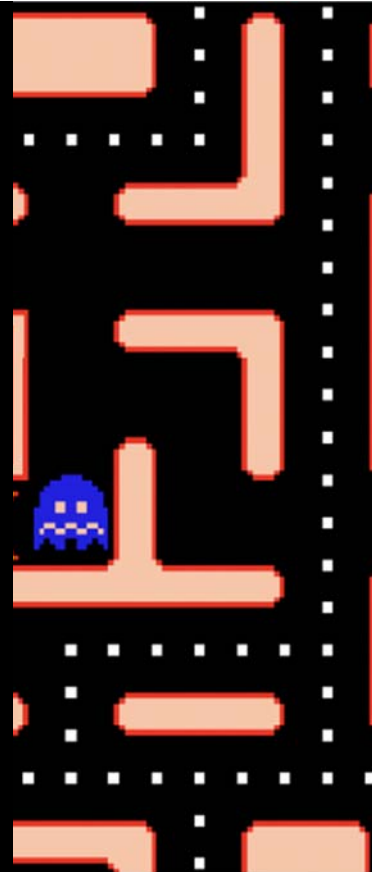


# Ms Pacman

---



## Sensores

- Los valores de los booleanos son ``true`` o ``false``
- Los de las distancias son un número  $\geq 0$  o ``-1`` en caso de que no aplique,
- los nombres de los fantasmas son en mayúsculas (BLINKY, INKY, PINKY, SUE) o ``null`` si no aplica, las posiciones o nodos son números
- enteros  $> 0$  o ``-1`` en caso de que no aplique y los movimientos son las direcciones en inglés y mayúsculas (LEFT, RIGHT, UP o DOWN).
- Las distancias están medidas teniendo en cuenta la distancia real (PATH) y el movimiento previo (solo tiene en cuenta movimientos legales).
- Los sensores de MsPacman están contenidos en `mspacmanrules.clp`. Son de los tipos INTEGER (para números enteros) y SYMBOL (para el resto).
- Hay 3 definiciones de templates: MSPACMAN, GHOSTS, INFO

# Templates para MsPacman

La interfaz del juego aserta Facts en la memoria de trabajo. Esa es la única información con la que cuento para tomar decisiones. Para MsPacman (MsPacman.clp) la información acerca del tick actual del juego se proporciona a través de hechos con la siguiente estructura:

```
(deftemplate MSPACMAN
(slot msPacManNode (type INTEGER))
(slot msPacManMove (type SYMBOL))
)
```

```
(deftemplate GHOSTS
(slot anyChasing (type SYMBOL))
(slot numberOfChasing (type INTEGER))
(slot closestChasing (type SYMBOL))
(slot distanceClosestChasing (type INTEGER))
(slot secondChasing (type SYMBOL))
(slot distanceSecondChasing (type INTEGER))
(slot thirdChasing (type SYMBOL))
(slot distanceThirdChasing (type INTEGER))
(slot fourthChasing (type SYMBOL))
(slot distanceFourthChasing (type INTEGER))
(slot anyEdible (type SYMBOL))
(slot allEdible (type SYMBOL))
(slot closestEdible (type SYMBOL))
(slot distanceClosestEdible (type INTEGER)))
```

```
(deftemplate INFO
(slot nearestPill (type INTEGER))
(slot distanceNearestPill (type INTEGER))
(slot nearestPPill (type INTEGER))
(slot distanceNearestPPill (type INTEGER))
(slot numberOfLives (type INTEGER))
(slot timeLeft (type INTEGER))
(slot level (type INTEGER))
(slot score (type INTEGER))
(slot numberPillsLeft (type INTEGER))
(slot numberPPillsLeft (type INTEGER))
)
```

