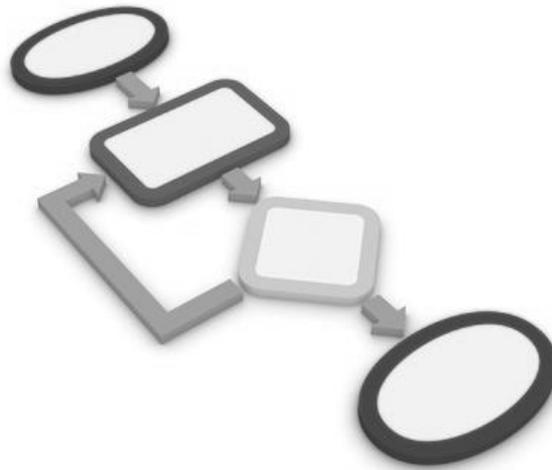


MANUAL DE EJERCICIOS PARA CURSOS BÁSICOS DE ALGORITMOS Y PROGRAMACIÓN.

Problemario

Elaborado por:

Enciso Almanza Adrián,
González Fraga José Ángel,
Álvarez Xochihua Omar,
González Sarabia José Alejandro



Octubre del 2016,
Ensenada, Baja California, México.

1 Introducción

Las primeras semanas de un curso de programación son definitivas en cuanto a los conocimientos que el estudiante adquiere sobre el área que se estudia, es durante este periodo en el que el estudiante aprende los principios estéticos de la disciplina y es más receptivo a nuevas ideas.

Siempre es un reto para los actores que participan en el proceso de enseñanza-aprendizaje el transmitir y recibir conocimiento nuevo, y se vuelve aún más complejo cuando estos conocimientos requieren de habilidades y destrezas para lograr diseñar un algoritmo: aprender a ordenar ideas, aprender a crear, aprender a plasmar un concepto en un conjunto de instrucciones ordenadas, es aprender a pensar mediante un razonamiento lógico y estructurado las ideas.

1.1 Los Minilenguajes o Micromundos

Los mini lenguajes o micro mundos, son lenguajes de programación simples y pequeños diseñados para ayudar a principiantes aprender programación. Un estudiante aprende programación controlando un actor la mayoría de los mini lenguajes lo utilizan, que puede ser una tortuga, un robot o cualquier otra entidad. Esta entidad actúa dentro del micro mundo.

Los mini lenguajes facilitan el aprendizaje de un lenguaje de programación que se usan hoy en día. Los mini lenguajes proveen una base sólida para resolver problemas sistemáticamente. No importa si el estudiante se enfoque o no en la programación en el futuro, aprender un mini lenguaje ayudara que el estudiante adquiriera un pensamiento algorítmico como una herramienta (Brusilovsky, 1997), las ventajas de estos mini leguajes es que son: pequeños, es decir tienen una sintaxis pequeña y una semántica simple, se basan en metáforas que son intrínsecamente interesantes y atractivos a la vista, las operaciones que el actor realiza siempre son visibles, lo cual revela la semántica de las construcciones del lenguaje, los mini lenguajes son especialmente diseñados para propósitos educativos.

