CS2163 Java bonus homework “autumn click” requirement

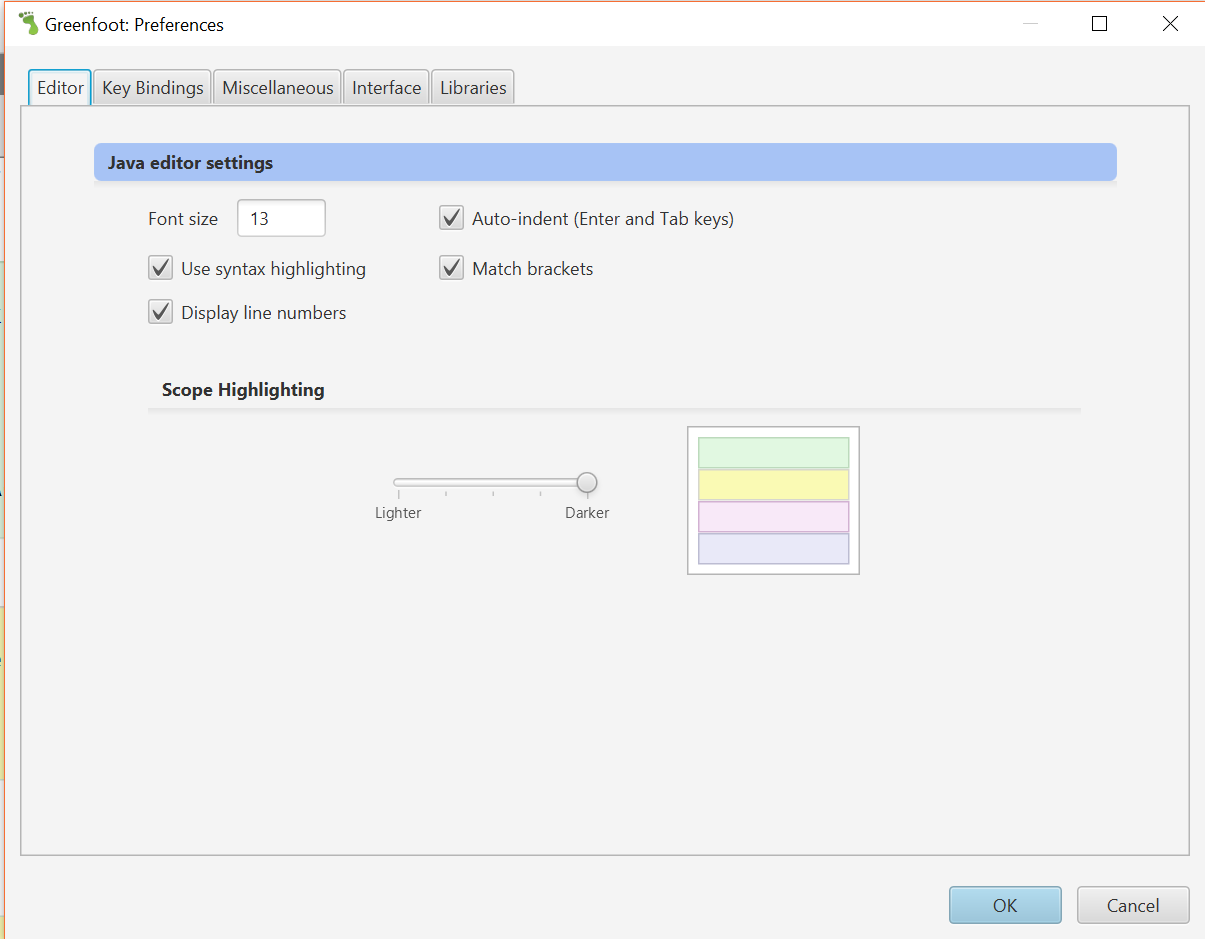
This is bonus homework “autumn click” needs to be finished on the Greenfoot platform, and it does not involve Eclipse. This bonus homework has three parts: part A, B, and C, and each part has 10 points. You can choose to finish all or some parts. If you finish all three parts correctly, you will receive 12 bonus points.

**Follow the steps below to finish Part A of this bonus homework:**

Based on the existing Greenfoot scenario “autumn-2” in textbook chapter 7, you need to do the following:

1. In Windows Filer Explorer or Mac Finder, make a copy of the folder “autumn-2”, and name the copy folder as ***JohnDoeAutumnClickA***, where JohnDoe should be replaced by your name.

2. Open this project in Greenfoot IDE and then double click class Block to view this class in Greenfoot source code editor. If the line numbers of the source code are not being displayed already, then in Greenfoot source code editor, click menu Option 🡪 Preference, and then this window will pop up, and then check the “Display Line Numbers” box.



3. go to line 49 in Block.java, and you will see this statement:

if (Greenfoot.mouseClicked(null))

Then visit greenfoot online api <https://www.greenfoot.org/files/javadoc/> , then click class [Greenfoot](https://www.greenfoot.org/files/javadoc/greenfoot/Greenfoot.html), then scroll down and click method [mouseClick](https://www.greenfoot.org/files/javadoc/greenfoot/Greenfoot.html#mouseClicked-java.lang.Object-), and then read the detail description of this method, and you can find out that the parameter of mouseClick can have three possible types, as listed below:

* the parameter of mouseClick is an Actor object
* the parameter of mouseClick is a World object
* the parameter of mouseClick is the null object

By reading the detail description of this method, you need to learn and understand what is the impact of using any one of the three parameters above as the parameter for method Greenfoot.mouseClicked ().

