Introduction to Programming with Greenfoot

Object-Oriented Programming in Java With Games and Simulations

Michael Kölling

To Krümel and Cracker—may their imagination never fade.

—mk

Education is not the filling of a pail, but the lighting of a fire.

—William Butler Yeats

Companion Website

Additional material and resources for this book can be found at

http://www.greenfoot.org/book/

For students:

- The Greenfoot software
- The scenarios discussed in this book
- The Greenfoot Gallery—a scenario showcase
- Tutorial videos
- A discussion forum
- Technical support

For teachers:

- A teacher discussion forum
- Additional exercises related to the book
- The "Green Room" containing worksheets and other teaching resources

Table of Contents

List of scenarios discussed in this book

Leaves and wombats (chapter 2)

This is a simple example showing wombats moving around on screen, occasionally eating leaves. The scenario has no specific purpose other than illustrating some important object-oriented concepts and Greenfoot interactions.

Asteroids 1 (chapter 2)

This is a simple version of a classic arcade game. You fly a spaceship through space and try to avoid being hit by asteroids. At this stage, we only use the scenario to make some small changes and illustrate some basic concepts.

Little crab (chapters 3, 4, 5)

This is out first full development. Starting from almost nothing, we develop a simple game slowly, adding may things such as movement, keyboard control, sound, and many other elements of typical games.

Piano (chapter 6)

An on-screen piano that you can really play.

Newton's lab (chapter 7)

Newton's Lab is a simulation of the motion of stars and planets in space. Gravity plays a central role here. We also make a variant of this that combines gravity with making music, ending up with musical output triggered by objects under gravitational movement.

Asteroids 2 (chapter 8)

We come back to the asteroids example from chapter 2. This time, we investigate more fully how to implement it.

Ants (chapter 9)

A simulation of ant colonies searching for food, communicating via drops of pheromones left on the ground.

The following scenarios are presented in Chapter 10, and selected aspects of them briefly discussed. They are intended as inspiration for further projects.

Marbles

A simulation of a marble board game. Marbles have to be cleared of the board within a limited number of moves. Contains simple physics.

Lifts

A start of a lift simulation. Incomplete at this stage—can be used as a start of a project.

Boids

A demo showing flocking behavior: A flock of birds flies across the screen, aiming to stick together while avoiding obstacles.

Circles

Make patterns in different colors on the screen with moving circles.

Explosion

A demo of a more sophisticated explosion effect.

Breakout

This is the start of an implementation of the classic Breakout game. Very incomplete, but with an interesting visual effect.

Platform jumper

A demo of a partial implementation of an ever-popular genre of games: platform jumpers.

Wave

This scenario is a simple demonstration of a physical effect: the propagation of a wave on a string.

Acknowledgements

This book is the result of more than five years of work, by a group of people. First and foremost involved are the people who contributed to the development of the Greenfoot environment, which makes this entire educational approach possible. Poul Henriksen started the implementation of Greenfoot as his Masters project and build the first prototype. He also took on the development of this prototype into a production system. For the first year or so we were a two-man project, and Poul's work led to the quality and robustness of the current system.

Bruce Quig and Davin McCall were the next developers to join the project, and Poul, Bruce and Davin jointly built most of Greenfoot as it is today. All three are exceptional software developers, and their contribution to the project cannot be overstated. It is a joy working with them.

Eventually, the whole "BlueJ Group" got involved in the Greenfoot project, including John Rosenberg and Ian Utting, and this book builds on contributions and joint work of all group members.

Colleagues in the Computing Laboratory at the University of Kent also helped me a great deal, especially our Head of Department, Simon Thompson, who saw the value of Greenfoot early on and supported and encouraged its further development.

Another important contribution, without which the development of Greenfoot (and ultimately, this book) would not have been possible, is the generous support of Sun Microsystems. Emil Sarpa, Katherine Hartsell, Jessica Orquina, Sarah Hammond, and many others within Sun believed in the value of our system and provided important support.

