Karel the Robot

A Gentle Introduction to the Art of Programming

Second Edition

Richard E. Pattis *Revised by* Jim Roberts • Mark Stehlik

KAREL THE ROBOT

A Gentle Introduction to the Art of Programming

Second Edition

P324

Richard E. Pattis

Revision By:

Jim Roberts
School of Computer Science
Carnegie Mellon University

Mark Stehlik
School of Computer Science
Carnegie Mellon University

John Wiley & Sons, Inc.
New York ▪ Chichester ▪ Brisbane ▪ Toronto ▪ Singapore

Acquisitions Editor	Steven Elliot
Marketing Manager	Susan Elbe
Assistant Marketing Manager	Laura McGayhey
Senior Production Editor	Nancy Prinz
Designer	Lynn Rogan
Manufacturing Manager	Susan Stetzer

This book was set in 10/12 Times Roman by Publication Services and was printed and bound by Courier Stoughton. The cover was printed by Phoenix Color Corp.

Recognizing the importance of preserving what has been written, it is a policy of John Wiley & Sons, Inc. to have books of enduring value published in the United States printed on acid-free paper, and we exert our best efforts to that end.

Library of Congress Cataloging-in-Publication Data

Pattis, Richard E.
Karel the robot : a gentle introduction to the art of programming / Richard E. Pattis.–2nd ed. / revision by Jim Roberts, Mark Stehlik.
p. cm.
Includes index.
ISBN 0-471-59725-2
1. Electronic digital computers–Programming. I. Roberts, Jim (Jim A.), 1946– . II. Stehlik, Mark. III. Title.
QA76.6.P38 1995
005.1–dc20 94-8087
CIP

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

KAREL THE ROBOT

To the Pattises, Shaffers, and Olshanskys—REP

To Linda—JR

To my parents, Lillian and Ladislav, and to Sylvia—MS

PREFACE

The programming landscape has changed significantly since the initial publication of *Karel the Robot* in 1981. Today there are new programming languages, new programming *paradigms*, and new and more powerful computers. Even though Pascal no longer enjoys the popularity it did in the 1980s, Karel is still as vibrant and valid an introduction to the programming and problem-solving processes as it was when first introduced.

We believe that most people will not actually have to program a computer as part of their everyday lives, either now or in the future. However, many people will need to be able to use a computer and will occasionally need to do something with the machine beyond the "ordinary." Simply put, they will have to solve some type of computer problem. We solve various kinds of problems every day; problem solving is part of our lives. This book will introduce you to problem-solving approaches that can be used with computers. Unfortunately, some people believe programming requires a "different" way of thinking. We don't agree with this statement. Instead of changing the way you think, this book will change how you apply your problem-solving skills to different kinds of problems.

For the experienced student programmer, this edition should provide insights into the problem-solving and program design processes that will make the student an even better programmer. It will also improve understanding of computer science concepts such as loop invariants and recursion. For individuals who want to begin a thorough sequence of training and education in programming, computer science, or both, Karel provides a solid foundation on which to begin your work.

For novice programmers, this book will give some insight into the programming process from two distinctly different points of view: the planner's and the implementer's. All the problems can be thought about, discussed, and planned in English. Once you have developed your plan, the actual syntax of the robot programming language has very few rules to get in your way as you become the implementer or programmer.

For individuals who do not want to program but need to have a feel for the process, Karel is an excellent tool for providing that insight.

Supplements. An instructor's manual is also available for the text that contains a detailed description of the changes from the first edition; numerous pedagogical suggestions for teaching the material based on many years of using Karel in introductory programming courses at the college level; and solutions to all the in-text exercises.

Software to simulate Karel is available on IBM PCs, Apple Macs, and a variety of "mainframes." The PC/Mac software can be directly purchased from John Wiley & Sons (accompanying copies of this book). Site licenses for all machines can be obtained by contacting:

Richard E. Pattis
2823 Broadway Avenue E.
Seattle, WA 98102

March 1994

Jim Roberts
Mark Stehlik

ACKNOWLEDGMENTS

This edition could never have existed without the first edition, so it is clearly appropriate to acknowledge the creative genius of Richard Pattis and the groundbreaking manuscript that is the first edition of *Karel the Robot*. It is always easier to modify wheels rather than reinvent them, and it is even easier when the wheel with which you are starting is almost perfectly circular.