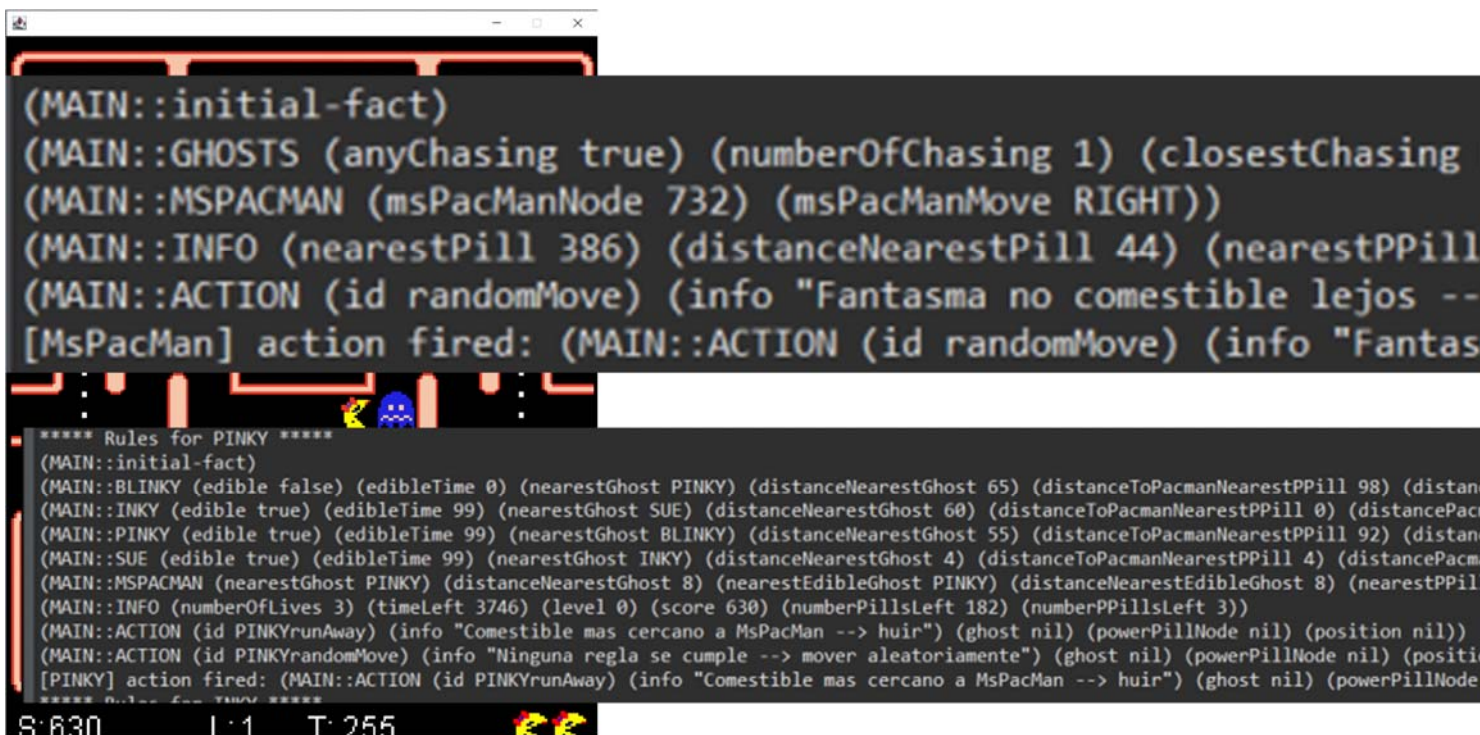
MSPACMAN: Contiene 2 datos acerca de MsPacMan en el tick actual de la partida.

1. `msPacManNode`: Entero que contiene la ubicación de MsPacMan en el tablero (no vamos a trabajar con estas posiciones)
2. `msPacManMove`: Último movimiento realizado por MsPacMan (LEFT, RIGHT, UP o DOWN)

GHOSTS: Contiene 14 datos acerca de los fantasmas en el tick actual de la partida.

