| /\* |  |
| --- | --- |
|  |  |
|  | \* AP(r) Computer Science GridWorld Case Study: |
|  | \* Copyright(c) 2005-2006 Cay S. Horstmann (http://horstmann.com) |
|  | \* |
|  | \* This code is free software; you can redistribute it and/or modify |
|  | \* it under the terms of the GNU General Public License as published by |
|  | \* the Free Software Foundation. |
|  | \* |
|  | \* This code is distributed in the hope that it will be useful, |
|  | \* but WITHOUT ANY WARRANTY; without even the implied warranty of |
|  | \* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the |
|  | \* GNU General Public License for more details. |
|  | \* |
|  | \* @author Cay Horstmann |
|  | \*/ |
|  |  |
|  | package info.gridworld.actor; |
|  |  |
|  | import info.gridworld.grid.Grid; |
|  | import info.gridworld.grid.Location; |
|  |  |
|  | import java.awt.Color; |
|  |  |
|  | /\*\* |
|  | \* A <code>Bug</code> is an actor that can move and turn. It drops flowers as |
|  | \* it moves. <br /> |
|  | \* The implementation of this class is testable on the AP CS A and AB exams. |
|  | \*/ |
|  | public class Bug extends Actor |
|  | { |
|  | /\*\* |
|  | \* Constructs a red bug. |
|  | \*/ |
|  | public Bug() |
|  | { |
|  | setColor(Color.RED); |
|  | } |
|  |  |
|  | /\*\* |
|  | \* Constructs a bug of a given color. |
|  | \* @param bugColor the color for this bug |
|  | \*/ |
|  | public Bug(Color bugColor) |
|  | { |
|  | setColor(bugColor); |
|  | } |
|  |  |
|  | /\*\* |
|  | \* Moves if it can move, turns otherwise. |
|  | \*/ |
|  | public void act() |
|  | { |
|  | if (canMove()) |
|  | move(); |
|  | else |
|  | turn(); |
|  | } |
|  |  |
|  | /\*\* |
|  | \* Turns the bug 45 degrees to the right without changing its location. |
|  | \*/ |
|  | public void turn() |
|  | { |
|  | setDirection(getDirection() + Location.HALF\_RIGHT); |
|  | } |
|  |  |
|  | /\*\* |
|  | \* Moves the bug forward, putting a flower into the location it previously |
|  | \* occupied. |
|  | \*/ |
|  | public void move() |
|  | { |
|  | Grid<Actor> gr = getGrid(); |
|  | if (gr == null) |
|  | return; |
|  | Location loc = getLocation(); |
|  | Location next = loc.getAdjacentLocation(getDirection()); |
|  | if (gr.isValid(next)) |
|  | moveTo(next); |
|  | else |
|  | removeSelfFromGrid(); |
|  | Flower flower = new Flower(getColor()); |
|  | flower.putSelfInGrid(gr, loc); |
|  | } |
|  |  |
|  | /\*\* |
|  | \* Tests whether this bug can move forward into a location that is empty or |
|  | \* contains a flower. |
|  | \* @return true if this bug can move. |
|  | \*/ |
|  | public boolean canMove() |
|  | { |
|  | Grid<Actor> gr = getGrid(); |
|  | if (gr == null) |
|  | return false; |
|  | Location loc = getLocation(); |
|  | Location next = loc.getAdjacentLocation(getDirection()); |
|  | if (!gr.isValid(next)) |
|  | return false; |
|  | Actor neighbor = gr.get(next); |
|  | return (neighbor == null) || (neighbor instanceof Flower); |
|  | // ok to move into empty location or onto flower |
|  | // not ok to move onto any other actor |
|  | } |
|  | } |