Dataflow

November 22nd, 2023
Ana Paiva, José Campos

In this recitation class, we are going to explore 'Dataflow Testing', *a white-box testing technique*, in the `jpacman` project.

Please make sure your machine is configured properly, i.e.:

- Java installed on your machine and available through the command line. Disclaimer: this tutorial has been validated under Java-11. It may or may not work on other versions of Java. Let us know whether it does not work under Java-X, where X is a version higher than 11.
- Apache Maven to be installed on your machine and available through the command line. In case Maven is not installed, please follow the following steps:
    - Download `apache-maven-3.9.4-bin.zip`
    - Extract `apache-maven-3.9.4-bin.zip`
    - On Windows, augment your environment variables with the full path to the `<extracted directory>/bin`. On Linux/MacOS, run `export PATH="<extracted directory>/bin:$PATH"`. (You might have to run the `export` everytime you restart the computer. For a more permanent solution, please consider adding that command to your bash profile.)

1. Perform 'Dataflow Testing'

Given the source code of the `jpacman` project, which you could find in here, we expect you to perform 'Dataflow Testing' on the following two functions.

In a nutshell, apply 'Dataflow Testing' to all **variables** in each function, i.e., identify all-defs, all-c-uses, all-p-uses, and all-uses, and write them in a txt file. Derive the tests and implement them in the JUnit framework.

1.1 `nextAiMove` function in the `nl.tudelft.jpacman.npc.ghost.Blinky` class

```java
/**
 * {@inheritDoc}
 *
```

```
 * <p>
 * When the ghosts are not patrolling in their home corners
(Blinky:
 * top-right, Pinky: top-left, Inky: bottom-right, Clyde:
bottom-left),
 * Blinky will attempt to shorten the distance between Pac-Man
and himself.
 * If he has to choose between shortening the horizontal or
vertical
 * distance, he will choose to shorten whichever is greatest. For
example,
 * if Pac-Man is four grid spaces to the left, and seven grid
spaces above
 * Blinky, he'll try to move up towards Pac-Man before he moves
to the left.
 * </p>
 */
@Override
public Optional<Direction> nextAiMove() {
  assert hasSquare();

  // TODO Blinky should patrol his corner every once in a while
  // TODO Implement his actual behaviour instead of simply
chasing.
  Unit nearest = Navigation.findNearest(Player.class, getSquare());
  if (nearest == null) {
    return Optional.empty();
  }
  assert nearest.hasSquare();
  Square target = nearest.getSquare();

  List<Direction> path = Navigation.shortestPath(getSquare(),
target, this);
  if (path != null && !path.isEmpty()) {
    return Optional.ofNullable(path.get(0));
  }
```

```
    return Optional.empty();
}
```

1.2 `render` function in the `nl.tudelft.jpacman.ui.BoardPanel` class

Java

```java
/**
 * Renders the board on the given graphics context to the given
dimensions.
 *
 * @param board
 *            The board to render.
 * @param graphics
 *            The graphics context to draw on.
 * @param window
 *            The dimensions to scale the rendered board to.
 */
private void render(Board board, Graphics graphics, Dimension window)
{
  int cellW = window.width / board.getWidth();
  int cellH = window.height / board.getHeight();

  graphics.setColor(BACKGROUND_COLOR);
  graphics.fillRect(0, 0, window.width, window.height);

  for (int y = 0; y < board.getHeight(); y++) {
    for (int x = 0; x < board.getWidth(); x++) {
      int cellX = x * cellW;
      int cellY = y * cellH;
      Square square = board.squareAt(x, y);
      render(square, graphics, cellX, cellY, cellW, cellH);
    }
  }
}
```

2. Exercise: write unit tests

Write unit test cases using the JUnit framework to every single test you found in section 1 of this tutorial. Note: in maven projects, tests must be developed under `src/test/java`.

3. What should you submit/deliver?

Zip the project's directory (including the txt file created in Section 1) and submit it here (M.EIC's moodle) or here (MESW's moodle).

Deadline: ~~End of the recitation class.~~ November 22, 2023, 11:59:00 pm.
Grades: available on November 29, 2023.

Miscellaneous

- Guide to Configuring Maven Plug-ins
- JUnit framework
- Learn how to write unit tests
- JUnit 5 User Guide
- Parameterized Tests and JUnit 5 Tutorial: Writing Parameterized Tests