

Estrategia de un Agente Inteligente en PYGOMAS

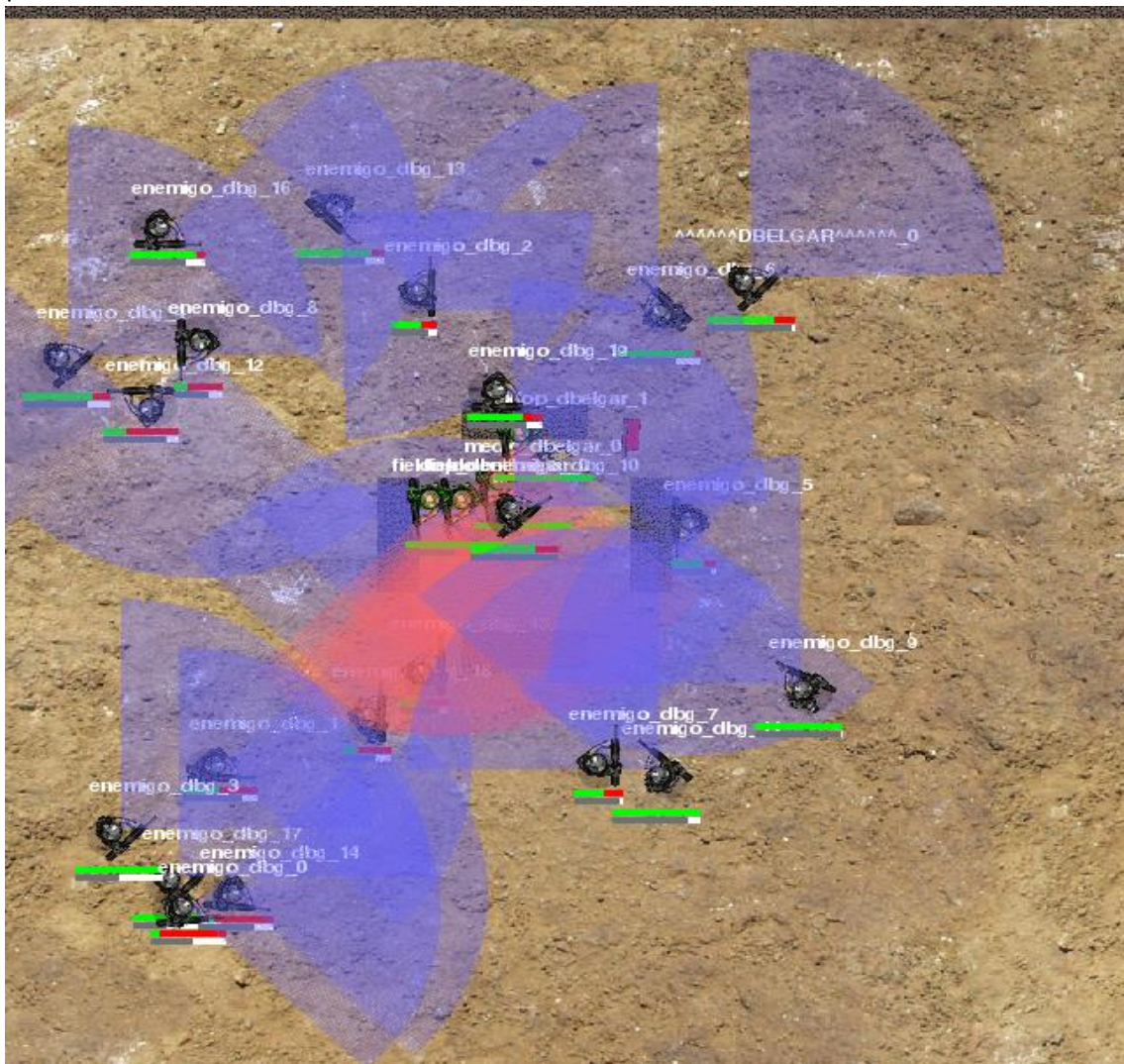
David Beltran Garceran

Grupo: L1-3CO21

En este documento se presenta la estrategia de un agente implementada en .asl/ para PYGOMAS. Para ello, se usan imágenes a modo de ejemplo.

