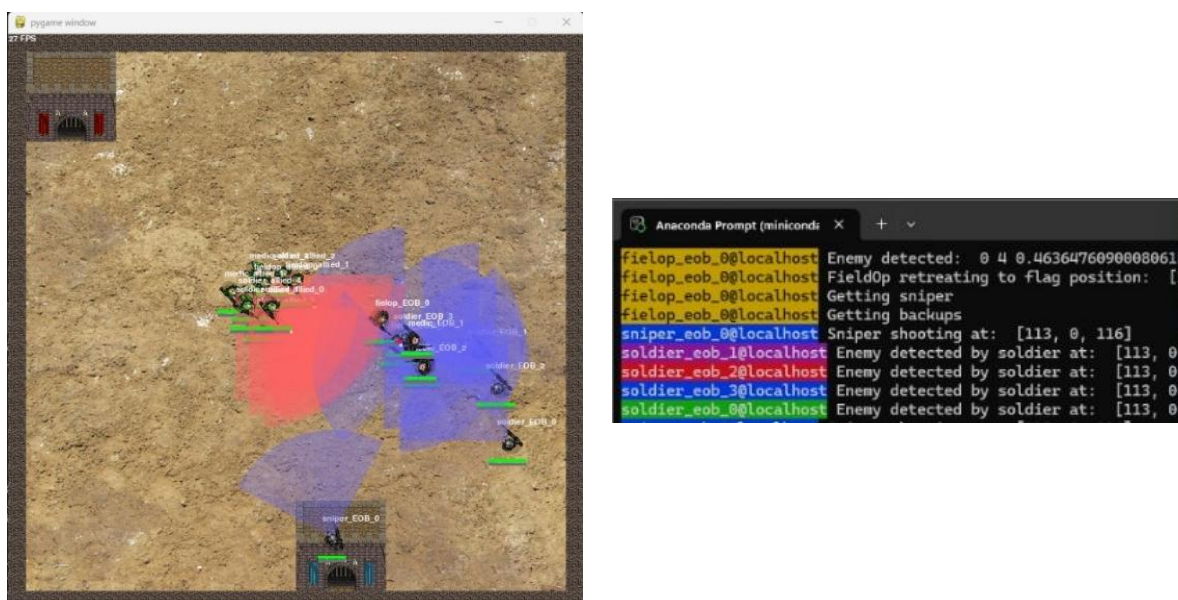


Figura 2: Retirada del *fieldop*, solicitud de refuerzos y actuación del *sniper* y los *soldier*.

La Figura 3 muestra la estrategia de solicitud de asistencia médica por parte del *fieldop_eob_0*. También muestra la actuación del *medic*, el cual notifica que ha recibido la solicitud y comienza a aproximarse al aliado que ha solicitado la asistencia.

```

fielop_eob_0@localhost Health is low, need a medic!
medic_eob_1@localhost Helping agent: fielop_eob_0@localhost

```

Figura 3: Solicitud de asistencia médica y actuación del *medic*.

La Figura 4 muestra la solicitud de refuerzos por parte del *medic_eob_2* y la solicitud de asistencia médica por parte del *soldier_eob_3*. También muestra la actuación del *soldier* y del *medic* respectivamente, los cuales rechazan la solicitud al ya estar ocupados.

```

medic_eob_2@localhost Enemy detected: 4 1 0.19739555984988044
soldier_eob_3@localhost Already fighting, cannot help!

soldier_eob_3@localhost Soldier needs healing!
fielop_eob_1@localhost Enemy detected: 0 2 0.7853981633974483
medic_eob_2@localhost Medic retreating to flag position: [184,
medic_eob_0@localhost Already helping someone! Cannot help aget

```

Figura 4: Rechazo de solicitudes de refuerzos y asistencia médica.

FINALIZACIÓN DEL PROGRAMA

En la Figura 5 se puede apreciar el final de la ejecución del programa. En esta se muestra como el equipo *axis* ha ganado debido a que el tiempo de partida ha expirado y el equipo *allied* no ha conseguido hacerse con la bandera ni eliminar a todos los enemigos.

