

Proyecto de Simulación

Agentes

Annie Hernández Pérez

1. Recopilación

En este proyecto se tienen uno o varios equipos de agentes (los equipos pueden ser de un solo agente). El objetivo de cada equipo es recolectar la mayor cantidad de recursos posibles en un número determinado de turnos. Los recursos deben ser transportados a un "almacén central".

Objetos:

- Agentes: Aparecen en posiciones aleatorias al comienzo de la ejecución. Pueden tener dos estados, vacío o lleno.
- Recursos: Inicialmente aparece una cantidad fija aleatoriamente distribuida por todo el mapa. A medida que transcurran los turnos irán apareciendo nuevos recursos en casillas aleatorias.
- Almacén Central: Aparecen en posiciones aleatorias del mapa, cada equipo tiene un único almacén central. Un almacén puede guardar una cantidad infinita de recursos.

Acciones que puede realizar los agentes:

- No hacer nada.
- Moverse (Arriba, Abajo, Derecha Izquierda): Si la acción implica salirse de las dimensiones del mapa, colocarse arriba de otro agente, almacén o recurso entonces no se ejecutará la acción.
- Almacén Central: Aparecen en posiciones aleatorias del mapa, cada equipo tiene un único almacén central. Un almacén puede guardar una cantidad infinita de recursos.
- Recopilar: el agente recopila un recurso de una posición adyacente. Se considera posición adyacente a las ocho casillas que rodean al agente. Si no hay un recurso en una posición adyacente o el agente ya está cargando un recurso entonces no se ejecutará la acción. Si hay más de un recurso que pueda ser recopilado entonces se escogerá, de entre los posibles, uno de forma aleatoria.
- Descargar: el agente descarga un recurso que está cargando en el almacén. Esta acción solamente se ejecutará si el almacén se encuentra en una casilla adyacente al agente y el agente está cargando un recurso.

Nota: Si el agente escoge una acción que no puede ser realizada entonces el ambiente permanecerá invariable y el agente perderá su turno.

Condición de Victoria:

Ganará el equipo que logre recopilar la mayor cantidad de recursos.

2. Solución

La solución al problema consiste en una implementación en *C#* de un programa que recibe la cantidad de agentes de cada tipo que se desea que haya y la cantidad de equipos en que se quiere que se dividan. Tiene como parámetros además la dimensión que tendrá la cuadrícula que representa al mapa y la cantidad de turnos que se efectuarán para determinar al equipo ganador.

Se implementó una clase *Map* que es la encargada de inicializar el mapa que representa el estado actual, distribuyendo aleatoriamente en él los recursos, los almacenes (uno por cada equipo) y los agentes deseados; además en cada turno coloca los nuevos recursos que aparecen aleatoriamente y controla que no se realicen acciones ilegales. La clase *Warehouse* representa el almacén central, esta tiene un contador de la cantidad de recursos que se han almacenado en él y el control del equipo al que pertenece. *Program* es la clase que controla la ejecución, manda a crear el mapa y todas las instancias necesarias y efectúa la cantidad de turnos seleccionados dándole la oportunidad de realizar sus acciones a cada agente, eligiendo el orden aleatoriamente en cada turno; también chequea al final cuál es el equipo ganador.

La clase *Agent* contiene la información básica, común a ambos tipos de agentes. Conoce la posición en que se encuentra el agente en todo momento, el equipo al que este pertenece y el almacén central del mismo, además de un estado para saber si el agente está lleno o vacío y una representación del mapa. Contiene un método para analizar las 8 casillas adyacentes al agente en busca de un elemento determinado (ya sea un recurso o el almacén) y otro que busca en las 4 direcciones en que puede moverse el agente para encontrar casillas vacías, ambos devuelven una lista con todas las posiciones disponibles. De esta heredan las clases *ReactiveAgent* y *ProactiveAgent*.

