Task 2

Coverage | Tests in 'jpacman.test'

Element	Class...	Method...	Line, %
nl.tudelft.jpacman	14% (8...	9% (30/3...	8% (93/11...
board	20% (2...	9% (5/53)	9% (14/141)
fuzzer	0% (0/1)	0% (0/6)	0% (0/32)
game	0% (0/3)	0% (0/14)	0% (0/37)
integration	0% (0/1)	0% (0/4)	0% (0/6)
level	15% (2...	6% (5/78)	3% (13/350)
npc	0% (0/...	0% (0/47)	0% (0/237)
points	0% (0/2)	0% (0/7)	0% (0/19)
sprite	66% (4...	44% (20/...	51% (66/1...
ui	0% (0/6)	0% (0/31)	0% (0/127)
Launcher	0% (0/1)	0% (0/21)	0% (0/41)
LauncherSmokeTe:	0% (0/1)	0% (0/4)	0% (0/29)
PacmanConfigurati	0% (0/1)	0% (0/2)	0% (0/4)

Original Test Coverage

Task 2.1 getDeltaX()

Element	Class, %	Method, %	Line, %
nl.tudelft.jpacman	14% (8/55)	9% (31/312)	8% (94/1151)
board	20% (2/10)	11% (6/53)	10% (15/141)
fuzzer	0% (0/1)	0% (0/6)	0% (0/32)
game	0% (0/3)	0% (0/14)	0% (0/37)
integration	0% (0/1)	0% (0/4)	0% (0/6)
level	15% (2/13)	6% (5/78)	3% (13/350)
npc	0% (0/10)	0% (0/47)	0% (0/237)
points	0% (0/2)	0% (0/7)	0% (0/19)
sprite	66% (4/6)	44% (20/45)	51% (66/128)
ui	0% (0/6)	0% (0/31)	0% (0/127)
Launcher	0% (0/1)	0% (0/21)	0% (0/41)
LauncherSmokeTest	0% (0/1)	0% (0/4)	0% (0/29)
PacmanConfigurationExcepti	0% (0/1)	0% (0/2)	0% (0/4)

```java
public class DirectionTest {
    /**
     * Do we get the correct delta when moving north?
     */
    @Test
    void testNorth() {
        Direction north = Direction.valueOf( name: "NORTH");
        assertThat(north.getDeltaY()).isEqualTo( expected: -1);
    }

    @Test
    void testSouth() {
        Direction north = Direction.valueOf( name: "SOUTH");
        assertThat(north.getDeltaY()).isEqualTo( expected: 1);
    }

    @Test
    void testEast() {
        Direction east = Direction.valueOf( name: "EAST");
        assertThat(east.getDeltaX()).isEqualTo( expected: 1);
    }

    @Test
    void testWest() {
        Direction east = Direction.valueOf( name: "WEST");
        assertThat(east.getDeltaX()).isEqualTo( expected: -1);
    }
}
```

The first thing I decided to test was to try and look at the other directions for DirectionTest. I realized that it did not matter what value (aka the four directions) was ensured to increase test coverage rather than just the method. The board method coverage went from 9% to 11%, and the line percent went from 9% to 10%. I ensured the value gotten from getDeltaX() is equal to the east or west direction.

Task 2.1 start()

Coverage | Tests in 'jpacman.test'

Element	Class, %	Method, %	Line, %
nl.tudelft.jpacman	21% (12/55)	14% (44/312)	13% (152/1154)
board	40% (4/10)	15% (8/53)	11% (17/142)
fuzzer	0% (0/1)	0% (0/6)	0% (0/32)
game	0% (0/3)	0% (0/14)	0% (0/37)
integration	0% (0/1)	0% (0/4)	0% (0/6)
level	30% (4/13)	20% (16/78)	19% (69/352)
npc	0% (0/10)	0% (0/47)	0% (0/237)
points	0% (0/2)	0% (0/7)	0% (0/19)
sprite	66% (4/6)	44% (20/45)	51% (66/128)
ui	0% (0/6)	0% (0/31)	0% (0/127)
Launcher	0% (0/1)	0% (0/21)	0% (0/41)
LauncherSmokeTest	0% (0/1)	0% (0/4)	0% (0/29)
PacmanConfigurationException	0% (0/1)	0% (0/2)	0% (0/4)

