

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 that 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, and 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 to make the experience interesting and pedagogically sound at the same time.

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, network communication, 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.

Programming is a creative discipline, and Greenfoot is a tool that helps you build what you invent.