Christopher Pybus

AP Computer Science

Grid World Case Study Part 3

**Do You Know?** (Set 5)

1. **Name three properties of every actor.**Three properties of every actor are color, direction, and location on the grid.
2. **What is the value of b after the following statement is executed?   
   Boolean b = loc1.equals(loc2);**When an actor is constructed, its default color is blue and the default direction is North.
3. **Why do you think that the Actor class was created as a class instead of an interface?**Because an interface doesn’t allow instance variables to de declared.
4. **Can an actor put itself into a grid twice without first removing itself? Can an actor remove itself from a grid twice? Can an actor be placed into a grid, remove itself, and then put itself back?**No, an actor cannot put itself into a grid if it is already in the grid. And an actor cannot remove itself from the grid twice either. And finally, an actor CAN be placed into the grid, remove itself, and then place itself back in.
5. **How can an actor turn 90 degrees to the right?**There are two ways to turn, you could either do turn() twice, or   
   setDirection(getDirection() + Location.RIGHT);

**Do you know?** (Set 6)

1. **Which statement(s) in the canMove method ensures that a bug does notr try and move out of its grid?**The following statement ensures that the location that the bug will move two is valid and on the grid:  
   if(!gr.isValid(next))  
    return false;
2. **Which statement(s) in the canMove method determines that a bug will not walk into a rock?**The following statements ensure that a bug will only move to the next location if it is a flower.  
   Actor neighbor = gr.get(next);  
   return (neighbor == null) || (neighbor insteaceof Flower);
3. **Which methods of the Grid interface are invoked by the canMove method?**The two methods of the Grid interface that are invoked by the canMove method are get and isValid.
4. **Which method of the Location class is invoked by the canMove method?**The Location method that is invoked by the canMove method is getAdjactentLocation().
5. **Which methods inherited from the Actor class are invoked in the canMove method?**The methods in the canMove method that are invoked from the Actor class are getGrid, getDirection, and getLocation.
6. **What happens in the move method when the location immediately in front of the bug is out of the grid?**  
   If the location in front of the bug is out of the grid, the bug will remove itself from the grid.
7. **Is the variable loc needed in the move method, or could it be avoided by calling getLocation() multiple times?**Yes, the loc variable is needed. The loc variable is what the bugs location is stored in before it moves.
8. **Why do you think the flowers that are dropped by a bug have the same color as the bug?**Because it makes it easier to see which bugs drop which flowers.
9. **When a bug removes itself from the grid, will it place a flower into its previous location?**No, a flower will not be placed in location if the removeSelfFromGrid method is called.
10. **Which statement(s) in the move method places the flower into the grid at the bug’s previous location?**Flower flower = new Flower(getColor());  
    flower.putSelfinGrid(gr, loc);
11. **If a bug needs to turn 180 degrees, how many times should it call the turn method?**A bug would need to call the turn() method 4 timrs because each time it turns 45 degrees.

**Group Activity**

1. **Specify-Each group creates class called Jumper, this actor can move forward two cells in each move. It “jumps” over rocks and flowers. It does not leave anything behind it when it jumps.**
2. **What will a jumper do if the location in front of it is empty, but the location two cells in front contains a flower/rock?**The jumper bug could turn, or it could proceed normally if the location is a flower. It could also not move forward 2 steps, but either 1 or 3.
3. **What will the jumper do if the location two cells in front of the jumper is out of the grid?**The jumper would have to turn and go another direction or remove itself from the grid.
4. **What will a jumper do if it is facing an edge of the grid?**Again, the jumper would have to turn.
5. **What will a jumper do if another actor is in the cell that is two cells in front of the jumper?**Again, the jumper could turn. Or it could “jump” either 1 step, or 3 steps instead of 2.
6. **What will a jumper do if it encounters another jumper in its path?**Again, the jumper would have to turn.
7. **Are there any other tests the jumper needs to make?**  
   No.
8. **Design – groups address important design decisions to solve the problem.**
9. **Which class should Jumper extend?** The actor class.
10. **Is there an existing class that is similar to Jumper class?** The bug class.
11. **Should there be a constructor?** Yes, a way for the uer to pass on variables like configurable color and variables like how many grid blocks the jumper will “jump.”
12. **Which methods should be overridden?** Act method.
13. **What methods, if any, should be added?** Methods like makeJump() and canJump(). One to make the bug jump, and one to verify If the bug can jump.
14. **What is the plan for testing the class?** Make the jumper go through all the situations discussed in question 1.
15. **Code.**
16. **Test.**