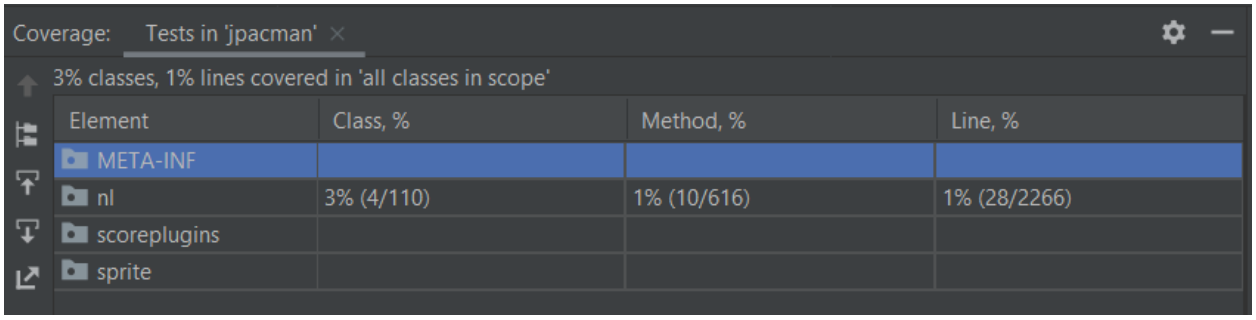
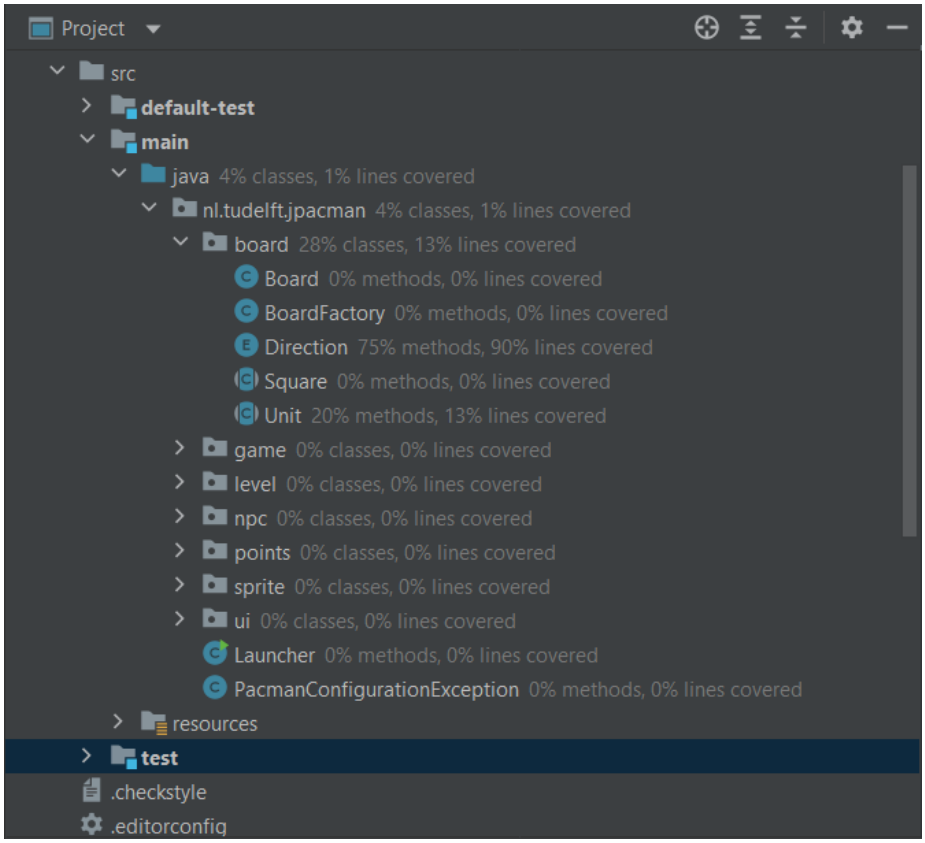
Miguel Baniqued
CS 472-1001
Feb 2, 2023