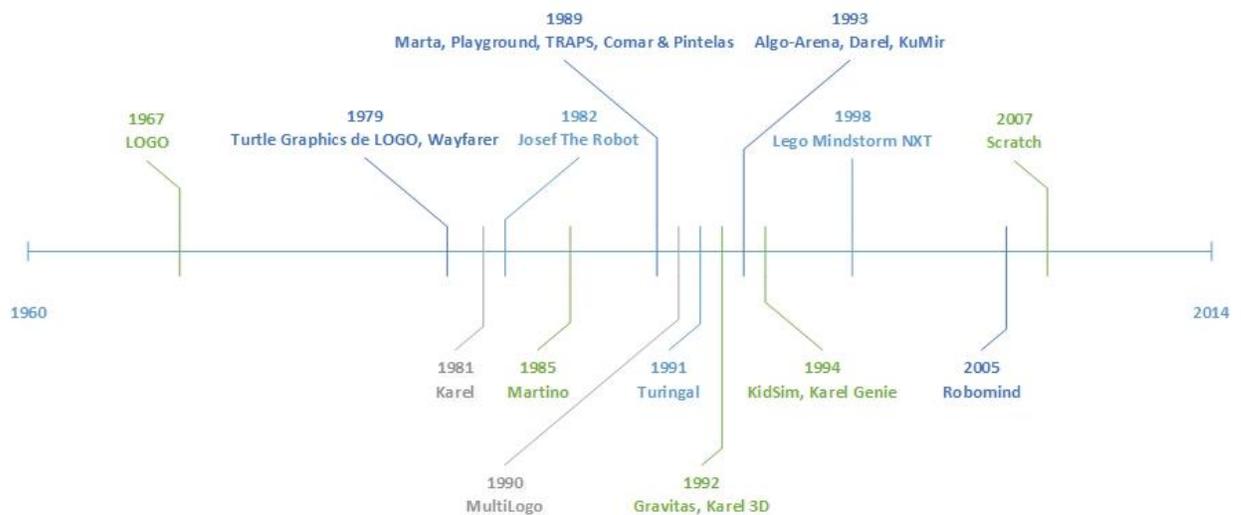


Figura 1, Aparición de los Minilenguajes

Un ejemplo de los mini lenguajes es el Robot karel de Richar E. Pattis, en su libro de programación el robot Karel (Richard. E. Pattis, 1994), describe de forma sencilla y divertida una serie de conceptos de programación y conocimientos que

el estudiante adquiere en la medida que el Robot Karel aprende nuevas instrucciones y estructuras, mientras que el estudiante se adentra en el mundo de Karel, asumiendo su papel de guía en la solución de problemas, incurriendo cada vez menos errores “paros por error” o “errores de intento” que pueden llegar a presentarse durante su programación.

A partir del 2008 la Facultad de Ciencias ha llevado a la práctica al robot Karel en un curso de diseño de algoritmos previo a la programación de computadoras, en el cual se ve con mucho agrado el nivel de abstracción y resolución de problemas que adquieren los estudiantes en un lenguaje enteramente imperativo. Sin embargo el grado de “motivación” decae hacia el final cuando no hay una forma de expresar y llevar a lo tangible estos algoritmos que en papel son expresados en cientos de líneas de código y al final son solo eso “papel”. Para ello han surgido una serie de aplicaciones desarrolladas por los propios estudiantes “Fans del Robot Karel” que ayudan al estudiante a detectar los errores y a visualizar paso a paso la solución del problema en un entorno simulado al mundo de Karel, de una manera visual, donde se le muestra al estudiante la “ejecución” de su algoritmo y conozca el porqué de una corrida exitosa o errónea de su programa, un ejemplo es el karel de la OMI.

2 Karel el Robot

Karel es una herramienta de aprendizaje de programación utilizada por diversas instituciones para introducir a los alumnos a la utilización de lenguajes de programación, fue desarrollado por Richard E. Pattis, y se dio a conocer junto a su libro “Karel The Robot: A Gentle Introduction to the Art of Programming” en 1994.

Karel vive en un mundo limitado, de manera informal se puede decir que el mundo de Karel, ver figura 1, es un conjunto de calles y avenidas por las que este puede desplazarse. También hay objetos especiales que puede sentir y manipular en su mundo las secciones de paredes y los zumbadores, los cuales son pequeñas bolas de plástico que emiten un sonido suave intermitente, y que pueden estar colocados en esquinas y pueden ser levantados, transportados y bajados de una mochila que trae consigo karel.

Karel como lo menciona Richard E Pattis en su libro posee rudimentarios sentidos de la vista, el sonido, la dirección, y el tacto, así con ánimos de recrearnos la imagen de karel usando la imaginación ante los hechos reales de lo que es karel en el papel, les puedo describir cada uno de ellos, tiene el sentido de la vista por que cuenta con tres cámaras web, las cuales se enfocan

en las direcciones de enfrente, a su derecha y a su izquierda, tiene el sentido de sonido por que tiene la habilidad de escuchar los zumbadores siempre y cuando estén ambos en la misma esquina, tiene el sentido de la dirección por que cuenta con una brújula interna, y finalmente tiene el sentido del tacto por que cuenta con un brazo mecánico que con el cual puede levantar y colocar los zumbadores.