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

import ...

public class StartLevelTest {
    3 usages
    private Level level;

    1 usage
    private final Board board = mock(Board.class);
    1 usage
    private final Ghost ghost = mock(Ghost.class);

    1 usage
    private final Square square1 = mock(Square.class);
    1 usage
    private final Square square2 = mock(Square.class);

    1 usage
    private final CollisionMap collisions = mock(CollisionMap.class);
    @Test
    void testStart() {
        level = new Level(board, Lists.newArrayList(ghost), Lists.newArrayList(
            square1, square2), collisions);
        level.start();
        assertThat(level.isInProgress()).isTrue();
    }
}
```

Learning this new thing of using methods to increase test coverage, I decided to test start(), ensuring when a level is started in progress is true. This tests a number of things, creating level objects, retrieving board, ghost, and square, collision objects. With those, I started a new level, creating arrays, using the Lists class. Finally I started the level and ensured it is in progress aka true. The test coverage went up way more than the previous, the entire class percentage went up from 14% to 21%, and the board class percentage went up from 20% to 40%. The level coverage essentially doubled in each category, except in line going from 3% to 19%.

Task 2.1 createBoard()

Coverage | Tests in 'jpacman.test'

Element	Class, %	Method,...	Line, %
nl.tudelft.jpacman	23% (13/...	17% (55/3...	16% (191/...
board	50% (5/10)	35% (19/53)	38% (56/1...
fuzzer	0% (0/1)	0% (0/6)	0% (0/32)
game	0% (0/3)	0% (0/14)	0% (0/37)
integration	0% (0/1)	0% (0/4)	0% (0/6)
level	30% (4/13)	20% (16/78)	19% (69/3...
npc	0% (0/10)	0% (0/47)	0% (0/237)
points	0% (0/2)	0% (0/7)	0% (0/19)
sprite	66% (4/6)	44% (20/...	51% (66/1...
ui	0% (0/6)	0% (0/31)	0% (0/127)
Launcher	0% (0/1)	0% (0/21)	0% (0/41)
LauncherSmokeTest	0% (0/1)	0% (0/4)	0% (0/29)
PacmanConfigurationExcept	0% (0/1)	0% (0/2)	0% (0/4)

```java
10 usages
private BoardFactory factory;

20 usages
private Square s1;

16 usages
private Square s2;
@Test
void worldIsRound() {
    PacManSprites sprites = mock(PacManSprites.class);
    factory = new BoardFactory(sprites);

    s1 = new BasicSquare();
    factory.createBoard(new Square[][]{{s1}});
    assertThat(Arrays.stream(Direction.values()).map(s1::getSquareAt)).containsOnly(s1);
}

@Test
void connectedEast() {
    PacManSprites sprites = mock(PacManSprites.class);
    factory = new BoardFactory(sprites);

    s1 = new BasicSquare();
    s2 = new BasicSquare();
    factory.createBoard(new Square[][]{{s1}, {s2}});
    assertThat(s1.getSquareAt(Direction.EAST)).isEqualTo(s2);
    assertThat(s2.getSquareAt(Direction.EAST)).isEqualTo(s1);
}

@Test
void connectedWest() {
    PacManSprites sprites = mock(PacManSprites.class);
    factory = new BoardFactory(sprites);

    s1 = new BasicSquare();
    s2 = new BasicSquare();
    factory.createBoard(new Square[][]{{s1}, {s2}});
    assertThat(s1.getSquareAt(Direction.WEST)).isEqualTo(s2);
    assertThat(s2.getSquareAt(Direction.WEST)).isEqualTo(s1);
}
```