Lab 2 Report

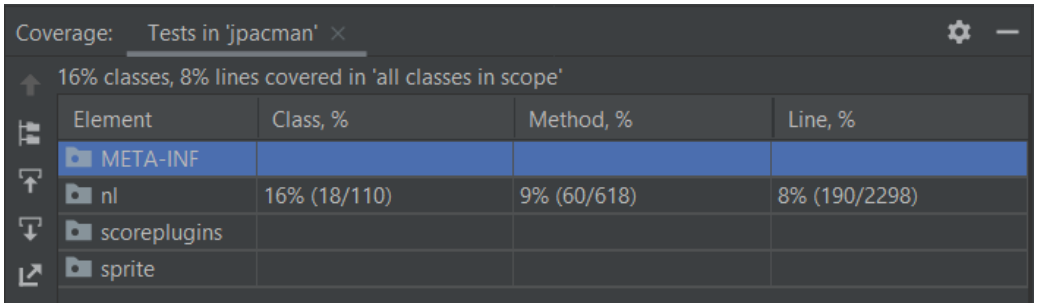
Forked Repository Link: https://github.com/mcebaniqued/jpacman
Main Repository: https://github.com/CivBuilder/cs472project

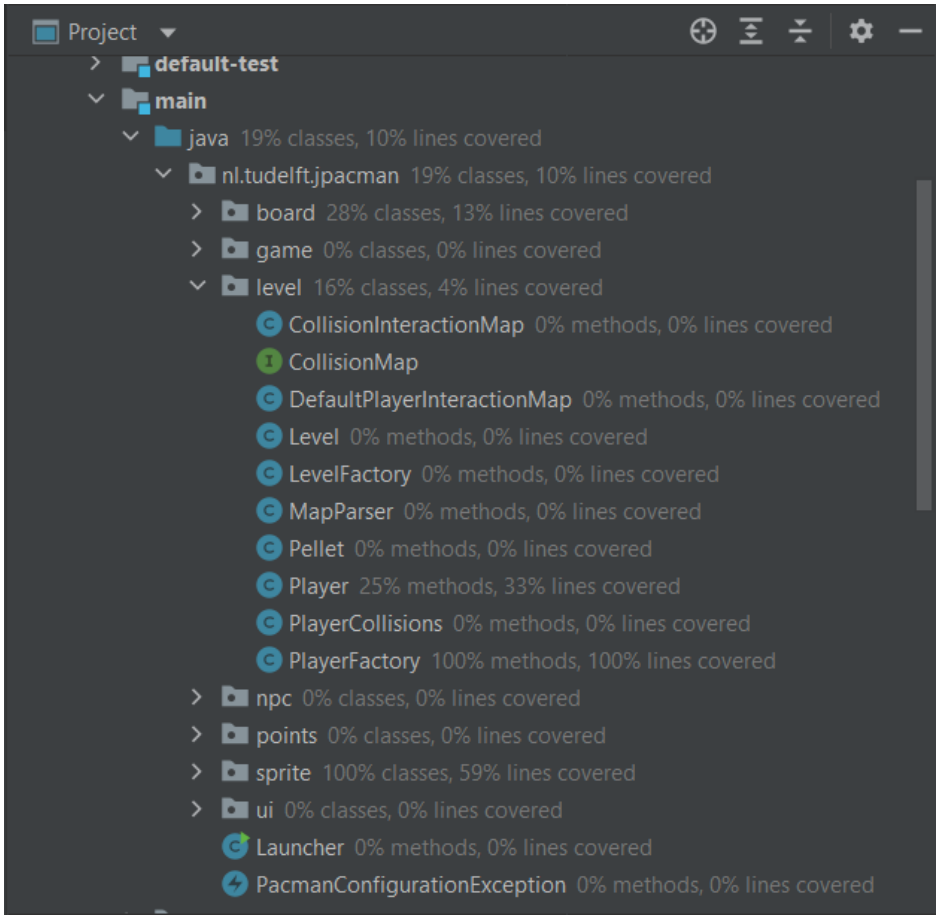
Task 1: Initial Test Coverage





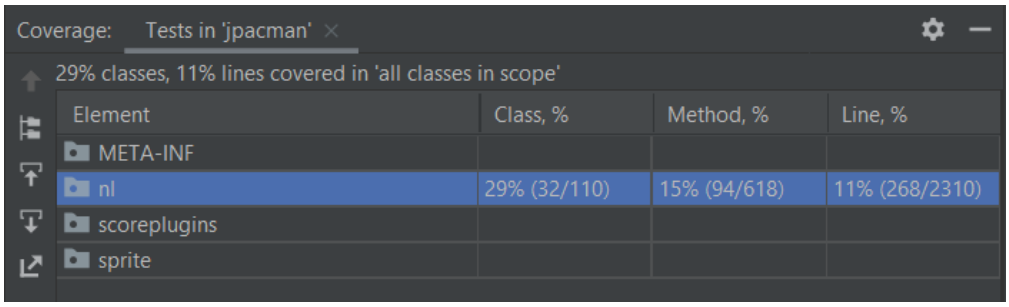
Task 2: Added isAlive() Test



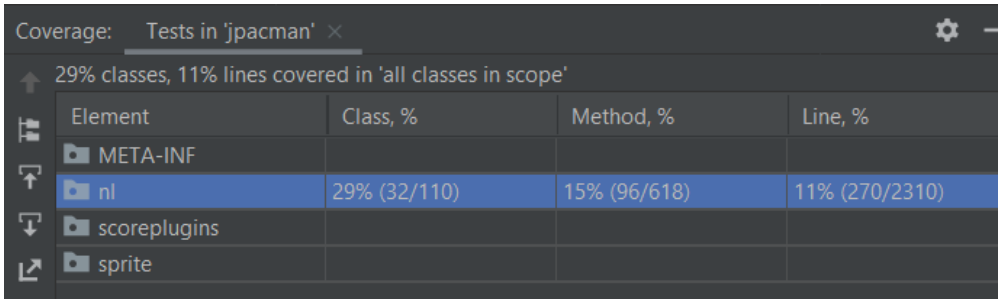


Task 2.1: Added hasSquare(), getDirection(), and isInProgress() Tests

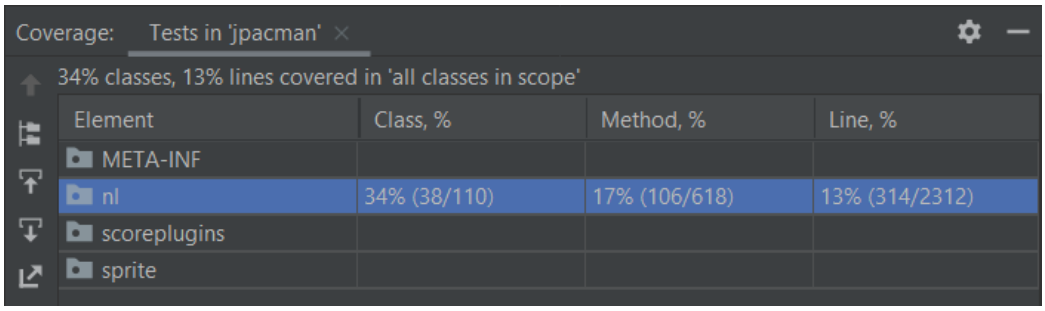
```java
package nl.tudelft.jpacman.level;

import org.junit.jupiter.api.Test;
import static org.assertj.core.api.Assertions.assertThat;

import nl.tudelft.jpacman.npc.ghost.GhostFactory;
import nl.tudelft.jpacman.sprite.PacManSprites;
import nl.tudelft.jpacman.npc.Ghost;

public class GhostHasSquareTest {
    private static final PacManSprites SPRITE_STORE = new PacManSprites();
    private GhostFactory Factory = new GhostFactory(SPRITE_STORE);

    private Ghost Blinky = Factory.createBlinky();
    private Ghost Clyde = Factory.createClyde();
    private Ghost Inky = Factory.createInky();
    private Ghost Pinky = Factory.createPinky();

    @Test
    void testBlinkyHasSquareIsFalse(){ assertThat(Blinky.hasSquare()).isEqualTo(false); }

    @Test
    void testClydeHasSquareIsFalse(){ assertThat(Clyde.hasSquare()).isEqualTo(false); }

    @Test
    void testInkyHasSquareIsFalse(){ assertThat(Inky.hasSquare()).isEqualTo(false); }

    @Test
    void testPinkyHasSquareIsFalse(){ assertThat(Pinky.hasSquare()).isEqualTo(false); }
}
```



```java
package nl.tudelft.jpacman.level;

import org.junit.jupiter.api.Test;
import static org.assertj.core.api.Assertions.assertThat;

