



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

AGENTES INTELIGENTES

Memoria de la práctica 1

Grado en Ingeniería Informática

Escuela Técnica Superior de Ingeniería Informática

Curso 2020-21



Fabián Scherle Carboneres

Lishuang Sun (María)

ÍNDICE

1. Estrategia -----	3
2. Agente deambulando -----	3
3. Amigos cerca de mí: disparando a amigos -----	4
4. Necesito paquetes -----	5

1. Estrategia

Nuestra estrategia consiste en que el agente deambule por el mapa en busca de soldados amigos, dispararles si los encuentra. Sin embargo, si la vida del amigo que ha visto se encuentra por debajo de la mitad debe ir a su posición.

Por un lado, si la vida de nuestro agente se ve reducida a casi por la mitad, debe ir a la base a por medicinas y por el camino si encuentra a más amigos debe dispararles sin acercarse a ellos. Por otro lado, si se queda sin munición también debe ir a la base a por municiones, pero ignorando a cualquier amigo que vea en el camino. Es posible que estén disparando a nuestro agente cuando vaya hacia la base a por paquetes.

2. Agente deambulando

```
// 200 -> indicador de que es del equipo "Eje"
+flag(F): team(200)
<-
?position(POS);
.create_control_points(POS,100,1,C);
.nth(0,C,A); // Guarda en A el elemento de la posición 0 en la lista C
.goto(A).
```

Nuestro agente al comienzo verifica si es del equipo Eje y se ha añadido una bandera, entonces guarda en POS su posición actual y crea 1 punto de control aleatorio a una distancia de 100 de la ubicación POS y dicho punto se guarda en C. Nuestro agente se dirige a ese punto de control.

```
+target_reached(T): goingpack
<-
?base(X);
-goingpack;
.create_control_points(X,8,1,C);
.nth(0,C,A); // Guarda en A el elemento de la posición 0 en la lista C
.goto(A);
.print("Looking for packs!").
+target_reached(T): outofammo | needhealing
<-
?base(X);
.create_control_points(X,8,1,C);
.nth(0,C,A); // Guarda en A el elemento de la posición 0 en la lista C
.goto(A);
.print("Looking for packs!").
+target_reached(T): shooting
<-
-shooting;
?position(POS);
.create_control_points(POS,100,1,C);
.nth(0,C,A); // Guarda en A el elemento de la posición 0 en la lista C
.goto(A);
.print("Enemy lost, going to: ", A).
+target_reached(T)
<-
?position(POS);
.create_control_points(POS,100,1,C);
.nth(0,C,A); // Guarda en A el elemento de la posición 0 en la lista C
.goto(A).
```

Al llegar a su destino pueden darse los siguientes casos:

- Había visto un paquete (munición o medicina), iba a por dicho paquete y por alguna razón no tomó el paquete. En este caso debe seguir buscando paquetes en la base.
- Ha llegado a la base en busca de paquetes y no ha visto ninguno, como en el anterior caso debe seguir buscando.
- Estaba disparando a un agente que vio y llegó a la posición de dicho agente, por lo que debe seguir deambulando.
- Estaba deambulando, debe seguir deambulando.

3. Amigos cerca de mí: disparando a amigos

```
+friends_in_fov(ID,Type,Angle,Distance,Health,Position):  
  not (outofammo | needhealing) & not shooting & Health <= 50  
  <-  
  +shooting;  
  .goto(Position);  
  .shoot(3,Position).  
+friends_in_fov(ID,Type,Angle,Distance,Health,Position):  
  not outofammo  
  <-  
  .shoot(3,Position).
```

Mientras nuestro agente esté deambulando por el mapa, verá a otros agentes de su equipo. Entonces, mientras no se quede sin munición, estará disparando a los agentes Eje que vea y a su vez estará yendo a la posición del primer agente que vio con menos de la mitad de vida, además, mientras esté yendo a esa posición (shooting), disparará 3 balas hacia la posición de los agentes que vea. Sin embargo, en caso de que necesite curarse solo disparará si ve a un enemigo sin ir a la posición del enemigo puesto que su prioridad es ir por paquetes de medicinas.

4. Necesito paquetes

```
+outofammo
<-
?base(X);
.goto(X);
.print("Going to base for ammo").
+needhealing
<-
?base(X);
.goto(X);
.print("Going to base for healing").

+ammo(Y): not outofammo & Y < 1
<-
+outofammo;
.print("Run out of ammo").
+health(X): X < 60 & not needhealing
<-
+needhealing;
.print("Need healing!").
```

Nuestro agente puede estar en peligro si se ha quedado sin munición (outofammo) y/o su vida baja de cierto umbral (needhealing), para ello hemos considerado que el agente debe ir a la base a curarse y recargar su munición.

```
+pack_taken(Type, N): Type == fieldops & outofammo & goingpack
<-
.print("Ammo taken");
-goingpack;
-outofammo.
+pack_taken(Type, N): Type == fieldops & outofammo
<-
.print("Ammo taken");
-outofammo.
```

Si ha recogido algún paquete dejará de estar en la necesidad de seguir buscando. No hemos considerado revisar si la vida ha superado el umbral después de recoger dicho paquete, puesto que en situaciones críticas no habrá suficientes paquetes para que el agente reestablezca su vida por encima del umbral.

```
+packs_in_fov(ID,Type,Angle,Distance,Health,Position):
Type == 1001 & needhealing & not goingpack
<-
+goingpack;
.goto(Position);
.print("Medic Pack seen").
+packs_in_fov(ID,Type,Angle,Distance,Health,Position):
Type == 1002 & outofammo & not goingpack
<-
+goingpack;
.goto(Position);
.print("Ammo Pack seen").
```

El agente puede ver paquetes en su campo de visión por lo que debe ir a la posición de dichos paquetes para poder recogerlos.