JUnit Tutorial

Instructor: Yongjie Zheng

Instructions:

1. Importing the Lunar Lander folder into Eclipse.

Open Eclipse, select *File* > *Import* > *General* > *Existing Projects into Worksapce*.

In the next panel, click *Browse* next to the option *Select root directory*. Select the unzipped *LunarLander* folder and click *Open*.

The lunar lander project will automatically appear in the available project list. Select it and click *Finish*.

After importing the lunar lander project into your Eclipse workspace, go through the code and make sure that you understand its basic code structure and how it works.

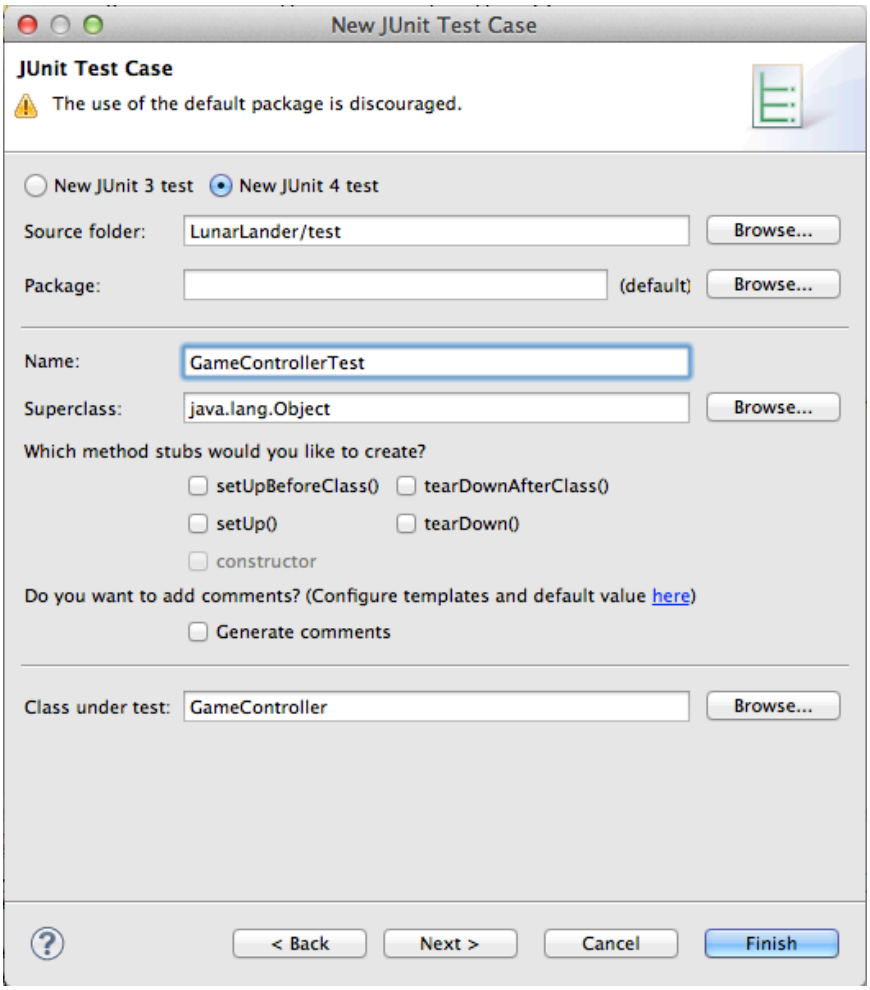
You can also run it as a Java Application. You need to open the Console view to play the game.

2. Creating a JUnit Test Case

It is recommended that you put all your test cases into a new folder (e.g. named *test*), and make it one of your source folders.

To do this, right click your project in the Eclipse workspace, select New > Source Folder, and enter the name “test”.

To create a test case, right click the class that you want to test and select New > Other > Java > JUnit > Junit Test Case and click Next.

