**chap 11 Simulations, chap 13 Additional scenario ideas** and **Final group project**

**Objectives**

* Understand how to implement simulations in Greenfoot
* Understand how to use the random number generation method in Greenfoot for simulation
* Learn the basic of exception handling

**Schedule:** This lesson covers Moodle folder “chap 11: Simulations; chap 13: Additional scenario ideas; Final project”, and we release the final project. Follow the steps below.

1, work on sec 11.1 ~ 11.6, which include scenarios: ants, ants-2, ants- 3, and foxes-and-rabbits. Go through the exercises in chapter 11, and cross reference the source code in these scenarios, and see how these requirements are implemented.

2, read sec 13.1 ~ 13. 8 that have 8 scenarios. Run these scenarios, and notice that they cover how to use image, sound, simulation, and map.

3, download and unzip “exceptionSrcCode.zip”, and study the included java source files, and learn the basic try-catch block to handle exception. Also, you should notice the difference between having exception handling and without, and benefit of having it. For the final project, it is optional to use exception handling.

4, Final project of the semester:

This is a group project. Each group has 2 students. The teacher will send out the grouping information in week 14.

The requirement of the final project: create an interactive game in Greenfoot, and it must have user interaction via keyboard or mouse, and it must incorporate sound and image. A score or notice must be displayed when the game is finished.

The game must be created from scratch, and it cannot be an improvement of an existing scenario from the textbook, or any existing scenario you download from Internet or any other sources. Attention: violation of this requirement will be considered academic dishonesty, and it will results in a grade of 0 for the final project.

Grading criteria:

1. user interaction, either through keyboard or mouse 10 points
2. use GreenfootImage class to create image objects and call APIs of these objects. 10 points
3. use GreenfootSound class to create sound objects and call APIs of these objects. 10 points
4. need to have a score system during the game, and a notice after the game is finished, 10 points
5. the game logic must be correct and be reflected in source code, 10 points. For example, the game must have a clear set of rules as how the user can obtain points, and/or lose points, and how the user can win or lose the game, and how the user interactions are associated with the winning and losing of points/game.

Further explanations for the criterion #2 and #3 above:

* You need to create at least two GreenfootImage objects and two GreenfootSound objects.
* For each GreenfootImage object, you need to call at least two APIs in GreenfootImage class
* For each GreenfootSound object, you need to call at least two APIs in GreenfootSound class
* If you don’t create any object from GreenfootImage class, you will receive 0 point for criterion #2
* If you don’t create any object from GreenfootSound class, you will receive 0 point for criterion #3
* If you don’t call any APIs from GreenfootImage class, you will receive 0 point for criterion #2
* If you don’t call any APIs from GreenfootSound class, you will receive 0 point for criterion #3
* Calling method ***Greenfoot.PlaySound(filename)*** is not considered as creating a GreenfootSound object

How to submit the final project:

After finishing the final project, coding and debugging and running and testing, and if everything meets the requirement, you need to write an README.TXT file, which briefly explains the rules of the your game, and how should the user play the game, including the winning or failing conditions. It does not need to be lengthy, and clarity is crucial. Place this README.TXT file inside the scenario folder.

Then you need to use Windows “File Explorer” or Mac “Finder” to zip the scenario folder into a zip file, and rename the zip file as “***JohnAndSueFinal.zip***”, where John and Sue need to be replaced by the names of the two group members.

Submit ***JohnAndSueFinal.zip*** into Moodle drop box “final project drop box”. For any submission that still has syntax error so that it cannot compile or run in Greenfoot, it will receive zero points. No re-submission is allowed after the due day. For both online and on-campus section, please click on the Final Project drop box in Moodle folder to see its due day.

For the remaining lessons, it will be used for group time working on the final project.

**For on-campus sections**, each group needs to present its group project during class meeting time in the last week of the semester. If class meets more than once per week, each group can choose which day is the presentation day. The file ***JohnAndSueFinal.zip*** is due one day earlier before each group’s presentation time.

**For online section**, each group needs to come to campus and present the final project, and a 10-minutes time slot need to be reserved in the last week of the semester as below:

* 11am~1:45pm on Monday or Wednesday of last week (May 11 or 13), or
* 1:30pm~2:50pm on Tuesday or Thursday of last week (May 12 or 14)

Each group may take 10 minutes or so to present the final project, including the question and answer time from instructor. When it comes close to presentation at the end of the semester, I will notify students on which classroom is reserved for presentation.

There is no need to create any ppt slides for the presentation. What I am looking at from the presentation is that, each student in each group needs to be able to explain his/her contribution to the project, and I will check the final project submission with the grading criteria listed in this document**. If a student claims a portion of the source code as his/her contribution but cannot explain the source code, then no point will be given on the final project for this student.**

Each student in the group must make contribution to the coding of the final project, and without coding contribution, no points will be given for a student’s final project score.

For online students, you need to communicate with each other through email, and participate and contribute in the final project.