

Control Car.

El programa consiste en poder controlar un vehículo por medio de una consola de comandos. Este vehículo podrá moverse de forma libre por todo el área disponible y también podrá ir de un punto a otro en un laberinto. Para resolver el laberinto deberá diseñar un algoritmo que encuentre el camino más corto, o también, podrá hacerlo moviéndose paso a paso por laberinto. El mapa que contiene el laberinto podrá leerse desde un archivo, o generarse aleatoriamente. Todos los movimientos y comandos necesarios deberán ser ingresados en un editor de texto simulando una consola. El programa debe incorporar el uso de asignación dinámica de memoria y listas enlazadas, así como también la documentación del código de forma adecuada utilizando *doxygen*.

Comandos

Los comandos deberán ingresarse en un editor de texto de 24 caracteres que puede almacenar hasta 1024. Los comandos se aceptan presionando la tecla **[ENTER]**, Si existe algún error debe indicarse.

COMANDO	DESCRIPCIÓN
setinicio(x, y)	Posiciona el punto de inicio de en el mapa.
setfinal(x, y)	Posiciona el punto final en el mapa.
resetmapa	Posiciona el vehículo en el punto de inicio. Si esta en modo gomapa lo detiene.
loadmapa(mapaname)	Carga un mapa desde archivo.
savemapa(mapaname)	Guarda un mapa a un archivo.
genmapa(x, y)	Genera un mapa aleatoriamente de x columnas por y filas.
resolver	Encuentra el camino más corto para ir de un punto de inicio a uno final en el mapa.
gomapa	Comienza a desplazar el vehículo por el laberinto siguiendo el camino encontrado
steptime(ms)	Configura el tiempo en ms para simulación del desplazamiento del vehículo.
clrscr	Borra toda el área de movimiento.
adelante(n)	Mueve hacia adelante n pasos.
atras(n)	Mueve hacia atras n pasos.
girarizquierda(n)	Gira a la izquierda n veces. Cada giro es de 90°
girarderecha(n)	Gira a la derecha n veces. Cada giro es de 90°
repetir(n) { comandos }	Permite repetir una secuencia de comandos n veces. Los comandos se encuentran entre las llaves. Eje.: repetir(4) { adelante(4) girarderecha(1) }

	}
brushon(color)	Activa el pincel que deja el rastro de los movimientos del vehículo.
brushoff	Desactiva el pincel.
loadsecuencia(seceuencianame)	Carga una secuencia de comandos desde un archivo.
gosecuencia(modos)	Ejecuta la última secuencia cargada. modo: GO o STEP. GO: Ejecuta toda la secuencia. STEP: Ejecuta la secuencia paso a paso, una vez iniciado este modo se puede continuar presionando la tecla [F10] .
resetsecuencia	Vuelve al inicio la última secuencia cargada. Si está ejecutándose una secuencia la detiene.
stop	Detiene la secuencia o laberinto que se se está ejecutando.
Para el Turno Agosto (31/07 al 04/08) agregar los siguientes comandos.	
si(condición) { comandos } sino { comandos }	Ejecución condicional. Puede existir el si sin el sino , pero no el sino sin el si . Condición puede ser cualquiera de estas: adelantePared atrasPared izquierdaPared derechaPared Operadores lógicos: no(condición): niega condición.
repetirmientras(condición) { comandos }	Repite una secuencia de comandos siempre que la condición sea verdadera.

Mapa

El mapa es un archivo BMP compuesto por los siguientes colores: **LIGHTGRAY** para indicar una pared, **ROJO** el punto de inicio, **AZUL** el punto final. El resto del mapa es **BLACK** que indica un camino libre. Los puntos de inicio y final pueden no aparecer en el archivo del mapa.

Las dimensiones máximas del mapa serán 80 columnas por 40 filas (80x40 pixels) y para ello se utilizará un archivo tipo BMP en formato RGB24.

El espacio libre debe tener 2x2 pixels de espesor, las paredes tienen 1 pixel de espesor.

Archivo Secuencia

Es un archivo de texto en donde cada línea representa un comando.

Vehículo

El vehículo será representado por cuatro pixels (2x2) dos de los cuales deben tener un color diferente para indicar cuál es el frente.

Movimientos

Los movimientos deben realizarse usando los comandos correspondientes. Tenga en cuenta que si existe un obstáculo durante el movimiento este solo podrá realizarse hasta ese lugar sin atravesar la pared, e indicar que hay un obstáculo. Simule el movimiento yendo casillero a casillero según el **steptime** configurado.

En la ejecución de una secuencia si existe algún error en los comandos debe detenerse la ejecución e indicar este error, siempre debe mostrarse cuál es el comando que se está ejecutando y si este está activo o no.