Everyone at Pearson Education worked very hard to get this book published on time, with a very tight schedule, and in sometimes difficult circumstances. Tracy Dunkelberger worked with me on this book from the beginning. She managed amazingly well to stay positive and excited while putting up with my repeated missed deadlines, and still encouraged me to continue writing. Melinda Haggerty did a whole lot of different things, including managing the reviews.

A special thank you needs to go to the reviewers of this book, who have provided very detailed, thoughtful and useful feedback. They are Carolyn Oates, Damianne President, Detlef Rick, Gunnar Johannesmeyer, Josh Fishburn, Mark Hayes, Marla Parker, Matt Jadud, Todd O'Bryan, Lael Grant, Jason Green, Mark Lewis, Rodney Hoffman, and Michael Kadri. They helped spotting many errors and pointed out many opportunities for improvement.

My good friend Michael Caspersen also deserves thanks for providing early feedback and encouragement that was very important to me. Partly because it helped improve the book, and more importantly because it encouraged me to believe that this work might be interesting to teachers, and worthwhile completing.

Introduction

Welcome to Greenfoot! In this book, we will discuss how to program graphical computer programs, such as simulations and games, using the Java Programming Language and the Greenfoot environment.

There are several goals in doing this: one is to learn programming, another is to have fun along the way. While the examples we discuss in this book are specific to the Greenfoot environment, the concepts are general: working through this book will teach you general programming principles in a modern, object-oriented programming language. However, it will also show you how to make your own computer game, a biology simulation, or an on-screen piano.

This book is very practically oriented. Chapters and exercises are structured around real, hands-on development tasks. First, there is a problem that we need to solve, then we look at language constructs and strategies that help us solve the problem. This is quite different from many introductory programming textbooks which are often structured around programming language constructs.

As a result, this book starts with less theory, and more practical activity than most programming books. This is also the reason we use Greenfoot: It is the Greenfoot environment that makes this possible. Greenfoot allows us to play. And that does not only mean playing computer games; it means playing with programming: we can create objects, move them around on screen, call their methods, observe what they do, all interactively and easily. This leads to a more hands-on approach to programming than what would be possible without such an environment.

A more practical approach does not mean that the book does not cover the necessary theory and principles as well. It's just that the order is changed. Instead of introducing a concept theoretically first, and then doing some exercises with it, we often jump right in and use a construct, initially explaining only as much as necessary to solve the task at hand, then come back to the theoretical background later. We typically follow a spiral approach: We introduce some aspects of a concept when we first encounter it, then revisit it later in another context, and gradually deepen our understanding.

The emphasis throughout is to make the work we do interesting, relevant, and enjoyable. There is no reason why computer programming has to be dry, formal, or boring. Having fun along the way is okay. We think we can manage making the experience interesting and pedagogically sound at the same time. This is an approach that has been called *serious fun*—we do something interesting, and learn something useful along the way.

This book can be used both as a self-study book, or as a textbook in a programming course. Exercises are worked into the text throughout the book—if you do them all, you will come out of this as a fairly competent programmer.

The projects discussed in this book are easy enough that they can be managed by high school students, but they are also open and extendable enough that even seasoned programmers can find interesting and challenging aspects to do. While Greenfoot is an educational environment, Java is not a toy language. Since Java is our language of choice for this book, the projects discussed here (and others you may want to create in Greenfoot) can be made as complex and challenging as you like.

While it is possible to create simple games quickly and easily in Greenfoot, it is equally possible to build highly sophisticated simulations of complex systems, possibly using artificial intelligence algorithms, agent technology, database connectivity, or anything else you can think of. Java is a very rich language that opens the whole world of programming, and Greenfoot imposes no restrictions as to which aspects of the language you can use.

In other words: Greenfoot scales well. It allows easy entry for young beginners, but experienced programmers can also implement interesting, sophisticated scenarios.

You are limited only by your imagination.