

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 {($res = (inicio \neq llegada \wedge todosEnRango(paredes \cup fantasmas \cup chocolates \cup \{inicio, llegada\}, largo, alto) \wedge \{inicio, llegada\} \cap (fantasmas \cup paredes) = \emptyset \wedge disjuntosDeAPares(paredes, fantasmas, chocolates)) \wedge_L res \Rightarrow {}_L m =_{obs} nuevoMapa(largo, alto, inicio, llegada, paredes, fantasmas, chocolates))$ }
Complejidad: $O(chocolates + fantasmas + paredes)$
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: -

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: -

Interfaz

se explica con: FICHÍN

géneros: fichin.

Operaciones básicas de fichin

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

Pre $\equiv \{true\}$

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

Complejidad: $O(n)$

Descripción: Genera un fichín

Aliasing: -

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

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

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

Complejidad: $O(n)$

Descripción: Inicia una nueva partida

Aliasing: -

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

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

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

Complejidad: $O(n)$

Descripción: Mueve en la dirección indicada

Aliasing: -

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

Pre $\equiv \{true\}$

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

Complejidad: $O(n)$

Descripción: Devuelve el ranking del fichin

Aliasing: -

OBJETIVO(**in** f : fichin, **out** o : tupla<jugador, nat>) $\rightarrow res$: res

Pre $\equiv \{true\}$

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

Complejidad: $O(n)$

Descripción: Devuelve una tupla con el oponente y su puntaje

Aliasing: -