Monty Hall Dilemma Kata

This is a ~30-minute kata to help try to prove the validity of the accepted conclusion in the classic “Monty Hall Dilemma” or “Monty Hall Problem.”

The basic idea is that you have 3 doors to pick from. One of them has a prize. Behind each of the other 2 doors is a non-prize. You are told to select one of the 3 doors. Upon selection, the game show host (Monty Hall) will open one of the other doors, showing a non-prize. You will then be given an option to switch to the remaining unopened door or stay with your original choice. The accepted conclusion is that you double your chances of winning by switching. This kata exists to help detail the forces at play here.

More information on the problem can be found at:

http://wikipedia.org/wiki/Monty_hall_problem

Step 1:

Write a test that asserts the number that is returned from the getWinningDoor method is 1 or 2 or 3. My getWinningDoor method returns 1. Assert this a large number of times for each time the tests are run (this will make more sense later.)

Step 2:

Write a test that asserts that the number returned from the getWinningDoor method isn’t the same every time. My getWinningDoor method returns 0 or 1 randomly. Assert this a large number of times for each time the tests are run. This breaks the door limits test. Then add 1 when returning from getWinningDoor.

Step 3:

Write a test that asserts that all of 1 and 2 and 3 are returned from getWinningDoor. Assert this a large number of times each time the tests are run. This caused me to change from returning 1 or 2 to returning 1 or 2 or 3.

Step 4:

Write a test that simulates the playing of the game without ever switching doors. Do this a large number of times: set your choice, determine the other door, (re) set your choice, determine player success. Assert that this happens around one-third of the time. I used a delta of .05 for the assertion.

This takes some refactoring of the game class to externalize the return of getWinningDoor to a field that is initialized in the constructor. The user choice should also be a field so that it can be compared to the winning door to determine player success.

Step 5:

Write a test that simulates the playing of the game with switching doors every time. Do this a large number of times: set your choice, determine the other door, set your choice to the other door, determine player success. Assert that this happens around two-thirds of the time. I used a delta of .05 for the assertion.

This necessitates the implementation of the determineOtherDoor method in the game class. If the user has originally selected the correct door, the other door will randomly be one of the other two doors. If the user was not correct, the other door will have to be the winning door (because Monty will *have* to reveal a non-prize door).

Issues with the kata:

- When using Java to do the kata, it can get very tiresome to keep writing code that repeats processes a large number of times.
- The determineOtherDoor method isn't well tested by the end of the kata.