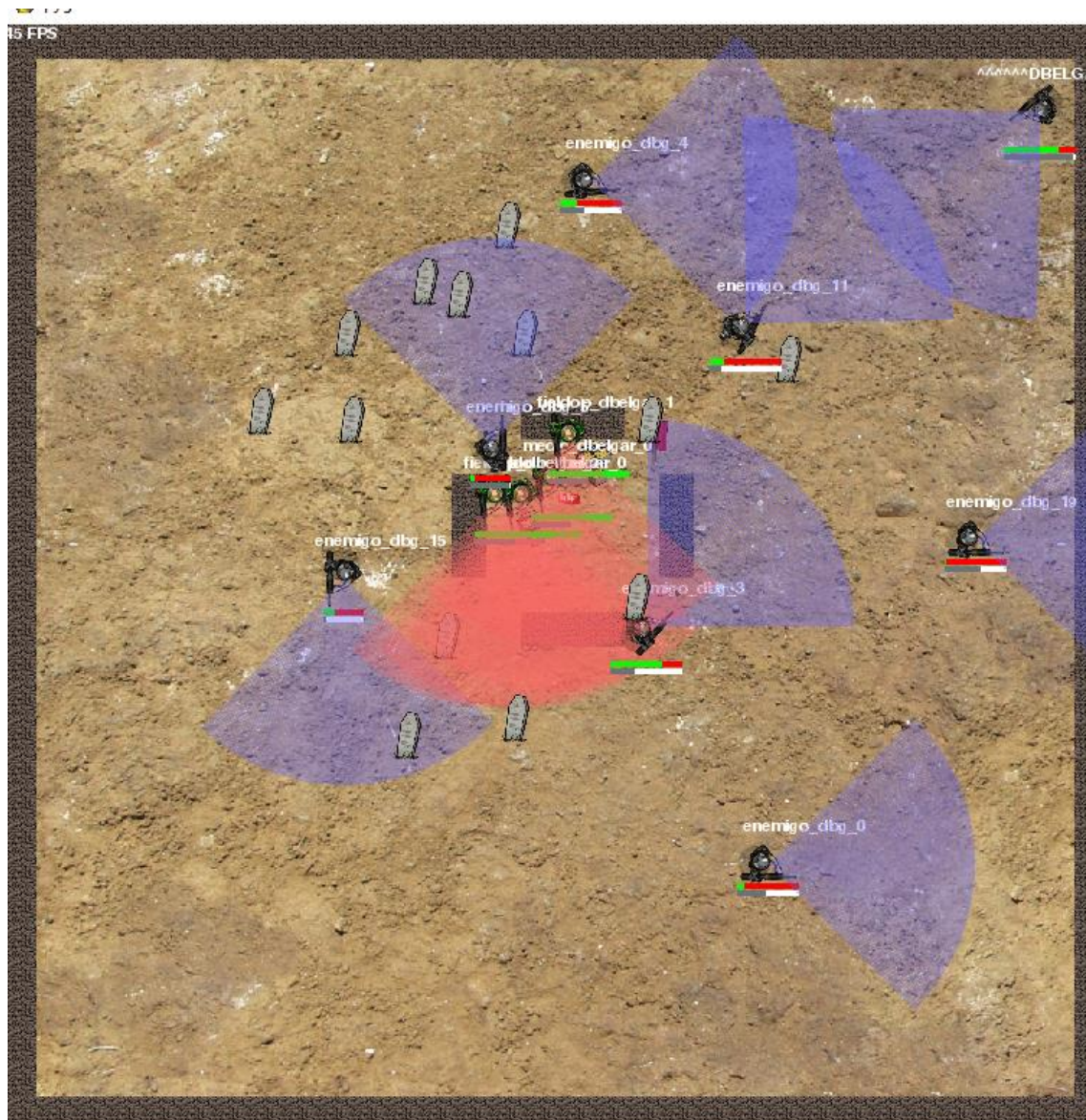
Estrategia:

El agente se dirige a la esquina más cercana (en este caso arriba a la derecha). Si se encontrase con enemigos dispararía sin dejar de dirigirse a la esquina. Podría darse el caso de que se encontrase con varios enemigos y no lograra sobrevivir, sin embargo este caso no es tan probable.