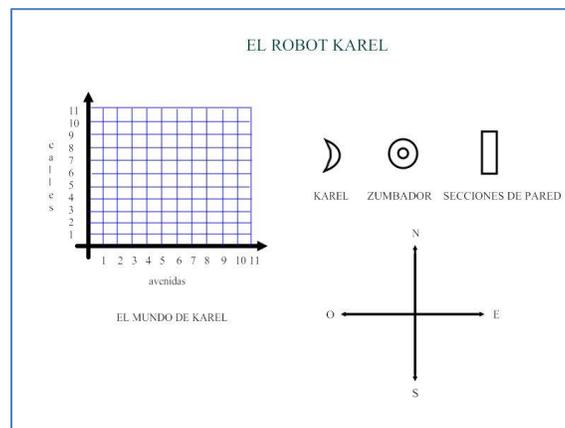


Figura 2, Mundo de Karel.

Instrucciones primitivas de karel

Las habilidades que tiene Karel, ver figura 2, son: puede moverse hacia adelante, la dirección hacia la cual esta volteado con la instrucción muévete, puede girar a su izquierda con la instrucción gira-a-la-izquierda, puede levantar un zumbador y ponerlo en su mochila con la instrucción levanta-zumbador y colocarlo con la instrucción de coloca-zumbador, y apagarse con la instrucción de apágate.

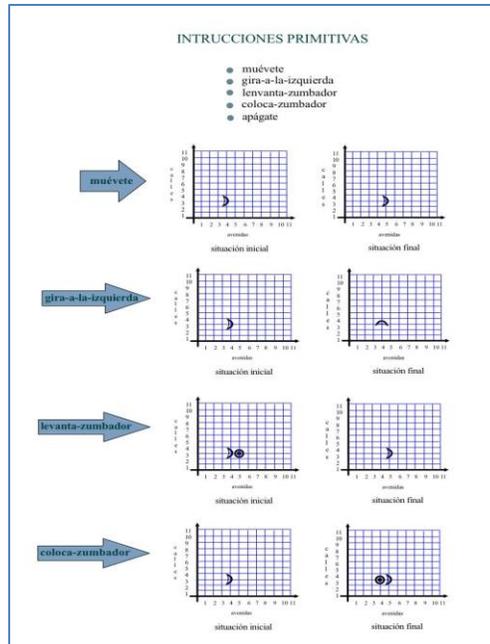


Figura 3, Habilidades de Karel.

Manejo de condicionales y ciclos

Una habilidad que tiene karel es precisamente investigar lo que pasa alrededor de él, como esta su entorno en un momento dado, y después decidir qué hacer a continuación, en base a la información obtenida.

Las instrucciones IF/THEN, IF/THEN/ELSE nos permitirán poder escribir programas mucho más generales, evitar que existan errores durante la ejecución del programa y permitirle a karel llevar a cabo tareas sin importar la situación inicial en la que se encuentra. Para ello utiliza una serie de pruebas, ver figura 4, que forman parte del vocabulario de karel y que le van a permitir verificar como está el mundo antes y después de realizar una acción.

Las instrucciones que repiten ITERATE Y WHILE también llamadas en el mundo de la programación como ciclos, permiten que karel repita una instrucción o un

conjunto de instrucciones un número determinado de veces como lo es el iterate, y repetir una instrucción o un conjunto de instrucciones tantas veces como la prueba sea verdadera como lo es el while, ambas instrucciones ejecutar en forma repetitiva cualquier conjunto de instrucciones que formen parte del vocabulario de karel.

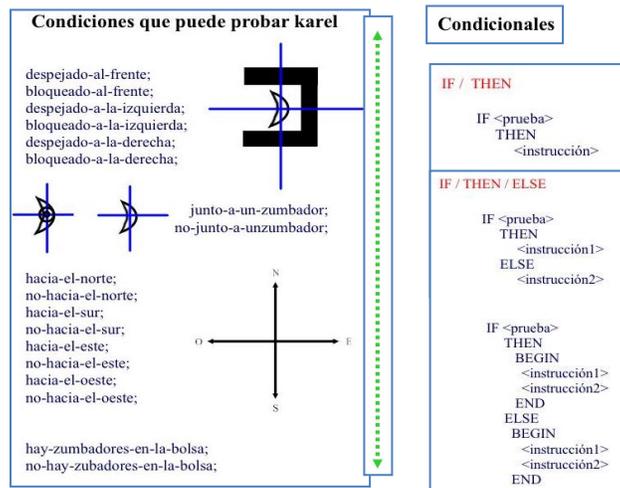


Figura 4, manejo de condicionales en Karel.

