

CONTENTS IN DETAIL

ACKNOWLEDGMENTS

xix

INTRODUCTION

xxi

Who Is This Book For?	xxii
About This Book.	xxiii
How to Use This Book.	xxiv
Line Numbers and Indentation	xxiv
Long Code Lines	xxv
Downloading and Installing Python	xxv
Starting IDLE	xxvi
Finding Help Online.	xxvii

1

THE INTERACTIVE SHELL

1

Some Simple Math.	2
Integers and Floating-Point Numbers	2
Expressions	3
Evaluating Expressions	3
Syntax Errors.	4
Storing Values in Variables	5
Summary	8

2

WRITING PROGRAMS

11

String Values	12
String Concatenation	13
Writing Programs in IDLE's File Editor	13
Creating the Hello World Program	14
Saving Your Program.	15
Running Your Program.	16
How the Hello World Program Works	17
Comments for the Programmer	17
Functions: Mini-Programs Inside Programs	18
The End of the Program	19
Naming Variables	20
Summary	20

3

GUESS THE NUMBER

21

Sample Run of Guess the Number	22
Source Code for Guess the Number	22
Importing the random Module	23
Generating Random Numbers with the random.randint() Function	24
Welcoming the Player	26

Flow Control Statements	26
Using Loops to Repeat Code.	26
Grouping with Blocks	27
Looping with for Statements	28
Getting the Player's Guess.	29
Converting Values with the int(), float(), and str() Functions	29
The Boolean Data Type.	31
Comparison Operators	32
Checking for True or False with Conditions.	32
Experimenting with Booleans, Comparison Operators, and Conditions	33
The Difference Between = and ==	34
if Statements	34
Leaving Loops Early with the break Statement	35
Checking Whether the Player Won	35
Checking Whether the Player Lost	35
Summary	36

4

A JOKE-TELLING PROGRAM

39

Sample Run of Jokes.	40
Source Code for Jokes	40
How the Code Works.	41
Escape Characters	41
Single and Double Quotes	42
The print() Function's end Keyword Parameter	43
Summary	44

5

DRAGON REALM

45

How to Play Dragon Realm	45
Sample Run of Dragon Realm	46
Flowchart for Dragon Realm	46
Source Code for Dragon Realm	47
Importing the random and time Modules	48
Functions in Dragon Realm	49
def Statements	49
Calling a Function.	49
Where to Put Function Definitions	50
Multiline Strings	50
How to Loop with while Statements	51
Boolean Operators.	52
The and Operator	52
The or Operator	53
The not Operator	53
Evaluating Boolean Operators	54
Return Values.	55
Global Scope and Local Scope	56
Function Parameters	57

Displaying the Game Results	58
Deciding Which Cave Has the Friendly Dragon	59
The Game Loop	60
Calling the Functions in the Program	60
Asking the Player to Play Again	61
Summary	61

6 USING THE DEBUGGER 63

Types of Bugs	64
The Debugger	65
Starting the Debugger	65
Stepping Through the Program with the Debugger	67
Finding the Bug	70
Setting Breakpoints	73
Using Breakpoints	73
Summary	75

7 DESIGNING HANGMAN WITH FLOWCHARTS 77

How to Play Hangman	78
Sample Run of Hangman	78
ASCII Art	79
Designing a Program with a Flowchart	80
Creating the Flowchart	81
Branching from a Flowchart Box	81
Ending or Restarting the Game	83
Guessing Again	83
Offering Feedback to the Player	85
Summary	86

8 WRITING THE HANGMAN CODE 87

Source Code for Hangman	88
Importing the random Module	91
Constant Variables	91
The Lists Data Type	92
Accessing Items with Indexes	92
List Concatenation	94
The in Operator	94
Calling Methods	94
The reverse() and append() List Methods	95
The split() String Method	95
Getting a Secret Word from the Word List	96
Displaying the Board to the Player	97
The list() and range() Functions	98
List and String Slicing	98
Displaying the Secret Word with Blanks	99

Getting the Player's Guess	101
The lower() and upper() String Methods	101
Leaving the while Loop	103
elif Statements	103
Making Sure the Player Entered a Valid Guess	104
Asking the Player to Play Again	104
Review of the Hangman Functions	105
The Game Loop	105
Calling the displayBoard() Function	106
Letting the Player Enter Their Guess	106
Checking Whether the Letter Is in the Secret Word	106
Checking Whether the Player Won	107
Handling an Incorrect Guess	107
Checking Whether the Player Lost	108
Ending or Resetting the Game.	108
Summary	109

9 EXTENDING HANGMAN 111

Adding More Guesses	112
The Dictionary Data Type	112
Getting the Size of Dictionaries with len()	113
The Difference Between Dictionaries and Lists	113
The keys() and values() Dictionary Methods	114
Using Dictionaries of Words in Hangman	115
Randomly Choosing from a List	115
Deleting Items from Lists	117
Multiple Assignment	118
Printing the Word Category for the Player	119
Summary	120