1. `anyChasing`: Booleano que indica si hay algún fantasma no comestible fuera del cuadrado central.
2. `numberOfChasing`: Número que indica la cantidad de fantasmas no comestibles fuera del cuadrado central.
3. `closestChasing`: Nombre del fantasma no comestible más cercano a MsPacMan.
4. `distanceClosestChasing`: Distancia que separa a MsPacMan del fantasma `closestChasing` (`-1` en caso de `closestChasing = null`).
5. `secondChasing`: Nombre del segundo fantasma no comestible más cercano a MsPacMan.
6. `distanceSecondChasing`: Distancia que separa a MsPacMan del fantasma `secondChasing`.
7. `thirdChasing`: Nombre del tercer fantasma no comestible más cercano a MsPacMan.
8. `distanceThirdChasing`: Distancia que separa a MsPacMan del fantasma `thirdChasing`.
9. `fourthChasing`: Nombre del cuarto fantasma no comestible más cercano a MsPacMan.
10. `distanceFourthChasing`: Distancia que separa a MsPacMan del fantasma `fourthChasing`.
11. `anyEdible`: Booleano que indica si hay algún fantasma comestible fuera del cuadrado central.
12. `allEdible`: Booleano que indica si todos los fantasmas de fuera del cuadrado central son comestibles.
13. `closestEdible`: Nombre del fantasma comestible más cercano a MsPacMan en mayúsculas.
14. `distanceClosestEdible`: Distancia que separa a MsPacMan del fantasma `closestEdible` (`-1` en caso de `closestEdible = null`).

- INFO: Contiene 10 datos adicionales acerca de como está la partida en el tick actual de la partida.
- 1. `nearestPill`: Nodo en el que se encuentra la pill más cercana a MsPacMan.
- 2. `distanceNearestPill`: Distancia que separa a MsPacMan de la `nearestPill`.
- 3. `nearestPPill`: Nodo en el que se encuentra la powerpill más cercana a MsPacMan.
- 4. `distanceNearestPPill`: Distancia que separa a MsPacMan de la `nearestPPill` (`-1` en caso de `nearestPPill = -1`).
- 5. `numberOfLives`: Número de vidas que le quedan a MsPacMan para perder.
- 6. `timeLeft`: Tiempo que le queda a MsPacMan para perder por tiempo (empieza por 4000 ticks).
- 7. `level`: Número del nivel en el que se encuentra la partida (empieza por 0).
- 8. `score`: Puntuación que ha conseguido MsPacMan en la partida.
- 9. `numberPillsLeft`: Número de pills que quedan en el nivel actual.
- 10. `numberPPillsLeft`: Número de powerpills que quedan en el nivel actual.



The image shows a Pac-Man game window with a terminal window overlaid. The terminal displays the game state and logic for the PINKY ghost.

```

(MAIN::initial-fact)
(MAIN::GHOSTS (anyChasing true) (numberOfChasing 1) (closestChasing
(MAIN::MSPACMAN (msPacManNode 732) (msPacManMove RIGHT))
(MAIN::INFO (nearestPill 386) (distanceNearestPill 44) (nearestPPill
(MAIN::ACTION (id randomMove) (info "Fantasma no comestible lejos --
[MsPacMan] action fired: (MAIN::ACTION (id randomMove) (info "Fantas

***** Rules for PINKY *****
(MAIN::initial-fact)
(MAIN::BLINKY (edible false) (edibleTime 0) (nearestGhost PINKY) (distanceNearestGhost 65) (distanceToPacmanNearestPPill 98) (distan
(MAIN::INKY (edible true) (edibleTime 99) (nearestGhost SUE) (distanceNearestGhost 60) (distanceToPacmanNearestPPill 0) (distancePac
(MAIN::PINKY (edible true) (edibleTime 99) (nearestGhost BLINKY) (distanceNearestGhost 55) (distanceToPacmanNearestPPill 92) (distan
(MAIN::SUE (edible true) (edibleTime 99) (nearestGhost INKY) (distanceNearestGhost 4) (distanceToPacmanNearestPPill 4) (distancePacm
(MAIN::MSPACMAN (nearestGhost PINKY) (distanceNearestGhost 8) (nearestEdibleGhost PINKY) (distanceNearestEdibleGhost 8) (nearestPPil
(MAIN::INFO (numberOfLives 3) (timeLeft 3746) (level 0) (score 630) (numberPillsLeft 182) (numberPPillsLeft 3))
(MAIN::ACTION (id PINKYrunAway) (info "Comestible mas cercano a MsPacMan --> huir") (ghost nil) (powerPillNode nil) (position nil))
[MAIN] action fired: (MAIN::ACTION (id PINKYrunAway) (info "Comestible mas cercano a MsPacMan --> huir") (ghost nil) (powerPillNode

```

At the bottom of the terminal, the game state is summarized:

```

S: 630    L: 1    T: 255

