

## Interfaz

se explica con: MAPA

géneros: mapa.

## Operaciones básicas de mapa

**NUEVOMAPA**(*in* largo: nat, *in* alto: nat, *in* inicio: coordenada, *in* llegada: coordenada, *in* fantasmas: conj(coordenada), *in* paredes: conj(coordenada), *in* chocolates: conj(coordenada), *out* m: mapa)  $\rightarrow$  res : bool  
**Pre**  $\equiv$  {true}  
**Post**  $\equiv$  { (res = (inicio  $\neq$  llegada  $\wedge$  todosEnRango(paredes  $\cup$  fantasmas  $\cup$  chocolates  $\cup$  {inicio, llegada}, largo, alto)  $\wedge$  {inicio, llegada}  $\cap$  (fantasmas  $\cup$  paredes) =  $\emptyset$   $\wedge$  disjuntosDeAPares(paredes, fantasmas, chocolates)))  $\wedge_L$  res  $\Rightarrow$   $_L m =_{\text{obs}}$  nuevoMapa(largo, alto, inicio, llegada, paredes, fantasmas, chocolates) ) }  
**Complejidad:**  $O(\text{chocolates} + \text{fantasmas} + \text{paredes})$   
**Descripción:** Genera un nuevo mapa  
**Aliasing:** -

**DISTCONFANTASMASMÁSCERCANO**(*in* m: mapa, *in* posicin: coordenada)  $\rightarrow$  res : nat  
**Pre**  $\equiv$  {true}  
**Post**  $\equiv$  { res = distConFantasmasMásCercano(fantasmas(m), posición) }  
**Complejidad:**  $O(n)$   
**Descripción:** Devuelve la distancia con el fantasma más cercano  
**Aliasing:** -

**ENRANGO**(*in* m: mapa, *in* posicin: coordenada)  $\rightarrow$  res : bool  
**Pre**  $\equiv$  {true}  
**Post**  $\equiv$  { res = enRango(posición, largo(m), alto(m)) }  
**Complejidad:**  $O(n)$   
**Descripción:** Devuelve true si la posición se encuentra en rango  
**Aliasing:** -

**CHOCOLATES**(*in* map: mapa)  $\rightarrow$  res : conj(coordenada)  
**Pre**  $\equiv$  {true}  
**Post**  $\equiv$  { res = chocolates(map) }  
**Complejidad:**  $O(n)$   
**Descripción:** Devuelve el conjunto de chocolates del mapa  
**Aliasing:** -

**PAREDES**(*in* map: mapa)  $\rightarrow$  res : conj(coordenada)  
**Pre**  $\equiv$  {true}  
**Post**  $\equiv$  { res = paredes(map) }  
**Complejidad:**  $O(n)$   
**Descripción:** Devuelve el conjunto de paredes  
**Aliasing:** -

**INICIO**(*in* map: mapa)  $\rightarrow$  res : coordenada  
**Pre**  $\equiv$  {true}  
**Post**  $\equiv$  { res = inicio(map) }  
**Complejidad:**  $O(1)$   
**Descripción:** Devuelve la coordenada de inicio del mapa  
**Aliasing:** -

**LLEGADA**(*in* map: mapa)  $\rightarrow$  res : coordenada  
**Pre**  $\equiv$  {true}  
**Post**  $\equiv$  { res = llegada(map) }  
**Complejidad:**  $O(1)$   
**Descripción:** Devuelve la coordenada de llegada del mapa  
**Aliasing:** -  
**Representación**

mapa se representa con  $mp$

donde  $casillero$  es tupla( $fantasma$ : bool,  $peligrosa$ : bool,  $pared$ : bool,  $inicio$ : bool,  $llegada$ : bool)

donde  $columna$  es array[0... $ancho$ ] de casillero

donde  $mp$  es tupla( $matriz$ : array[0... $alto$ ] de columna,  $chocolates$ : conj(coordenada),  $alto$ : nat,  $ancho$ : nat)