Ideas are never formed in a vacuum, and we would like to acknowledge the Intro Programming Group at Carnegie Mellon University for providing a positive torr environment in which to test out ideas, see which ones work, and run with those that do. There are many individual people who help make an idea into a book. First and foremost, the authors would like to thank Steven Elliot, acquisitions editor at John Wiley, for believing in this project and getting it started. The authors would like to express their appreciation to the following people who reviewed all versions of the second edition manuscript: Roland Untch, Clemson University; Joe Kmoch, Washington High School, Milwaukee, Wisconsin; Billibon Yoshimi, Columbia University; Peter Henderson, SUNY/Stony Brook; Georgette Geotsi, University of Bridgeport; Tim Thurman, University of Kansas; and Peter Casey, Central Oregon Community College. We also acknowledge the help provided by Maria Fischer in getting everything out the door on time.

We also thank the following people who provided feedback on the first version of the manuscript and helped point us in the right direction: Phyllis Sturman; Douglas Grannes; Terry Brink, Lock Haven University of Pennsylvania; and Gordon Bugby, University of Pittsburgh.

Thanks also to Linda and Syl.

JR
MS

I want to take this opportunity to thank Jim Roberts and Mark Stehlik for writing the second edition of *Karel the Robot*. As long-time teachers and experimenters with *Karel,* they have brought to this book many fresh ideas, new insights, and interesting examples and problems.

I also want to thank Phil Miller, for being a tireless promoter (and amplifier) of the ideas embodied in *Karel*; Douglas B. Stein, for translating the Karel Simulator to run on PCs; and Steven Elliot, for having the persistence and clout to ensure the second edition was published.

I attribute the lengthy success of the first edition to a seminal idea (thanks to Seymour Papert) and the excellent feedback I received while translating this idea into a book. For their invaluable comments, I remain indebted to Marsha Berger, Jim Boyce, Denny Brown, Michael Clancy, Kalen Delaney, Tom Dietterich, Joseph Faletti, Bob Filman, John Gilbert, Brent Hailpern, Wayne Harvey, Mike Kenniston, Jock Mackinlay, Keith L. Phillips, Mark Tuttle, and David Wall.

Finally, I want to thank my original editor, Gene Davenport, and the staff at Wiley who helped me create a book that was successful enough to spawn a second edition: Elaine Rauschal, Loretta Saracino, and Rosemary Wellner.

REP

CONTENTS

Chapter One The Robot World 1

1.1 Karel's World 1
1.2 Karel's Capabilities 3
1.3 Tasks and Situations 4
1.4 Problem Set 6

Chapter Two Primitive Instructions and Simple Programs 7

2.1 Changing Position 7
2.2 Handling Beepers 9
2.3 Finishing a Task 9
2.4 A Complete Program 10
2.4.1 Executing a Program 11
2.4.2 The Form of Robot Programs 11
2.5 Error Shutoffs 13
2.6 Programming Errors 14
2.6.1 Bugs and Debugging 17
2.7 Problem Set 19

Chapter Three Extending Karel's Vocabulary 25

3.1 Creating a More Natural Programming Language 25
3.2 A Mechanism That Defines New Instructions 26
3.3 Block Structuring 27
3.4 The Meaning and Correctness of New Instructions 28
3.5 Defining New Instructions in a Program 29
3.6 Boxing: How Karel Understands a Program 32
3.7 An Ungrammatical Program 34

3.8 Tools for Designing and Writing Karel Programs 35
- 3.8.1 Stepwise Refinement—A Technique for Planning, Implementing, and Analyzing Robot Programs 36
  - 3.8.1.1 The First Step—An Overall Plan 37
  - 3.8.1.2 The Second Step—Planning **harvest-2-rows** and **position-for-next-harvest** 39
  - 3.8.1.3 The Third Step—Planning **harvest-1-row** and **go-to-next-row** 40
  - 3.8.1.4 The Final Step—Verifying That the Complete Program Is Correct 42
- 3.8.2 The Design Tree—A Planning Technique 43
  - 3.8.2.1 The First Step—An Overall Plan 44
  - 3.8.2.2 The Second Step—Adding More Branches 45
- 3.8.3 Implementing the Plan Using Vertical Slicing 47
  - 3.8.3.1 The First Slice 48
  - 3.8.3.2 The Second Slice 49
  - 3.8.3.3 The Third and Fourth Slices 50
  - 3.8.3.4 The Last Slice 52