```java
/**
 * Verifies a chain of cells is connected to the north.
 */
@Test
void connectedNorth() {
    PacManSprites sprites = mock(PacManSprites.class);
    factory = new BoardFactory(sprites);

    s1 = new BasicSquare();
    s2 = new BasicSquare();
    factory.createBoard(new Square[][]{{s1, s2}});
    assertThat(s1.getSquareAt(Direction.NORTH)).isEqualTo(s2);
    assertThat(s2.getSquareAt(Direction.NORTH)).isEqualTo(s1);
}

/**
 * Verifies a chain of cells is connected to the south.
 */
@Test
void connectedSouth() {
    PacManSprites sprites = mock(PacManSprites.class);
    factory = new BoardFactory(sprites);

    s1 = new BasicSquare();
    s2 = new BasicSquare();
    factory.createBoard(new Square[][]{{s1, s2}});
    assertThat(s1.getSquareAt(Direction.SOUTH)).isEqualTo(s2);
    assertThat(s2.getSquareAt(Direction.SOUTH)).isEqualTo(s1);
}
```

The final test I did was createBoard(). Learning from level, I realized I have to find all the elements to execute a function and execute that function. This is testing a bunch of methods such as creating a square, ensuring different placements of squares. The class test coverage had gone up from 21% to 23%. The board percents from 40% to 50% in class, 15% to 35% in method, 11 to 38% in lines.

Task 3

Coverage | Tests in 'jpacman.test'

Element	Class, %	Method, %	Line, %
nl.tudelft.jpacman	23% (13/55)	17% (55/312)	16% (191/1156)
board	50% (5/10)	35% (19/53)	38% (56/144)
fuzzer	0% (0/1)	0% (0/6)	0% (0/32)
game	0% (0/3)	0% (0/14)	0% (0/37)
integration	0% (0/1)	0% (0/4)	0% (0/6)
level	30% (4/13)	20% (16/78)	19% (69/352)
npc	0% (0/10)	0% (0/47)	0% (0/237)
points	0% (0/2)	0% (0/7)	0% (0/19)
sprite	66% (4/6)	44% (20/45)	51% (66/128)
ui	0% (0/6)	0% (0/31)	0% (0/127)
Launcher	0% (0/1)	0% (0/21)	0% (0/41)
LauncherSmokeTest	0% (0/1)	0% (0/4)	0% (0/29)
PacmanConfigurationException	0% (0/1)	0% (0/2)	0% (0/4)

jpacman

jpacman

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
nl.tudelft.jpacman.level		67%		57%	74	155	104	344	21	69
nl.tudelft.jpacman.npc.ghost		71%		55%	56	105	43	181	5	34
nl.tudelft.jpacman.ui		77%		47%	54	86	21	144	7	31
default		0%		0%	12	12	21	21	5	5
nl.tudelft.jpacman.board		86%		58%	44	93	2	110	0	40
nl.tudelft.jpacman.sprite		86%		59%	30	70	11	113	5	38
nl.tudelft.jpacman		69%		25%	12	30	18	52	6	24
nl.tudelft.jpacman.points		60%		75%	1	11	5	21	0	9
nl.tudelft.jpacman.game		87%		60%	10	24	4	45	2	14
nl.tudelft.jpacman.npc		100%		n/a	0	4	0	8	0	4
Total	1,213 of 4,694	74%	293 of 637	54%	293	590	229	1,039	51	268

I got way more coverage in jcoco than intellej. It is strange because jcoco counts way more lines than intellej. For example, stop is counted fully in jcoco but not in intellej. My hypothesis is that jcoco tracks every line a code went over in a test, while intellej only tracks the methods used in the test code.
Example: My start() code had to eventually terminate the game, so jcoco tracks it, while in intellej I didnt explicitly stop() so it does not track it.

