

[Download ebook] Realm of Racket: Learn to Program, One Game at a Time!

Realm of Racket: Learn to Program, One Game at a Time!

*Von Matthias Felleisen, David Van Horn, Conrad Barski, Northeastern University Students
ebooks | Download PDF | *ePub | DOC | audiobook*



 Download

 Read Online

Produktinformation -Verkaufsrank: #363620 in eBooksVerffentlicht am: 2013-06-13Erscheinungsdatum: 2013-06-13File Name: B00DHRV584 | File size: 16.Mb

Von Matthias Felleisen, David Van Horn, Conrad Barski, Northeastern University Students : Realm of Racket: Learn to Program, One Game at a Time! before purchasing it in order to gage whether or not it would be worth my time, and all praised Realm of Racket: Learn to Program, One Game at a Time!:

KundenrezensionenHilfreichste Kundenrezensionen4 von 4 Kunden fanden die folgende Rezension hilfreich. mixed feelingsVon B LehnertA lot has been written about LISP and it's magical powers and how it has done all the things modern languages do 50 years ago and so on and so on. Wise men praise LISP for decades but the mainstream resists to follow. To finally understand the whole debate a little better I decided to read a book about it (well, I tried the little elephants book before and that shure taught me recursion but it's not the same). Not planning to make it my main language Racket appeared to be the ideal dialect (a dialect of scheme which is a dialect of LISP but what the heck, I want to get the ideas behind all that detail...). It is great for test-driving as it is easy to install, comes with batteries

included and leaves enough room for further development towards other interesting topics (planning to learn some logic programming in the future?) You just don't have to worry where to get the GUI libraries, database connection, what IDE to use... - it's all just there and it has very good documentation at your fingertips and it has passed the test of time (back when it was called PLT Scheme...). The backside of Racket: If you like learning from books, you have little choice. How to design Programs (HtDP) is an old book, the new version being online to read for years and Realm of Racket. I like dead tree that I can read in a hotel with fragile internet connection and on the beach with no internet connection. So this is, what you get (go learn Java if you need a lot of choices in books...). The title with "Games" and the pictures leave the impression of a programming book for newbies. Well, it is not. The fast pace in which concepts like a REPL, recursion or functions being passed to functions or the lot of (for) functions is handled is good for people with a little experience in functional programming. The games are not trivial but their construction is not described in the book. It's a "read the code to see more, we'll only explain the new concepts herein" approach. Again, that's not for a newbie programmer and experienced programmers may like this or not. I passed the "read the code to the whole program" part and was busy enough reading the book. Games are used as examples in order not to shy those away who are bored of computing fibonacci numbers with each new language again. Even for someone with no interest in writing games this is not a bad approach. For someone with sincere interest in game development: Games are a means of teaching in this book. It is not really a book about game development. I'm having real problems trying to figure, whom this book is for. It is easy enough for light reading but heavy in content. Some important aspects are covered very shortly. There is a kind of appendix called "Close Paren" where object orientation in Racket is explained in 4 pages and macros are described in another 4 pages. Not enough for me to grasp, why macros apparently are what helped Paul Graham beating the averages. So in a way, the book did not do what I read it for (see above). Why still 4 stars? It is not a 5 star book, but the book made me start programming in Racket. I finished the first useful program within three or four days. It used lots of things that are not covered in the book (drawing on canvas, building input forms, making menus,...) but I was able to read the built-in help and draw the necessary information out of it for everything I planned to do. I now plan to write something new in Racket, that was originally planned to be programmed in another language. So in a way it helped me a lot towards Racket and that's worth 4 stars. If you are patient enough you can learn a lot about the main authors intentions on this book on youtube here: [...] and you can preview some of the games that are not truly programmed within the book here: [...] 1 von 5 Kunden fanden die folgende Rezension hilfreich. Racket a Scheme descendant Von Claus Von Der Kueste I was looking for a compact documentation of the Racket language and IDE to understand some code snippets in the dissertation of Neil Toronto about "Trustworthy, Useful Languages for Probabilistic Modeling and Inference". Racket is an extension of Scheme which is an extension of Lisp - the language of symbolic AI -. RoR is the only reference to Racket in book form. Unfortunately it is not addressed to academics but to newbies in programming. Similar in style to Conrad Barski's "Land of Lisp" it is devoted to "Learn to Program, One Game at a Time!" (this is the subtitle). Concepts range from simple functional programming to higher functions with nested lambdas, lazy evaluation, memoization, to imperative programming with states and GUIs. I would be glad if I had such an introduction nearly forty years ago. At that time I had to learn imperative Algol for solving dull numerical problems. Learning Racket makes it necessary to switch to a different language when doing some serious programming afterwards. Clojure is a natural follow up language. Scala is not bad too. Martin Odersky feels himself in the tradition of Abelson Sussmans seminal book "Structure and Interpretation of Computer Programs". So beware, when learning Racket you later have to move on to a second production language. What's the rating? 5 stars for the didactic concept, minus 2 stars for the verbose sophomore-orientated style and the missing references.

