

Quell

El juego *Quell* es un juego de acertijos donde el jugador maneja una **burbuja** en un escenario que contiene diversos elementos. Algunos de los elementos que se pueden encontrar en el tablero son bordes, paredes, trampas y dispositivos. Los bordes pueden ser **esquinas**, **ángulos** o **límites** rectos. Las trampas pueden ser **espinas**, **multiespinas** o **espinas rotables**. Los dispositivos pueden ser **interruptores** o **compuertas**. Algunos de estos elementos tienen diferentes atributos, según su naturaleza. Por ejemplo, los bordes, las espinas, las espinas rotables y los interruptores tienen una dirección que indica hacia dónde apunta el elemento. Las compuertas tienen un atributo que indica si están abiertas o cerradas.

En el siguiente tablero podemos encontrar un escenario básico de Quell, donde podemos observar una serie de bordes que conforman un espacio cerrado, algunas paredes, y un par de espinas. Notar que los bordes apuntan hacia adentro de la zona de juego.



La representación puntual en términos de bolitas no es relevante, pues contamos con operaciones primitivas. Para esto se dividieron los elementos en categorías, aquellos que son simples (burbujas, gemas y multiespinas que no tienen atributos) los direccionables (que tienen un atributo de dirección, como los límites, esquinas y ángulos, las espinas, las espinas rotables y los interruptores) y los cerrables (que se abren y cierran, como las compuertas).

Para manipular todos estos elementos se definieron los siguiente tipos:

```
type ElementoSimple is record {
  /* PROPÓSITO: Modela los datos de un elemento que
    posee atributos de dirección.
    INV. REP.
    * tipoDeElemento es Burbuja, Gema o Multiespinas
  */
  field tipoDeElemento // TipoDeElemento
}
```

```
type ElementoDireccional is record {
  /* PROPÓSITO: Modela los datos de un elemento que
    posee atributos de dirección.
    INV. REP.
    * tipoDeElemento es Límite, Ángulo, Esquina, Espina,
    EspinaRotable o Interruptor
  */
  field tipoDeElemento          // TipoDeElemento
  field direcciónALaQueApunta  // Dirección
}
```

```
type ElementoCerrable is record {
  /* PROPÓSITO: Modela los datos de un elemento que
    posee atributos de dirección.
    INV. REP.
    * tipoDeElemento es Compuerta
  */
  field tipoDeElemento // TipoDeElemento
  field abierto        // Booleano (Verdadero = abierto, Falso = cerrado)
}
```

```
type TipoDeElemento is variant {
  /* PROPÓSITO: Modela los posibles tipos de elemento */
  case Burbuja          {}
  case Gema              {}
  case Multiespinas      {}
  case Esquina           {}
  case Ángulo            {}
  case Límite            {}
  case Espinas           {}
  case EspinasRotables   {}
  case Interruptor       {}
  case Compuerta         {}
}
```

```
type FormaDeElemento is variant {
  /* PROPÓSITO: Modela las diferentes formas de un elemento */
  case Simple            {}
  case Direccional       {}
  case Cerrable          {}
}
```

Y las siguientes funciones y procedimientos:

```
hayElemento
PROPÓSITO: Indica si hay un elemento cualquiera en la celda actual.
TIPO: Booleano.
PRECONDICIONES: Ninguna
```

formaDelElemento

PROPÓSITO: Describe la forma del elemento en la celda actual.

TIPO: FormaDeElemento.

PRECONDICIONES: Hay un elemento en la celda actual.

elementoSimple

PROPÓSITO: Describe el elemento simple en la celda actual.

TIPO: ElementoSimple.

PRECONDICIONES: Hay un elemento simple en la celda actual.

elementoDireccional

PROPÓSITO: Describe el elemento direccional en la celda actual.

TIPO: ElementoDireccional.

PRECONDICIONES: Hay un elemento direccional en la celda actual.

elementoCerrable

PROPÓSITO: Describe el elemento cerrable en la celda actual.

TIPO: ElementoCerrable.

PRECONDICIONES: Hay un elemento cerrable en la celda actual.

LimpiarCelda

PROPÓSITO: Quita cualquier elemento de la celda actual,
dejando la celda vacía.

PRECONDICIONES: Ninguna

ColocarElementoSimple_

PROPÓSITO: Pone en la celda actual el elemento simple dado

PARÁMETROS: ElementoSimple - El elemento a poner

PRECONDICIONES: La celda actual está vacía

ColocarElementoDireccional_

PROPÓSITO: Pone en la celda actual el elemento direccional dado

PARÁMETROS: ElementoDireccional - El elemento a poner

PRECONDICIONES: La celda actual está vacía

ColocarElementoCerrable_

PROPÓSITO: Pone en la celda actual el elemento cerrable dado

PARÁMETROS: ElementoCerrable - El elemento a poner

PRECONDICIONES: La celda actual está vacía

Tenga en cuenta que la disponibilidad de todos estos no implica que vaya a utilizar todos en las soluciones que realice. Deberá elegir con cuidado cuales son convenientes para solucionar el problema.

Actividades Castillo Final

Ejercicio 1)

Escribir el procedimiento **GirarEnSentidoHorario** que gira el elemento direccionable en la celda actual en sentido horario. Se asume que existe un elemento direccionable en la celda actual.

Ejercicio 2)

En ocasiones, por ejemplo, cuando se presiona un interruptor, se deben girar todas las espinas rotables que hayan en el juego. Se pide que realice el procedimiento **GirarTodasLasEspinRotables** que gira en sentido horario todas las espinas rotables que haya en el juego.

Ejercicio 3)

También es necesario poder determinar si en algún lugar del juego se encuentra la burbuja. Así, se pide que escriba la función **hayBurbujaEnElJuego** que indica si hay una burbuja en algún lugar del juego.

Ejercicio 4)

Otra de las cosas que se desea analizar es saber si hay la misma cantidad de espinas rotables que de interruptores en el juego. Para ello se desea realizar la función **hayMismaCantidadDeEspinRotablesQueDeInterruptores**

Además, se pide que estructure su solución realizando la subtarea **hayElementoDireccionableDeTipo_** que dado un tipo de elemento indique si en la celda actual hay un elemento direccionable con el tipo dado.