**Algoritmo: movimiento del enemigo.**

**Entrada: entero n que representa la dirección del movimiento.**

1:arriba   
2:izquierda   
3:abajo   
4:derecha

boolean mov = true  
i ← enemigo.getFila()  
j ← enemigo.getColumna()  
cantidadColumnas ← matriz.length  
cantidadFilas ← matriz[0].length

Mientras (mov)

Si n es:

1: si i-1>=0 y matriz[i-1][j] == vacio entonces

tanqueEnemigo.setColumna (i-1)

matriz[i][j] 🡨 vacio

matriz[i-1][j] 🡨 enemigo

sino

mov 🡨 false

2: si j-1>=0 y matriz[i][j-1] == vacio entonces

tanqueEnemigo.setFila (j-1)

matriz[i][j] 🡨 vacio

matriz[i][j-1] 🡨 enemigo

sino

mov 🡨 false

3: si i+1 < cantidadFilas y matriz[i+1][j] == vacio entonces

tanqueEnemigo.setColumna (i+1)

matriz[i][j] 🡨 vacio

matriz[i+1][j] 🡨 enemigo

sino

mov 🡨 false

4: si j+1<cantidadColumnas y matriz[i][j+1] == vacio entonces

tanqueEnemigo.setFila (j+1)

matriz[i][j] 🡨 vacio

matriz[i][j+1]<-enemigo

sino

mov 🡨 false

int m 🡨 random entre 1 y 4

invocar este algoritmo con m