Yidong Fang
CS 472
Lab Testing

https://github.com/Thienguen/Barbell/tree/main/jpacman

Task 1 – JPacman Test Coverage

Coverage: Tests in 'jpacman.test'

Element	Class, %	Method, %	Line, %
nl	3% (4/110)	1% (10/626)	1% (28/2276)
tudelft	3% (4/110)	1% (10/626)	1% (28/2276)
jpacman	3% (4/110)	1% (10/626)	1% (28/2276)
board	20% (4/20)	9% (10/106)	9% (28/282)
fuzzer	0% (0/2)	0% (0/12)	0% (0/64)
game	0% (0/6)	0% (0/28)	0% (0/74)
integration	0% (0/2)	0% (0/8)	0% (0/12)
level	0% (0/26)	0% (0/156)	0% (0/690)
npc	0% (0/20)	0% (0/96)	0% (0/476)
points	0% (0/4)	0% (0/14)	0% (0/38)
sprite	0% (0/12)	0% (0/90)	0% (0/238)
ui	0% (0/12)	0% (0/62)	0% (0/254)
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)

Note that coverage in this test is extremely bad, it covered very few lines.

Task 2 – Increasing Coverage on JPacman

Coverage: Tests in 'jpacman.test'

Element	Class, %	Method, %	Line, %
nl	16% (18/11...	9% (60/626)	8% (190/2308)
tudelft	16% (18/11...	9% (60/626)	8% (190/2308)
jpacman	16% (18/11...	9% (60/626)	8% (190/2308)
board	20% (4/20)	9% (10/106)	9% (28/282)
fuzzer	0% (0/2)	0% (0/12)	0% (0/64)
game	0% (0/6)	0% (0/28)	0% (0/74)
integration	0% (0/2)	0% (0/8)	0% (0/12)
level	15% (4/26)	6% (10/156)	3% (26/700)
npc	0% (0/20)	0% (0/96)	0% (0/476)
points	0% (0/4)	0% (0/14)	0% (0/38)
sprite	83% (10/12)	44% (40/90)	52% (136/260)
ui	0% (0/12)	0% (0/62)	0% (0/254)
Launcher	0% (0/1)	0% (0/21)	0% (0/41)
LauncherSmokeTest	0% (0/1)	0% (0/4)	0% (0/29)
PacmanConfigurationExceptio	0% (0/1)	0% (0/2)	0% (0/4)

Task 2.1

testGetKiller:

```
@Test
void testGetKiller() {

    if (player.isAlive())
        assertThat(player.getKiller()).isNull();
    else
        assertThat(player.getKiller() instanceof Unit).isEqualTo( expected: true);
}
```

Coverage: Tests in 'jpacman.test'

Element	Class, %	Method, %	Line, %
nl	16% (18/11...	9% (62/626)	8% (192/2308)
tudelft	16% (18/11...	9% (62/626)	8% (192/2308)
jpacman	16% (18/11...	9% (62/626)	8% (192/2308)
board	20% (4/20)	9% (10/106)	9% (28/282)
fuzzer	0% (0/2)	0% (0/12)	0% (0/64)
game	0% (0/6)	0% (0/28)	0% (0/74)
integration	0% (0/2)	0% (0/8)	0% (0/12)
level	15% (4/26)	7% (12/156)	4% (28/700)
npc	0% (0/20)	0% (0/96)	0% (0/476)
points	0% (0/4)	0% (0/14)	0% (0/38)
sprite	83% (10/12)	44% (40/90)	52% (136/260)
ui	0% (0/12)	0% (0/62)	0% (0/254)
Launcher	0% (0/1)	0% (0/21)	0% (0/41)
LauncherSmokeTest	0% (0/1)	0% (0/4)	0% (0/29)
PacmanConfigurationExceptio	0% (0/1)	0% (0/2)	0% (0/4)

After *testGetKiller*

testCreatePacMan:

```
@Test
void testCreatePacMan() {
    assertThat(player.getScore()).isEqualTo( expected: 0);
    assertThat(player.isAlive()).isEqualTo( expected: true);

    assertThat(player.getSprite()).isExactlyInstanceOf(factory.getSprites().getPacManDeathAnimation().getClass());

    assertThat(player.getSprite()).isInstanceOf(AnimatedSprite.class);
}
```

Coverage: Tests in 'jpacman.test'