### Invariante de representación

$Rep : mp \rightarrow \text{boolean}$

$$\begin{aligned}
 Rep(e) \equiv \text{True} \iff & (\forall i: \text{nat})(0 \leq i \leq e.alto) \Rightarrow_L ( \\
 & (\forall j: \text{nat})(0 \leq j \leq e.ancho) \Rightarrow_L ( \\
 & \quad (\beta(e.matriz[i][j].inicio) + \\
 & \quad \beta(e.matriz[i][j].llegada) + \\
 & \quad \beta(e.matriz[i][j].pared) + \\
 & \quad \beta(e.matriz[i][j].fantasma) + \\
 & \quad \beta(\langle i, j \rangle \in e.chocolates) = 1) \wedge \\
 & (e.matriz[i][j].peligrosa \Rightarrow_L (\exists n, m: \text{nat})(0 \leq n \leq e.alto \wedge 0 \leq m \leq e.ancho) \wedge_L \\
 & \quad (e.matriz[n][m].fantasma \wedge \text{distancia}(\langle i, j \rangle, \langle n, m \rangle) \leq 3)) \wedge \\
 & ((\forall n, m: \text{nat}) (0 \leq n \leq alto \wedge 0 \leq m \leq ancho) \Rightarrow_L \\
 & \quad ((e.matriz[n][m].inicio \wedge e.matriz[i][j].inicio) \Rightarrow n = i \wedge m = j)) \wedge \\
 & ((\forall n, m: \text{nat}) (0 \leq n \leq alto \wedge 0 \leq m \leq ancho) \Rightarrow_L \\
 & \quad ((e.matriz[n][m].llegada \wedge e.matriz[i][j].llegada) \Rightarrow n = i \wedge m = j))) ) )
 \end{aligned}$$

### Función de abstracción

$Abs : mpe \rightarrow \text{Mapa}$

$\{Rep(e)\}$

$(\forall e : mp) Abs(e) =_{\text{obs}} m: \text{mapa} \mid \text{largo}(m) = e.alto \wedge$   
 $\text{ancho}(m) = e.ancho \wedge$

$$\begin{aligned}
 & \text{chocolates}(m) = e.chocolates \wedge \\
 & (\forall i: \text{nat})(0 \leq i \leq e.alto) \Rightarrow_L ( \\
 & \quad (\forall j: \text{nat})(0 \leq j \leq e.ancho) \Rightarrow_L ( \\
 & \quad \quad e.matriz[i][j].inicio = inicio(m) \wedge \\
 & \quad \quad e.matriz[i][j].llegada = llegada(m) \wedge \\
 & \quad \quad e.matriz[i][j].fantasma \iff \langle i, j \rangle \in \text{fantasmas}(m) \wedge \\
 & \quad \quad e.matriz[i][j].paredes \iff \langle i, j \rangle \in \text{paredes}(m) \wedge \\
 & \quad \quad e.matriz[i][j].peligrosa \iff \text{distConFantasmasMsCercano}(\text{fantasmas}(m), \langle i, j \rangle) \leq 3))
 \end{aligned}$$

## Interfaz

se explica con: PARTIDA

géneros: partida.

## Operaciones básicas de partida

NUEVAPARTIDA(**in**  $m$ : mapa)  $\rightarrow res$  : partida

**Pre**  $\equiv \{\text{true}\}$

**Post**  $\equiv \{res = \text{nuevaPartida}(m)\}$

**Complejidad:**  $O(n)$

**Descripción:** Genera una nueva partida

**Aliasing:** -

MOVER(**in/out**  $p$ : partida, **in**  $d$ : dirección)