Existen infinidad de situaciones o problemas a las que karel se puede encontrar en su mundo, pero hay una en particular, el cual será objeto de estudio y consiste en programar a karel para que salga de un laberinto.

El Problema del laberinto

El problema consiste en programar a karel para que escape de un laberinto que no contiene islas, la salida está marcada con un zumbador colocado en la parte de afuera del laberinto, el cual servirá determinar si salió o no de él. En la figura

5, se muestra la situación inicial del problema, podemos observar que siempre a la derecha de karel existe un pared, por lo que esta tarea se puede llevar a cabo ordenándole a karel que se desplace por el laberinto con la invariante de que a su lado derecho siempre haya una pared, hasta que encuentre el zumbador.

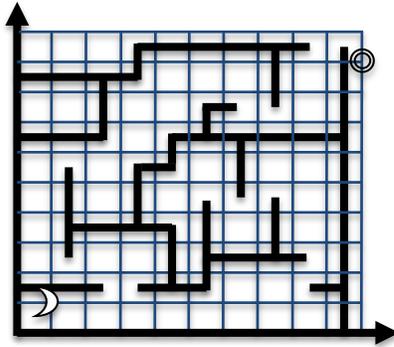


Figura 5, Situación inicial, salir del laberinto

Solución al problema del laberinto

A continuación se muestra el programa completo para la solución del problema del laberinto, como podemos observar la instrucción sigue-pared-derecha se encarga de conducir a karel por todo el contorno de la pared que tiene a su derecha , la cual termina hasta que se encuentre el zumbador que se encuentra fuera del laberinto y determina el fin del programa.

```
BEGINNING-OF-PROGRAM  
DEFINE-NEW-INSTRUCTION gira-a-la-derecha AS  
  ITERATE 3 TIMES  
    gira-a-la-izquierda;  
  
DEFINE-NEW-INSTRUCTION media-vuelta AS
```

```
ITERATE 2 TIMES
  gira-a-la-izquierda;

DEFINE-NEW-INSTRUCTION sigue-pared-derecha AS
BEGIN
  WHILE bloqueado-a-la-derecha DO
    IF despejado-al-frente THEN
      muévete;
    ELSE
      gira-a-la-izquierda;
      gira-a-la-derecha;
      muévete;
  END;

BEGINNING-OF-EXECUTION
  WHILE no-junto-a-zumbador DO
    sigue-pared-derecha;
    apágate;
  END-OF-EXECUTION
END-OF-PROGRAM
```

Con esto, mostramos la manera en que karel de Richar E. Pattis pretende enseñar de una forma divertida los conceptos básicos de la programación, utilizando instrucciones simples, ampliando su vocabulario con nuevas instrucciones (funciones o procedimientos), dotándolo de las estructuras de control como el IF/THEN/ELSE para evitar paros o errores durante su ejecución (condicionales) y las instrucciones que repiten y que ayudan a realizar un programa más general (ciclos). Todas estas en su conjunto los conocemos como las estructuras de control en un lenguaje de programación, en el libro el robot karel de Richar E. Pattis nos introduce a un lenguaje de programación de alto nivel, motivando y haciendo que el alumno pierda la fobia a la computación en un primer acercamiento a la programación de computadoras.

3 Problemas.

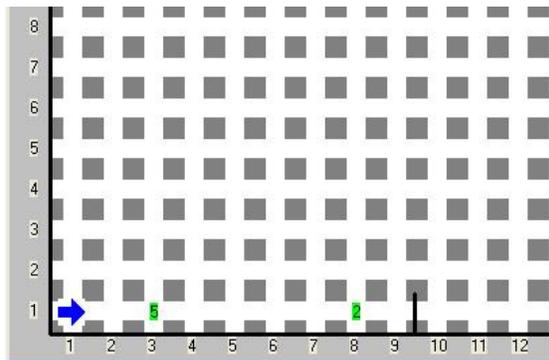
PROBLEMA 1 “RECOGIENDO JUGUETES”

KAREL TIENE COMO MISION RECOGER TODOS LOS JUGUETES DEL PISO DE SU CASA Y LLEVARLOS A SU RECAMARA (Origen).

