Fork Repository: https://github.com/martie34/jpacman

**Task 1:**

1. **Is the coverage good enough?**
    a. No, currently only one of the packages is getting tested, and it does not get tested fully.
    b. The package is "nl.tudelft.jpacman.board" and it only has 9% method coverage.

**Task 2 Coverage Update:**

1. My coverage after adding the isAlive() to my tests was 9% of overall methods.
2. This is due to a 6% method coverage for the "jpacman.level" package
3. Along with a now 44% method coverage for the "jpacman.sprite" package.

**Task 2.1 Coverage Updates:**

1. For the first method I tested the Player.setAlive(<bool>) one in the Player.java class.
    a. I made sure to test both branches the function could take (when the Boolean is True and False.)
    b. This increased the method coverage in level to 8%, and the method coverage in sprite went up to 46%.
2. For the second method I tested the Player.addPoints(<int>) method in the Player.java class.
    a. I made sure to test we could correctly add a positive number of points, and a negative number of points.
    b. This increased the method coverage in level to 11%.
3. For the second method I tested was Direction.getDeltaX() in the Direction.java class.
    a. I also updated the DirectionTest.java to test each direction ("North, "East", "South", "West") as this method test was easy.
    b. This increased the board method coverage to 11%.

**Task 3:**

1. **Are the coverage results from JaCoCo similar to the ones you got from IntelliJ in the last task? Why so or why not?:**
    a. The JaCoCo coverage results are not really like the ones that I got due to the JaCoCo tests including the ones ran in the "src/default_tests" directory.
2. **Did you find helpful the source code visualization from JaCoCo on uncovered branches?:**
    a. I really like the source code visualization on uncovered branches since I feel like it would be easy to miss that information when developing tests for methods that have several branches. I was able to get my IntelliJ to show something similar to this, and uncovered branches would appear on the right hand side of the IDE.
3. **Which visualization did you prefer and why? IntelliJ's coverage window or JaCoCo's report?:**
    a. I can see both being useful in different use cases. For quick testing and making sure every "method" is being tested, I prefer the IntelliJ window.
    b. However, to make sure every branch and instruction is covered I prefer the JaCoCo since it shows you possible missed branches you could have missed.
    c. They both display useful information, the other one doesn't. For example, JaCoCo only shows the methods and missed methods, but does not provide a percentage of those misses.