1.	Record Nr.	UNINA9910595029003321
	Autore	Morazan Marco T.
	Titolo	Animated Program Design : Intermediate Program Design Using Video Game Development / / by Marco T. Morazán
	Pubbl/distr/stampa	Cham : , : Springer International Publishing : , : Imprint : Springer, , 2022
	ISBN	9783031043178 9783031043161
	Edizione	[1st ed. 2022.]
	Descrizione fisica	1 online resource (515 pages)
	Collana	Texts in Computer Science, , 1868-095X
	Disciplina	794.81526
	Soggetti	Software engineering Computer programming Video games - Programming Software Engineering Programming Techniques Game Development Disseny de videojocs Programació (Ordinadors) Llibres electrònics
	Lingua di pubblicazione	Inglese
	Formato	Materiale a stampa
	Livello bibliografico	Monografia
	Nota di contenuto	Part I: Basic Problem Solving and Program Design -- 1. The Science of Problem Solving -- 2. The N-Puzzle Problem -- 3. Randomness -- Part II: Generative Recursion -- 4. Introduction to Generative Recursion -- 5. Sorting -- 6. Searching -- 7. N-Puzzle Version 2 -- 8. N-Puzzle Version 3 -- Part III: Accumulative Recursion -- 9. Accumulators -- 10. N-Puzzle Versions 4 and 5 -- 11. Iteration -- 12. N-Puzzle Version 6 -- 13. Continuation-Passing Style -- Part IV: Mutation -- 14. Sharing Values -- 15. Mutation Sequencing -- 16. Vectors -- 17. In-Place Operations -- 18. The Chicken and the Egg Paradox -- Part V: Epilogue -- 19. Where to go from here.
	Sommario/riassunto	This textbook presents a systematic methodology for program development by using design recipes, i.e. a series of steps, each with a specific outcome, that takes a problem solver from a problem

statement to a working and tested programmed solution. It introduces the reader to generative recursion, heuristic searching, accumulative recursion, tail recursion, iteration, mutation, loops, program correctness, and vectors. It uses video game development to make the content fun while at the same time teaching problem-solving techniques. The book is divided into four parts. Part I presents introductory material on basic problem solving and program design. It starts by reviewing the basic steps of a design recipe using structural recursion on a list. It then proceeds to review code refactoring–a common technique used to refine programs when a better or more elegant way is found to solve a problem–and introduces the reader to randomness. Next, Part II explores a new type of recursion called generative recursion. It navigates the reader through examples involving fractal image generation, efficient sorting, and efficient searching techniques such as binary, depth-first, and breadth-first search. Part III then explores a new type of recursion called accumulative (or accumulator) recursion. Examples used include finding a path in a graph, improving insertion sorting, and list-folding operations. Finally, Part IV explores mutation. To aid the reader in properly sequencing mutations it presents Hoare Logic and program correctness. In addition, it introduces vectors, vector processing, in-place operations, and circular data. Throughout the whole book complexity analysis and empirical experimentation is used to evaluate solutions. This textbook targets undergraduates at all levels as well as graduate students wishing to learn about program design. It details advanced types of recursion, a disciplined approach to the use of mutation, and illustrates the design process by developing a video game exploiting iterative refinement.