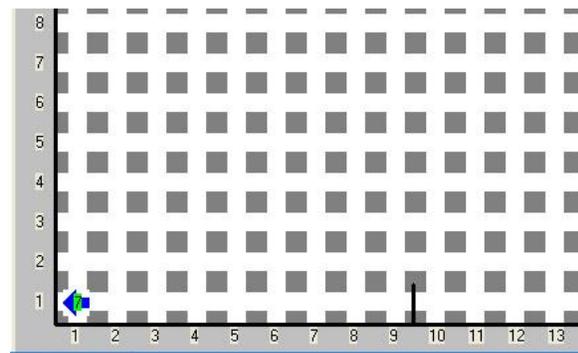
- NO SABEMOS CUANTOS JUGUETES HABRA EN CADA LUGAR
- KAREL SABE QUE TERMINO DE RECOGERLOS AL LLEGAR A LA BARRA DE SU CASA
- AL TERMINAR DE DEJAR LOS JUGUETES NO IMPORTA HACIA DONDE QUEDA APUNTANDO KAREL

EJEMPLO:

MUNDO INICIAL



MUNDO FINAL



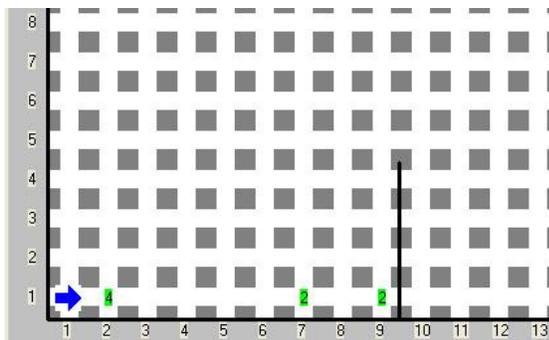
PROBLEMA 2 “AYUDANDO A MAMA”

KAREL HA CRECIDO UN POCO Y AHORA SU MAMA LE HA PEDIDO QUE RECOJA LA BASURA (zumbadores) DE SU CASA Y SE LOS TIRE AL VECINO

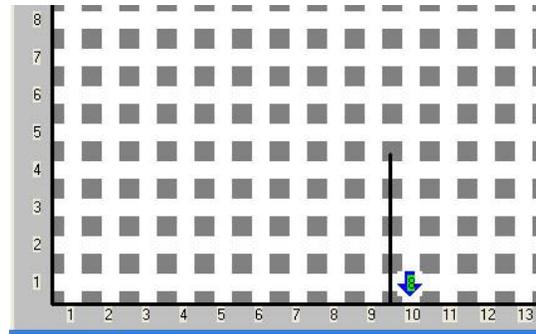
- NO SABEMOS CUANTA BASURA HABRA EN CADA LUGAR
- KAREL SABE QUE TERMINO DE RECOGER LA BASURA AL LLEGAR A LA BARRA DE SU CASA
- LA BARRA TIENE UNA ALTURA INDETERMINADA.
- KAREL DEBERA DEJAR LA BASURA EN EL PISO DE LA CASA DEL VECINO
- NO IMPORTA HACIA DONDE QUEDA APUNTANDO KAREL

EJEMPLO:

MUNDO INICIAL



MUNDO FINAL



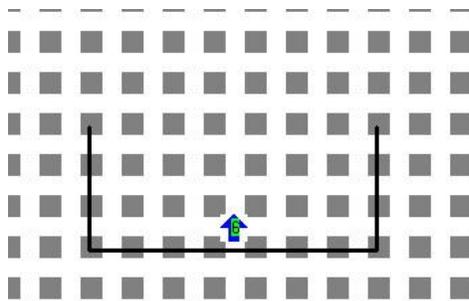
PROBLEMA 3 “REPARTIENDO JUGUETES”

