

# Cleaning Robots Simulation

Rodrigo García Gómez CC-412

## El problema

Se tiene un ambiente de información completa dado por un tablero rectangular de  $N \times M$  casillas. Cada  $t$  turnos el ambiente realiza un cambio aleatorio dado por los movimientos de los niños y sus respectivas generaciones de suciedad. Cada turno los agentes ejecutan sus acciones modificando el medio. Cada vez que se alcanza el turno de cambios aleatorios en el ambiente, estos ocurren junto con las acciones de los agentes en una misma unidad de tiempo. Los elementos que pueden existir en el ambiente son obstáculos, suciedad, niños, corral y los robots (que son los agentes).

1. Obstáculos:

Ocupan una única casilla. Pueden ser movidos sólo por los niños. No pueden ser movidos a ninguna casilla ocupada por algún otro elemento del ambiente.

2. Suciedad:

Ocupan una única casilla del ambiente. Es generada por los niños y sólo puede surgir en una casilla previamente vacía. También pueden aparecer en el estado inicial del tablero.

3. Corral:

El corral ocupa casillas adyacentes en número igual al total de niños presentes en el ambiente. El corral no puede ser desplazado. En una casilla del corral puede coexistir un único niño. En una casilla vacía del corral puede entrar un robot. En una casilla del corral pueden coexistir al mismo tiempo un niño y un robot sólo si el robot está cargando al niño o si lo acaba de soltar.

4. Niño:

Ocupa una única casilla. Durante el turno de cambio en el ambiente se mueven de forma aleatoria de a una casilla adyacente ser posible (No se puede mover a casillas ocupadas por otros niños, robots, suciedad o el corral). Si la casilla objetivo a moverse está ocupada por un obstáculo, este se desplaza en la dirección de la que venía el niño. En el caso de haber más obstáculos en esa dirección, todos son desplazados a menos que el más lejano de estos no pueda ser desplazado por estar ocupada la casilla objetivo. Luego de mover un obstáculo el niño ocupa su posición. Los niños son responsables de que aparezca la suciedad. La suciedad es generada de forma aleatoria en la cuadrícula de  $3 \times 3$  centrada en la casilla del niño, luego de que este se mueva hacia alguna de sus casillas adyacentes. Si antes de realizar su movimiento, la cuadrícula tenía sólo

al niño en la que se centraba, entonces se genera hasta una suciedad. Si además del niño en el centro había otro niño en la cuadrícula, se generan hasta 3 suciedades. Si había más de un niño además del del centro, se generan hasta 6 suciedades. Al estar encerrados en una casilla del corral, o atrapados por un robot, los niños no pueden moverse ni ensuciar.

#### 5. Robot de casa:

Es el encargado de limpiar y controlar a los niños. El robot decide en cada momento que acción realizar. Si no carga a un niño, puede desplazarse sólo una a una casilla adyacente. Si carga a un niño, puede desplazarse hasta dos casillas consecutivas. Puede también realizar acciones de limpiar y cargar niños. Si se mueve a una casilla con suciedad, en el próximo turno puede decidir si limpiar o moverse. Si se mueve a una casilla ocupada por un niño, inmediatamente lo carga. Si se mueve a una casilla vacía del corral y lleva a un niño cargado, puede dedicar un turno a soltarlo o puede moverse y mantenerlo cargado. Al ser dejado el niño en una casilla coexisten hasta el otro turno el robot y el niño, sin este ser cargado por el primero.

El Objetivo del robot es mantener la casa limpia. La casa se considera limpia si menos del 60 porciento de las casillas vacías están ocupadas por suciedad.

## Instrucciones de uso

Para ejecutar el proyecto debe importar el script *main.hs* ubicado dentro de */app*. Luego se llama a la función **main**.

---

```
*Main Lib Paths_agents> main
```

---

A continuación se pedirá al usuario dos entradas. Primero un número entero que se utiliza para seleccionar el tablero. El número 0 generará un tablero aleatorio de tamaño 8x8. Los números 1-4 se utilizan para seleccionar uno de los tableros perdefinidos en el cuerpo de la función: **selectInitialBoard :: Int -> IO Board**. Cualquier número mayor que 4 ingresado generará un tablero completamente vacío.

La segunda entrada representará la cantidad de turnos que deben pasar durante la simulación (en cada turno los robots de limpieza hacen sus movimientos) antes de que ocurra un cambio aleatorio en el ambiente (Los niños harán sus movimientos y provocarán el surgimiento de nuevas casillas con suciedad)

---

```

"insert index of initial board"
0
"insert number of iterations to shuffle"
5

```

---

A partir de este punto, cada vez que el usuario presione "enter", se llevará a cabo un turno y se imprimirá en la consola una representación del tablero con los resultados de los movimientos de los agentes (y del cambio en el ambiente si corresponde) durante el mismo. Los símbolos en el tablero representan los siguientes elementos:

$Empty >$  Casilla vacía

$(Obstacle) >$  Casilla ocupada por un obstáculo

$\sim\sim Dirt \sim\sim >$  Casilla con suciedad

$\{Corral\} >$  Casilla ocupada por un corral

$\{- - C + K - -\} >$  Casilla ocupada por un corral con un niño dentro

$\{[C + R]\} >$  Casilla ocupada por un corral con un robot dentro

$\{[- - C + K + R - -]\} >$  Casilla ocupada por un corral con un robot dentro mientras el robot lleva un niño

$\{[- - C + R => K - -]\} >$  Casilla ocupada por un corral con un robot dentro que acaba de soltar al niño que llevaba

$- - Kid - - >$  Casilla ocupada por un niño

$[Robot] >$  Casilla ocupada por un robot

$[\sim\sim R + D \sim\sim] >$  Casilla con suciedad ocupada por un robot

$[- - R + K - -] >$  Casilla ocupada por un robot que lleva a un niño

$[- - \sim\sim R + K + D \sim\sim - -] >$  Casilla con suciedad ocupada por un robot que lleva a un niño

Para detener la simulación, antes de presionar "enter" durante cualquier turno puede escribir `abort` en la consola. Además, la simulación se detendrá automáticamente si la cantidad de casillas con suciedad supera el 60 por ciento del total de casillas del tablero o si todos los niños se encuentran dentro del corral o siendo llevados por robots y no queda ninguna casilla sucia (al estar todos los niños atrapados, ya no se podrá generar suciedades nuevas).

## Detalles de implementación

El ciclo principal se ejecuta en la función `mainLoop :: Int -> Int -> Board -> IO ()`. En cada iteración se realiza primero las acciones de los agentes mediante la función `moveRobots :: [(Int, Int)] -> Board -> Board` que recibe