Element	Class, %	Method, %	Line, %
nl	16% (18/110)	10% (68/626)	8% (200/2308)
tudelft	16% (18/110)	10% (68/626)	8% (200/2308)
jpacman	16% (18/110)	10% (68/626)	8% (200/2308)
board	20% (4/20)	11% (12/106)	10% (30/282)
fuzzer	0% (0/2)	0% (0/12)	0% (0/64)
game	0% (0/6)	0% (0/28)	0% (0/74)
integration	0% (0/2)	0% (0/8)	0% (0/12)
level	15% (4/26)	10% (16/156)	4% (34/700)
npc	0% (0/20)	0% (0/96)	0% (0/476)
points	0% (0/4)	0% (0/14)	0% (0/38)
sprite	83% (10/12)	44% (40/90)	52% (136/260)
ui	0% (0/12)	0% (0/62)	0% (0/254)
Launcher	0% (0/1)	0% (0/21)	0% (0/41)
LauncherSmokeTest	0% (0/1)	0% (0/4)	0% (0/29)
PacmanConfigurationExceptior	0% (0/1)	0% (0/2)	0% (0/4)

After *testCreatePacMan*

testGetDeltaX:

```
new *
@Test
void testGetDeltaX() {
    Direction dirN = Direction.NORTH;
    Direction dirW = Direction.WEST;
    Direction dirE = Direction.EAST;
    Direction dirS = Direction.SOUTH;

    assertThat(dirN.getDeltaX()).isEqualTo( expected: 0);
    assertThat(dirW.getDeltaX()).isEqualTo( expected: -1);
    assertThat(dirE.getDeltaX()).isEqualTo( expected: 1);
    assertThat(dirS.getDeltaX()).isEqualTo( expected: 0);
}
```

Coverage: Tests in 'jpacman.test'

Element	Class, %	Method, %	Line, %
nl	16% (18/110)	11% (70/626)	8% (202/2308)
tudelft	16% (18/110)	11% (70/626)	8% (202/2308)
jpacman	16% (18/110)	11% (70/626)	8% (202/2308)
board	20% (4/20)	13% (14/106)	11% (32/282)
fuzzer	0% (0/2)	0% (0/12)	0% (0/64)
game	0% (0/6)	0% (0/28)	0% (0/74)
integration	0% (0/2)	0% (0/8)	0% (0/12)
level	15% (4/26)	10% (16/156)	4% (34/700)
npc	0% (0/20)	0% (0/96)	0% (0/476)
points	0% (0/4)	0% (0/14)	0% (0/38)
sprite	83% (10/12)	44% (40/90)	52% (136/260)
ui	0% (0/12)	0% (0/62)	0% (0/254)
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)

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%	73	155	103	344	20	69	4	12
nl.tudelft.jpacman.npc.ghost		71%		55%	56	105	43	181	5	34	0	8
nl.tudelft.jpacman.ui		78%		47%	54	86	21	144	7	31	0	6
default		0%		0%	12	12	21	21	5	5	1	1
nl.tudelft.jpacman.sprite		86%		59%	30	70	11	113	5	38	0	5
nl.tudelft.jpacman.board		86%		58%	44	93	2	110	0	40	0	7
nl.tudelft.jpacman		67%		25%	12	30	18	52	6	24	1	2
nl.tudelft.jpacman.points		59%		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	1,242 of 4,755	73%	293 of 637	54%	292	590	228	1,039	50	268	6	47

- The coverage results from JaCoCo has a more interactive and visual interface compared to gradlew results from IntelliJ. The biggest difference is that JaCoCo offers branch coverage visualization, which is more intuitive than just the line coverage percentage. The JaCoCo counts assembly-level instructions, and gradlew counts source code lines.

```java
    public Board createBoard(Square[][] grid) {
        assert grid != null;

        Board board = new Board(grid);

        int width = board.getWidth();
        int height = board.getHeight();
        for (int x = 0; x < width; x++) {
            for (int y = 0; y < height; y++) {
                Square square = grid[x][y];
                for (Direction dir : Direction.values()) {
                    int dirX = (width + x + dir.getDeltaX()) % width;
                    int dirY = (height + y + dir.getDeltaY()) % height;
                    Square neighbour = grid[dirX][dirY];
                    square.link(neighbour, dir);
                }
            }
        }

        return board;
    }
```

- I found the source code visualization from JaCoCo very helpful, it clearly tells me which branches are not covered and the source code displayed explains why.
- I prefer JaCoCo because it's the easiest way to achieve 100% coverage.

