

Guber

Guber es una novedosa app que permite a los usuarios solicitar un vehículo que pase a recogerlos por donde se encuentran y los lleve al lugar que desean (similar a otros servicios actuales, que compiten contra los taxis o remises). Como novedad, aquí no hay conductores, sino que el vehículo es autónomo (se maneja solo). En esta práctica integradora usted deberá programar un simulador de este sistema en Gobstones

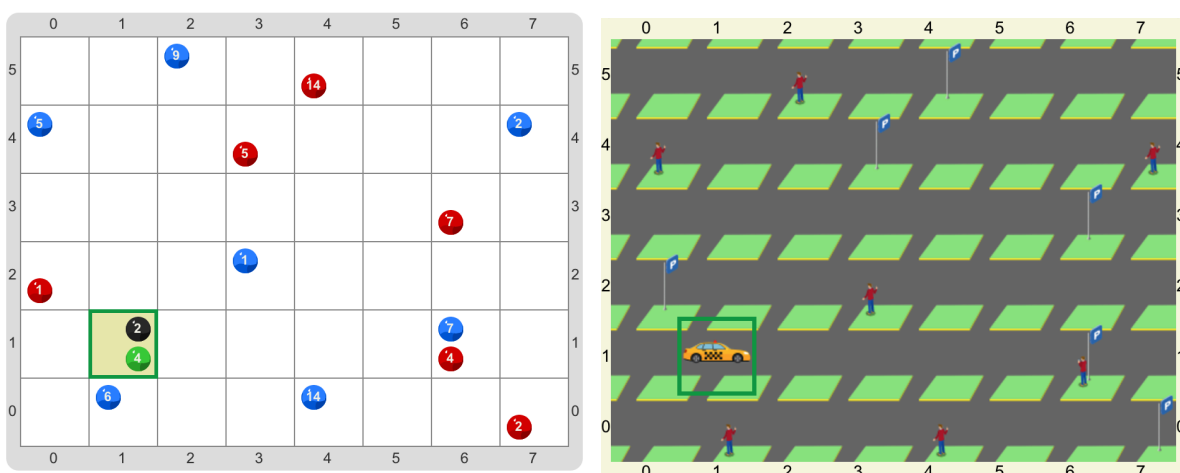
El tablero representará un mapa, donde cada celda representa una ubicación específica del mismo. En el mapa se encuentra el vehículo que deberemos manipular. En algunas de las ubicaciones del mapa puede haber solicitantes. Un solicitante es una persona que ha pedido un vehículo para que pase a recogerlo, y que debe ser llevado a su destino. Los solicitantes se identifican con un número único al que llamamos ID. Solo puede haber un único solicitante por ubicación.

El vehículo puede entonces recoger solicitantes (que una vez arriba del vehículo, pasamos a denominar pasajeros), con la intención de llevarlos posteriormente a su destino y bajarlo en dicho lugar. El vehículo no puede dejar al pasajero en cualquier lado, solo en su destino. Tampoco puede recoger más de un solicitante al mismo tiempo (es decir, solo lleva un pasajero a la vez). Una vez un pasajero baja en su destino, tanto el pasajero como el destino dejan de ser relevantes para la simulación, y por tanto desaparecerán del tablero.

Se ha elegido a los efectos prácticos la siguiente representación:

- El vehículo vacío se representa con una bolita de color Negro.
- Un solicitante se representa con tantas bolitas de color Azul como su ID.
- El vehículo con un pasajero se representa con dos bolitas de color Negro, y tantas bolitas de color Verde como el ID del pasajero que recogió.
- El destino de un pasajero se representa con tantas bolitas de color Rojo como el ID del pasajero que tiene esa ubicación como destino.

El siguiente es un tablero inicial de ejemplo, pero otros tableros iniciales son posibles.



