



## Restricciones para programar cada fase

### Fase 0

Los mapas de fase 0 son mapas preparados para que el jugador pueda iniciarse en la resolución de mapas y al funcionamiento de Plman. El jugador se familiariza con el uso de la regla **do/1** y con las posibles acciones que puede realizar Plman

En esta fase se pueden usar los predicados **do/1** y **see/3**, pero se recomienda usar sólo **do/1**.

### Fase 1

Mapas de tipo determinista en los que el comportamiento de los objetos y enemigos siempre es el mismo en cada ejecución, es decir, si en el mapa hubiera un enemigo que se mueve éste siempre realizará el mismo movimiento cada vez que se ejecute Plman.

Se aprende a usar la regla **see/3** que permite a Plman recorrer el mapa condicionando el movimiento según lo que ve.

Los mapas de esta fase se deben resolver fácilmente usando sólo los predicados **see/3** y **do/1**.

### Fase 2

Los mapas de la Fase 2 son mapas deterministas, de dificultad media. Plman podrá realizar acciones condicionadas no sólo a lo que ve, sino a lo que lleva. Para ello se introducen las variables y el predicado **havingObject/0** y **havingObject/1**.

Se utilizan subreglas para modular el código.

Además se permiten los predicados **do/1** y **see/3**.

### Fase 3 y Fase 4

Aparece indeterminismo en los mapas. Los enemigos y objetos móviles pueden tener comportamientos diferentes y aparecer en lugares aleatorios en cada ejecución. Se introducen los predicados dinámicos y se siguen utilizando subreglas. En todos los mapas hay algún componente indeterminista.

- Estudiamos los diferentes comportamientos de Plman en el mapa de tal forma que para cada uno definimos sub-reglas con diferentes acciones.
- Cada conjunto de sub-reglas conformará un estado.
- Cuando Plman cambie de comportamiento se cambiará de estado.
- El estado se declara en el programa mediante un hecho con predicado dinámico con el fin de que su/s argumento/s pueda variar durante la ejecución.

### ¿Cuántos estados se pueden/deben usar para resolver un mapa?

Tantos como se estimen oportunos según los diferentes comportamientos de Plman.

En cada momento de la resolución se debe saber el estado en que se encuentra Plman.



Tus soluciones pueden utilizar cualquier técnica de programación en lenguaje Prolog, respetando las siguientes restricciones:

- Debes seguir las normas del juego PLMan.
- No debes utilizar estrategias de hacking para alterar el funcionamiento del juego.
- Los predicados dinámicos sólo están permitidos en fases 3 y 4.
- Las listas sólo están permitidas en fase 4.

Cualquier solución que no siga estas normas será anulada. Si, además, la solución utilizaba mecanismos poco éticos (como hacking), podría haber acciones disciplinarias posteriores.