import nl.tudelft.jpacman.npc.ghost.GhostFactory;
import nl.tudelft.jpacman.sprite.PacManSprites;
import nl.tudelft.jpacman.npc.Ghost;
import nl.tudelft.jpacman.board.Direction;

public class GhostGetDirection {
    private static final PacManSprites SPRITE_STORE = new PacManSprites();
    private GhostFactory Factory = new GhostFactory(SPRITE_STORE);

    private Ghost Blinky = Factory.createBlinky();
    private Ghost Clyde = Factory.createClyde();
    private Ghost Inky = Factory.createInky();
    private Ghost Pinky = Factory.createPinky();

    @Test
    void testBlinkyGetDirectionIsEast(){ assertThat(Blinky.getDirection()).isEqualTo(Direction.EAST); }

    @Test
    void testClydeGetDirectionIsEast(){ assertThat(Clyde.getDirection()).isEqualTo(Direction.EAST); }

    @Test
    void testInkyGetDirectionIsEast(){ assertThat(Inky.getDirection()).isEqualTo(Direction.EAST); }

    @Test
    void testPinkyGetDirectionIsEast(){ assertThat(Pinky.getDirection()).isEqualTo(Direction.EAST); }
}
```



```java
package nl.tudelft.jpacman.level;

import nl.tudelft.jpacman.board.Board;
import nl.tudelft.jpacman.board.Square;
import nl.tudelft.jpacman.npc.Ghost;

import static org.mockito.Mockito.mock;

import org.assertj.core.util.Lists;
import org.junit.jupiter.api.Test;
import static org.assertj.core.api.Assertions.assertThat;

public class LevelIsInProgress {
    private Level level;
    private Ghost ghost = mock(Ghost.class);
    private Board board = mock(Board.class);
    private Square square1 = mock(Square.class);
    private Square square2 = mock(Square.class);
    private final CollisionMap collisions = mock(CollisionMap.class);