10 TIC-TAC-TOE 121

Sample Run of Tic-Tac-Toe	122
Source Code for Tic-Tac-Toe	123
Designing the Program	127
Representing the Board as Data	127
Strategizing with the Game AI	128
Importing the random Module	129
Printing the Board on the Screen	129
Letting the Player Choose X or O	130
Deciding Who Goes First	131
Placing a Mark on the Board	131
List References	132
Using List References in makeMove()	135
Checking Whether the Player Won	135
Duplicating the Board Data	137
Checking Whether a Space on the Board Is Free	138
Letting the Player Enter a Move	138

Short-Circuit Evaluation	139
Choosing a Move from a List of Moves	141
The None Value	142
Creating the Computer's AI	142
Checking Whether the Computer Can Win in One Move	143
Checking Whether the Player Can Win in One Move	144
Checking the Corner, Center, and Side Spaces (in That Order)	144
Checking Whether the Board Is Full	145
The Game Loop	145
Choosing the Player's Mark and Who Goes First	146
Running the Player's Turn	146
Running the Computer's Turn	147
Asking the Player to Play Again	148
Summary	148

11 THE BAGELS DEDUCTION GAME 149

Sample Run of Bagels	150
Source Code for Bagels	151
Flowchart for Bagels	152
Importing random and Defining getSecretNum()	153
Shuffling a Unique Set of Digits	154
Changing List Item Order with the random.shuffle() Function	154
Getting the Secret Number from the Shuffled Digits	154
Augmented Assignment Operators	155
Calculating the Clues to Give	156
The sort() List Method	157
The join() String Method	158
Checking Whether a String Has Only Numbers	158
Starting the Game	159
String Interpolation	159
The Game Loop	160
Getting the Player's Guess	161
Getting the Clues for the Player's Guess	161
Checking Whether the Player Won or Lost	161
Asking the Player to Play Again	162
Summary	162

12 THE CARTESIAN COORDINATE SYSTEM 163

Grids and Cartesian Coordinates	164
Negative Numbers	166
The Coordinate System of a Computer Screen	167
Math Tricks	168
Trick 1: A Minus Eats the Plus Sign on Its Left	168
Trick 2: Two Minuses Combine into a Plus	169
Trick 3: Two Numbers Being Added Can Swap Places	169
Absolute Values and the abs() Function	170
Summary	170

13

SONAR TREASURE HUNT

171

Sample Run of Sonar Treasure Hunt	173
Source Code for Sonar Treasure Hunt	175
Designing the Program	180
Importing the random, sys, and math Modules	180
Creating a New Game Board	180
Drawing the Game Board	181
Drawing the X-Coordinates Along the Top of the Board	182
Drawing the Ocean	183
Printing a Row in the Ocean	184
Drawing the X-Coordinates Along the Bottom of the Board	184
Creating the Random Treasure Chests	184
Determining Whether a Move Is Valid	185
Placing a Move on the Board	185
Finding the Closest Treasure Chest	186
Removing Values with the remove() List Method	189
Getting the Player's Move	190
Printing the Game Instructions for the Player	191
The Game Loop	192
Displaying the Game Status for the Player	193
Handling the Player's Move	193
Finding a Sunken Treasure Chest	194
Checking Whether the Player Won	194
Checking Whether the Player Lost	195
Terminating the Program with the sys.exit() Function	195
Summary	196

14

CAESAR CIPHER

197

Cryptography and Encryption	198
How the Caesar Cipher Works	199
Sample Run of Caesar Cipher	200
Source Code for Caesar Cipher	201
Setting the Maximum Key Length	202
Deciding to Encrypt or Decrypt the Message	202
Getting the Message from the Player	203
Getting the Key from the Player	203
Encrypting or Decrypting the Message	203
Finding Passed Strings with the find() String Method	204
Encrypting or Decrypting Each Letter	205
Starting the Program	206
The Brute-Force Technique	206
Adding the Brute-Force Mode	207
Summary	208

15

THE REVERSEGAM GAME

209

How to Play Reversegam	210
Sample Run of Reversegam	213

Source Code for Reversegam	215
Importing Modules and Setting Up Constants.	220
The Game Board Data Structure.	220
Drawing the Board Data Structure on the Screen.	221
Creating a Fresh Board Data Structure.	222
Checking Whether a Move Is Valid	222
Checking Each of the Eight Directions	223
Finding Out Whether There Are Tiles to Flip Over	224
Checking for Valid Coordinates.	225
Getting a List with All Valid Moves	226
Calling the bool() Function	227
Getting the Score of the Game Board.	227
Getting the Player's Tile Choice	228
Determining Who Goes First	228
Placing a Tile on the Board	229
Copying the Board Data Structure	229
Determining Whether a Space Is on a Corner	230
Getting the Player's Move	230
Getting the Computer's Move	232
Strategizing with Corner Moves	232
Getting a List of the Highest-Scoring Moves	233
Printing the Scores to the Screen	234
Starting the Game	234
Checking for a Stalemate.	234
Running the Player's Turn	235
Running the Computer's Turn	236
The Game Loop	237
Asking the Player to Play Again	238
Summary	238