```

# Templates para los fantasmas

- Están contenidos en Xrules.clp siendo X => blinky, inky, pinky o sue (en minúsculas). Los hechos sobre el juego se asertan con hechos de los siguientes templates:

```
(deftemplate BLINKY ;iguales para INKY, PINKY, SUE
  (slot edible (type SYMBOL))
  (slot edibleTime (type INTEGER))
  (slot nearestGhost (type SYMBOL))
  (slot distanceNearestGhost (type INTEGER))
  (slot distanceToPacmanNearestPPill (type INTEGER))
  (slot distancePacman (type INTEGER))
)
```

```
(deftemplate MSPACMAN
  (slot nearestGhost (type SYMBOL))
  (slot distanceNearestGhost (type INTEGER))
  (slot nearestEdibleGhost (type SYMBOL))
  (slot distanceNearestEdibleGhost (type INTEGER))
  (slot nearestPPill (type INTEGER))
  (slot distanceNearestPPill (type INTEGER)))
```

```
(deftemplate INFO
  (slot numberOfLives (type INTEGER))
  (slot timeLeft (type INTEGER))
  (slot level (type INTEGER))
  (slot score (type INTEGER))
  (slot numberPillsLeft (type INTEGER))
  (slot numberPPillsLeft (type INTEGER)))
```

## Templates para BLINKY-INKY-PINKY-SUE:

Contienen 6 datos, cada uno aplicado a uno de los 4 fantasmas (24 datos en total).

- 1. `edible`: Booleano que indica si el fantasma es comestible.
- 2. `edibleTime`: Número que indica el tiempo que le queda al fantasma para dejar de ser comestible. Si no es comestible es 0.
- 3. `nearestGhost`: Nombre del fantasma más cercano al fantasma X (`null` si está dentro del cuadrado central o es el único que está fuera).
- 4. `distanceNearestGhost`: Distancia que separa al fantasma X del fantasma `nearestGhost` (`-1` en caso de `nearestGhost = null` o esté en el cuadrado central).
- 5. `distanceToPacmanNearestPPill`: Distancia que separa al fantasma X de la powerpill más cercana a MsPacMan (`-1` en caso de que esté en el cuadrado central o no queden powerpills).
- 6. `distancePacman`: Distancia que separa al fantasma X de MsPacMan (`-1` en caso de que esté en el cuadrado central).

- MSPACMAN: Contiene 6 datos acerca de la relación de MsPacMan con los fantasmas en el tick actual de la partida.
- 1. `nearestGhost`: Nombre del fantasma más cercano a MsPacMan (`null` si todos están en el cuadrado central).
- 2. `distanceNearestGhost`: Distancia que separa a MsPacMan del fantasma `nearestGhost` (`-1` si `nearestGhost = null`).
- 3. `nearestEdibleGhost`: Nombre del fantasma comestible más cercano a MsPacMan (`null` si todos están en el cuadrado central o no hay comestibles).
- 4. `distanceNearestEdibleGhost`: Distancia que separa a MsPacMan del fantasma `nearestEdibleGhost` (`-1` si `nearestEdibleGhost = null`).
- 5. `nearestPPill`: Nodo en el que se encuentra la powerpill más cercana a MsPacMan. (`-1` si no quedan powerpills).
- 6. `distanceNearestPPill`: Distancia que separa a MsPacMan de la `nearestPPill` (`-1` si `nearestPPill = -1`).

- - INFO: Contiene 6 datos adicionales acerca de como está la partida en el tick actual de la partida.
- 1. `numberOfLives`: Número de vidas que le quedan a MsPacMan para perder.
- 2. `timeLeft`: Tiempo que le queda a MsPacMan para perder por tiempo (empieza por 4000 ticks).
- 3. `level`: Número del nivel en el que se encuentra la partida (empieza por 0).
- 4. `score`: Puntuación que ha conseguido MsPacMan en la partida.
- 5. `numberPillsLeft`: Número de pills que quedan en el nivel actual.
- 6. `numberPPillsLeft`: Número de powerpills que quedan en el nivel actual.