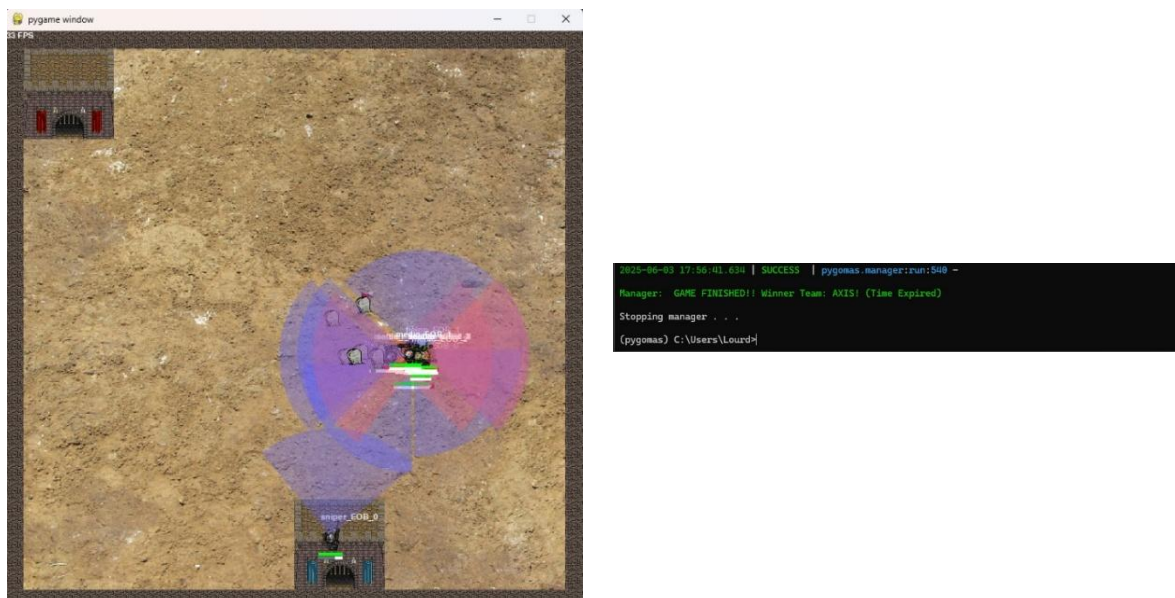


Figura 5: Programa tras su finalización. Victoria del equipo *axis*.

1.6. MODIFICACIONES PARA EL FRANCOTIRADOR

MANAGER.PY

Para simular el daño que causaría un disparo de francotirador, se ha modificado el archivo *manager.py* utilizando la clase *CLASS_SNIPER* definida en el archivo *bditroop.py*. En esta implementación, se ha establecido que el daño causado por cada bala del *sniper* sea de 4 puntos, el doble que el daño causado por un *soldier*.

BDITROOP.PY

Como se ha mencionado anteriormente, se ha creado una nueva clase *CLASS_SNIPER* en el archivo *bditroop.py* para modificar individualmente el daño del agente *SNIPER* sin afectar a los demás agentes.

1.7. ARCHIVOS DE LA ENTREGA

A la hora de realizar el trabajo, se ha dividido la entrega en cuatro carpetas:

- **EOB_code:** En esta carpeta se encuentra la lógica principal de la entrega. Entre los archivos de esta se encuentran los siguientes.
 - Archivos **bdiagent.asl** con las estrategias implementadas para cada tipo de agente del equipo allied aportado por el profesorado.
 - Archivos **bdiagent_EOB.asl** con las estrategias implementadas para cada tipo de agente del equipo axis implementado por el alumnado.
 - Archivo **EOB_behavior.py** con el comportamiento implementado para cada tipo de agente.
 - Archivo **EOB.json** con la distribución de los agentes.
- **EOB_documentation:** En esta carpeta se encuentra el archivo *PDF* en que se encuentra redactado el proyecto.
- **EOB_help_resources:** En esta carpeta se encuentran dos archivos.
 - **execution_help.py:** Este archivo permite comprobar que no haya ningún carácter no deseado en los archivos que se deben ejecutar. Su presencia ha sido clave para evitar fallos de ejecución por tildes o comillas no permitidas en la ejecución de programas mediante *pygomas*.
 - **execution_info.txt:** Este archivo permite acceder rápidamente a los comandos necesarios para ejecutar el código correctamente.
- **EOB_sniper_modifications:** En esta carpeta se encuentran los archivos **manager.py** y **bdiroop.py** modificados.