

Proyecto Agentes

José A. Labourdette - Lartigue Soto C-12

[Enlace al repositorio de GitHub](#)

Problema

Dado un conjunto de robots contenidos en un tablero de $n \times m$, que a su vez contiene niños, cunas, obstáculos y suciedades; se desea simular la acción de dichos robots para mantener la limpieza en el tablero. El tablero nunca debe exceder el 30% de suciedad y para ello los robots pueden limpiar suciedades o llevar niños hacia las cunas sin pasar por los obstáculos; siendo los niños responsables de generar suciedad a su paso. Los niños que estén siendo transportados, o que hayan sido dejados en las cunas pierden la capacidad de moverse y por tanto producir suciedad. Los niños que tengan la capacidad de moverse aún, lo harán luego que sucedan t turnos de los robots, a la par que se genera una cantidad de suciedad preestablecida independientemente de la generada por los niños movidos.

Ideas para los Robots (Agentes)

La idea general será que los agentes realicen operaciones (tanto de limpieza como de llevar niños a su cuna). Dichas operaciones se definen como objetivos del robot. El robot entonces cumplirá dicho objetivo y proseguirá al siguiente. Este proceso continuará hasta q logren controlar a todos los niños (ganar) o verse desbordados por suciedad (perder)

Sobre el movimiento

El movimiento de los robots admite solamente las posiciones superior, inferior, izquierda y derecha de la casilla que actualmente ocupa, con excepción de los robots que estén cargando algún niño: capaces de avanzar 2 casillas consecutivas en dichas direcciones.

Sobre la selección de objetivos

Quizás la idea más básica que se pueda ocurrir es mover al robot de manera aleatoria por el terreno hasta que se vea encima de una suciedad o un niño y entienda que hay que limpiarlo o llevarlo a la cuna respectivamente. Sin embargo se desaprovecha la propiedad de que cada robot conoce exactamente como se encuentra distribuido el terreno en ese instante, para todos los instantes. Si se selecciona una casilla que implique una tarea y se decide mover al robot hasta dicha casilla, entonces perderíamos la aleatoriedad del movimiento, que en los peores casos podría nunca encontrar nada. La forma en que se selecciona dicho objetivo es un tema debatible, pues formas distintas suponen tanto ventajas como desventajas. En este proyecto se selecciona ese objetivo de manera greedy,

pues los robots seleccionan el objetivo que más cerca le quede, siendo la cantidad de pasos de distancia la función de costo.

Sobre la comunicación entre Agentes

Pero qué sucede si dos o más Agentes seleccionan la misma casilla como objetivo. Eventualmente uno de los Agentes llegará a dicha casilla, realizará la tarea y los demás agentes computarán un nuevo objetivo al que saldrán a cumplir. El resto de los agentes perdió tiempo en un objetivo que podían saber irrealizable antes que su colega lo cumpliera. Para evitar estos movimientos innecesarios cada Agente calcula su próximo objetivo conociendo los objetivos de sus colegas. En caso de que alguno lo tenga, comprueba quien tenga menor cantidad de pasos hasta llegar ahí, ese se queda con el objetivo y el otro tiene que recalcular sus objetivos: buscando el que más cerca esté, tal que no sea el de su compañero.

Sobre recalcular sus objetivos

```
| |B| |  
|A| |a|  
| |b| |
```

Veamos este ejemplo: los agentes A y B han calculado sus objetivos y los resultados son desplazarse hasta a y b respectivamente, ambos con costo 2. Cuando se mueve el agente A este se desplaza una casilla a la derecha. Entonces el agente B ya no tendría su objetivo b a 2 pasos, y podría tener algún tercero más cerca que la nueva cantidad de pasos. Por esto, además de la rectificación de objetivos propia de la comunicación entre agentes se rectifica además justo antes de moverse. Además de evitar errores como este es capaz de identificar objetivos que antes podían haber estado obstruidos.

Ideas para simular Ambiente

Cuando un turno del ambiente se ejecuta los niños pueden moverse, pero además generarán suciedad en una cuadrícula de 3x3. Atendiendo a la cantidad de niños en dicha cuadrícula será la cantidad de suciedad que se pueda generar en ella. En este trabajo ningún niño se queda sin incluir en alguna cuadrícula.