```
216     public void stop() {
217         synchronized (startStopLock) {
218             if (!isInProgress()) {
219                 return;
220             }
221             stopNPCs();
222             inProgress = false;
223         }
224     }
225
```

```
212.     /**
213.      * Stops or pauses this level, no l
214.      * and stopping all NPCs.
215.      */
216.     public void stop() {
217.         synchronized (startStopLock) {
218.             if (!isInProgress()) {
219.                 return;
220.             }
221.             stopNPCs();
222.             inProgress = false;
223.         }
224.     }
225.
```

I find the source code visualization extremely useful on tracking uncovered branches, breaking down code by methods and each method by the line and if it was tested. I enjoy how a lot of color is in my face telling me what I have and have not tested

I much prefer jcoco over the intellej! The jacpacman allowed me to go line by line much easier than jacoco, as the entire lines are highlighted even showing red and yellow for the ones not yet done. However, I think I am suspicious of jcoco of its interestingly high coverage

Task 4

```
Test Account Model
- Test creating multiple Accounts
- Test Account creation using known data
- Test account to delete
- Test account to find name
- Test account to fromdict
- Test the representation of an account
- Test account to dict
- Test account to update successfully
- Test account to updated FAIL

Name                  Stmts   Miss  Cover   Missing
-----------------------------------------------------
models\__init__.py        7      0   100%
models\account.py        40      0   100%
-----------------------------------------------------
TOTAL                    47      0   100%
----------------------------------------------------------------------
Ran 9 tests in 0.595s

OK
```

```python
def test_find(self):
    """ Test account to find name"""
    data = ACCOUNT_DATA[self.rand] # get a random account
    account = Account(**data)
    account.create()
    test_account = Account.find(account.id)
    result = test_account.name
    self.assertEqual(result, account.name)
```

```python
def test_update(self):
    """ Test account to update successfully """
    data = ACCOUNT_DATA[self.rand] # get a random account
    account = Account(**data)
    account.create()

    testName = "name"
    account.name = testName
    account.update()

    new_account = Account.find(account.id)

    self.assertEqual(new_account.name, testName)

def test_update_fail(self):
    """ Test account to updated FAIL """
    data = ACCOUNT_DATA[self.rand] # get a random account
    account = Account(**data)
    with self.assertRaises(DataValidationError):
        account.update()

def test_from_dict(self):
    """ Test account to fromdict """
    data = ACCOUNT_DATA[self.rand] # get a random account
    account = Account(**data)
    result = account.to_dict()
    test_account = Account(**data)
    test_account.from_dict(result)
    self.assertEqual(account.name, test_account.name)
    self.assertEqual(account.email, test_account.email)
    self.assertEqual(account.phone_number, test_account.phone_number)
    self.assertEqual(account.disabled, test_account.disabled)
    self.assertEqual(account.date_joined, test_account.date_joined)
```

Task 5

```
src
    counter.py
    status.py
tests
    test_counter
venv
.coverage
README.md
requirements.tx
setup.cfg
External Libraries
Scratches and Con
```

```python
    def test_update_a_counter(self):
        """It should update counter"""
        result = self.client.post('/counters/update')
        self.assertEqual(result.status_code, status.HTTP_201_CREATED)
        self.assertEqual(result.json['update'], 0)

        result = self.client.put('/counters/update')
        self.assertEqual(result.status_code, status.HTTP_200_OK)
        self.assertEqual(result.json['update'], 1)

        result = self.client.put('/counters/doesNotExist')
        self.assertEqual(result.status_code, status.HTTP_204_NO_CONTENT)
```

Terminal Local

```
- It should return an error for duplicates
- It should update counter (FAILED)

======================================================================
FAIL: It should update counter
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\neonf\Documents\tdd\tdd\tests\test_counter.py", line 47, in test_update_a_counter
    self.assertEqual(result.status_code, status.HTTP_200_OK)
AssertionError: 405 != 200
-------------------- >> begin captured logging << --------------------
src.counter: INFO: Request to create counter: update
--------------------- >> end captured logging << ---------------------

Name              Stmts   Miss  Cover   Missing
---------------------------------------------
src\counter.py       11      0   100%
src\status.py         6      0   100%
---------------------------------------------
TOTAL                17      0   100%
----------------------------------------------------------------------
```

Created test_update_a_counter(self): that creates a counter, ensure it came back created, make sure it is baseline, updates, ensure successful code is returned, then make sure it is more than 0. If there is an item that does not exist there is an error

I got RED phase, with the error AssertionError: 405 != 200