(Agente ^^DBELGAR^^ arriba a la derecha dirigiéndose a la esquina más cercana)

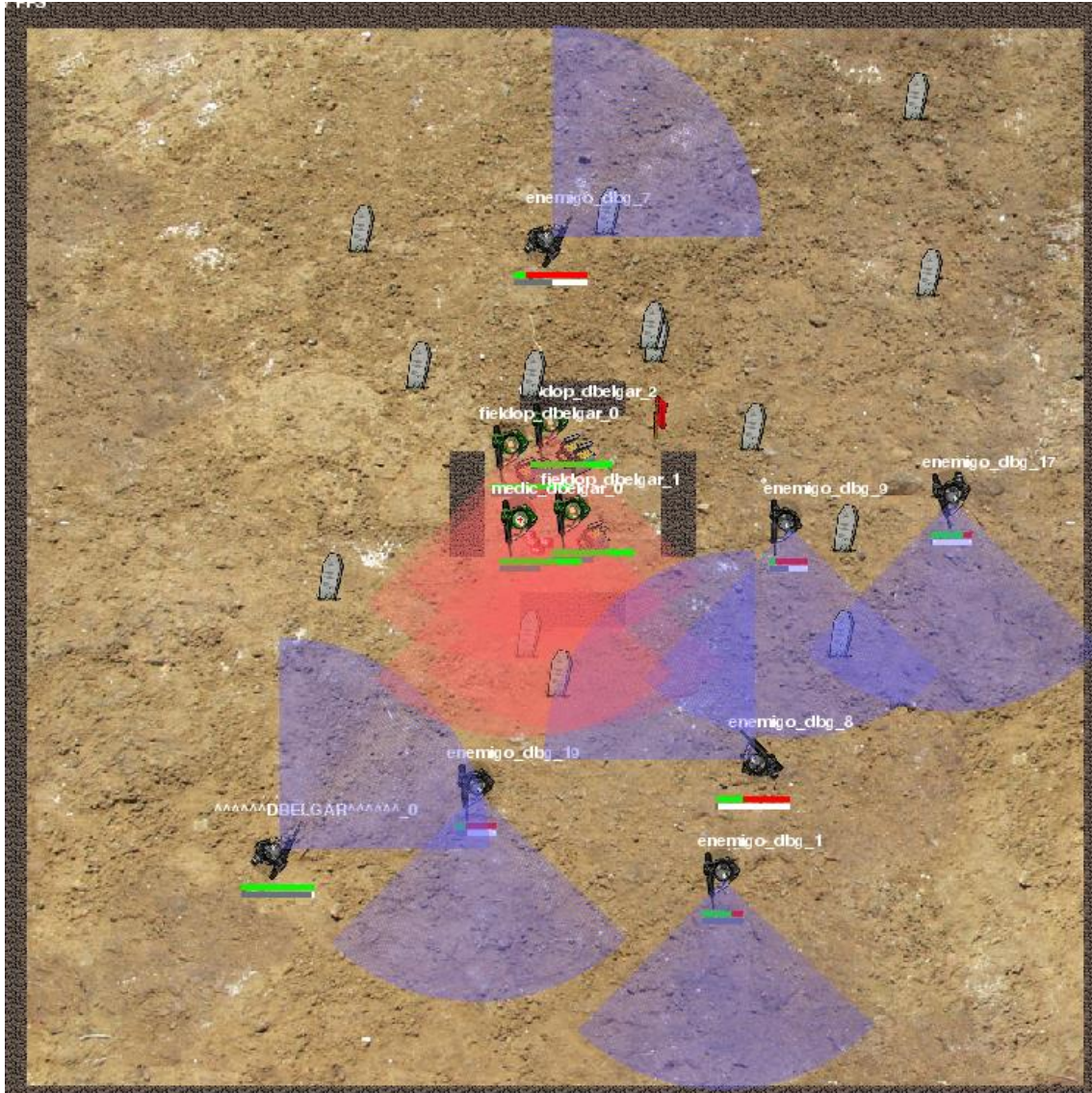
Podemos observar que el agente ya ha llegado a la esquina más cercana, desde ahí controlará si se acerca algún enemigo. En el caso de que se quedase sin munición, el agente se dirigiría al centro a reponerla.



