Simulación de la Agente Aspiradora (Proyecto-5)

El proyecto consiste en escribir una agente inteligente para el mundo de una aspiradora, similar a la que vimos en los capítulos 2, 3 y 4 del libro. El ambiente de tarea de la aspiradora esta implementado en un simulador. Como la agente aspiradora del libro, esta agente es capaz de aspirar toda la suciedad de la cuadricula donde se encuentra. La aspiradora (agente) y su entorno tienen las siguientes características:

- La aspiradora se ocupa de un cuarto de 5 x 5 cuadriculas; ya no solo de dos cuadriculas como en el libro.
- La agente tiene una localización inicial posiblemente diferente cada que la simulación comienza (diferente cuadricula dentro del cuarto).
- Hay obstáculos en el cuarto; el agente no sabe de esos obstáculos con anticipación antes de que los detecte con sus sensores.
- La agente tiene sensores para detectar obstáculos inmediatamente en frente de donde se encuentra.
- La agente puede detectar si accidentalmente se golpeó con un obstáculo.
- La agente puede detectar si hay suciedad en la cuadricula en donde se encuentra.
- La agente es indicada por la letra "**A**"; cuando orientada hacia el norte, con el símbolo "**>**" cuando orientada hacia el este, con el símbolo "**<**", cuando orientado hacia el oeste y la letra "**V**" cuando orientado hacia el sur.
- Una cuadricula sucia es indicada por "*****", y cada obstáculo es indicado por "**X**".
- Las paredes del cuarto son indicadas igual con "**X**" alrededor del perímetro.
- Estas indicaciones están implementadas en el simulador que se entrega como parte del proyecto.

Ejemplo de un "output" del simulador con la agente orientada al norte:

```
  0  1  2  3  4  5  6
 +--+--+--+--+--+--+--+
0|X |X |X |X |X |X |X |
 +--+--+--+--+--+--+--+
1|X |  |  |  |  |X |X |
 +--+--+--+--+--+--+--+
2|X |  |* |  |* |X |X |
 +--+--+--+--+--+--+--+
3|X |  |  |A |  |X |X |
 +--+--+--+--+--+--+--+
4|X |X |  |  |  |  |X |
 +--+--+--+--+--+--+--+
5|X |X |X |  |* |  |X |
 +--+--+--+--+--+--+--+
6|X |X |X |X |X |X |X |
 +--+--+--+--+--+--+--+
Location: (3,3)  Facing: NORTH
```

Ejemplo de un “output” del simulador con la agente orientada al este:

```
  0  1  2  3  4  5  6
 +--+--+--+--+--+--+--+
0|X |X |X |X |X |X |X |
 +--+--+--+--+--+--+--+
1|X |  |  |> |  |X |X |
 +--+--+--+--+--+--+--+
2|X |  |* |  |* |X |X |
 +--+--+--+--+--+--+--+
3|X |  |  |  |  |X |X |
 +--+--+--+--+--+--+--+
4|X |X |  |  |  |  |X |
 +--+--+--+--+--+--+--+
5|X |X |X |  |* |  |X |
 +--+--+--+--+--+--+--+
6|X |X |X |X |X |X |X |
 +--+--+--+--+--+--+--+
Location: (3,1)  Facing: EAST
```

La aspiradora tiene la capacidad de escoger entre las siguientes acciones:

(1) Puede aspirar la suciedad en la cuadricula que se encuentra

new SuckDirt();

(2) Puede moverse una cuadricula en la dirección que esta direccionada

new GoForward();

(3) Puede girar 90 grados a la izquierda

new TurnLeft();

(4) Puede girar 90 grados a la derecha

new TurnRight();

(5) Cuando la agente cree que todo el cuarto está limpio puede apagarse a si misma

new ShutOff();

Se te entrega el código fuente del simulador. Este código incluye la clase “myagent.VacAgent”. Esta clase extiende la clase abstracta “Agent”. La clase “myagent.VacAgent” es la clase que debes desarrollar. Los métodos que tienes que implementar son los siguientes:

1. public void see(Percept p)
2. public Action selectAction()

La agente solo cuenta con la información que el simulador le da atreves del método “public void see(Percept p)”. El simulador invoca ese método y también a “public Action selectAction()”. El método “public void see(Percept p)” ha sido desarrollado para obtener todas las percepciones que el simulador le manda al agente.

El simulador califica el desempeño del agente; un “performance measure”. Para más detalles de cómo se determina el desempeño del agente en términos del puntaje, por favor leer el método donde se calcula la medida de rendimiento en el método “getPerformanceMeasure()”; en la clase “VacuumWorld”. Para más detalles de cómo se integra la clase “myagent.VacAgent” al simulador, leer los comentarios de la clase “agent.Agent”. Esta clase es la súper clase de “myagent.VacAgent”.

El simulador usa los siguientes parámetros para configurar su comportamiento:

- “java vacworld.VacuumWorld [-batch]” hace que el simulador corra sin pausa hasta el final; sin ese argumento el agente corre pausando después de cada acción, esperando que se teclee la tecla “Enter” para seguir con la siguiente acción.
- “java vacworld.VacuumWorld [-rand *seed*]” hace que el agente genere estados pseudoaleatorios; la *seed* es la semilla que es utilizada para el generador de números pseudoaleatorios. Es importante probar el código con diferentes semillas, “seeds”, para asegurarse que la solución es robusta.

La simulación con cualquiera de las opciones se termina automáticamente si es que el numero de movimientos acumulados de la agente llega a 200 movimientos. Si el programa es corrido manualmente, sin la opción –batch, el usuario del simulador puede teclear la tecla “Q” o “q” para “quit”.

La idea es obviamente obtener el mejor desempeño de la agente según la métrica del simulador, a la vez que el cuarto quede completamente limpio. El código inicial en la clase “myagent.VacAgent” es solo para ayudarles a entender el mecanismo del simulador. Cambiar el código **solo** en esa clase.