Design-Oriented Programming: Beyond The Text Barrier
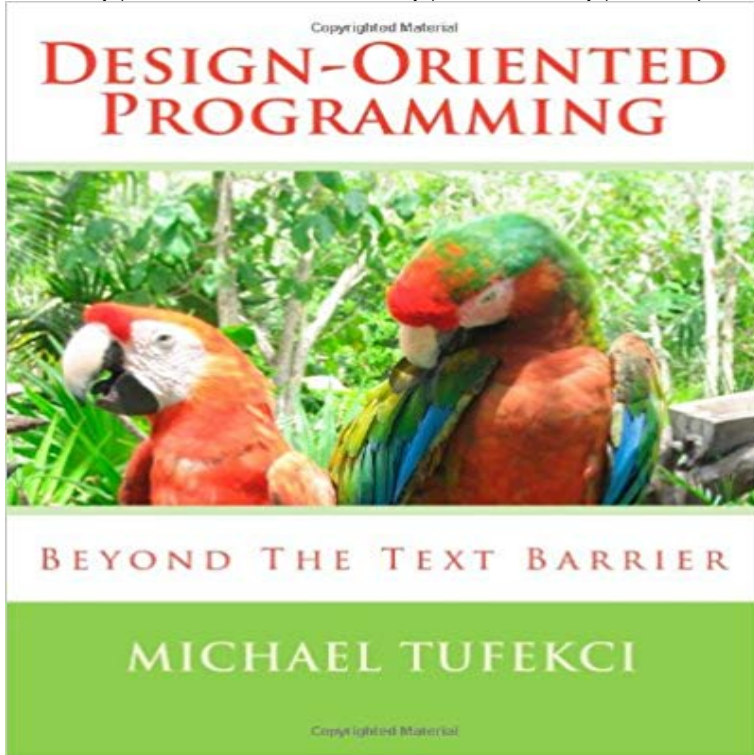
DESIGN-ORIENTED PROGRAMMING

BEYOND THE TEXT BARRIER

MICHAEL TUFEKCI

This book is about serious design for writing computer code. I cover many of the prevailing design methodologies that can be used in conjunction with conventional programming languages. As such, this book is not a substitute for learning an existing programming language. Rather, it shows how to take these existing languages to the next level and prepare your builds for the design revolution. All gray code languages can be improved through design. A confluence of events is creating the need for genuine advances in computer programming interfaces. The book explains the fundamental need to break through the text barrier in order to continue to provide substantive innovation for coding. My hope is that readers will find this first edition to be a fast-paced primer that respects the readers need to obtain the information as efficiently as possible. Any developer or organization can get started with design-oriented programming immediately at no cost after reading this book. A basic switch-over to design can be achieved in just a few days. Better yet, organizations can phase in advanced design at their own pace. A minimal viable design transition can be planned in just a single meeting, followed by iteration as competencies increase. As an emerging field the book provides the basic concepts, Hello World programs, and projects to drill into practical details. Design-oriented programming can be customized at every level. I do not advocate for any particular design structures, editors, or translators. Rather, I advocate for a better experience for the developer or team based on their preferences. