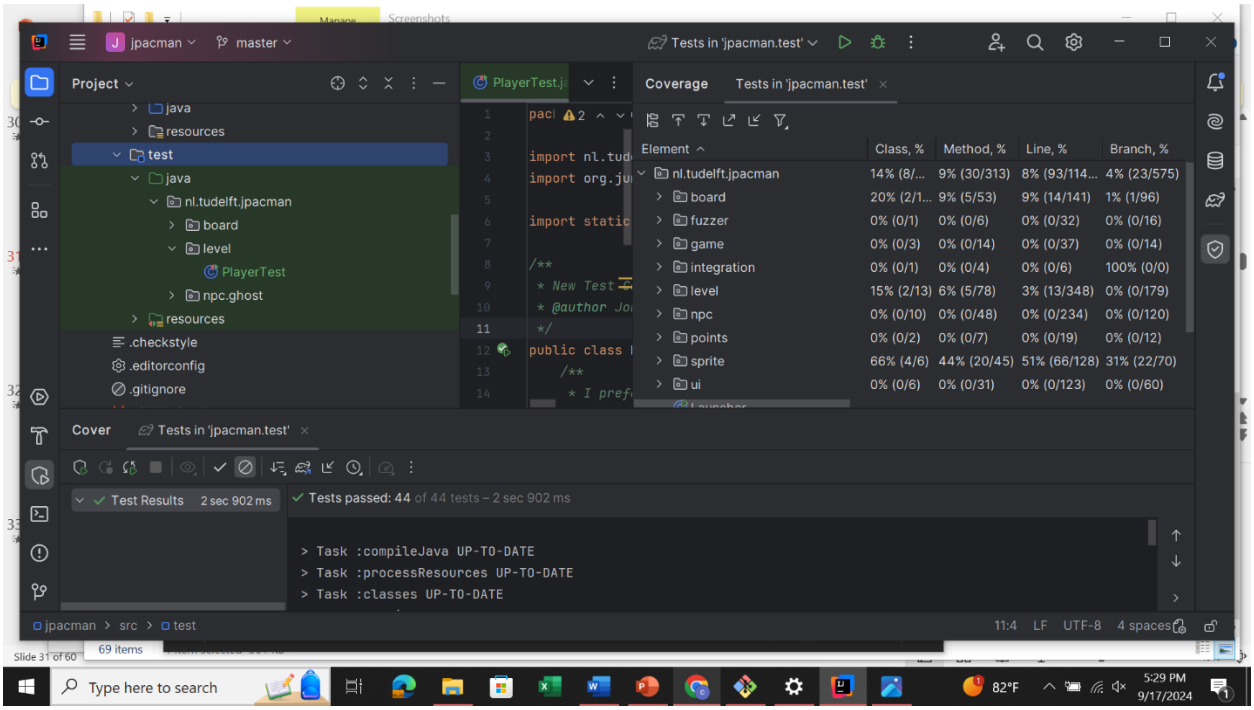
Unit Test Report

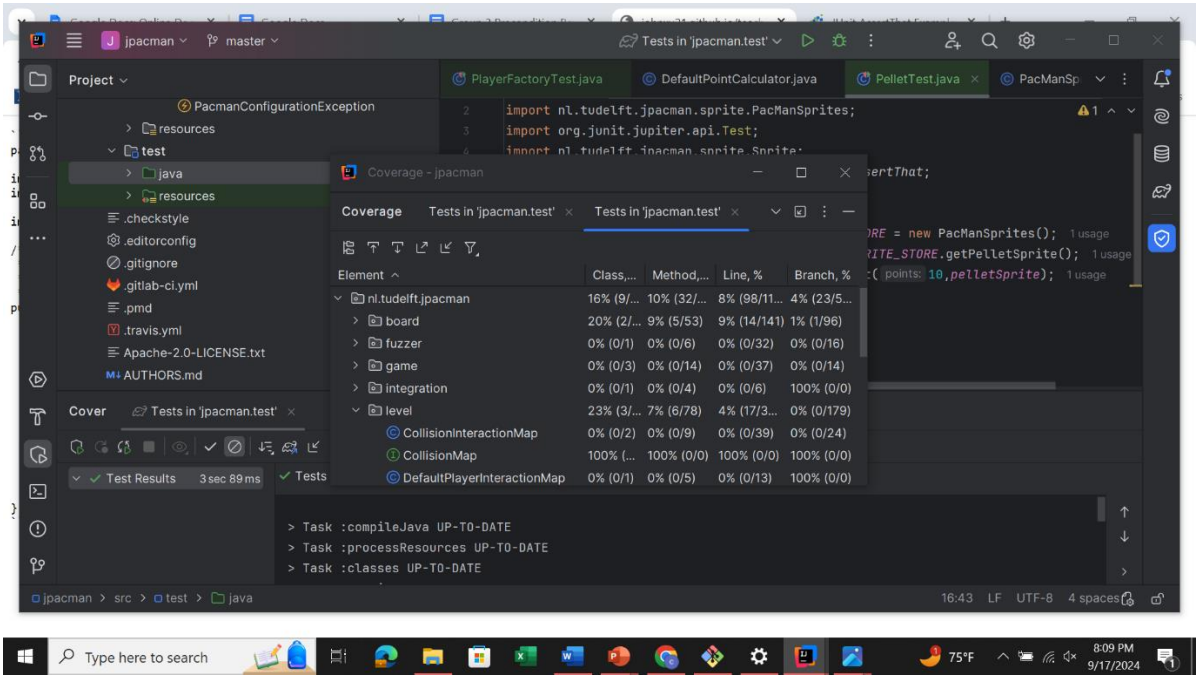
For the first unit test I tested the createPacman function in the Player Factory class. It was a simple test just to make sure that Player object was created.

```java
package nl.tudelft.jpacman.level;
import nl.tudelft.jpacman.sprite.PacManSprites;
import org.junit.jupiter.api.Test;
import nl.tudelft.jpacman.sprite.Sprite;


import static org.assertj.core.api.Assertions.assertThat;

public class PlayerFactoryTest {

    private static final PacManSprites SPRITE_STORE = new PacManSprites();
    private PlayerFactory Factory = new PlayerFactory(SPRITE_STORE);
    private Player ThePlayer = Factory.createPacMan();


    @Test
    void testCreatePacman()
    {
        assertThat(ThePlayer != null);

    }
}
```



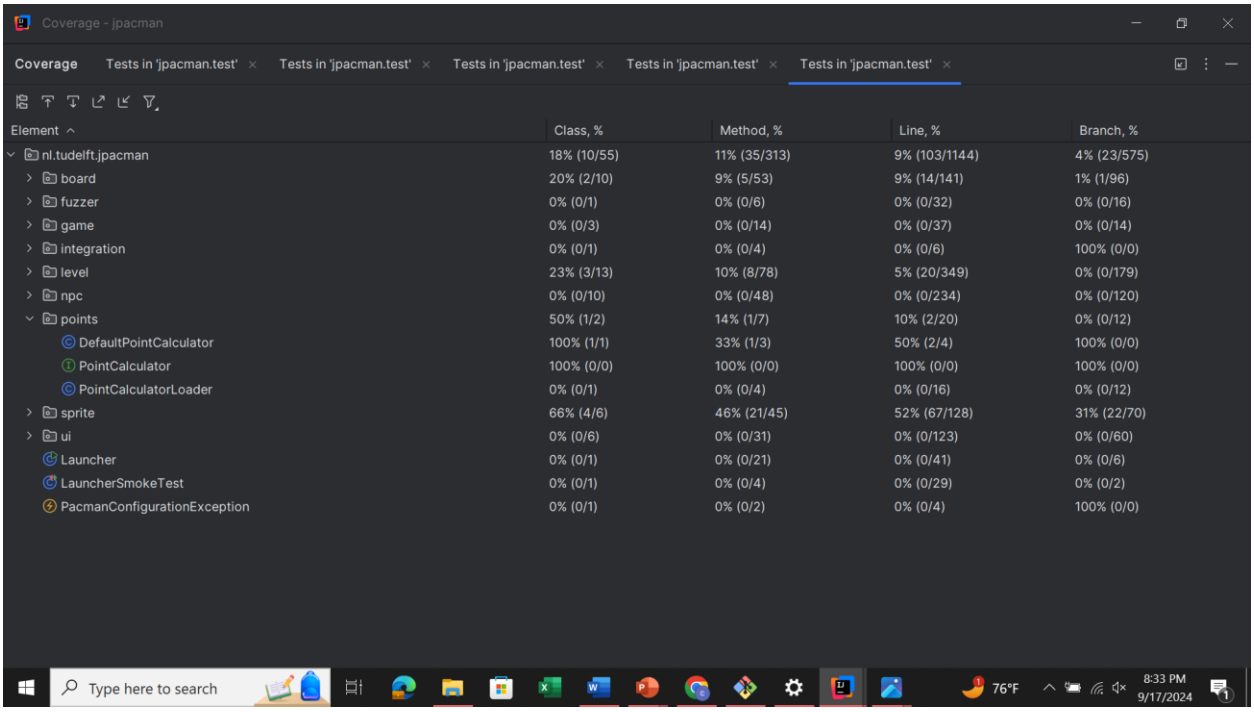
Furthermore for my second test I wanted to test the consuming of the pellets and ensure the score was correctly updated for the player.

```java
package points;
import nl.tudelft.jpacman.level.Pellet;
import nl.tudelft.jpacman.level.Player;
import nl.tudelft.jpacman.level.PlayerFactory;
import nl.tudelft.jpacman.points.DefaultPointCalculator;
import nl.tudelft.jpacman.sprite.PacManSprites;
import org.junit.jupiter.api.Test;
import nl.tudelft.jpacman.sprite.Sprite;
import static org.assertj.core.api.Assertions.assertThat;

public class DefaultPointCalculatorTest {
    private static final PacManSprites SPRITE_STORE = new PacManSprites();
    private PlayerFactory Factory = new PlayerFactory(SPRITE_STORE);
    private Player ThePlayer = Factory.createPacMan();
    private static final Sprite pelletSprite= SPRITE_STORE.getPelletSprite();
    private static final Pellet pellet=new Pellet(10,pelletSprite);
    private static final DefaultPointCalculator pointCalc= new
DefaultPointCalculator();
    @Test
     void testConsumedAPellet(){
pointCalc.consumedAPellet(ThePlayer,pellet);
assertThat(ThePlayer.getScore()).isEqualTo(10);
    }
}
```



For the last unit test I tested getting the value of individual pellets and ensuring the values were correctly being stored.

```java
package nl.tudelft.jpacman.level;
import nl.tudelft.jpacman.sprite.PacManSprites;
import org.junit.jupiter.api.Test;
import nl.tudelft.jpacman.sprite.Sprite;
import static org.assertj.core.api.Assertions.assertThat;

public class PelletTest {
    private static final PacManSprites SPRITE_STORE = new PacManSprites();
    private static final Sprite pelletSprite= SPRITE_STORE.getPelletSprite();
    private static final Pellet pellet=new Pellet(10,pelletSprite);


    @Test
    void testGetValue(){

        assertThat(pellet.getValue()==10);
    }
}
```



In looking over the Jacoco files the coverage was different most likely to Jacoco considering the branch coverage as well. I do prefer the Jacoco representation as it gives a lot more information and easier navigate by quickly seeing code and methods.