First Practical Exercise

Programming

Grado en Ingeniería Informática

Universidad Carlos III de Madrid

Programming – Grado en Ingeniería Informática	
Authors *Of the English version:* Juan Gómez Romero *Based on the work by:* Ángel García Olaya Manuel Pereira González Gustavo Fernández-Baillo Cañas	

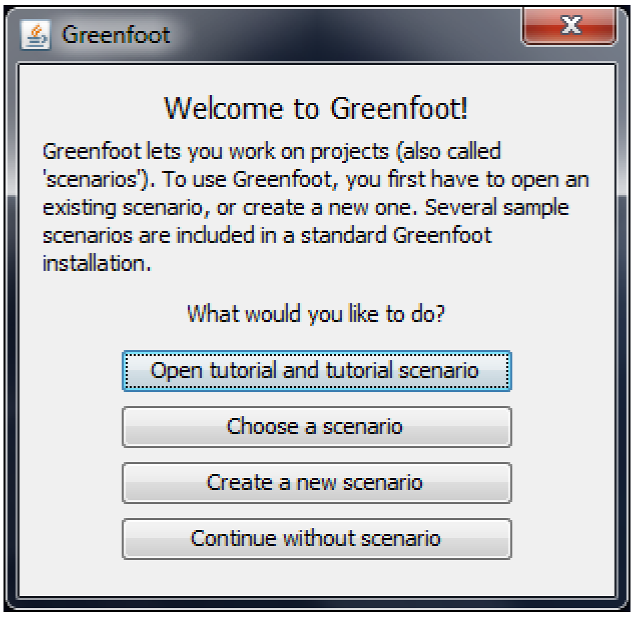
1. First practical exercise: Greenfoot

1.1 Introduction

The main objective of this practical exercise is to introduce the basic concepts of imperative and object-oriented programming. In that regard, some exercises to be developed with Greenfoot are proposed. Greenfoot is a development environment created by the University of Kent and the University of Deakin, and can be freely downloaded from the web page http://greenfoot.org/download/.

Greenfoot is a software tool designed to let beginners get experience with object-oriented programming. It supports development of graphical applications in the Java™ Programming Language. (Source: Greenfoot Tutorial)

Students are recommended to finish the tutorial included in the Greenfoot distribution before starting the development of the practical exercise, in order to get used to the environment, the syntax and the concepts that will be studied. The tutorial files (html + scenario) are installed with the environment and can be started when the Greenfoot software is executed for the first time (select *Open Tutorial and tutorial scenario*). The tutorial html files can be also found in the "tutorial" folder of the Greenfoot installation folder.


Greenfoot

Welcome to Greenfoot!

Greenfoot lets you work on projects (also called 'scenarios'). To use Greenfoot, you first have to open an existing scenario, or create a new one. Several sample scenarios are included in a standard Greenfoot installation.

What would you like to do?

Open tutorial and tutorial scenario
Choose a scenario
Create a new scenario
Continue without scenario

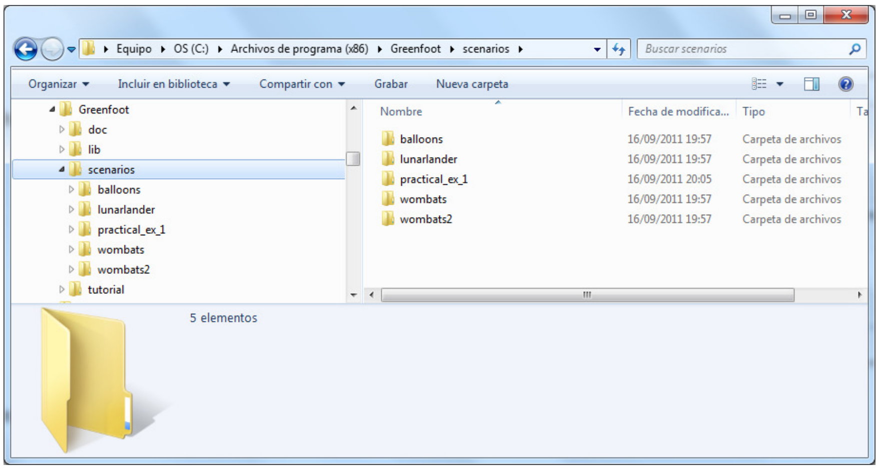
NOTE: Students are recommended to try to solve only the "Getting started" section of the tutorial. "Programming" and next sections are focused on more advanced topics.

This practical exercise consists of a single exercise divided in several tasks. Students must develop the tasks described in section 2.2 in a single Greenfoot scenario.