Los agentes reactivos van a agregar un indicador para saber si ya han encontrado su almacén y dos tipos de marcas que realizan en el mapa para encontrar los mejores caminos desde cualquier lugar del mapa hasta su almacén. Estos van a moverse aleatoriamente hacia cualquier casilla vacía hasta encontrar un recurso en una casilla adyacente y entonces lo recogerán. Luego volverán a moverse aleatoriamente en busca de su almacén, pero ahora tendrán en cuenta las marcas en el mapa: si hay una marca de tipo 2 en una casilla a la que puede moverse se moverán hacia allí y va a convertirla en una de tipo 1 (para que los caminos no sean permanentes y se queden los más cortos), si no la hay pero hay una de tipo 1 pues va hacia esta y borra la marca, sino se mueve aleatoriamente hasta encontrar el almacén donde descargará el recurso. Una vez que esto ocurra además de salir aleatoriamente a encontrar recursos van a comenzar a dejar marcas de tipo 2 por cada casilla del mapa que pase hasta que encuentre

un recurso y tengan que regresar. En resumen el comportamiento de un agente reactivo es el siguiente:

- Si está vacío y hay al menos un recurso en una posición adyacente: elegir uno aleatoriamente y recogerlo.
- Si está vacío y hay al menos una de las 4 casillas a la que puede moverse que esté vacía y sin marcas: elegir una aleatoriamente y moverse a ella. Si además ya encontró el almacén hacer en la casilla una marca de tipo 2.
- Si está vacío y hay al menos una de las 4 casillas a la que puede moverse que tiene una marca tipo 1: elegir una aleatoriamente y moverse a ella. Si además ya encontró el almacén convertir esta en una marca de tipo 2.
- Si está vacío y hay al menos una de las 4 casillas a la que puede moverse que tiene una marca tipo 2: elegir una aleatoriamente y moverse a ella.
- Si está lleno y el almacén de su equipo se encuentra en una posición adyacente: descargar el recurso.
- Si está lleno y hay al menos una de las 4 casillas a la que puede moverse que tiene una marca tipo 2: elegir una aleatoriamente y moverse a ella. Convertir esta marca en una de tipo 1.
- Si está lleno y hay al menos una de las 4 casillas a la que puede moverse que tiene una marca tipo 1: elegir una aleatoriamente y moverse a ella. Además borrar esta marca.
- Si está vacío y hay al menos una de las 4 casillas a la que puede moverse que esté vacía y sin marcas: elegir una aleatoriamente y moverse a ella.

Por otro lado los agentes proactivos van a tener como información adicional en que turno se encuentran y si en el turno anterior cambiaron su estado (cambiaron de llenos a vacíos o viceversa). Estos agentes van a buscar el recurso hacia el que tienen el camino más corto y van a empezar a realizar ese camino, si en ese proceso se encuentran un recurso en sus casillas adyacentes (nuevo recurso que apareció al final de un turno) van a recogerlo. Cada tres turnos el agente va a volver a buscar el recurso más cercano ya que el anterior puede haber sido tomado por otro agente. Una vez lleno el agente va a buscar el camino más corto hacia su almacén y va a moverse por el mismo. En resumen el comportamiento de un agente proactivo es el siguiente:

- Si está vacío y hay al menos un recurso en una posición adyacente: elegir uno aleatoriamente y recogerlo.
- Si está vacío y acabó de vaciarse o el turno es un múltiplo de 3: calcular el camino más corto hacia el recurso más cercano y moverse en el primer paso de este camino.
- Si está vacío y la casilla a la que corresponde moverse en el camino está vacía moverse hacia ella, sino recalcular el camino hacia el recurso más cercano y moverse por él.
- Si está lleno y el almacén de su equipo se encuentra en una posición adyacente: descargar el recurso.

- Si está lleno y acabó de llenarse o el turno es un múltiplo de 3: calcular el camino más corto hacia el almacén y moverse en el primer paso de este camino.
- Si está lleno y la casilla a la que corresponde moverse en el camino está vacía moverse hacia ella, sino recalcular el camino hacia el almacén y moverse por él.

En general los agentes proactivos tuvieron un mejor comportamiento que los reactivos, ganando en más ocasiones. Sin embargo en ejemplos pequeños y medianos donde había un número considerable de recursos ambos modelos de agentes presentaron un comportamiento similar, fue en ejemplos relativamente grandes donde los recursos se presentaban esparcidos y se dejaban correr varios turnos, que los agentes proactivos tuvieron un mejor comportamiento.