3.9 Advantages of Using New Instructions 53
- 3.9.1 Avoiding Errors 54
- 3.9.2 Future Modifications 54
- 3.9.3 A Program Without New Instructions 56

3.10 Writing Understandable Programs 57

3.11 Problem Set 59

Chapter Four Conditionally Executing Instructions 65

4.1 The **IF/THEN** Instruction 65

4.2 The Conditions Karel Can Test 66

4.3 Simple Examples of the **IF/THEN** Instruction 67
- 4.3.1 The **harvest-1-row** Instruction 67
- 4.3.2 The **face-north-if-facing-south** Instruction 69
- 4.3.3 The **face-north** Instruction 70
- 4.3.4 Boxing the **IF/THEN** Instruction 72

4.4 The **IF/THEN/ELSE** Instruction 74

4.5 Nested **IF** Instructions 76

4.6 More Complex Tests 79

4.7 When to Use an **IF** Instruction 80

4.8 Transformations for Simplifying **IF** Instructions 81

4.9 The Dangling **ELSE** 84

4.10 Problem Set 87

Chapter Five Instructions That Repeat 93

5.1 The ITERATE Instruction 93
5.2 The WHILE Instruction 95
5.2.1 Why WHILE Is Needed 95
5.2.2 The Form of the WHILE Instruction 96
5.2.3 Building a WHILE Loop—The Four-Step Process 97
5.2.4 A More Interesting Problem 98
5.3 Errors to Avoid with WHILE Loops 100
5.3.1 The Fence Post Problem 101
5.3.2 Infinite Execution 102
5.3.3 When the Test of a WHILE Is Checked 103
5.4 Nested WHILE Loops 104
5.4.1 A Good Example of Nesting 104
5.4.2 A Bad Example of Nesting 107
5.5 WHILE and IF Instructions 112
5.6 Reasoning About Loops 112
5.7 A Large Program Written by Stepwise Refinement 116
5.8 When to Use a Repeating Instruction 126
5.9 Problem Set 128

Chapter Six Advanced Techniques for Karel 141

6.1 Recursion 141
6.2 Searching 145
6.3 Doing Arithmetic 149
6.4 Problem Set 152

Appendix 155

Technical Term Index 157

Instruction Index 159

1 THE ROBOT WORLD

This chapter introduces Karel[1] the robot and sketches the world it inhabits. In later chapters, where a greater depth of understanding is necessary, we will amplify this preliminary discussion.

1.1 KAREL'S WORLD

Karel lives in a world that is unexciting by present-day standards (there are no volcanoes, Chinese restaurants, or symphony orchestras), but it does include enough variety to allow the robot to perform simply stated, yet interesting, tasks. Informally, the world is a grid of streets that Karel can traverse. It also contains special objects that Karel can sense and manipulate.

Figure 1–1 is a map illustrating the structure of Karel's world, whose shape is a great flat plane with the standard north, south, east, and west compass points. This world is bounded on its west side by an infinitely long vertical wall extending northward. To the south, it is bounded by an infinitely long horizontal wall extending eastward. These boundary walls are made from solid *neutronium*, an impenetrable metal that restrains Karel from falling over the edge of the world.

Crisscrossing the world are horizontal streets (running east–west) and vertical avenues (running north–south) at regular, one-block intervals. To help you distinguish between streets and avenues, remember that the A in Avenue points north and the V in aVenue points south. A corner, sometimes referred to as a street corner or an intersection, is located wherever a street and an avenue intersect. Karel can be positioned on any corner, facing one of the four compass points. Both streets and avenues are numbered; consequently, each corner is uniquely

[1] Karel is named after Czechoslovakian dramatist Karel Čapek, who popularized the word "robot" in his play R.U.R. (Rossum's Universal Robots). The word "robot" was derived from the Czech word "robota," meaning forced labor.