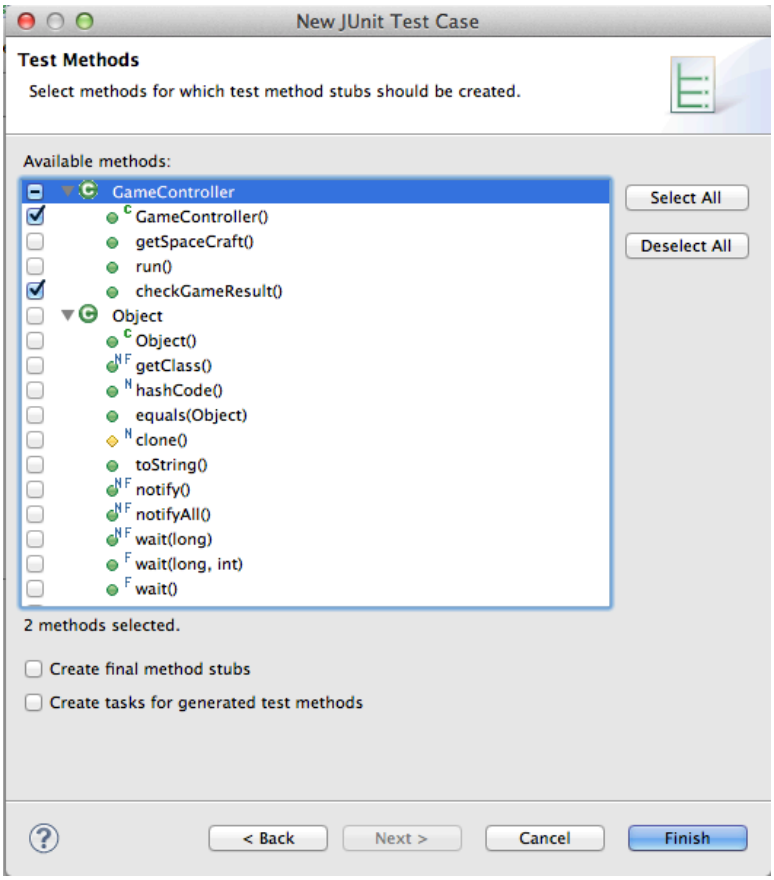


In the next panel shown above, choose the created *test* folder as your source folder.

Make sure that you are using *New JUnit 4 test*.

If necessary, you can also check the checkboxes of *setUp()*, *tearDown()*, etc.

Click Next. In the next panel as shown below, select the method that you want to test and click *FInish*.



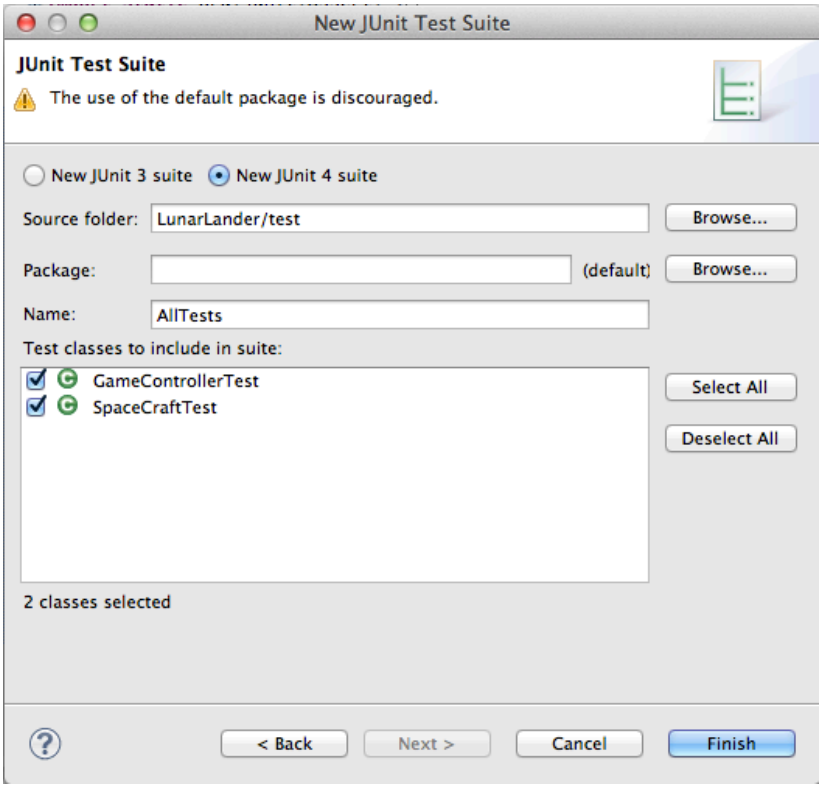
You may be asked to *Add JUnit 4 library to the build path*. Just click OK to confirm. A new test case will be created in the folder you specified (i.e. test).

