

MEMORIA PRÁCTICA 1 AIN

AUTOR:

ÁNGEL IGUALADA MORAGA

Ejercicio 1:

Se ha modificado init y perform_look_action , consultamos la creencia my_position que nos devuelve tres variables y las printamos.

Init printará al iniciar y perform_look_action cada vez que mire.

Ejercicio 2:

En este caso, en perform_look_action consultamos la creencia fovObjects que nos devuelve una lista de objetos que printamos.

Ejercicio 3:

En este ejercicio creamos un test goal llamado contador_esquinas con el valor 0 para saber a que esquina vamos y el test goal primera_esquina , únicamente sirve par realizar prints con sentido.

Dependiendo del numero de esquina de contador_esquinas, asignamos sus coordenadas, tras esto, vamos probando posiciones en diagonal desde la asignada hasta que alguna sea valida (safe_pos daba problemas).

Cuando la encontramos, lanzamos la tarea go_to_target en esa posición y actualizamos contador_esquinas.

Ejercicio 4:

Modificamos perform_look_action y recorremos la lista proporcionada por fovObjects, si alguno es un agente y su equipo es el 200 (somos AXIS) lo guardamos en un test goal y tras esto, comprobamos si encontramos algún agente de nuestro equipo, si es así, lo recuperamos y comprobamos si su posición es válida y si lo es lanzamos una tarea de tipo go_to_target a su posición

CÓDIGO:

EJ 1:

```
+!perform_look_action
  <- ?debug(Mode); if (Mode<=3) {
    //muestro mi posición cada vez que miro
    ?my_position(X, Y, Z);
    .println("ESTOY EN LA POSICION : X = ", X, " Z = ", Z)
  }
.
```

```
+!init
  <- ?debug(Mode); if (Mode<=3) {
    // Muestra la posición inicial
    .println("LISTO");
    ?my_position(X, Y, Z);
    .println("ESTOY EN LA POSICION : X = ", X, " Z = ", Z)
  }.
.
```

EJ 2:

```
+!perform_look_action
  <- ?debug(Mode); if (Mode<=3) {
    ?fovObjects(FOVObjects);
    .println(FOVObjects)
  }
.
```

EJ 3:

```
+!init
  <- ?debug(Mode);
  if (Mode<=3) {
    .println("VOY A INTENTAR IR A LA PRIMERA ESQUINA");
  }
  +contador_esquinas(0);
  +primera_esquina(1);
.
```

```

+!update_targets
  <- ?debug(Mode);

  ?contador_esquinas(N);
  ?my_position(X, Y, Z);
  ?primera_esquina(C);
  .my_name(MyName);
  if (C==1){
    -+primera_esquina(0);
  } else{
    .println("ESQUINA ALCANZADA","\n\nCALCULANDO SIGUIENTE ESQUINA ...");
  }
  /*
  | 3   2 |   | (0,255)   (255,255) | | |
  |   |   |   |   |   |
  | 0   1 |   | (0,0)     (255,0)  |
  -----
  RENDER
  | 0   1 |   | (0,0)     (255,0)  | | |
  |   |   |   |   |   |
  | 3   2 |   | (0,255)   (255,255) |
  -----

  */
  if (N == 1 | N == 2) {
    +newX(255);
  } else {
    +newX(0);
  }
  if (N <= 1) {
    +newZ(0);
  } else {
    +newZ(255);
  }

+position(invalid);
while(position(invalid)) {
  -position(_);
  ?newX(NX);
  ?newZ(NZ);
  if (N == 1 | N == 2) {
    -+newX(NX - 1);
  } else {
    -+newX(NX + 1);
  }
  if (N <= 1) {
    -+newZ(NZ + 1);
  } else {
    -+newZ(NZ - 1);
  }

  ?newX(NX2);
  ?newZ(NZ2);
  check_position(pos(NX2, Y, NZ2));
  ?position(P);
  if (Mode<=3) {
    .println("COMPROBANDO... (" , NX2, ", ", NZ2, ") is ", P);
  }
}

?newX(NX3);
?newZ(NZ3);
if (Mode<=3) {
  .println("ME DIRIJO A LA ESQUINA ",N," (" ,NX3," ",NZ3,")");
}
!add_task(task(9999, "TASK_GOTO_POSITION",MyName,pos(NX3,Y,NZ3),""));
-newX(_);
-newZ(_);

if (N == 3) {
  -+contador_esquinas(0);
} else {
  -+contador_esquinas(N + 1);
} .

```

EJ 4:

```
+!perform_look_action
<- ?debug(Mode);
  ?fovObjects(FOVObjects);
  .length(FOVObjects, Length);
  if (Length > 0) {
    +bucle(0);
    --targeted("false");
    while (targeted("false") & bucle(X) & (X < Length)) {

      //println("En el bucle, y X vale:", X);
      .nth(X, FOVObjects, Object);
      // Object structure
      // [#, TEAM, TYPE, ANGLE, DISTANCE, HEALTH, POSITION ]
      .nth(2, Object, Type);
      if (Type < 1000) {
        // Object may be an enemy
        .nth(1, Object, Team);
        ?my_formattedTeam(MyTeam);
        if (Team == 200) { // Only if I'm AXIS
          if (Mode<=2) {
            .println("Seeing a dude. . .");
          }
          +targeted("true");
          +targeted_agent(Object);
        }
      }
      --bucle(X+1);
    }
    if (targeted(Objective) & Objective == "true") {
      ?targeted_agent(Agente);
      .nth(6, Agente, NewDestination);
      check_position(NewDestination);
      if(position(valid)){
        !add_task(task(9999,"TASK_GOTO_POSITION",MyName,NewDestination,""));
      }
      --state(standing);
    }
  }
}
-targeted(_);
-targeted_agent(_);
-bucle(_)
```