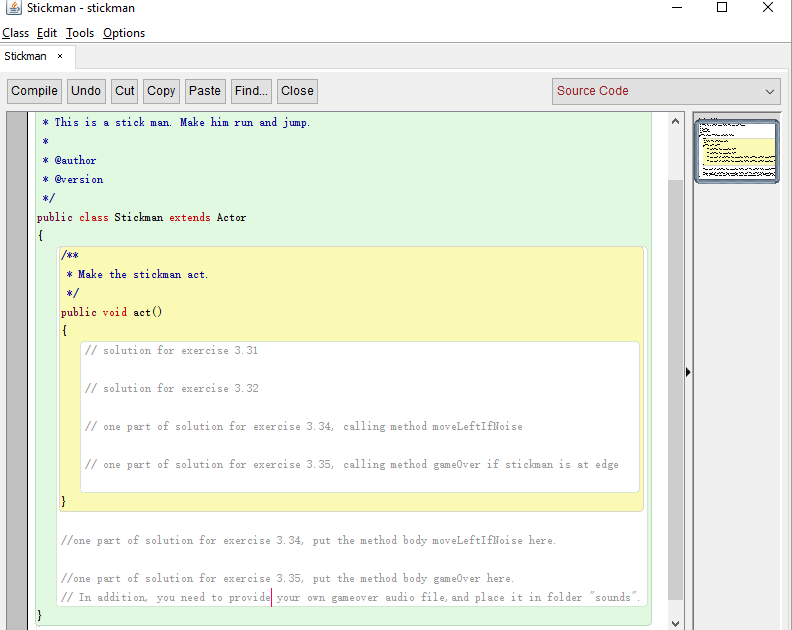
CS2163 Java Homework 3 requirement

Homework 3 needs to be finished on the Greenfoot platform. Homework 3 does not involve Eclipse.

**Follow the steps below to finish homework 3:**

* work on these four exercises in textbook: 3.31, 3.32, 3.34 and 3.35 (exercise 3.33 is NOT included). It requires you to open the existing Greenfot scenario folder “stickman”, and then modify its source code.
* provide the solutions for all these 4 exercises in one scenario project “stickman”, more specifically, all in this class Stickman:
  + exercise 3.31 and 3.3.2: solutions should be in method act() of class Stickman
  + exercise 3.34: provide another method **moveLeftIfNoise** in class Stickman, after method **act**, then in method **act**, invoke this method **moveLeftIfNoise** .
  + exercise 3.35: provide another method **gameOver** in class Stickman, after method **act**, then in method **act**, invoke this method **gameOver** when it is the correct condition to trigger the invocation.



The comment lines starting with // should be presented in your solution, so that they serve as separate lines for the four exercises, and for different methods.

* **Hint for this homework:** when you work on the solution of one exercise, you can use block comment to comment out the solutions for the other three exercises, so that you can focus on one solution at a time. When all four solutions have been run and tested correctly one by one, then you need to remove all the block comments, so that only the line comments stays in the code to serve as separate lines for the solutions of the four exercises. Then make sure your code compiles before you generate the zip file

Block comment look like this, and it can comment out multiple lines.

/\*\*

multiple lines can go

inside a block comment

\*/

* Please understand that the four exercises in this homework are logically independent from each other, and please do NOT try to connect them together logically. Therefore, the solution for each exercise must be tested alone by itself, without the interference from the rest three solutions. In order to do so, you need to use the block comment in your source code to comment out the other three exercises, when you are debugging and testing one exercise. For example, when you are testing your code for exercise 3.32, then you need to use block comment to comment out your code for exercise 3.31, 3.34, and 3.35, so on and so forth, until all four exercises have been tested individually and successfully. Then you need to remove all the block comments in your source code, and then you need to add the line comments to separate the solution of the four exercises, so that I know which part of your source code is for which exercise. Then you can zip the greenfoot project and submit it. A sample line comment look like this, and it is also shown on the picture in page 1 of this document:

// solution for exercise 3.31

* After the block comments for all four exercises are removed from your source code, and the line comments are added in your source code, your source code still need to be compilable.
* To finish exercise 3.34, you can finish exercise 3.33 first, then you can modify the code of exercise 3.33 to fulfill the requirement of exercise 3.34. Exercise 3.33 is not required in this homework, because if one can finish exercise 3.34, then the solution for exercise 3.34 already includes the solution for exercise 3.33.
* In exercise 3.32, it mentioned that “use an if-statement and the microphone input method you found above”, and the “found above” means exercise 3.26. You may need to read exercise 3.26 and understand which method to call in class Greenfoot, in order to get the microphone level.
* When you are testing your solution for exercise 3.32 or 3.34, and if your code in Geenfoot does not receive the microphone input very well, you can follow the instruction in textbook sec 3.9 and open the Recorder in your Greenfoot IDE: go to Greenfoot IDE, select menu item:

***Tools*** 🡪 ***Show Sound Recorder*** 🡪 and then click the ***Record*** button, while you are debugging your code. This action will make sure that the Greenfoot IDE is listening to the microphone.

* Zip the entire scenario folder “stickman”, and rename the zip file as “***JohnDoeHw3.zip***”, where JohnDoe needs to be replaced by your first and last name
* Submit the zip file ***JohnDoeHw3.zip*** to Moodle “homework 3 drop box”.

Exercises 3.26 ~ 3.30 are the exercises before the four exercise required in this homework, and if you choose to review these preceding exercises, it will help you to find solutions for this homework, but you don’t need to include the solution for Exercise 3.26 ~ 3.30 in this homework.

**After finishing this homework, how to verify the correctness of your submitted zip files:**

1. Download the zip files you have uploaded to Moodle homework drop box.
2. Unzip the zip file to a different local folder in your computer, other than the original local folder where the zip files are generated.
3. Run the Greenfoot project from the unzip folder, and make sure it compiles and runs correctly.
4. If your submitted zip file in the Moodle drop box

**cannot be downloaded,** or

**cannot be unzipped,** or

**cannot compile,** or

**cannot run,**

then you need to figure out the reason and fix the error, and then submit the corrected zip file to the Moodle drop box. Then start this verification process again until you can download, unzip, compile and run successfully. To upload a corrected zip file to the Moodle drop box, you need to delete the previous submitted zip file from the Moodle drop box first.

In the first page of file “chap1-schedule.docx”, you can find the instructions on how to zip and unzip files.

**Grading components:**

* Each of these four exercises has 5 points, thus results in a total of 20 points for this homework

**For any submitted zip file that still has syntax error and it cannot compile or run in Greenfoot, it will receive ZERO point**. No re-submission is allowed after the homework due day.

Please click the Moodle homework drop box to see the due day of this homework.

When coding in Eclipse and/or Greenfoot, please read document “RulesForIndentAndAlignCode.docx” in Moodle folder “chap 1”, and follow all the rules in code alignment and indentation.