Greenfoot - A Highly Graphical IDE for Learning Object-Oriented Programming

Michael Kölling
Computing Laboratory,
University of Kent
mik@kent.ac.uk

ABSTRACT
Greenfoot is an educational IDE that makes it easy to write interactive graphical applications. It helps to solve several problems in the teaching and learning of object-oriented programming: It provides educational tools that aid in understanding fundamental object-oriented concepts, and it is highly motivational through instant graphical feedback.

Categories and Subject Descriptors
D.2.3 [**Software Engineering**]: Coding Tools and Techniques.
K.3.2 [**Computers & Education**]: Computer & Information Science Education - *Computer Science Education*

General Terms
Design, Experimentation, Human factors.

Keywords
Computing education, IDE, Java, Greenfoot.

1. INTRODUCTION
One of the important challenges we face in introductory programming courses is to stir the interest and curiosity of our students. We do not only have to teach the concepts, we also have to motivate students to develop an interest in the field of computer science in general or programming in particular.

Finding an interesting and engaging entrance to programming, however, does not only serve to increase participation of otherwise unmotivated students. Those who are already well motivated also greatly benefit from working in an engaging and stimulating context.

2. GREENFOOT
The Greenfoot environment [1] provides a platform that allows a highly interactive, visual and engaging approach to teaching object-oriented programming in Java. Greenfoot is two things:

- a pedagogical integrated development environment designed to be easy to use for beginners. As such, it integrates the common tools (such as an editor, compiler, virtual machine) with educational tools (such as interactive object invocation, inspection, class visualisation).
- a simulation and micro-world framework. Using Greenfoot, existing micro-worlds, such as turtle graphics, Karel the robot, or Gridworld, can easily be implemented, while providing a much higher degree of interaction than in their original implementations. In addition, other graphical applications, such as simulations and games, can also be written.

The Greenfoot system makes it very easy to write applications that use two-dimensional animated graphics. A typical example are computer games, but scientific simulations or interactive animations also fall into this category. The Greenfoot system encapsulates the actual graphics code, so that programmers can concentrate fully on programming object behaviour. This design provides an effective educational tool that focuses students on the important concepts while providing immediate visual feedback.

The educational tools (such as object interaction and object inspection) are aligned with the BlueJ environment [2] to facilitate easy migration to that system at a later stage.

Greenfoot can be used with many different scenarios from different topic areas. Thus, it can be tailored to specific target groups to serve special interest areas.

The system can be used at various levels of education:

- High school/CS0 courses: Since the target audience at this stage has not necessarily made a decision to study computer science, motivation is especially important. Linking programming learning to known application areas, such as computer games, is effective in achieving this.
- CS1 courses, which benefit from the pedagogical tools that support the understanding of object-oriented concepts.
- Advanced courses. Since the language is full Java, no limit exists to the complexity that Greenfoot programs may take.

Greenfoot is available free from www.greenfoot.org.

3. REFERENCES
[1] Henriksen, P. and Kölling, M., Greenfoot: Combining object visualisation with interaction. in *Companion to the 19th annual ACM SIGPLAN OOPSLA conference*, (Vancouver, Canada, 2004), ACM, 73-82.

[2] Kölling, M., Quig, B., Patterson, A. and Rosenberg, J. The BlueJ System And Its Pedagogy. *Journal of Computer Science Education, Special issue on Learning and Teaching Object Technology*, *13* (4). 249-268.

ITiCSE'08, June 30–July 2, 2008, Madrid, Spain.
ACM 978-1-60558-115-6/08/06.