16 REVERSEGAM AI SIMULATION 239

Making the Computer Play Against Itself	240
Sample Run of Simulation 1	240
Source Code for Simulation 1	241
Removing the Player Prompts and Adding a Computer Player.	242
Making the Computer Play Itself Several Times.	243
Sample Run of Simulation 2	243
Source Code for Simulation 2.	244
Keeping Track of Multiple Games	245
Commenting Out print() Function Calls.	245
Using Percentages to Grade the AIs	246
Comparing Different AI Algorithms.	247
Source Code for Simulation 3.	248
How the AIs Work in Simulation 3	249
Comparing the AIs	252
Summary	254

17

CREATING GRAPHICS

255

Installing pygame.	256
Hello World in pygame	256
Sample Run of pygame Hello World	257
Source Code for pygame Hello World	257
Importing the pygame Module	259
Initializing pygame.	259
Setting Up the pygame Window	260
Tuples	260
Surface Objects	261
Setting Up Color Variables	261
Writing Text on the pygame Window.	262
Using Fonts to Style Text.	262
Rendering a Font Object	263
Setting the Text Location with Rect Attributes	264
Filling a Surface Object with a Color	266
pygame's Drawing Functions.	266
Drawing a Polygon	266
Drawing a Line	267
Drawing a Circle	268
Drawing an Ellipse	268
Drawing a Rectangle	268
Coloring Pixels	269
The blit() Method for Surface Objects	270
Drawing the Surface Object to the Screen	270
Events and the Game Loop	270
Getting Event Objects	271
Exiting the Program	271
Summary	272

18

ANIMATING GRAPHICS

273

Sample Run of the Animation Program	274
Source Code for the Animation Program	274
Moving and Bouncing the Boxes	276
Setting Up the Constant Variables	277
Constant Variables for Direction	277
Constant Variables for Color	278
Setting Up the Box Data Structures	278
The Game Loop	279
Handling When the Player Quits.	279
Moving Each Box	280
Bouncing a Box	281
Drawing the Boxes on the Window in Their New Positions.	282
Drawing the Window on the Screen	282
Summary	283

19		285
COLLISION DETECTION		
Sample Run of the Collision Detection Program	286	
Source Code for the Collision Detection Program	287	
Importing the Modules	289	
Using a Clock to Pace the Program	289	
Setting Up the Window and Data Structures	290	
Setting Up Variables to Track Movement	291	
Handling Events	292	
Handling the KEYDOWN Event	293	
Handling the KEYUP Event	294	
Teleporting the Player	295	
Adding New Food Squares	295	
Moving the Player Around the Window	296	
Drawing the Player on the Window	297	
Checking for Collisions	297	
Drawing the Food Squares on the Window	298	
Summary	299	

20		301
USING SOUNDS AND IMAGES		
Adding Images with Sprites	302	
Sound and Image Files	302	
Sample Run of the Sprites and Sounds Program	303	
Source Code for the Sprites and Sounds Program	304	
Setting Up the Window and the Data Structure	306	
Adding a Sprite	307	
Changing the Size of a Sprite	307	
Setting Up the Music and Sounds	307	
Adding Sound Files	308	
Toggling the Sound On and Off	308	
Drawing the Player on the Window	309	
Checking for Collisions	309	
Drawing the Cherries on the Window	310	
Summary	310	

21		311
A DODGER GAME WITH SOUNDS AND IMAGES		
Review of the Basic pygame Data Types	312	
Sample Run of Dodger	313	
Source Code for Dodger	313	
Importing the Modules	317	
Setting Up the Constant Variables	318	
Defining Functions	319	
Ending and Pausing the Game	319	
Keeping Track of Baddie Collisions	320	
Drawing Text to the Window	320	

Initializing pygame and Setting Up the Window	321
Setting Up Font, Sound, and Image Objects	322
Displaying the Start Screen	323
Starting the Game	324
The Game Loop	325
Handling Keyboard Events	325
Handling Mouse Movement	327
Adding New Baddies	327
Moving the Player's Character and the Baddies	328
Implementing the Cheat Codes	329
Removing the Baddies	330
Drawing the Window	330
Drawing the Player's Score	331
Drawing the Player's Character and Baddies	331
Checking for Collisions	332
The Game Over Screen	332
Modifying the Dodger Game	333
Summary	334

INDEX

335