

Lenguaje de Programación: Ciclos y Condicionales (condicionales)

José Luis Alonzo Velázquez

Universidad de Guanajuato

Sesión 6

Condicionales

Cuando el Dr. Isaac Karelov construyo a Karel, le dio la habilidad de reconocer si había una pared o si estaba parado sobre un beeper, hacia que dirección estaba mirando, y saber si contaba con beepers o no en su beeper bag.

Estas habilidades le permitirán a Karel reconocer un poco de su mundo y tomar decisiones acordes a su entorno sin necesidad de conocerlas previamente.

Karel puede conocer si hay pared a un paso de distancia a través de su cámara frontal. La condicional **frontIsClear** es una función que pregunta al sensor de Karel, si hay una pared enfrente de él. A diferencia de las instrucciones que definimos anteriormente, una condicional no ejecuta un movimiento, en lugar de ello, proporciona una respuesta de **SI** o **NO** a la pregunta hecha. Una condicional retorna el valor de uno cuando la respuesta a la pregunta es un **SI** y retorna un cero cuando la respuesta es **NO**.

Definición

Los condicionales también se conocen como *Funciones Booleanas* debido a sus dos posibles respuestas.

La sintaxis de if es:

```
if(<condición>)  
  {  
    <instrucción>  
    <instrucción>  
    :  
    <instrucción>  
  }
```

Nota:

Cuando sólo queremos que se realice una sola instrucción, es posible omitir las llaves “{” y “}”. Sin embargo recomiendo que se utilicen al principio, aunque sólo sea una instrucción.

Ejemplo: Cosechando la calle.

En una calle se encuentran beepers(a lo más uno en cada esquina) y queremos que Karel los recolecte. Sin embargo, la calle puede o no contener beepers en cada esquina y en un principio solo sabemos que debe recorrer 10 esquinas.

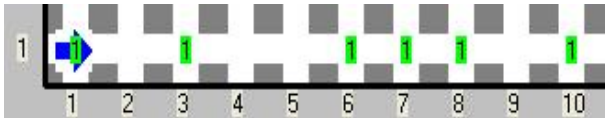


Figura: Calle de beepers.

Completando Pilas

Nuestro amigo Karel quiere colocar pilas de dos beepers como se muestra en la figura. Es decir, poner dos beepers en la avenida donde no hay.

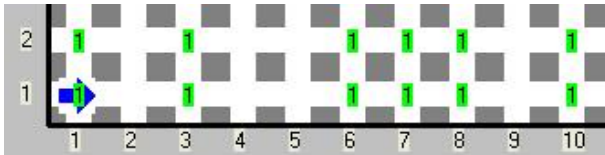


Figura: Pilas de beepers que deben ser completadas.

frontIsClear	es verdadero si el frente de Karel esta libre de pared
frontIsBlocked	es verdadero si existe una pared frente a Karel
leftIsClear	es verdadero si el lado izquierdo de Karel esta libre de pared
leftIsBlocked	es verdadero si en el lado izquierdo de Karel existe una pared
rightIsClear	es verdadero si el lado derecho de Karel esta libre de pared
rightIsBlocked	es verdadero si en el lado derecho de Karel existe una pared
nextToABeeper	es verdadero si existe un beeper donde Karel esta parado
notNextToABeeper	es verdadero si no existe un beeper donde Karel esta parado
facingNorth	es verdadero si Karel esta volteando al norte
notFacingNorth	es verdadero si Karel no esta volteando al norte
facingSouth	es verdadero si Karel esta volteando al sur
notFacingSouth	es verdadero si Karel no esta volteando al sur
facingEast	es verdadero si Karel esta volteando al este
notFacingEast	es verdadero si Karel no esta volteando al este
facingWest	es verdadero si Karel esta volteando al oeste
notFacingWest	es verdadero si Karel no esta volteando al oeste
anyBeeperInBeeperBag	es verdadero si existe algún beeper en la beeper bag



Edgar Alfredo Duñez Guzmán & Edgar Said Hernández
Sánchez & Marte Alejandro Ramírez Ortega *Los Dilemas de
Karel*. CIMAT, Mayo 2006.