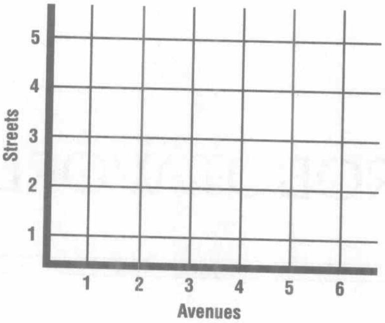

Figure 1–1 Karel's World

identified by its street and avenue numbers. The corner where 1st Street and 1st Avenue intersect is named the origin. The absolute location of the origin is the intersection of 1st Street and 1st Avenue. The origin also has a relative location; it is the most southwesterly corner in Karel's world. We will use both absolute and relative locations to describe the position of Karel and objects in Karel's world.

Besides Karel, there can be two other types of objects in the world. The first type is a wall section. Wall sections are also fabricated from the impenetrable metal neutronium, and they can be manufactured in any desired length and pattern. They are positioned between adjacent street corners, effectively blocking

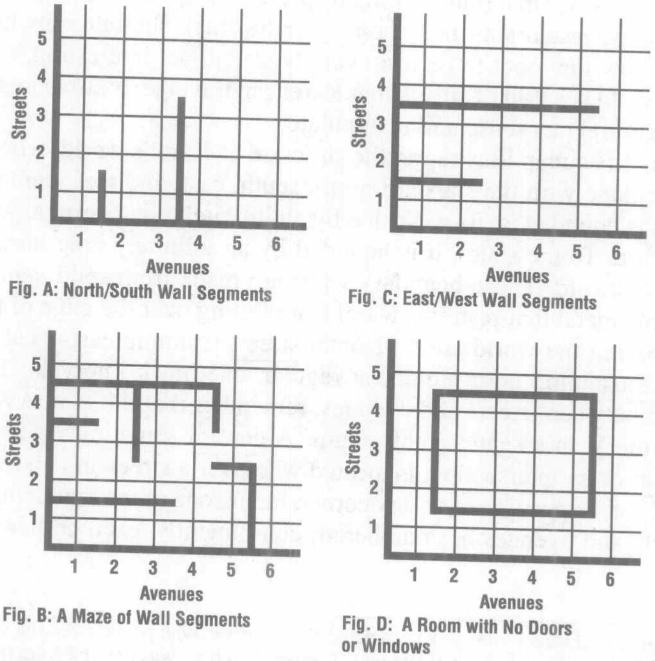

Fig. A: North/South Wall Segments

Fig. C: East/West Wall Segments

Fig. B: A Maze of Wall Segments

Fig. D: A Room with No Doors or Windows

Figure 1–2 Different Wall Segment Arrangements in Karel's World

Karel's direct path from one corner to the next. Wall sections are used to represent obstacles around which Karel must navigate such as hurdles and mountains. Enclosed rooms, mazes, and other barriers can also be constructed from wall sections. Figure 1–2 shows some typical wall arrangements Karel might find in the world.

The second type of object in Karel's world is a beeper. Beepers are small plastic cones that emit a quiet beeping noise. They are situated on street corners and can be picked up, carried, and put down by Karel. Some of Karel's tasks involve picking up or putting down beepers arranged in patterns or finding and transporting beepers. Figure 1–3 shows one possible beeper arrangement.

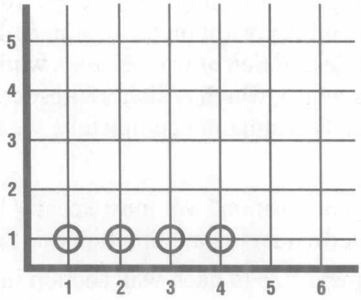

Figure 1–3 One Pattern of Beepers in Karel's World

1.2 KAREL'S CAPABILITIES

We now shift our attention away from Karel's world and concentrate on Karel itself. Karel is a mobile robot; it can move forward in the direction it is facing, and it can turn in place. Karel can also perceive its immediate surroundings using rudimentary senses of sight, sound, direction, and touch.