Sobre la selección de cuadrículas

Dado un niño se buscarán todas las cuadrículas posibles y se seleccionará una al azar entre ellas. Las cuadrículas solo son posibles si todas las casillas están inscritas en el tablero

Sobre asociar niños a las cuadrículas

Dada una cuadrícula se buscan todos los niños inscritos en ella y justo sobre ellos se aplica el movimiento aleatorio. Luego se procede a generar la suciedad

atendiendo a los niños que fueron seleccionados en la cuadrícula. Los niños que fueron procesados se retiran de la lista de niños a procesar, pues en esa lista solo estarán aquellos que no han sido incluidos en alguna. Se selecciona un niño de estos, se obtiene una de sus cuadrículas y se repite el proceso hasta agotar los niños.

Proceso de elegir los objetivos

Para elegir el objetivo de un agente habíamos visto que se selecciona de todos las tareas posibles para él, aquella cuya cantidad de pasos sea mínima (y además no exista otro agente con esa tarea y menor costo). Pero para esto tenemos que computar la cantidad de pasos que da un agente hasta cada una de las posibles tareas. Para lograr esto se hace un recorrido BFS partiendo del robot y hasta que se alcance un objetivo válido. De no encontrarse ninguna tarea, entonces se entiende que el robot no puede realizar ninguna acción en este turno y para ello se genera un objetivo con tipo especial que le indica que no va a realizar acción alguna. Son objetivos siempre válidos la limpieza de suciedades. Sin embargo la recogida de niños es solo válida si el robot no está cargando algún otro actualmente. La acción para dejar niños en el corral solo es válida con una condición totalmente contraria a la anterior. De esta forma cada robot sabe cuál es la casilla donde está la tarea que va a realizar. Sin embargo no sabe cuál es el camino correcto hasta ella. Para solucionar esto computaremos la primera casilla a la cual tiene que moverse para realizar un camino con cantidad de pasos mínima. La forma de realizar este cómputo es realizando precisamente un recorrido BFS partiendo de la casilla donde está la tarea y hasta llegar a la casilla del robot actual. El camino hasta llegar a la tarea tiene que tener en cuenta si el robot está cargado o no. De esta forma se tienen en cuenta los pasos dobles o no.

Forma de mostrar el Tablero

Para mostrar el tablero cada casilla se compone de cinco espacios, cada uno representa, por la letra de la inicial, el elemento que se encuentra en esta casilla. Estos elementos pueden ser: * k (Kid) : Niño * r (Robot) : Robot * o (Obstacle) : Obstáculo * d (Dirt) : Suciedad * c (Crib) : Cuna

Todos los elementos se representan con letra minúscula a no ser que se encuentren cargando un niño. Por esta razón los elementos que también pueden ser representados por la letra inicial Mayúscula son el robot y la cuna. Esto es un tablero de 7x7 impreso por el programa:

```

|   |   |   | o |   |   |   |   | | |
|   |   | o |   | d |   | d |   |   |
|   |   |   |   | r |   |   | d |   |
|   |   |   | k | C |   | c | k | C |   |
| r |   |   |   |   |   | o |   |   |   |
|   | kR |   | d |   |   |   |   |   |

```

| | | | a | | |

Para ejecutar simulaciones

Para correr el proyecto ejecutar en una terminal abierta en la raíz del proyecto el comando:

```
$ stack run
```

Si se desean modificar parámetros de la simulación entonces modificar Simulation Config en el archivo app/Main.hs

Conclusión del Ambiente

El Ambiente en este proyecto podemos concluir posee las siguientes características:

- Accesible : En todo momento los agentes tienen completa información sobre el estado del ambiente
- No Determinista : La dirección de movimiento de los niños es de manera manera aleatoria, además de la generación de suciedades tanto en las cuadrículas de 3x3 como en el tablero en general. Estrictamente hablando no es una aleatoriedad, sino una pseudo-aleatoriedad pues conociendo el seed con el que se generan los números, entonces es posible predecir el tablero hasta el propio final de la simulación. Pero la intención es seleccionar elementos de manera aleatoria.
- Dinámico : El ambiente se somete a cambios cada una cantidad prefijada de turnos de los agentes
- Discreto : Las posibles modificaciones al ambiente se suceden de una lista conocida de posibles modificaciones y alteraciones.

Conclusión de los Agentes

Podemos definir los Agentes como:

- Reactivos : Pues perciben el ambiente y responden de un modo oportuno a los cambios que ocurren para lograr sus objetivos.
- Pro-Activos : Pues cuando se fijan un objetivo realizan las acciones necesarias para vencerlo. Dichas acciones pueden ser caminar hacia el niño, cargarlo y llevarlo a la cuna. Todo como parte del mismo proceso
- Sociables : A la hora de seleccionar sus objetivos los agentes negocian y cooperan entre ellos, para lograr lo mejor, el bien común

De esta manera se puede concluir que los agentes de este proyecto son agentes inteligentes.