

VIAJE INTERGALÁCTICO

Simularemos en Gobstones un escenario que muestra, dentro de una galaxia, planetas y el viaje de una nave espacial.

Representaremos a los planetas con bolitas azules, tantas como id numérico que describa más información sobre el planeta en cuestión, esto no es relevante p/ nosotros. Nuestra nave, también representada con tantas bolitas de color rojo como el identificador numérico, se alimenta de materia oscura, señalaremos esa cantidad con bolitas negras. Este combustible particular permite el viaje intergaláctico. Cada unidad de este material brinda una autonomía de 1 año/luz de distancia, que en nuestro tablero de GS, es una celda de distancia. Todas las distancias mencionadas en este dominio son ortogonales, no contabilizamos trayectorias diagonales.

Ejercicio #01) A continuación se muestra el contrato de dos funciones muy importantes para el desarrollo de nuestra simulación. Se pide definir las s/ representación.

function hayUnPlanetaA_Hacia_(distancia, dirección)

Propósito. Indica si hay un planeta a exactamente *distancia* celdas en sentido *dirección*.

Precondición.

Hay al menos *distancia* celdas en dirección *dirección*.

Parámetros.

distancia. Número. La cantidad de años luz a la cual se desea buscar un planeta.

dirección. Dirección. La dirección hacia la cual mirar el planeta.

Tipo. Booleano.

function combustibleRestante


Propósito. Indica la cantidad de combustible que le queda a la nave.

Precondición. El cabezal está sobre la nave.

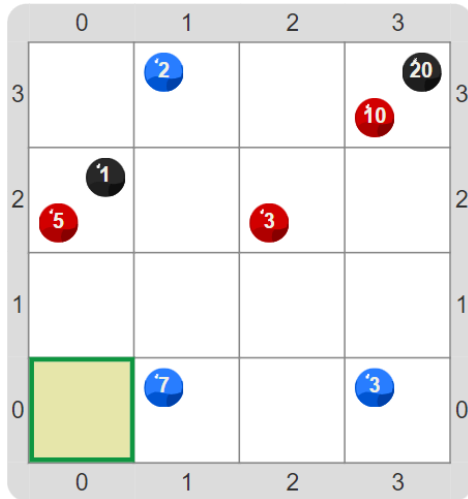
Tipo. Número.

Ejercicio #02) Definir **sePuedeAterrizarA_HaciaEl_(distanciaAPlaneta, direcciónAPlaneta)**, que asumiendo que el cabezal se encuentra sobre la nave y hay al menos distanciaAPlaneta celdas en dirección direcciónAPlaneta, indica si hay un planeta a exactamente distanciaAPlaneta en la dirección direcciónAPlaneta y si el combustible es suficiente para llegar al mismo.

Ejercicio #03) Sabiendo que el cabezal se encuentra sobre la nave se pide que escriba la función **hayUnPlanetaRecto**, que indica que existe un planeta en cualquiera de las direcciones, a cualquier distancia desde la nave.

 / **Ejercicio #04)** Definir **puedeLaNave_OrbitarAlgunoPorEmergencia**, que nos indica si, para un id de nave determinado; en caso de una emergencia, es posible aunque sea orbitar un planeta. Definimos orbitar un planeta a estar a 1 año luz de distancia. Definimos falta de combustible cuando la nave tiene exactamente 1 unidad de materia oscura. Asumimos que hay en La Galaxia una nave con el Id indicado y que no se encuentra sobre un planeta ni orbitando uno.

Por ejemplo:



`puedeLaNave_OrbitarAlgunoPorEmergencia(5)`
indicaría Verdadero, puesto orbita el planeta 2.

`puedeLaNave_OrbitarAlgunoPorEmergencia(10)`
indicaría Falso, pues no cumple la precondición, no está en Estado de Emergencia.

`puedeLaNave_OrbitarAlgunoPorEmergencia(3)`
indicaría Falso, pues no cumple la precondición.
Directamente no tiene combustible para moverse.

🦴 / **Ejercicio #05)** Nuestra nave es libre de girar a 90° y avanzar, siempre en busca del planeta. Así podemos trazar una trayectoria de donde está una nave y un planeta, en nuestra galaxia. Se pide definir **AterrizarNave_EnPlaneta_SiEsPosible** que, de ser posible, lleve la nave indicada hasta el planeta indicado. Se recomienda para resolver este problema la utilización de primitivas que manipulen coordenadas cartesianas. No se pide recrear el recorrido de la nave, simplemente situar la nave en el planeta destino, sacándola de donde estaba. Asumimos que existe tanto el planeta como la nave en cuestión.