

Tetrix Project (Part 3).docx

Continue to work with your partner on this assignment. Don't forget to switch roles often.

Exercise 3.0: Spin Cycle

Next, we'll make our tetrads rotate. We'll have them rotate clockwise by 90 degrees around the location of the tetrad's first block (`blocks[0]`) in the tetrad. The following diagram shows a single rotation of a T-shaped tetrad.



Given a block at position $(row, column)$, there is a surprisingly simple formula (shown below) to find its new location $(row', column')$, following a 90 degree clockwise rotation about a point at (row_0, col_0) .

$$\begin{aligned} row' &= row_0 - col_0 + col \\ col' &= row_0 + col_0 - row \end{aligned}$$

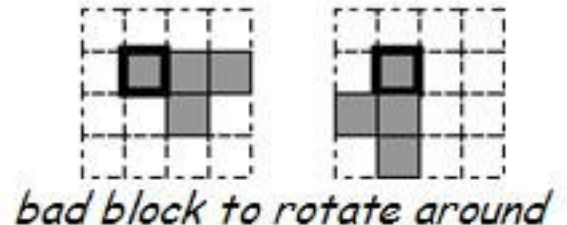
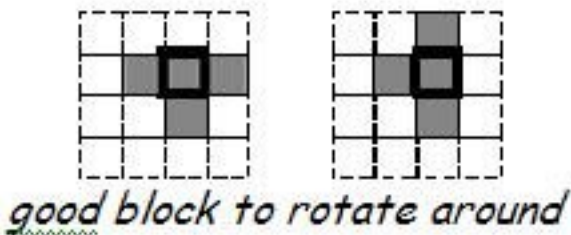
Use this formula to complete the following `rotate` method in the `Tetrad` class. Your `rotate` code should wind up looking very similar to your `translate` code.

```
//postcondition: Attempts to rotate this tetrad
//                clockwise by 90 degrees about its
//                center, if the necessary positions
//                are empty; returns true if successful
//                and false otherwise.
public boolean rotate()
```

Now modify the `Tetris` class so that it calls `rotate` to rotate the active tetrad clockwise by 90 degrees whenever the up arrow is pressed (instead of shifting the tetrad up). Test to make sure your tetrads rotate appropriately (but sometimes off-center), and that your game prevents you from rotating the tetrad off the edge of the window.

Exercise 3.1: A Pivotal Mistake

You've probably noticed that some of your tetrads seem to be rotating around the wrong block. What we'd like to do is make sure that each tetrad rotates around one of its center-most blocks (in other words, the block that contains the tetrad's *centroid*, or center of gravity), as shown below for the T-shaped tetrad.



Fortunately, this problem is easily fixed by going back to the `Tetrad` class's constructor, and simply changing the order of the locations for each block, so that the first location in each `Tetrad` is approximately at the center of the `Tetrad`. Make these changes and test that your tetrads rotate correctly now.

Exercise 3.2: The Sky Is Falling

Modify the method `play` in `Tetris`, which should repeatedly (forever)

1. pause for 1 second,
2. move the active tetrad down one row, and
3. redraw the display.

We can tell Java to pause for a second with this hideous line of code:

```
try { Thread.sleep(1000); } catch(Exception e) {}
```

When you test your program, you should find that you can still shift and rotate the tetrad, but that it will now slowly drop on its own. When it gets to the bottom, the tetrad should stop falling (although you'll still be able to slide it around for now).

Exercise 3.3: Tetrad Comrades

Now modify the `play` method so that, when it is unable to shift the active tetrad down any further, it creates a new active tetrad. (Hint: Check the `translate` method's return value, but make sure you only call `translate` once per loop!) Test your game, and see how much you've accomplished!

When you are finished, demonstrate your work for me so that I will know that you have completed this part of the assignment.