



Cómo crear un contador de balas

Se declara un predicado dinámico para llevar cuenta de las balas que quedan después de cada disparo. El número de balas se mostrará en el argumento del predicado cuyo valor cambiará en tiempo de ejecución, cada vez que se dispare a un enemigo.

Para resolver el mapa: **plman/maps/fase3/mapa5.pl** definimos el predicado dinámico “balas/1”

`:- dynamic balas/1.`

`% La pistola tiene 3 balas
balas(3).`

```
disparo :-      balas(N),  
               M is N-1,  
               retractall(balas(_)),  
               assert(balas(M)),  
               write("Me quedan  "), write(M), tab(2), writeln(' balas').
```

`% El predicado “disparo/0” se encargará de actualizar el argumento del predicado “balas/1”.`

`/*Explico:`

`Al comenzar la ejecución, el argumento del predicado balas/1 tiene el valor N=3;`

`M = 3-1 = 2,`

`retractall/1 eliminará del código el hecho balas(3).`

`assert/1 insertará en el código el hecho balas(M) (calculado antes)`

`Cuando vuelvas a disparar la pistola tendrá M balas.`

`*/`

`% nos interesa disparar mientras tengamos balas y parar en cuanto el número de balas sea cero.`

`% Declaramos reglas para controlar si se puede disparar`

`% regla para disparar si la pistola tiene balas (N>0) y el enemigo está, por ejemplo, a la derecha.
do(use(right)) :- havingObject, balas(N), N>0, see(normal, right, 'E'), disparo.`

`% regla que tendrá éxito cuando ya no queden balas. Plman se mueve, por ej., a la derecha porque ve un coco.`

`do(move(right)) :- havingObject, balas(0), see(normal, right, '.').`