

A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the date.

30-5-2017

# JGOMAS-AIN

Entrega práctica 2

Several thin, curved lines in dark blue and light grey originate from the bottom left corner and curve upwards and to the right.

RAMON RUIZ DOLZ  
SALVADOR MARTI ROMAN

## ALLIED TEAM

Esta segunda entrega del laboratorio de agentes inteligentes consiste en hacer un equipo que mediante la comunicación y técnicas del acuerdo consiga bien hacerse con el control de la bandera (ALLIED) o bien defenderla (AXIS). En nuestro caso hemos escogido ALLIED y hemos implementando una estrategia de ataque para tomar la bandera.

Para ello tenemos dos tipos de ficheros, el `jasonAgent_ALLIED_COMANDANTE.asl` una unidad de tipo médico que cumple el rol de comandante de las tropas dando órdenes y asignando posiciones, y el `jasonAgent_ALLIED_COBERTURA.asl` que serán los otros 7 soldados que se encargarán de cubrir al comandante.

Para cubrir el problema de las formaciones hemos utilizado un método iterativo de manera que independientemente del número de agentes estos realizaran la formación correctamente, permitiendo así nuestro código que sea funcional con más de los 8 agentes preestablecidos para la competición de la práctica. El trozo de código es el siguiente:

```
+!a_formar(Batallon)
  <-
    .length(Batallon, Len);
    +loop(0);
    ?my_position(X,Y,Z);
    if(Len>0){
      while( loop(I) & (I<Len) ){
        .println("Orden enviada", I);
        ?posicionesform(K);
        .nth(I, Batallon, Soldado);
        .concat("en_posicion(",X+K," ",Y," ",Z+5,")",Posi);
        .send_msg_with_conversation_id(Soldado,tell,Posi,"INT");
        -+posicionesform(K+4);
        -+loop(I+1);
      }
      .println("fin del envio de ordenes");
      -loop(_);
      -posicionesform(_);
    }
  .
```

Como podemos observar por cada agente en batallón envía un mensaje con la posición destino a la que este debe dirigirse, en K tenemos el desplazamiento entre una posición de la formación y la siguiente por eso en cada iteración es modificada. De esta forma conseguimos indicar diferentes posiciones sin la necesidad de fijar el número a un determinado número de soldados.

Por otra parte, cabe destacar que la estrategia que hemos tomado consiste en tomar la bandera lo más rápido posible y retirarse a nuestra base corriendo. Para conseguir este “rush” lo que hacen los agentes es dentro del plan `update_targets` comprueban si tienen la bandera y, en este caso realizan una limpieza de tareas y añaden la tarea de volver a la base inmediatamente. Esto lo podemos observar en el siguiente trozo de código:

```
if(LT==1 & Flag==on){
    .println("Vamos a por todas!");
    .nth(0,TASKS,Vuelta);
    -+Vuelta;
    ?task(,_,_, PosicionBase,_);
    .println(PosicionBase);
    -+PosicionBase;
    ?pos(XB,YB,ZB);
    .my_team("comandante",Com);
    .my_team("cobertura",Cov);
    .concat(Cov,Com,Team);
    // Decirles a todos que vayan a la posición de la
bandera para luego seguirme //

    .concat("victoria(",XB,"",YB,"",ZB,"")",AcabemosEsto);

    .send_msg_with_conversation_id(Team,tell,AcabemosEsto,"INT");

}
```

Finalmente, y cumpliendo con todos los requisitos solicitados, nuestro equipo aliado además de formar, comunicarse y tomar la bandera, también evitaran el fuego amigo. Para ello mediante el cálculo de la distancia euclidiana entre el agente y el enemigo apuntado, el agente y los aliados y los aliados y el enemigo, somos capaces de saber si hay un aliado en medio de la línea de tiro del agente que va a disparar y el agente enemigo.

```
+!distancia(pos(X1,Y1,Z1), pos(X2,Y2,Z2))
<-
//Distancia euclidea entre dos puntos
D=math.sqrt((X1-X2)*(X1-X2)+(Y1-Y2)*(Y1-Y2)+(Z1-Z2)*(Z1-Z2));
-+distancia(D);
.
```

Con este cálculo tan solo insertando otro loop dentro del método get agent to aim, donde se recorran los aliados en el campo de visión y si calculen sus distancias euclídeas en cada iteración respecto al agente que dispara y al enemigo, y solamente en el caso de darse que la distancia del agente que dispara con el enemigo sea mayor o igual que la suma de las distancias euclídeas del agente que dispara con el enemigo y el aliado con el enemigo, en este caso deberá dejar de disparar a este enemigo y buscará a otro mejor en su lista de objetos avistados.

```
if( DistEn+5>=DistAl+DistAlEn ){
    -aimed_agent(Object);
    -+aimed("false");
    ?debug(Mode); if
(Mode<=3) { .println("Aiming enemy canceled, in middle . . .", Object2); }

}
```

De esta forma el fuego amigo es sustancialmente reducido ya que los agentes rápidamente se dan cuenta de que están impactando en un aliado.