



Predicado see/1

Plman puede condicionar sus acciones dependiendo de lo que está viendo.

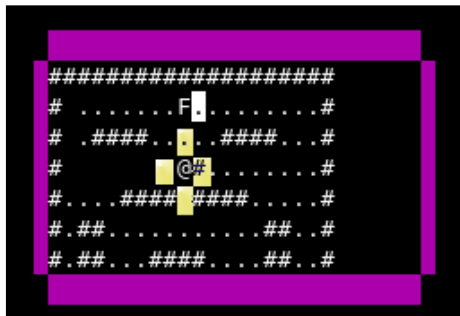
Predicado de visión que funciona como un sensor: **see/3**

Sintaxis: **see(Sensor, Dir, Objeto).**

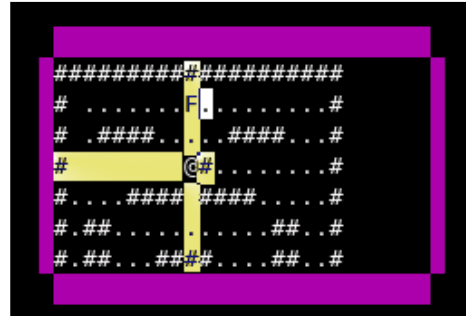
- **Sensor:** argumento que restringe la visión de Plman de tal forma que si Sensor es:
 - **normal:** Plman ve en la casilla inmediata a la que se encuentra, según la dirección Dir. Este sensor se puede usar en todas las fases
 - **list:** Plman ve toda la fila/columna hasta el siguiente objeto sólido, según la dirección Dir. Los objetos se guardan en una lista que Prolog almacena. Este sensor **SÓLO** se puede usar en la fase 4
- **Dir:** {up, down, left, right, here, up-left, up-right, down-left, down-right}.
(la visión diagonal admite viceversa, por ejemplo, up-left = left-up)
- **Objeto:** declaración del objeto.

El predicado see/3 se escribe en el cuerpo de la regla. Puedes añadir todos los que creas convenientes para condicionar la acción que debe realizar Plman, pero sin abusar. Los separas con comas.

Ejemplo: en la siguiente Figura tenemos marcado lo que Plman **ve** según el tipo de sensor:



Sensor normal : see(normal, _, _)



Sensor de lista : see(list, _, _)

El sensor **normal** permite ver el contenido de las 8 celdas que Plman tiene a su alrededor y el de la celda donde se encuentra, por ejemplo, lo que Plman ve en la casilla superior, es decir, en la dirección "up", según el predicado see/3, se implementa como, see(normal, up, X), y el valor de X será X = '.'

El sensor de **list** (sólo para fase 4) proporciona una lista con todo lo que se puede ver hasta el primer objeto sólido, por ejemplo, lo que Plman ve hacia arriba, en la dirección "up", se almacena en la lista ['.', 'F', '#']. Para tener acceso a los elementos de una lista, en el tercer argumento del predicado see/3, se declara una variable en la que se almacenarán estos elementos. Si usamos la variable L, escribimos see(list, up, L), y en este caso el valor de L será, L = ['.', 'F', '#']



Más ejemplos del predicado `see/3` para cada mapa:

?- `see(normal, up, X)`.
`X = '.'`

?- `see(normal, right, X)`.
`X = '#'`

?- `see(normal, left, '.')`.
No

?- `see(normal, down, ' ')`.
Yes

?- `see(list, up, X)`.
`X = ['.', 'F', '#']`

?- `see(list, right, ['#'])`.
Yes

?- `see(list, left, [' '])`.
No

?- `see(list, down, X)`.
`X = [' ', '.', '#']`