    @Test
    void testLevelIsInProgressIsFalse(){
        level = new Level(board, Lists.newArrayList(ghost), Lists.newArrayList(square1, square2), collisions);
        assertThat(level.isInProgress()).isEqualTo(false);
    }
}
```



Task 3: JaCoCo Report on JPacman

jpacman

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
nl.tudelft.jpacman.level		67%		57%	74	155	104	344	21	69	4	12
nl.tudelft.jpacman.npc.ghost		71%		55%	56	105	43	181	5	34	0	8
nl.tudelft.jpacman.ui		77%		47%	54	86	21	144	7	31	0	6
default		0%		0%	12	12	21	21	5	5	1	1
nl.tudelft.jpacman.board		86%		59%	43	93	2	110	0	40	0	7
nl.tudelft.jpacman.sprite		86%		59%	30	70	11	113	5	38	0	5
nl.tudelft.jpacman		69%		25%	12	30	18	52	6	24	1	2
nl.tudelft.jpacman.points		60%		75%	1	11	5	21	0	9	0	2
nl.tudelft.jpacman.game		87%		60%	10	24	4	45	2	14	0	3
nl.tudelft.jpacman.npc		100%		n/a	0	4	0	8	0	4	0	1
Total	1,212 of 4,694	74%	292 of 637	54%	292	590	229	1,039	51	268	6	47

The result from IntelliJ's coverage differs greatly from JaCoCo's coverage. JaCoCo show's 74% coverage while IntelliJ shows 34% coverage. I do find JaCoCo's UI to be helpful because it adds a visualization of how much missed instruction and branches for each element, and you can compare it with other elements. It also gives more information about lines, methods, classes, and the overall for each element. I think the most helpful out of all would be the highlighting of missed and covered lines. It tells you exactly which lines are being used for tests so it's easy to point out which line or method still needs to be tested. Overall, I would prefer JaCoCo's report over IntelliJ.