```
tests
    test_counter
venv
.coverage
README.md
requirements.tx
setup.cfg
External Libraries
Scratches and Con
```

```python
@app.route('/counters/<name>', methods=['PUT'])
def update_counter(name):
    """Update a counter"""
    app.logger.info(f"Request to update counter: {name}")
    if name in COUNTERS:
        COUNTERS[name] += 1
        return {name: COUNTERS[name]}, status.HTTP_200_OK
    return {"Message":f"Counter {name} does not exist"}, status.HTTP_204_NO_CONTENT
```

Terminal Local

```
FAILED (failures=1)

PS C:\Users\neonf\Documents\tdd\tdd> nosetests

Counter tests
- It should create a counter
- It should return an error for duplicates
- It should update counter

Name              Stmts   Miss  Cover   Missing
-----------------------------------------------
src\counter.py       18      0   100%
src\status.py         6      0   100%
-----------------------------------------------
TOTAL                24      0   100%
----------------------------------------------------------------------
Ran 3 tests in 0.152s

OK

PS C:\Users\neonf\Documents\tdd\tdd>
```

REFACTOR counter, by updating the counter to add one if it is found. If it is not found an error is returned

Went to GREEN

```
.idea
src
    counter.py
    status.py
tests
    test_counter
venv
.coverage
README.md
requirements.tx
```

```python
53      def test_read_a_counter(self):
54          """It should read a counter"""
55          result = self.client.post('/counters/read')
56          self.assertEqual(result.status_code, status.HTTP_201_CREATED)
57
58          result = self.client.get('/counters/read')
59          self.assertEqual(result.status_code, status.HTTP_200_OK)
60
61          result = self.client.get('/counters/doesNotExist')
62          self.assertEqual(result.status_code, status.HTTP_409_CONFLICT)
```

Terminal Local

```
PS C:\Users\neonf\Documents\tdd\tdd> nosetests

Counter tests
- It should create a counter
- It should return an error for duplicates
- It should read a counter (FAILED)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\neonf\Documents\tdd\tdd\tests\test_counter.py", line 59, in test_read_a_counter
    self.assertEqual(result.status_code, status.HTTP_200_OK)
AssertionError: 405 != 200
-------------------- >> begin captured logging << --------------------
src.counter: INFO: Request to create counter: read
-------------------- >> end captured logging << --------------------

Name              Stmts   Miss  Cover   Missing
---------------------------------------------
src\counter.py       18      0   100%
src\status.py         6      0   100%
---------------------------------------------
TOTAL                24      0   100%
----------------------------------------------------------------------
Ran 4 tests in 0.156s

FAILED (failures=1)
```

RED

I created a read counter that creates a counter, then tries to get it successfully, then tries to get a counter unsuccessfully

I got error AssertionError: 405 != 200

```
tests
    test_counter
venv
.coverage
README.md
requirements.tx
setup.cfg
External Libraries
```

```python
@app.route('/counters/<name>', methods=['GET'])
def read_counter(name):
    """Read a counter"""
    app.logger.info(f"Request to read counter: {name}")
    if name in COUNTERS:
        return {name: COUNTERS[name]}, status.HTTP_200_OK
    return {"Message":f"Counter {name} does not exist"}, status.HTTP_409_CONFLICT
```

Terminal Local

```
PS C:\Users\neonf\Documents\tdd\tdd> nosetests

Counter tests
- It should create a counter
- It should return an error for duplicates
- It should read a counter
- It should update counter

Name              Stmts   Miss  Cover   Missing
----------------------------------------------
src\counter.py       24      0   100%
src\status.py         6      0   100%
----------------------------------------------
TOTAL                30      0   100%
----------------------------------------------------------------------
Ran 4 tests in 0.173s

OK

PS C:\Users\neonf\Documents\tdd\tdd>
```

REFACTOR
I created a function that returns the counter if it is found,
If it is not found, I return an error
There i got GREEN