4, copy and paste line 49 as a new line in the file, which is line 50, and then use line comment // to comment out line 49. And then remove the parameter of method mouseClick() in line 50, which is null. Then **you need to supply a suitable parameter for mouseClick() method,** so that mouseClick() method **only** responds to your click on the block object itself.

Run and test your program and make sure it works.

5. copy and paste line 50 as a new line in the file, which is line 51, and then use line comment // to comment out line 50. And then remove the parameter of method mouseClick() in line 51. Then **you need to supply a suitable parameter for mouseClick() method**, so that mouseClick() method **only** responds to your click on the background World object.

Run and test your program and make sure it works.

When you finish line 49, 50, and 51 should look like these three lines:

// if (Greenfoot.mouseClicked (null) ) // line 49

// if (Greenfoot.mouseClicked (a suitable parameter provided by you) ) // line 50

if (Greenfoot.mouseClicked (a suitable parameter provided by you) ) // line 51

The parameter you provided in line 50 should satisfy the requirement for step 4.

The parameter you provided in line 51 should satisfy the requirement for step 5

Step 4 above has 2 points. Step 5 above has 2 points. **Total points for part A: 4.**

**Hint for the homework**: you need to read the online API of mouseClick() method carefully, and understand what impact of the three different parameters can bring.

**Part A is finished.**

**Follow the steps below to finish Part B of this bonus homework:**

Based on the existing Greenfoot scenario “autumn-2” in textbook chapter 7, you need to do the following:

1. In Windows Filer Explorer or Mac Finder, make a copy of the folder “autumn-2”, and name the copy folder as ***JohnDoeAutumnClickB***, where JohnDoe should be replaced by your name.

2. Open this project in Greenfoot IDE and then double click class Block to view this class in Greenfoot source code editor. If the line numbers of the source code are not being displayed already, then in Greenfoot source code editor, click menu Option 🡪 Preference, and then check the “Display Line Numbers” box.

3. use block comment below to comment out line 49 to 58 of Block.java.

/\*\* line 49 to 58 goes between this block comment.

One block comment can comment out multiple lines (a block)

\*/

After you place the block comment as indicated, method checkMouseClick in line 47 has an empty body.

4. Now you need to supply code in method checkMouseClick(), so that whenever you click a single leaf, this single leaf will change color(change image), and all other leaves does not change colors (images).

Run and test your program and make sure it works. **Total points for part B: 4.**

**Part B is finished.**

**Follow the steps below to finish Part C of this bonus homework:**

Based on the existing Greenfoot scenario “autumn-2” in textbook chapter 7, you need to do the following:

1. In Windows Filer Explorer or Mac Finder, make a copy of the folder “autumn-2”, and name the copy folder as ***JohnDoeAutumnClickC***, where JohnDoe should be replaced by your name.

2. Open this project in Greenfoot IDE and then double click class Block to view this class in Greenfoot source code editor. If the line numbers of the source code are not being displayed already, then in Greenfoot source code editor, click menu Option 🡪 Preference, and then check the “Display Line Numbers” box.

3. use block comment below to comment out line 49 to 58 of Block.java.

/\*\* line 49 to 58 goes between this block comment.

One block comment can comment out multiple lines (a block)

\*/

After you place the block comment as indicated, then method checkMouseClick in line 47 has an empty body.

4. Now you need to supply code in method checkMouseClick(), so that whenever you click any single leaf, then all leaves in the background change color (change image), including the one being clicked and all others that are not being clicked.

Run and test your program and make sure it works. **Total points for part C: 4.**

**Part C is finished.**

**Hint for part B and C**: the reason why I ask you to comment out line 49~58 in the original source code is to let you create your own code to meet the requirements of part B and C. The new code should have a different logic than the original code in line 49~58. Nevertheless, you can still use similar code in original line 51 and 52 to obtain a list of leaves and save it in an object of ***List<Leaf>*** class type, and you can still use a for-each loop similar to line 54 to navigate through this list of leaves.

**How to submit part A, B, and C together in one zip file**:

First, make sure that these three Greenfoot scenario folders: ***JohnDoeAutumnClickA, JohnDoeAutumnClickB,*** and ***JohnDoeAutumnClickC*** are under one parent folder, for example, folder “\book-scenarios\chapter07\”.

Then under this parent folder “chapter07”, first click folder ***JohnDoeAutumnClickA*** , then hold down the ***ctrl*** key in windows or the Command key ⌘  in mac, then click the other two folders: ***JohnDoeAutumnClickB*** and ***JohnDoeAutumnClickC***.

Now all three folders are selected, then right click any one of the selected folders, then in the context menu displayed, choose menu item “Send to 🡪 compressed (zipped) folder” in windows; or choose menu item “Compress Items” in Mac.

Now a zip will be created including all three folders. Rename this zip file ***JohnDoeAutumnClick.zip***, where JohnDoe should be your name.

Submit the zip file ***JohnDoeAutumnClick.zip*** to Moodle “bonus homework autumn drop box”.

**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 projects from the unzip folder, and make sure it compiles and runs correctly.
4. If your submitted zip file in the Moodle drop box
   1. **cannot be downloaded,** or
   2. **cannot be unzipped,** or
   3. **cannot compile,** or
   4. **cannot run,**
   5. 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.

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