KAREL ESTA JUGANDO CON SU HERMANA KAREN, TIENEN JUGUETES Y LOS VAN A REPARTIR, AYUDALE A KAREL

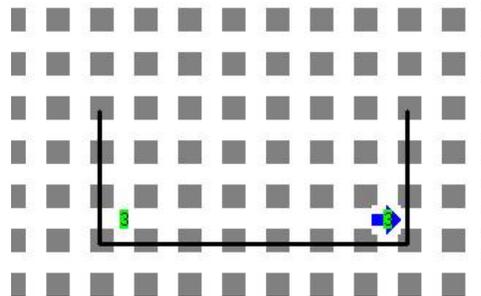
- KAREL INICIA APUNTANDO HACIA EL NORTE SOBRE UN MONTON DE JUGUETES (representado por zumbadores).
- KAREL DEBERA RECOGERLOS Y REPARTIRLOS, LOS JUGUETES DE KAREL EN LA ESQUINA IZQUIERDA Y LOS DE SU HERMANA KAREN EN LA ESQUINA DERECHA
- SI EXISTE UN NUMERO IMPAR DE JUGUETES, KAREL DECIDE COLOCAR EL ULTIMO JUGUETE EN SU ESQUINA.
- NO IMPORTA HACIA DONDE QUEDA APUNTANDO KAREL

EJEMPLO:

MUNDO INICIAL



MUNDO FINAL



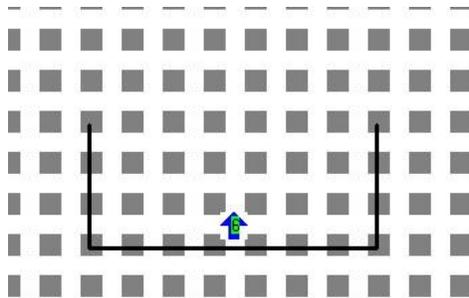
PROBLEMA 4 “REPARTIENDO JUGUETES EN TORRE”

KAREL ESTA JUGANDO CON SU AMIGO ARMANDO, TIENEN JUGUETES Y LOS VAN A REPARTIR, AYUDALE A KAREL

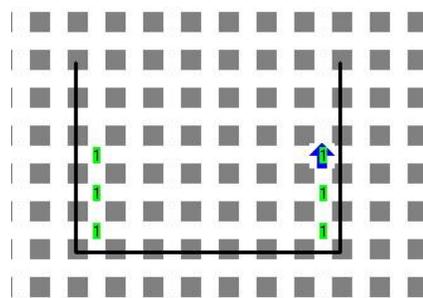
- KAREL INICIA APUNTANDO HACIA EL NORTE SOBRE UN MONTON DE JUGUETES (representado por zumbadores).
- KAREL DEBERA RECOGERLOS Y REPARTIRLOS, LOS JUGUETES DE KAREL EN LA ESQUINA IZQUIERDA FORMANDO UNA TORRE Y LOS DE ARMANDO EN LA ESQUINA DERECHA TAMBIEN FORMANDO UNA TORRE
- SI EXISTE UN NUMERO IMPAR DE JUGUETES, KAREL DECIDE COLOCAR EL ULTIMO JUGUETE EN SU ESQUINA.
- NO IMPORTA HACIA DONDE QUEDA APUNTANDO KAREL

EJEMPLO:

MUNDO INICIAL



MUNDO FINAL



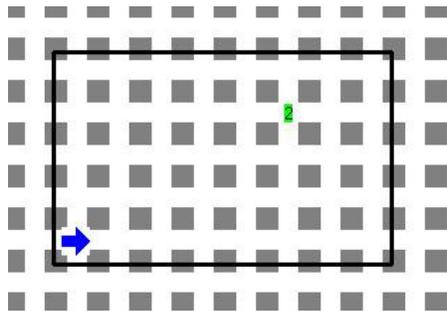
PROBLEMA 5 “EL SECUESTRO”

KAREL ESTA MAS QUE ENOJADO POR EL PROBLEMA EN EL QUE ARMANDO LO A METIDO Y HA DECIDIDO TOMAR CARTAS EN EL ASUNTO, SE HA ARMADO DE VALOR Y DE ZUMBADORES (8) Y SE HA IDO A BUSCAR A MANDO (2) PARA SECUESTRARLO (encerrarlo entre zumbadores), AYUDALE A KAREL.