(Agente ^^DBELGAR^^ arriba a la derecha jugando defensivo)

Transcurrido un minuto, el agente se moverá al centro del mapa para tratar de eliminar los enemigos restantes patrullando alrededor de unos puntos de control aleatorios.

Si se quedara con poca vida, se dirigiría de vuelta a la esquina para jugar defensivo.



(Agente ^^DBELGAR^^ abajo a la izquierda dirigiéndose al centro a patrullar)

ANEXO

```
+flag(F): team(200)
    <-
    .print("A POR ELLOS!! -.-");
    +vida(85);
    +ir_a;//ve a la esquina mas cercana
    .wait(10000);//espera 10 segundos
    .create_control_points(F,100,3,C);
    .length(C,L);
    +control_points(C);
    +total_control_points(L);
    +patrolling;//Patrullar en busca de enemigos
```

```
//-----IR A LA ESQUINA MAS CERCANA-----
```

```
//Esquina DownR
```

```
+ir_a: position([X,Y,Z]) & X >= 128 & Z >= 128 & not(centro)
```

```
<-
```

```
-ir_a;
```

```
+esquina;
```

```
.print("Coordenadas! [240, 0, 240]");
```

```
.goto([240,0,240]).//muevete a la esquina
```

```
//Esquina DownL
```

```
+ir_a: position([X,Y,Z]) & X >= 128 & Z <= 128 & not(centro)...
```

```
//Esquina UpR
```

```
+ir_a: position([X,Y,Z]) & X <= 128 & Z >= 128 & not(centro)...
```

```
//Esquina UPL
```

```
+ir_a: position([X,Y,Z]) & X <= 128 & Z <= 128 & not(centro)...
```

```
//Juega Defensivo si tiene poca vida
```

```
+health(X): vida(Y) & Y > X & X <= 45
```

```
<-
```

```
-patrolling;
```

```
-vida(Y);
```

```
+vida(X);
```

```
.print("POCA VIDA!!! HUYENDOOO! X(");
```

```
.look_at([128, 0, 128]);//mira al centro
```

```
+ir_a;//vuelve a la esquina
```

```
//Si tiene poca munición y esta en la esquina, ve a reponer
```

```
+ammo(A): A <= 20 & esquina
```

```
<-
```

```
.wait(20000); //espera 20s
```

```
.print("A POR MUNICION!!! (◡_◡)");
```

```
.goto([128,0,128]); //ve al centro
```

```
+centro.
```

```
//Ha Llegado a la esquina
```

```
+target_reached(T): esquina
```

```
<-
```

```
-ir_a;
```

```
-esquina;
```

```
.look_at([128,0,128]); //mira al centro
```

```
.print("Campeando (-- . --)");
```

```
.wait(60000); //espera 1min
```

```
+patrullar.
```

```
//Patrullar en busca de enemigos
```

```
+patrullar: position([X,Y,Z]) & esquina
```

```
<-
```

```
.goto([128, 0, 128]); //dirigete al centro
```

```
.print("Buscando enemigos! Al centro! (--...--)").
```

```
//Patrullar en busca de enemigos
```

```
+patrullar: position([X,Y,Z]) & not(esquina)
```

```
<-
```

```
.print("Buscando enemigos! (--...--)");
```

```
+patroll_point(0);
```

```
+patrolling.
```