Guia practica de nivel avanzado, para ganar todas las batallas.
De alumnos para alumnos :)

General

!: se desea alcanzar algo.
?: se desea probar algo.
;: indica secuencia.
←: indica implicación.
$\sim l$: *strong negation*: el agente cree que el literal l es explícitamente falso.
not l: el agente no cree que el literal l sea verdadero.
not $\sim l$: el agente no cree que el literal l sea falso.
& |: expresiones de contexto *and* y *or*.
$Comentarios$: una línea // varias líneas /* */.
Todas las variables deben empezar por letra mayúscula.

Eventos de Disparo

+: de adición de creencias.
−: de borrado de creencias.
+!: de adición de objetivos a alcanzar (*achievement goals*).
−!: de borrado de objetivos a alcanzar (*achievement goals*).
+?: de adición de objetivos a testear (*test goals*).
−?: de borrado de objetivos a testear (*test goals*).

$creencia[source(percept)]$: creencia percibida del entorno.
$creencia[source(agenteX)]$: creencia comunicada por el agente X.
$creencia[source(self)]$: creencia por el propio agente.

BDI

$.drop\_\{desire, intention, event\}$: elimina un deseo/intención/evento de un agente.
$.drop\_all\_\{desires, intentions, events\}$: elimina todos los deseos/intenciones/eventos de un agente.

Base de creencias

$.abolish$: elimina algunas creencias.
$.findall$: encuentra una lista de creencias de algún tipo.

Biblioteca de Planes

$.add\_plan$: añade nuevos planes.
$.remove\_plan$: elimina un plan.

Comunicación

$.send$: envía mensajes.
$.broadcast$: envía mensajes *broadcast*.
$.my\_name$: devuelve el nombre del agente.
$.all\_names$: devuelve el nombre de todos los agentes del sistema.

Listas y Sets

$.member$: listar miembros.
$.length$: tamaño de una lista.
$.concat$: concatenar listas.
$.delete$: eliminar miembros de una listaa.
$.nth$: enésimo elemento de una lista.
$.max$: máximo valor de una lista.
$.min$: mínimo valor de una lista.
$.sort$: ordenar listas.
$.list$: comprobar si un argumento es una lista.

Strings

$.concat$: concatenar strings.
$.delete$: eliminar caracteres de un string.
$.substring$: testea substrings de un string.
$.string$: comprobar si un argumento es un string.

Control execution

$.if$: implementación de *if*.
$.while$: implementación de *while*.
$.for$: implementación de *for*.

Variado

$.at$: añadir un evento futuro.
$.wait$: esperar un evento.
$.create\_agent$: crear un agente nuevo.
$.kill\_agent$: matar un agente.
$.stopMAS$: parar todos los agentes.
$.date$: devuelve la fecha actual.
$.time$: devuelve la hora actual.
$.random$: produce números aleatorios.

Guia practica de nivel avanzado, para ganar todas las batallas.
De alumnos para alumnos :)

Comunicacion:
.send(agenteReceptor, prformativa , obj) : comando para mandar mensajes de un agente a otro

Performativa:
- '*tell*' : Instancia una creencia que se declara en obj para que el agente receptor la ejecute
- '*achieve*' : Instancia un objetivo a conseguir obj en el agente receptor

Servicios:
- *.register_service('nombreServicio')* : pide registrar un servicio de nombre 'nombreServicio que otros agentes podían usar
- *.get_service('nombreServicio')* : solicita al agente de servicios la lista de los agentes que ofrecen el servicios *nombreServicio,* de sevolverá en *+nombreServicio(Agents_that_offer_this_service_list)*

- *.get_medics* : Solicita al agente de servicios la lista de los medicos pertenecientes a su equipo, se retornara en una creencia nueva *myMedics(Medics_list)*
- *.get_fieldops* : solicita al agente de servicios que devuelva la lista de operadores de mi equipo activos, se retornaran en *myFieldops(Fieldops_list)*
- *.get_backups* : solicita al agente de servicios que devuelva la lista de soldados de mi equipo activos en una lista: se retornaran en *myBackups(Backups_list)*

Acciones internas Python interesantes:
Atributos BDITroop:
- *is_objective_carried* : Indica si lleva la bandera o no (True / False)
- *fov_objects* : lista de objetos actualmente en el campo de vision del agente
- *aimed_agent* : agente al que actualmente esta apuntando (id / None)
- *health* : salud del agente actual
- *ammo* : municion actual del agente

- *is_fighting* : indica si el agente esta luchando en este momento (True / False)
- *is_escaping* : indica si el agente está escapando en este momento (True / False)
- *map.can_walk(X,Z)* : indica si es posible pisar la posicion (X, 0 , Z) -> (True / False)

- *self.soldiers_count* : indica la cantidad de soldados con vida
- *self.medics_count* : indica la cantidad de medics con vida
- *self.fieldops_count* : indica la cantidad de fieldops con vida
- *self.team_count* : indica la cantidad de agentes del team con vida

Acción interna con Python, ejemplo fácil:

```
@actions.add_function(".distance", (tuple,tuple, ))
def _distancia(p1, p2):
    return ((p1[0]-p2[0])**2+(p1[2]-p2[2])**2)**0.5
```

- *(tuple, tuple,)* : recibira dos tuplas y devolverá algo, por lo tanto se llamaria desde el agente como *.nombre(tupla, tupla, return)*
- "*.distance*" : nombre de la instruccion nueva añadida

- *@actions.add_function("nombre", 0)* : la accion se ejecutara con .nombre y no devolverá nada

RECORDATORIO GENERAL BDI

- 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).
- 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.
- 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].

Guia practica de nivel avanzado, para ganar todas las batallas.
De alumnos para alumnos :)

- 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]).
- .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.
- .shoot(N,[X,Y,Z]): Disparar N disparos a [X,Y,Z].

DE INTERES:

- *.cure* : si se usa en medico, crea botiquines y los lanza abajo, ojo con el mana del agente
- *.reload* : si es fieldop, crea municion que lanza al campo, ojo con el mana del agente.