Karel sees by using any one of three television cameras that point straight ahead, to the left, and to the right. These three cameras are focused to detect walls exactly half a block away from Karel. Karel also has the ability to hear a beeper, but only if the beeper is on the same corner as the robot. (The beepers beep quietly.) By consulting an internal compass, Karel can determine which direction is being faced. Finally, Karel is equipped with a mechanical arm that can be used to pick up and put down beepers. To carry these beepers, Karel wears a soundproof beeper-bag around its waist. Karel can also determine if it is carrying any beepers in this bag by probing the bag with the mechanical arm.

Whenever we want Karel to accomplish a task in the world, we must supply a detailed set of instructions that explains how to perform the task. Karel is able to read and follow such a set of instructions, which is called a program.

What language do we use to program (here we use "program" to mean "write instructions for") Karel? Instead of programming Karel in English, a natural language for us, we write the program in a special programming language, which was designed to be useful for writing robot programs. Karel's robot programming language has a vocabulary, punctuation marks, and rules of grammar. This language is simple enough for Karel to understand; yet it is powerful and concise enough to allow us to write brief and unambiguous programs for Karel.

1.3 TASKS AND SITUATIONS

A task is just something that we want Karel to do. The following examples are tasks for Karel: move to the corner of 15th Street and 10th Avenue, run a hurdle race (with wall sections representing hurdles), escape from an enclosed room that has a door, find a beeper and deposit it on the origin, and escape from a maze.

A situation is an exact description of what Karel's world looks like. Besides the basic structure of Karel's world, which is always present, wall sections and beepers can be added. To specify a situation completely, we must state the following information.

- What is Karel's current position? We must specify both Karel's location (in absolute or relative terms) and what direction it is facing.
- What is the location and size of each wall section in the world?
- What is the location of each beeper in the world? This information includes specifying the number of beepers in Karel's beeper-bag.

We specify situations in this book by a small map or a brief written description. If we know the number of beepers that Karel has in the beeper-bag, then the maps in Figure 1–4 completely specify different situations. The initial situation for any task is defined to be the situation in which Karel is placed at the start of the task. The final situation is the situation that Karel is in when it turns itself off. Unless told otherwise, you may assume that Karel starts all tasks with an empty beeper-bag.

Figure 1–4 shows six initial situations that are typical for tasks that Karel will accomplish in the coming chapters.

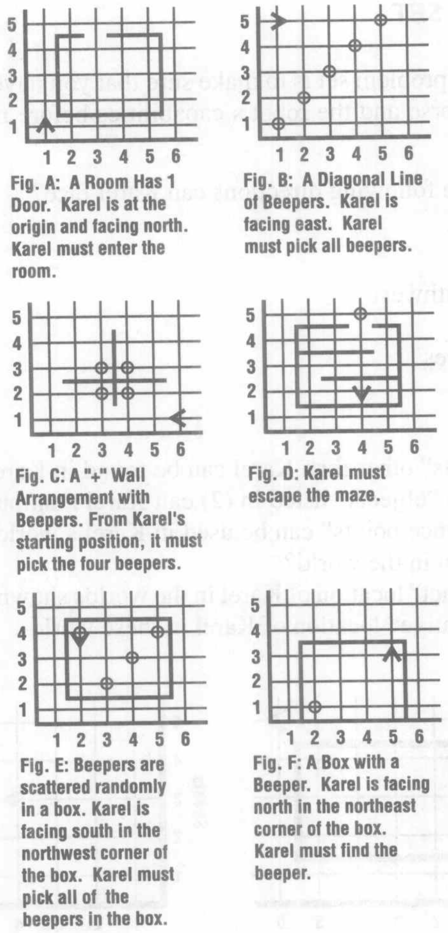

Fig. A: A Room Has 1 Door. Karel is at the origin and facing north. Karel must enter the room.

Fig. B: A Diagonal Line of Beepers. Karel is facing east. Karel must pick all beepers.

Fig. C: A "+" Wall Arrangement with Beepers. From Karel's starting position, it must pick the four beepers.

Fig. D: Karel must escape the maze.

Fig. E: Beepers are scattered randomly in a box. Karel is facing south in the northwest corner of the box. Karel must pick all of the beepers in the box.

Fig. F: A Box with a Beeper. Karel is facing north in the northeast corner of the box. Karel must find the beeper.

Figure 1–4 Six Different Tasks for Karel to Perform