Courseware   Updates & News   Calendar   Wiki   Discussion   Progress

PROBLEM 1: RECTANGULARROOM CLASS (10/10 points)

You will need to design two classes to keep track of which parts of the room have been cleaned as well as the position and direction of each robot.

In `ps2.py`, we've provided skeletons for the following two classes, which you will fill in in Problem 1:

RectangularRoom
  Represents the space to be cleaned and keeps track of which tiles have been cleaned.

Robot
  Stores the position and direction of a robot.

We've also provided a complete implementation of the following class:

Position
  Stores the *x*- and *y*-coordinates of a robot in a room.

Read `ps2.py` carefully before starting, so that you understand the provided code and its capabilities.

PROBLEM 1

In this problem you will implement two classes, `RectangularRoom` on this page and `Robot` on the next.

For the `RectangularRoom` class, decide what fields you will use and decide how the following operations are to be performed:

- Initializing the object

- Marking an appropriate tile as cleaned when a robot moves to a given position (casting floats to ints - and/or the function `math.floor` - may be useful to you here)

- Determining if a given tile has been cleaned

- Determining how many tiles there are in the room

- Determining how many cleaned tiles there are in the room

- Getting a random position in the room

- Determining if a given position is in the room

Complete the `RectangularRoom` class by implementing its methods in `ps2.py`.

Although this problem has many parts, it should not take long once you have chosen how you wish to represent your data. For reasonable representations, *a majority of the methods will require only a couple of lines of code*.)

Hint:
During debugging, you might want to use `random.seed(0)` so that your results are reproducible.

Enter your code for `RectangularRoom` below.

```
1 # Enter your code for RectangularRoom in this box
2 # === Problem 1
3 class RectangularRoom(object):
4     """
5     A RectangularRoom represents a rectangular region containing clean or dirty
6     tiles.
7
8     A room has a width and a height and contains (width * height) tiles. At any
9     particular time, each of these tiles is either clean or dirty.
10    """
11    #CLEANED, DIRTY = 0, 1
12
13    def __init__(self, width, height):
14        """
15        Initializes a rectangular room with the specified width and height.
16
```

Correct

Test results

CORRECT

Test: 1 class creation

```
room = RectangularRoom(5, 5)
room.getNumTiles()
```

Output:

```
Successfully created a room of size 25
```

Test: 2 test getNumTiles

This test creates a number of randomly sized rooms and
checks the number of tiles by calling room.getNumTiles().

Output:

```
room = RectangularRoom(20, 7)
room = RectangularRoom(20, 3)
room = RectangularRoom(10, 4)
room = RectangularRoom(20, 4)
room = RectangularRoom(9, 5)
Successfully created a room of size 45
```

Test: 3 unclean tiles

```
room = RectangularRoom(4, 6)
```
This tests that all squares are properly marked as unclean
by calling the isTileCleaned() and getNumCleanedTiles() methods.

Output:

```
Successfully created a room of size 24
Number of cleaned tiles:
```

Test: 4 test cleaningTiles

This test creates a randomly sized room and checks the
number of clean tiles by calling room.getNumCleanTiles(). Then, the
tiles in the room are cleaned and the check is performed again.

Output:

```
room = RectangularRoom(4, 5)
Successfully created a room of size 20
Number of clean tiles: 0
After cleaning, number of clean tiles: 20
```

Test: 5 unclean tiles

room = RectangularRoom(6, 8)
This test cleans a subset of the tiles by calling the cleanTileAtPosition method,
then checks that those tiles are marked as cleaned by calling the isTileCleaned method.

Output:

```
Successfully created a room of size 48
After cleaning, number of cleaned tiles:
```

Test: 6 cleaning tiles repeatedly

This test cleans the same tiles repeatedly, which should result in
the tiles continuing to be marked clean.

Output:

```
room = RectangularRoom(13, 18)
Done with iteration 0, number of cleaned tiles is 24
Done with iteration 1, number of cleaned tiles is 24
Done with iteration 2, number of cleaned tiles is 24
Done with iteration 3, number of cleaned tiles is 24
Done with iteration 4, number of cleaned tiles is 24
```

Test: 7 getRandomPosition

Test getRandomPosition by calling it 100 times and ensuring the outputs are sensible.

Output:

Test: 8 isPositionInRoom

Test isPositionInRoom by testing random positions inside and outside of a room.

Output:

Test: 9 isPositionInRoom

Test getRandomPosition and isPositionInRoom by generating random positions
then testing if they are within the room.

Output:

Hide output

Check   Save   *You have used 7 of 30 submissions*

Show Discussion   New Post

edX

EdX offers interactive online classes and MOOCs from the world's best universities. Online courses from MITx, HarvardX, BerkeleyX, UTx and many other universities. Topics include biology, business, chemistry, computer science, economics, finance, electronics, engineering, food and nutrition, history, humanities, law, literature, math, medicine, music, philosophy, physics, science, statistics and more. EdX is a non-profit online initiative created by founding partners Harvard and MIT.

© 2014 edX, some rights reserved.

Terms of Service and Honor Code

Privacy Policy (Revised 4/16/2014)

About & Company Info

About

News

Contact

FAQ

edX Blog

Donate to edX

Jobs at edX

Follow Us

Twitter

Facebook

Meetup

LinkedIn

Google+