1.2 Exercise

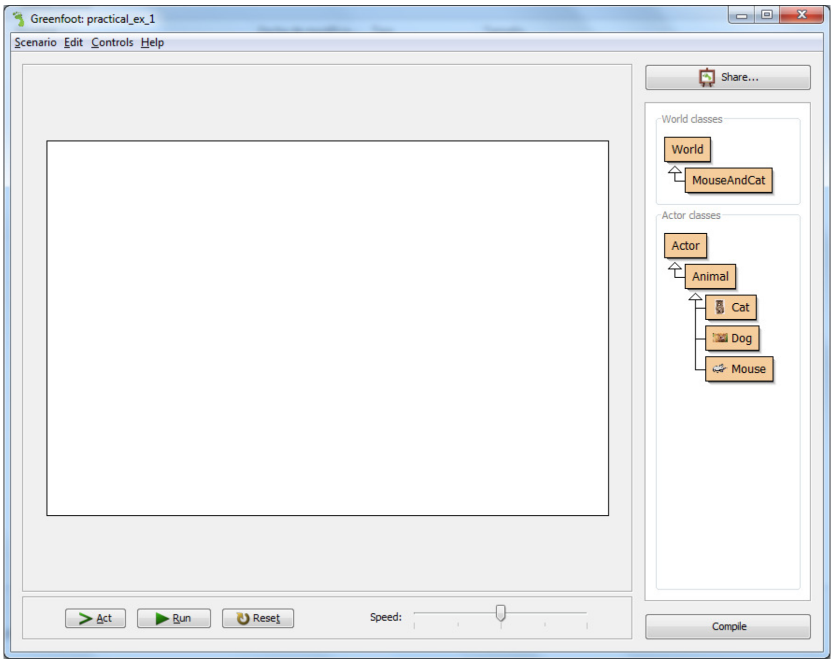
In this section, we will present some important concepts of the Java programming language, and how they are realized in Greenfoot. To do so, we will start from a Greenfoot scenario with some classes representing animals. We will modify the code of the scenario to change the behavior of the animals.

Please download the file “practical_ex_1.zip” from Aula Global. Unzip the file in the “scenarios” folder of the Greenfoot installation. Start Greenfoot and open “practical_ex_1” scenario.



Nombre	Fecha de modifica...	Tipo
balloons	16/09/2011 19:57	Carpeta de archivos
lunarlander	16/09/2011 19:57	Carpeta de archivos
practical_ex_1	16/09/2011 20:05	Carpeta de archivos
wombats	16/09/2011 19:57	Carpeta de archivos
wombats2	16/09/2011 19:57	Carpeta de archivos

NOTE: In Windows 7, you may experience some problems with the permissions of the files in the Greenfoot installation folder. If this is your case, copy the “scenarios” folder to a different location of your hard disk (e.g., Desktop) and use the newly created files.



1.2.1 Exercise 1

As seen in the tutorial, the first step to create a scenario in Greenfoot is to insert the objects that participate in the world and study their behavior. Open the "practical_ex_1" scenario and proceed to solve the following tasks:

Task	**1.5**
• Change the background of the scenario: ○ Right click on the *MouseAndCat* class ○ Select *Set image* ○ Select *backgrounds > weave* • Create two *Mouse* objects and leave them on the scenario at different positions • Run the *act()* method on one of the new *Mouse* objects ○ Right click on the *Mouse* object ○ Select *void act()* • Run the *move()* method on one of the new *Mouse* objects. Is there any difference with respect to *act()*? Why? ○ Right click on the *Mouse* object ○ Select *inherited from Animal > void move()* • Click the *Act* button of the scenario and study what happens • Click the *Run* button of the scenario and study what happens • Click the *Pause* button	

1.2.2 Exercise 2

Task	**1.5**
• Create another *Mouse* object and put it on the central section of the scenario • Run the *void turn(int angle)* method on the new *Mouse* object [inherited from Animal]. Use different parameter values, both positive and negative. What happens? Why? • Run the methods *void turnLeft()* and *void turnRight()* [inherited from Animal] • Run the method *boolean atWorldEdge()* [inherited from Animal]. What happens? • Drag and drop the mouse to the edge of the board • Run again *boolean atWorldEdge()* [inherited from Animal]. What happens?	

1.2.3 Exercise 3

Task	**2**
• Edit the *Animal* class ○ Right click on *Animal* class ○ Select *Open editor* • Study the Java code that implements the methods *turnRight()* and *turnLeft()*, used in the previous exercise • Modify these two methods to make any animal turn 17º instead of 45º • Add comments to the methods to explain how they are implemented	

- Close the editor and go back to the scenario window
- Click the *Compile* button
- Create a *Mouse* object and put it on the central section of the scenario
- Run the methods *void turnLeft()* and *void turnRight()* to test the new implementation [inherited from Animal]

1.2.4 Exercise 4

Task	**3**

- Edit the *Mouse* class
- Study the Java code that implements the methods *act()* used in exercise 1
- Modify this method to make mice turn to right (instead of turning to left) when they reach the scenario edge
- Modify this method to make mice randomly turn just 10% of the times (instead of 50% of the times) before moving
- Add comments to the methods to explain how they are implemented
- Close the editor and go back to the scenario window
- Click the *Compile* button
- Create some *Mouse* objects on the scenario
- Click the *Run* button. Is there any difference with respect to the previous implementation?

1.2.5 Exercise 5

Task	**2**

- Edit the *MouseAndCat* class
- Study the Java code that implements the methods *MouseAndCat,* in particular the last but one sentence:

  // populateWorld();
- Remove the comment to activate this sentence
- Close the editor and go back to the scenario window
- Click the *Compile* button
- Click the *Run* button. Is there any difference with respect to the previous implementation?
- As you can see, the scenario is initialized with 2 dogs, 1 cat and 5 mice. Modify the *MouseAndCat* class to create 3 dogs and 8 mice
- Close the editor and go back to the scenario window
- Click the *Compile* button
- Select *Scenario* > *Scenario information* on the upper menu and read the game instructions
- Click the *Run* button. Play the game! Reset the game to try to defeat all the crazy dogs