Kurzbeschreibung Racket is a descendant of Lisp, a programming language renowned for its elegance, power, and challenging learning curve. But while Racket retains the functional goodness of Lisp, it was designed with beginning programmers in mind. Realm of Racket is your introduction to the Racket language. In Realm of Racket, you'll learn to program by creating increasingly complex games. Your journey begins with the Guess My Number game and coverage of some basic Racket etiquette. Next you'll dig into syntax and semantics, lists, structures, and conditionals, and learn to work with recursion and the GUI as you build the Robot Snake game. After that it's on to lambda and mutant structs (and an Orc Battle), and fancy loops and the Dice of Doom. Finally, you'll explore laziness, AI, distributed games, and the Hungry Henry game. As you progress through the games, chapter checkpoints and challenges help reinforce what you've learned. Offbeat comics keep things fun along the way. As you travel through the Racket realm, you'll: Master the quirks of Racket's syntax and semantics Learn to write concise and elegant functional programs Create a graphical user interface using the 2htdp/image library Create a server to handle true multiplayer games Realm of Racket is a lighthearted guide to some serious programming. Read it to see why Racketeers have so much fun! **Kurzbeschreibung** Racket is a descendant of Lisp, a programming language renowned for its elegance, power, and challenging learning curve. But while Racket retains the functional goodness of Lisp, it

was designed with beginning programmers in mind. Realm of Racket is your introduction to the Racket language. In Realm of Racket, you'll learn to program by creating increasingly complex games. Your journey begins with the Guess My Number game and coverage of some basic Racket etiquette. Next you'll dig into syntax and semantics, lists, structures, and conditionals, and learn to work with recursion and the GUI as you build the Robot Snake game. After that it's on to lambda and mutant structs (and an Orc Battle), and fancy loops and the Dice of Doom. Finally, you'll explore laziness, AI, distributed games, and the Hungry Henry game. As you progress through the games, chapter checkpoints and challenges help reinforce what you've learned. Offbeat comics keep things fun along the way. As you travel through the Racket realm, you'll:

- Master the quirks of Racket's syntax and semantics
- Learn to write concise and elegant functional programs
- Create a graphical user interface using the `2htdp/image` library
- Create a server to handle true multiplayer games

Realm of Racket is a lighthearted guide to some serious programming. Read it to see why Racketeers have so much fun!

ber den Autor und weitere Mitwirkende

Professor Matthias Felleisen is one of the original authors of the Racket language and a co-author of *The Little Schemer* and *How to Design Programs*. In 1995, Felleisen launched the TeachScheme! Project, reaching out to high schools with a radically novel computing curriculum. Felleisen is the 2012 recipient of the ACM Special Interest Group on Programming Languages (SIGPLAN) Lifetime Achievement Award as well as ACM's 2009 Karl V. Karlstrom Outstanding Educator Award. He is currently a Trustee Professor in the College of Computer and Information Science at Northeastern University in Boston, Massachusetts.

Conrad Barski has an M.D. from the University of Miami and nearly 20 years of programming experience. The author of *Land of Lisp*, Barski is also an avid cartoonist, having created the popular alien Lisp mascot and many graphical tutorials.

David Van Horn is a research professor at Northeastern University who has programmed in Racket and Scheme for over a decade.