

**SI169 - Algoritmos Avanzados**

**Campeonato de Algoritmos - Estándar**

**Ciclo:** 2013-2

**Sección:** Todas

**Fecha de Entrega:** Semana 17

1. **Condiciones generales**

* El laberinto será de máximo 50 x 50 casillas.
* Deberá permitir la conexión de dos jugadores y establecer los controles de cada uno.
* Debe asegurar el cumplimiento de la totalidad de reglas.
* No puede haber más de un robot recolector en la misma casilla en un instante determinado.
* Si puede haber más de un robot explorador en la misma casilla en un instante determinado.

1. **Implementaciones por Casilla**
   1. **Origen:**

* La casilla O1 y O2 representarán las Bases o Orígenes de los jugadores.
  1. **Casilla Libre:**
* Esta casilla se identificará por los números 1,2,3,4 y 5.
* La casilla libre representa un espacio libre por donde cualquiera de los dos robots puede atravesar.
* Estas casillas pueden ser atravesadas por los robos descubridores y recolectores
* El número 1,2,3,4,5 representa la rigurosidad del terreno. Esto implica el tiempo el segundos que demorará el robot recolector en pasar el terreno según la tabla adjunta. La aplicación debe tener un mantenimiento para cambiar estos valores en caso sea necesario.

|  |  |
| --- | --- |
| Rigurosidad | Tiempo (s) |
| 1 | 0.5 |
| 2 | 0.7 |
| 3 | 1.0 |
| 4 | 1.4 |
| 5 | 2.4 |

* Los robots voladores demoran en atravesar todas las casillas un total de 0.4(s)
  1. **Casilla Muro**
* Estas casillas están representadas por el símbolo \*, estas son un muro que no puede ser traspasado por ningún robot.
  1. **Casilla Tesoro**
* Estas casillas están representadas por el símbolo T seguido de dos números P-V. Donde P representa el peso del Tesoro, y V representa el Valor del tesoro.
* Por ejemplo T-6-2
* Las casillas solo pueden ser atravesadas por los robots descubridores, el robot recolector no puede pasar por una casilla donde exista un tesoro.
* Cuando un robot recolector, recoja un tesoro, la casilla quedará libre y tendrá una rigurosidad de terreno de 1 para todos los casos.
* Solo los robots recolectores (terrestres) pueden recoger la nave.
  1. **Casilla Moneda**
* Identifica por “C”.
* La casilla moneda permite al robot descubridor, ganar puntaje para su equipo. Si el robot encuentra una moneda y la lleva a su lugar de origen ganará +2 de valor (este valor debe ser parametrizable).
* El robot descubridor solo puede cargar una moneda a la vez. Por lo tanto si quiere llevarse otra deberá haber dejado la anterior en su base.
* Por defecto, todas las casillas donde existan monedas tendrán rigurosidad de 1.
* Solo los robots descubridores pueden cargas las monedas (estas están flotando en el aire).
* El robot descubridor puede optar por llevarse o no la moneda cuando este en alguna de estas casillas.
* Cuando un robot recolector, recoja un tesoro, la casilla quedará libre y tendrá una rigurosidad de terreno de 1 para todos los casos.
  1. **Casilla Fuego**
* Se identifica por “F”.
* La casilla fuego no permite que ingrese ninguno de los 2 robots, ya que en caso suceda esto, los robots quedarán destruidos.
* La casilla de fuego ira creciendo conforme pase el juego, cada 15 segundos esta casilla incendiará las que están a su alrededor. Este parámetro será configurable.
* Si el fuego agarra algún robot, este será eliminado y un nuevo robot deberá ingresar en su reemplazo en la base. Esto solo se podrá hacer 15 segundos después de la destrucción. (parámetro configurable)
* El fuego elimina los tesoros y monedas.
  1. **Casilla Bomba**
* La casilla bomba únicamente al robot recolector cuando este pisa alguna de estas casillas.
* Al igual que con el fuego, el robot destruido deberá esperar 15 segundos para volver a entrar a la base.
* Como las bombas están en el piso esto no genera ningún problema para el robot descubridor (volador).

1. **Mensajería**

Para que el algoritmo funcione correctamente, se deberá construir la siguiente lógica de mensajería que enviará el aplicativo cliente al programa.

* 1. **Inicio de Sesión:** El Robot iniciará sesión y el tablero lo colocará en su posición de Origen

Método para robot terrestre y volador.

|  |  |  |
| --- | --- | --- |
|  | Envío | Respuesta |
| Formato | Deberá enviar el mensaje INI, seguido por el ID del Jugador | Si todo fue conforme devolver TRUE seguido por los siguientes valores separados por coma:   * Ancho (relativo a X) * Largo (relativo a Y) * Coordenada X * Coordenada Y * Capacidad Robot Recolector (kg) * TiempoMoverRecolector1(ms) * TiempoMoverRecolector2(ms) * TiempoMoverRecolector3(ms) * TiempoMoverRecolector4(ms) * TiempoMoverRecolector5(ms)MaximaCantidadBombas   De lo contrario FALSE  Para excepciones retornará EX |
| Ejemplo | INI,Jugador1 | TRUE,25,25,0,2,30,500,700,1000,1400,2400,5 |

* 1. **Consulta:** El Robot consulta sobre las posiciones a su alrededor

Método para SOLO para robot volador.