# Actuadores para MsPacMan

```
;DEFINITION OF THE ACTION FACT
(deftemplate ACTION
  (slot id)
  (slot info (default ""))
  (slot pillNode (type INTEGER))
  (slot powerPillNode (type INTEGER))
  (slot ghost (type SYMBOL))
  (slot closestChasing (type SYMBOL))
  (slot secondChasing (type SYMBOL))
)
```

- Están contenidos en mspacmanrules.clp.
- Vienen definidos por el template ACTION, que tiene 7 slots:
- 1. `id`: Es el más importante, es el nombre que hace que el sistema detecte la acción que se quiere realizar. Hay 6 posibles ids => `randomMove`, `runAway`, `survive`, `goToPill`, `goToPowerPill`, `chaseGhost`.
- 2. `info`: Cadena opcional que describe la acción, muy útil a la hora de debuggear.
- 3. `pillNode`: Entero que indica la posición de la pill hacia la que se quiere ir al ejecutar la acción `goToPill`.
- 4. `powerPillNode`: Entero que indica la posición de la powerpill hacia la que se quiere ir al ejecutar la acción `goToPowerPill`.
- 5. `ghost`: Nombre del fantasma del que se quiere huir o perseguir al ejecutar las acciones `runAway` y `chaseGhost` respectivamente.
- 6. `closestChasing`: Nombre del fantasma del que se quiere huir más urgentemente al ejecutar la acción `survive`.
- 7. `secondChasing`: Nombre del fantasma del que se quiere huir menos urgentemente al ejecutar la acción `survive`.

En los Action facts que se pueden asertar al sistema para realizar acciones rellenar el slot `id` es obligatorio para todas las acciones, el de `info` es recomendable, y los demás son obligatorios o no dependiendo de la acción a realizar como se describe a continuación.

## ACTION IDS (relleno para el slot ID) PARA MSPACMAN

- **randomMove**
  - Descripción => Movimiento aleatorio
  - Extra Facts necesarios => ninguno
  - MsPacMan ejecutará un movimiento aleatorio de entre los posibles por la dirección en la que va. No necesita rellenar slots adicionales.
- **runAway**
  - Descripción => Huir de un fantasma dado
  - Extra Facts necesarios => ghost
  - MsPacMan ejecutará el movimiento posible que más la aleje del fantasma dado en el slot `ghost`.
- **survive**
  - Descripción => Huir de los fantasmas no comestibles mas cercanos
  - Extra Facts necesarios => closestChasing, secondChasing
  - MsPacMan ejecutará el movimiento posible que más la aleje tanto del fantasma dado en el slot `closestChasing` como del dado en el slot `secondChasing`. En caso de que no exista un movimiento que lo aleje de ambos priorizará alejarse del fantasma `closestChasing`.

- **goToPill:**
  - Descripcion => Ir hacia la ubicacion de una pill
  - Extra Facts necesarios => pillNode
  - MsPacMan ejecutará el movimiento posible que más la acerque a la pill cuya posición ha sido indicada en el slot `pillNode`.
- **goToPowerPill:**
  - Descripcion => Ir hacia la ubicacion de una powerPill
  - Extra Facts necesarios => powerPillNode
  - MsPacMan ejecutará el movimiento posible que más la acerque a la powerpill cuya posición ha sido indicada en el slot `powerPillNode`.
- **chaseGhost:**
  - Descripcion => Perseguir a un fantasma sea comestible o no
  - Extra Facts necesarios => ghost
  - MsPacMan ejecutará el movimiento posible que más la acerque al fantasma indicado por el slot `ghost` sin importar si es comestible o no.

## Ejemplos de reglas para MsPacman

```
(defrule MSPACMANrandomMove
  (GHOSTS (allEdible true))
=>
(assert
(ACTION (id randomMove)
(info "Fantasmas no peligrosos --> mover aleatoriamente"))
))
```

```
(defrule MSPACMANrunsAway
  (GHOSTS
    (anyChasing true)
    (closestChasing ?c)
    (distanceClosestChasing ?d))
    (test (<= ?d 50))
=>
(assert
(ACTION (id runAway)
(info "Fantasma no comestible cerca --> huir de el")
(ghost ?c))) )
```

```
(defrule MSPACMANrandomMove
  (GHOSTS (anyChasing true) (distanceClosestChasing ?d))
  (test (> ?d 50))
=>
(assert
(ACTION (id randomMove)
  (info "Fantasma no comestible lejos --> mover aleatoriamente"))))
```



