

MINECRAFT

Minecraft es un juego de construcción abierto, que se compone de objetos tridimensionales cúbicos. Se centra en la creación y destrucción de estos bloques, que representan principalmente distintos elementos de la naturaleza, como tierra, piedra, minerales, troncos de un árbol, entre otros. Los jugadores pueden realizar construcciones mediante estos cubos, pueden explorar el entorno, recolectar recursos y crear objetos con distintas utilidades, combatir monstruos u otros jugadores. Son libres de desplazarse por su entorno y modificarlo mediante la creación, recolección y transporte de los bloques que componen al juego.

Se tiene la siguiente base de conocimiento en cuanto a los jugadores, con información de cada uno sobre su nombre, ítems que posee, y su nivel de hambre (0 a 10).

También se tiene información sobre el mapa del juego, particularmente de las distintas secciones del mismo, los jugadores que se encuentran en cada uno, y su nivel de oscuridad (0 a 10).

Por último, se conoce cuáles son los ítems comestibles.

```
jugador(stuart, [piedra, piedra, piedra, piedra, piedra, piedra, piedra, piedra, piedra], 3).
```

```
jugador(tim, [madera, madera, madera, madera, madera, pan, carbon, carbon, carbon, pollo, pollo], 8).
```

```
jugador(steve, [madera, carbon, carbon, diamante, panceta, panceta, panceta], 2).
```

```
lugar(playa, [stuart, tim], 2).
```

```
lugar(mina, [steve], 8).
```

```
lugar(bosque, [], 6).
```

```
comestible(pan).
```

```
comestible(panceta).
```

```
comestible(pollo).
```

```
comestible(pescado).
```



Todos los predicados que se piden a continuación deben ser completamente inversibles.

1) Jugando con los ítems

- a. Relacionar un jugador con un ítem que posee. **tieneItem/2**
- b. Saber si un jugador se preocupa por su salud, esto es si tiene entre sus ítems más de un tipo de comestible. (Tratar de resolver sin findall) **sePreocupaPorSuSalud/1**
- c. Relacionar un jugador con un ítem que existe (un ítem existe si lo tiene alguien), y la cantidad que tiene de ese ítem. Si no posee el ítem, la cantidad es 0. **cantidadDelItem/3**
- d. Relacionar un jugador con un ítem, si de entre todos los jugadores, es el que más cantidad tiene de ese ítem. **tieneMasDe/2**

```
?- tieneMasDe(steve, panceta).  
true.
```

2) Alejarse de la oscuridad

- a. Obtener los lugares en los que hay monstruos. Se sabe que los monstruos aparecen en los lugares cuyo nivel de oscuridad es más de 6. **hayMonstruos/1**
- b. Saber si un jugador corre peligro. Un jugador corre peligro si se encuentra en un lugar donde hay monstruos; o si está hambriento ($\text{hambre} < 4$) y no cuenta con ítems comestibles. **correPeligro/1**
- c. Obtener el nivel de peligrosidad de un lugar, el cual es un número de 0 a 100 y se calcula:
 - Si no hay monstruos, es el porcentaje de hambrientos sobre su población total.
 - Si hay monstruos, es 100.
 - Si el lugar no está poblado, sin importar la presencia de monstruos, es su nivel de oscuridad *

10. **nivelPeligrosidad/2**

```
?- nivelPeligrosidad(playa, Peligrosidad).  
Peligrosidad = 50.
```



3) A construir

El aspecto más popular del juego es la construcción. Se pueden construir nuevos ítems a partir de otros, cada uno tiene ciertos requisitos para poder construirse:

- Puede requerir una cierta cantidad de un ítem simple, que es aquel que el jugador tiene o puede recolectar. Por ejemplo, 8 unidades de piedra.
- Puede requerir un ítem compuesto, que se debe construir a partir de otros (una única unidad).

Con la siguiente información, se pide relacionar un jugador con un ítem que puede construir. **puedeConstruir/2**



```
item(horno, [ itemSimple(piedra, 8) ] ).
item(placaDeMadera, [ itemSimple(madera, 1) ] ).
item(palo, [ itemCompuesto(placaDeMadera) ] ).
item(antorcha, [ itemCompuesto(palo), itemSimple(carbon, 1) ] ).
```

```
?- puedeConstruir(stuart, horno).
true.
?- puedeConstruir(steve, antorcha).
true.
```

Aclaración: Considerar a los componentes de los ítems compuestos y a los ítems simples como excluyentes, es decir no puede haber más de un ítem que requiera el mismo elemento.

4) Para pensar sin bloques

- ¿Qué sucede si se consulta el nivel de peligrosidad del desierto? ¿A qué se debe?
- ¿Cuál es la ventaja que nos ofrece el paradigma lógico frente a funcional a la hora de realizar una consulta?