Open the generated test case, and fill in specific testing logics to complete the test case.

3. Creating JUnit Test Suite

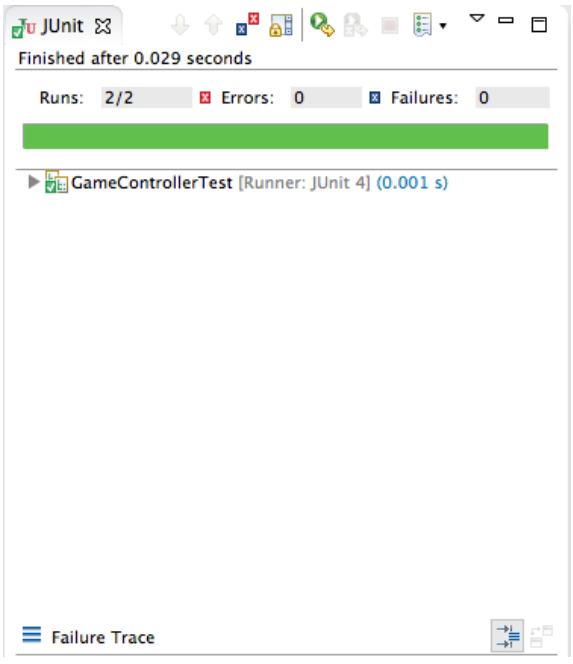
To create a test suite, select all the test cases that you want to include. Right click your mouse and select *New > Other > Java > JUnit > JUnit Test Suite*. Click Next and you will see the following panel.

Again, use *test* as your source folder. Click *FInish*.



4. Running JUnit tests in Eclipse

To run the test cases you created, right click your test case and select Run As > JUnit Test. The result can be checked in the Eclipse JUnit view as shown below.



Exercise

JUnit API: http://junit.sourceforge.net/javadoc

Task 1: Create a JUnit test case for GameController of the lunar lander game. The test case is initially implemented as follows. **You need to add one more test into this test case**.

```
private GameController gc;

@Before
public void setUp() throws Exception {
      gc = new GameController();
}

@After
public void tearDown() throws Exception {
      gc = null;
}

@Test
public void testGameController() {
      assertNotNull(gc.getSpaceCraft());
}

@Test
public void testCheckGameResult() {
      gc.getSpaceCraft().setAltitude(10);
      assertEquals(0,gc.checkGameResult());
}
```

Task 2: Create a JUnit test case for SpaceCraft of the lunar lander game. The initial implementation is as follows. Again, add one more test into this test case.

```
@Test
public void testCalcNewValues() {
      SpaceCraft sc = new SpaceCraft();
      sc.init();
      assertEquals(0, sc.getBurnRate());

      sc.setBurnRate(1000);
```

```
        try {
                sc.calcNewValues();
                fail("Exception should be thrown");
        } catch (Exception e){
                ;//Exception expected
        }

        sc.setBurnRate(5);
        sc.setFuel(45);
        try{
                sc.calcNewValues();
        } catch (Exception e){
                ;//Ignore this time
        }
        assertEquals(40, sc.getFuel());
}

@Ignore
public void oldTest(){
        fail("If executed, this test should fail.");
}
```

Task 3: Create a test suite that includes the two test cases you created above.

Task 4: Run your test cases and test suite.