# Actuadores para los Fantasma

- Están en Xrules.clp siendo X => blinky, inky, pinky o sue (en minúsculas). Vienen definidos por el template ACTION, que tiene 5 slots:

```
(deftemplate ACTION
  (slot id)
  (slot info (default ""))
  (slot ghost (type SYMBOL))
  (slot powerPillNode (type INTEGER))
  (slot position (type INTEGER))
)
```

En los Action facts el slot `id` es obligatorio para todas las acciones, el de `info` es recomendable, y los demás son obligatorios o no dependiendo de la acción a realizar como se describe a continuación.

1. `id`: Es el más importante, es el nombre que hace que el sistema detecte la acción que se quiere realizar.
2. Hay 28 posibles ids, es decir, 7 acciones distintas por cada fantasma => `XrandomMove`, `XchasePacman`, `XchaseGhost`, `XmoveToPPill`, `XrunAway`, `XrunAwayFromGhost`, `XrunAwayFromPosition` siendo X => BLINKY, INKY, PINKY o SUE (en mayúsculas).
3. `info`: Cadena opcional que describe la acción, muy útil a la hora de debuguear.
4. `ghost`: Nombre del fantasma del que debe huir o perseguir el fantasma X al ejecutar las acciones `XrunAwayFromGhost` y `XchaseGhost` respectivamente.
5. `powerPillNode`: Entero que indica la posición de la powerpill hacia la que debe ir el fantasma X al ejecutar la acción `XmoveToPPill`.
6. `position`: Entero que indica la posición de la que debe huir el fantasma X al ejecutar la acción `XrunAwayFromPosition`.

## ACTION IDS PARA FANTASMAS

- `XrandomMove`: El fantasma X ejecutará un movimiento aleatorio de entre los posibles por la dirección en la que va. No necesita rellenar slots adicionales.
- `XrunAway`: El fantasma X ejecutará el movimiento posible que más lo aleje de MsPacMan. No necesita rellenar slots adicionales.
- `XrunAwayFromGhost`: El fantasma X ejecutará el movimiento posible que más lo aleje del fantasma dado en el slot `ghost`.
- `XrunAwayFromPosition`: El fantasma X ejecutará el movimiento posible que más lo aleje de la posición indicada en el slot `position`.
- `XmoveToPPill`: El fantasma X ejecutará el movimiento posible que más lo acerque a la powerpill cuya posición ha sido indicada en el slot `powerPillNode`.
- `XchasePacman`: El fantasma X ejecutará el movimiento posible que más lo acerque a MsPacMan. No necesita rellenar slots adicionales.
- `XchaseGhost`: El fantasma X ejecutará el movimiento posible que más lo acerque al fantasma indicado por el slot `ghost` sin importar si es comestible o no.



```

(defrule PINKYchase
  (PINKY (edible false) (distancePacman ?d))
  (MSPACMAN (distanceNearestPPill ?n))
  (test (< ?d 40)) (test (> ?n 80))
  =>
  (assert
   (ACTION (id PINKYchasePacman) (info "MsPacMan vulnerable --> perseguir")))
  )
)

```

```

(defrule PINKYrunsAway
  (PINKY (edible true))
  =>
  (assert
   (ACTION (id PINKYrunAway)
    (info "Comestible --> huir")))
  ))

```

```

(defrule PINKYrandomMove
  (PINKY (edible false))
  =>
  (assert
   (ACTION (id PINKYrandomMove)
    (info "MsPacMan lejos o peligroso -> mover aleatoriamente")))
  )
)

```

```

(defrule BLINKYchase
  (BLINKY (edible false) (distancePacman ?d))
  (MSPACMAN (distanceNearestPPill ?n))
  (test (< ?d 40)) (test (> ?n 80))
  =>
  (assert
   (ACTION (id BLINKYchasePacman) (info "MsPacMan vulnerable -->
perseguir")))
  )
)

```

```

(defrule SUErunsAway
  (SUE (edible true))
  =>
  (assert
   (ACTION (id SUErunAway) (info "Comestible -->
huir")))
  )
)

```