**Pre**  $\equiv \{p_0 = p\}$

**Post**  $\equiv \{p = \text{mover}(p_0, d)\}$

**Complejidad:**  $O(n)$

**Descripción:** Mueva la posición del jugador un casillero

**Aliasing:** -

GANÓ?(**in**  $p$ : partida)  $\rightarrow res$  : bool

**Pre**  $\equiv \{\text{true}\}$

**Post**  $\equiv \{res = \text{ganó?}(p)\}$

**Complejidad:**  $O(n)$

**Descripción:** Devuelve true si el jugador gana la partida

**Aliasing:** -

PERDIÓ?(**in**  $p$ : partida)  $\rightarrow res$  : bool

**Pre**  $\equiv \{\text{true}\}$

**Post**  $\equiv \{res = \text{perdió?}(p)\}$

**Complejidad:**  $O(n)$

**Descripción:** Devuelve true si el jugador perdio la partida

**Aliasing:** -

JUGADOR(**in**  $p$ : partida)  $\rightarrow res$  : coordenada

**Pre**  $\equiv \{\text{true}\}$

**Post**  $\equiv \{res = \text{perdió?}(p)\}$

**Complejidad:**  $O(n)$

**Descripción:** Devuelve la posición del jugador

**Aliasing:** -

CANTMOV(**in**  $p$ : partida)  $\rightarrow res$  : nat

**Pre**  $\equiv \{\text{true}\}$

**Post**  $\equiv \{res = \text{cantMov}(p)\}$

**Complejidad:**  $O(n)$

**Descripción:** Devuelve la cantidad de movimientos del jugador

**Aliasing:** -

**Representación**

partida se representa con  $pt$

donde  $pt$  es tupla( $mapa$ : mp,  $jugador$ : coordenada,  $chocolates$ : conj(chocolates),  $cantMov$ : nat,  $inmunidad$ : nat)

## Interfaz

se explica con: FICHÍN

géneros: fichin.

### Operaciones básicas de fichin

NUEVOFICHIN(**in**  $m : \text{mapa}$ )  $\rightarrow res : \text{fichin}$

**Pre**  $\equiv \{\text{true}\}$

**Post**  $\equiv \{res = \text{nuevoFichin}(m)\}$

**Complejidad:**  $O(n)$

**Descripción:** Genera un fichín

**Aliasing:** -

NUEVAPARTIDA(**in/out**  $f : \text{fichin}$ , **in**  $j : \text{jugador}$ )  $\rightarrow res : \text{bool}$

**Pre**  $\equiv \{f_0 = f\}$

**Post**  $\equiv \{res = \neg \text{alguienJugando?}(f) \wedge_L res \Rightarrow_L f = \text{nuevaPartida}(f_0, j)\}$

**Complejidad:**  $O(n)$

**Descripción:** Inicia una nueva partida

**Aliasing:** -

MOVER(**in/out**  $f : \text{fichin}$ , **in**  $d : \text{dirección}$ )  $\rightarrow res : \text{bool}$

**Pre**  $\equiv \{f_0 = f\}$

**Post**  $\equiv \{res = \text{alguienJugando?}(f) \wedge_L res \Rightarrow_L f = \text{mover}(f_0, d)\}$

**Complejidad:**  $O(n)$

**Descripción:** Mueve en la dirección indicada

**Aliasing:** -

VERRANKING(**in**  $f : \text{fichin}$ )  $\rightarrow res : \text{ranking}$

**Pre**  $\equiv \{\text{true}\}$

**Post**  $\equiv \{res = \text{ranking}(f)\}$

**Complejidad:**  $O(n)$

**Descripción:** Devuelve el ranking del fichin

**Aliasing:** -

OBJETIVO(**in**  $f : \text{fichin}$ , **out**  $o : \text{tupla} \langle \text{jugador}, \text{nat} \rangle$ )  $\rightarrow res : \text{res}$

**Pre**  $\equiv \{\text{true}\}$

**Post**  $\equiv \{(res = \text{alguienJugando?}(f) \wedge \text{def?}(\text{jugadorActual}(f), \text{ranking}(f))) \wedge_L res \Rightarrow_L o = \text{objetivo}(f)\}$

**Complejidad:**  $O(n)$

**Descripción:** Devuelve una tupla con el oponente y su puntaje

**Aliasing:** -

**Representación**

partida se representa con  $fch$

donde  $fch$  es  $\text{tupla}(\text{mapa} : \text{mp}, \text{alguienJugando} : \text{bool}, \text{jugadorActual} : \text{string}, \text{partidaActual} : \text{pt}, \text{ranking} : \text{dicc}(\text{string}, \text{nat}))$