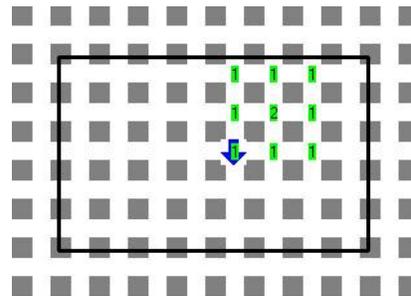
- EL MUNDO ES EN FORMA RECTANGULAR DELIMITADO POR PAREDES
- KAREL INICIA APUNTANDO HACIA EL ESTE EN LA ESQUINA INFERIOR IZQUIERDA
- KAREL DEBERA IR A DONDE ESTA ARMANDO (2) Y ENCERRARLO
- NO IMPORTA HACIA DONDE QUEDA APUNTANDO KAREL

EJEMPLO:

MUNDO INICIAL



MUNDO FINAL



PROBLEMA 6 "ATINALE AL MONTON"

Historia

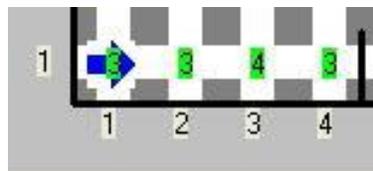
Para contentar a Gretel por haberse comido su helado, Karel ha decidido intentar ganar un peluche para su hermana en el juego "Atínale al montón". El juego consiste en escoger de entre fila de montones de zumbadores, el montón que tiene más zumbadores. Todos los montones tienen la misma cantidad de zumbadores excepto el que tiene el peluche que tiene exactamente uno más.

Problema

Escribe un programa que haga que Karel termine en la posición del montón que tiene el peluche (el montón de más zumbadores).

Consideraciones

- Los montones estarán entre dos paredes sobre una misma calle.
- Karel inicia sobre el primer montón mirando al resto de los montones.
- Puedes recoger los zumbadores si quieres.
- No importa la orientación del peluche ni de Karel.



Ejemplo de caso de prueba

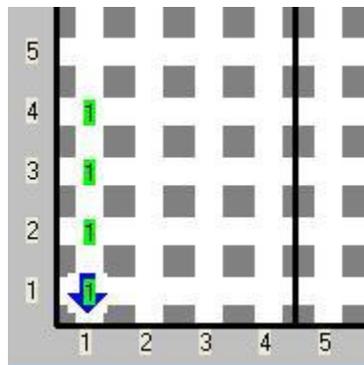
PROBLEMA 6 “MIDIENDO LA DISTANCIA”

Problema

Debes escribir un programa que permita que Karel mida la distancia entre su posición inicial y una pared. Como resultado, Karel deberá dejar en la esquina inferior izquierda del mundo una pila de zumbadores de altura igual a la distancia entre la posición inicial de Karel y la pared.

Consideraciones

- Inicialmente Karel se encuentra orientado hacia el ESTE en la esquina inferior izquierda de su mundo.
- A la derecha de Karel, a una distancia desconocida hay una pared vertical.
- Karel Tiene 100 zumbadores en su *mochila*.
- No importa la dirección y posición con la que termine Karel.



Solución cuando la barrera está a 4 avenidas

REFERENCIAS

- Richard E. Pattis, (1981). Introducción gradual a la programación, el robot karel
- Brusilovsky, P., Calabrese, E., Hvorecky, J., Kouchnirenko, A., and Miller, P. (1997) Mini-languages: A Way to Learn Programming Principles. Education and Information Technologies. <http://www.contrib.andrew.cmu.edu/~plb/papers/minilang.html>
- Tom, Mitchell. (July 27, 2000) Karel The Robot <http://karel.sourceforge.net/>
- Tom, Mitchell. (July 27, 2000) Karel The Robot http://karel.sourceforge.net/doc/html_mono/karel.html
- Richard E. Pattis, "Introducción Gradual a la programación, El Robot Karel", Ed. México: Limusa, 2006, pp. Prologo.
- OMI, Olimpiada Mexicana de Informática.