Del lado izquierdo, sin vestimentas, del lado derecho, con vestimentas. En este tablero se aprecia el vehículo bajo el cabezal, el cual tiene un pasajero (el de ID 4). El destino de dicho pasajero se encuentra 5 lugares al Este, en la celda 6,1. En ese mismo lugar espera un solicitante, el de ID 7, cuyo destino es la celda 6,3. En total hay 7 solicitantes. Notar como los solicitantes 6 y 9 no tienen un destino asociado en el mapa. Decimos que su destino está “fuera de la ciudad”, y estos casos tendrán relevancia más adelante.

Primera parte (Programas, Contratos, Procedimientos y Repeticiones Simples):

Ejercicio 1)

Realice el procedimiento **MoverVehículoVacíoAlNorte** que mueve el vehículo al Norte una única ubicación, asumiendo que el cabezal está sobre el vehículo, y este se encuentra vacío.

Ejercicio 2)

Realice el procedimiento **RecogerSolicitanteConId5** que sube el solicitante con el ID 5 al vehículo. Para ello se asume que el cabezal se encuentra sobre el vehículo, y en la ubicación del vehículo se encuentra el solicitante con el ID mencionado.

Ejercicio 3)

Se pide que realice el procedimiento **LlevarPasajeroConId5ActualADestinoA5AlNorte** que lleva el pasajero con el Id 5 a su destino (el cual se encuentra 5 celdas al Norte de la actual). Para esto se asume que el cabezal se encuentra sobre el vehículo y que tiene al pasajero con Id 5 a bordo. También se presupone que el destino del mencionado pasajero se encuentra en línea recta hacia el Norte a 5 celdas de la actual. Cabe aclarar que una vez que se llega al destino debe bajarse al pasajero en el destino. Bajarlo consiste simplemente en sacar las bolitas que representan al mismo y dejando al vehículo vacío.

Para resolver este problema deberá hacer uso de la siguiente primitivas:

MoverVehículoConPasajeroConId_A1_

PROPÓSITO: Mueve el vehículo con el pasajero con el Id dado una ubicación en el mapa hacia la dirección dada.
El cabezal termina sobre el vehículo.

PRECONDICIONES:

- * El cabezal se encuentra sobre el vehículo.
- * El vehículo tiene al pasajero con el Id dado.

Un ej. de uso de la primitiva anterior sería **MoverVehículoConPasajeroConId_A1_(3, Sur)**, lo cual mueve el vehículo con el pasajero con Id 3 una posición al Sur.

Segunda parte (Parámetros, Expresiones y Tipos, Alternativas Condicionales, Funciones Simples):

Ejercicio 4)

Escriba el procedimiento **RecogerSolicitanteConId_** que dado un número de ID, sube al vehículo el solicitante con el ID mencionado. Se asume que el cabezal se encuentra sobre el vehículo, y que en la ubicación del mismo se encuentra el solicitante con el ID dado.

Ejercicio 5)

Se pide escriba el procedimiento **RecogerSolicitanteEnUbicaciónSiHay** que sube al pasajero que hay en la ubicación actual al vehículo, si hubiera uno y el vehículo se encontrara vacío. Para esto se asume que el cabezal se encuentra sobre el vehículo.

Para resolver este problema puede hacer uso de las funciones primitivas **haySolicitanteEnUbicación** y **idDeSolicitanteEnUbicación** cuyos contratos se dan a continuación:

haySolicitanteEnUbicación

PROPÓSITO: Indica sí en la celda actual hay un solicitante.

TIPO: Booleano.

PRECONDICIONES:

* Ninguna.

idDeSolicitanteEnUbicación

PROPÓSITO: Describe el ID del solicitante esperando en la ubicación actual.

TIPO: Número.

PRECONDICIONES:

* Hay un solicitante en la ubicación actual.

Ejercicio 6)

Se pide escriba la función **esDestinoDePasajero** que indica si la ubicación actual es el destino del pasajero en el vehículo. Para esto puede asumir que el cabezal se encuentra sobre el vehículo y que este tiene un pasajero.

Ejercicio 7)

Se pide escriba el procedimiento **MoverVehículoAl_** que mueve el vehículo en la celda actual a dirección dada. Tenga en consideración que sí el vehículo está vacío, lo mueve vacío, pero sí tuviera un pasajero, debe mover al vehículo con su correspondiente pasajero.