|  |  |  |
| --- | --- | --- |
|  | Envío | Respuesta |
| Formato | Deberá enviar mensaje CON | Responde con la información de las posiciones aledañas en el siguiente Orden: Centro,Arriba, Derecha, Abajo, Izquierda.  Devolverá según la especificación del archivo posibles valores de (1,2,3,4,5,\*,F,T#,C,O1,O2,B), separados por compas según el orden anterior.  Si alguna casilla esta fuera del mapa, se reportará como N.  Si el robot está muerto, o el fuego lo alcanza en la posición actual se dolverá el mensaje: DEAD  Si el robot aún no ha iniciado error, o algún otro problema se reportara ERROR  Para excepciones retornará EX |
| Ejemplo | CON | O1,\*,1,\*,N |

* 1. **Mover:** El Robot tratará de moverse.

Método para robot terrestre y volador.

|  |  |  |
| --- | --- | --- |
|  | Envío | Respuesta |
| Formato | Deberá enviar mensaje MOV.  Después una coma indicado el robot que se moverá.  R = Recolector  E = Explorador  Después una compa seguido por el destino del movimiento:  1-Arriba  2-Derecha  3-Abajo  4-Izquierda | Si el robot se mueve con éxito a una casilla libre o disponible según reglas definidas se devuelve TRUE.  Si el robot no puede moverse porque existe un obstáculo, muro, o sale del mapa devolverá FALSE.  Si el **robot recolector** no puede moverse debido a que existe otro robot recolector en la casilla destino. Se retornará ROBOT.  Si el robot está muerto, o muere con el movimiento, o el fuego lo alcanza mientras no hacia movimientos devolverá DEAD. Solo para el caso del robot recolector, se devolverá DEAD cuando este pise una bomba.  Si por las reglas del juego (rigurosidad del terreno, o tiempo de espera entre movimientos, o tiempo de espera después de una muerte) el robot no puede moverse aún se devolverá: WAIT  Para excepciones retornará EX |
| Ejemplo | MOV,R,1 | DEAD |

* 1. **Coger Moneda:** El Robot coge la moneda

Método para SOLO para robot volador.

|  |  |  |
| --- | --- | --- |
|  | Envío | Respuesta |
| Formato | Deberá enviar mensaje COM | Si el robot tiene capacidad para cargar la moneda y está en una casilla donde hay una moneda se devolverá TRUE.  Si no existe moneda en la casilla se devolverá FALSE.  Si no existe capacidad para cargar la moneda se devolverá ERROR  Si el fuego lo alcanza en la posición actual se devolverá el mensaje: DEAD  Para excepciones retornará EX |
| Ejemplo | COM | ERROR |

* 1. **Dejar Moneda:** El Robot tratará dejará la moneda en la base.

Método para SOLO para robot volador.

|  |  |  |
| --- | --- | --- |
|  | Envío | Respuesta |
| Formato | Deberá enviar mensaje DEM | Si el robot está en la posición de Origen y tiene una moneda devolverá TRUE. Se debe sumar el puntaje logrado.  De lo contrario se volverá FALSE.  Para excepciones retornará EX |
| Ejemplo | DEM | TRUE |

* 1. **Coger Tesoro:** El Robot recoje el tesoro

Método para SOLO para robot recolector/terrestre.

|  |  |  |
| --- | --- | --- |
|  | Envío | Respuesta |
| Formato | Deberá enviar mensaje COT | Si el robot tiene capacidad disponible para poder recoger el tesoro y adicionalmente, está en una casilla donde hay un tesoro se devolverá TRUE, seguido por el peso y valor del tesoro separado por comas  Si no existe moneda en la casilla se devolverá FALSE.  Si no hay suficiente capacidad para poder recoger el tesoro se devolverá ERROR  Si el fuego lo alcanza en la posición actual se devolverá el mensaje: DEAD  Para excepciones retornará EX |
| Ejemplo | COT | TRUE,6,2 |

* 1. **Dejar Tesoros:** El Robot tratará de dejar todos los tesoros recolectados en la base.

Método para SOLO para robot recolector/terrestre.

|  |  |  |
| --- | --- | --- |
|  | Envío | Respuesta |
| Formato | Deberá enviar mensaje DET | Si el robot está en la posición de Origen y tiene uno o más tesoros, se devolverá el mensaje TRUE. Se debe sumar el puntaje logrado.  De lo contrario se volverá FALSE.  Para excepciones retornará EX |
| Ejemplo | DET | TRUE |

* 1. **Dejar Bomba en Camino:** El Robot recolector dejará bombas en camino.

Método para SOLO para robot recolector/terrestre.

|  |  |  |
| --- | --- | --- |
|  | Envío | Respuesta |
| Formato | Deberá enviar mensaje DEB | Si el robot está en la posición de Origen o de algún tesoro, bomba y/o moneda, devolverá FALSE.  Si no tiene bombas disponibles se devolverá: NOBOMBA  Si el fuego lo alcanza en la posición actual se devolverá el mensaje: DEAD  Si la casilla es libre, se devolverá el mensaje TRUE. |
| Ejemplo | DEB | TRUE |

1. **Archivo de Texto**

El archivo de Texto una primera línea donde se indica la dimensión del tablero (Fila Columna) y a continuación el detalle de cada una de las celdas.