Task 4 – Working with Python Test Coverage

```python
new *
def test_from_dict(self):
    accdata = ACCOUNT_DATA[len(ACCOUNT_DATA) - 1]
    acc = Account()
    acc.from_dict(accdata)

    for key, val in accdata.items():
        self.assertEqual(getattr(acc, key), val)

new *
def test_update(self):
    data = ACCOUNT_DATA[self.rand]
    acc = Account(**data)
    acc.create()
    tmp = Account(**data)
    tmp.create()

    acc.update()
    self.assertEqual(acc.name, tmp.name)

    acc.id = False
    with self.assertRaises(models.account.DataValidationError):
        acc.update()
```

```python
new *
def test_delete(self):
    data = ACCOUNT_DATA[self.rand]
    acc = Account(**data)
    acc.create()

    acc.delete()
    self.assertEqual(len(acc.all()), 0)

new *
def test_find(self):
    data = ACCOUNT_DATA[self.rand]
    acc = Account(**data)
    acc.create()
    data = ACCOUNT_DATA[len(ACCOUNT_DATA) - 1]
    account = Account(**data)
    account.create()

    self.assertEqual(account.find(acc.id), acc)
```

```
PS C:\Users\Aaron's\OneDrive\Documents\GitHub\test_coverage> nosetests

Test Account Model
- Test creating multiple Accounts
- Test Account creation using known data
- delete
- find
- from dict
- Test the representation of an account
- Test account to dict
- update

Name                  Stmts   Miss  Cover   Missing
-----------------------------------------------------
models\__init__.py        7      0   100%
models\account.py        40      0   100%
-----------------------------------------------------
TOTAL                    47      0   100%
----------------------------------------------------------------------
Ran 8 tests in 0.489s

OK
```

Task 5 - TDD

Test_update_a_counter

Test:

```
new *
def test_update_a_counter(self):
    self.setUp()
    returned = self.client.post('/counters/update')
    self.assertEqual(returned.status_code, status.HTTP_201_CREATED)

    returned = self.client.put('/counters/update')
    self.assertEqual(returned.status_code, status.HTTP_200_OK)
    self.assertEqual(COUNTERS.get('update'), 1)
```

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

update_counter

```
new *
@app.route('/counters/<name>', methods=['PUT'])
def update_counter(name):
    app.logger.info(f"Request to update counter: {name}")
    COUNTERS[name] += 1
    return {name: COUNTERS[name]}, status.HTTP_200_OK
```

```
PS C:\Users\Aaron's\OneDrive\Documents\GitHub\tdd> nosetests

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

Name              Stmts   Miss  Cover   Missing
----------------------------------------------
src\counter.py       16      0   100%
src\status.py         6      0   100%
----------------------------------------------
TOTAL                22      0   100%
----------------------------------------------------------------
Ran 3 tests in 0.172s

OK
```

test_read__a_counter

red

```
new *
def test_read__a_counter(self):
    self.setUp()
    returned = self.client.post('/counters/read')
    returned = self.client.put('/counters/read')
    returned = self.client.get('/counters/read')
    self.assertEqual(returned.status_code, status.HTTP_200_OK)
```

```
======================================================================
FAIL: test_read__a_counter (test_counter.CounterTest)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\Aaron's\OneDrive\Documents\GitHub\tdd\tests\test_counter.py", line 52, in test_read__a_counter
    self.assertEqual(returned.status_code, status.HTTP_200_OK)
AssertionError: 404 != 200
```

test_read__a_counter
green/refactor

```
new *
@app.route('/counters/<name>', methods=['GET'])
def read_counter(name):
    app.logger.info(f"Request to read counter: {name}")
    return {name: COUNTERS[name]}, status.HTTP_200_OK
```

```
PS C:\Users\Aaron's\OneDrive\Documents\GitHub\tdd> nosetests

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

Name              Stmts   Miss  Cover   Missing
----------------------------------------------
src\counter.py       20      0   100%
src\status.py         6      0   100%
----------------------------------------------
TOTAL                26      0   100%
----------------------------------------------------------------
Ran 4 tests in 0.157s

OK
```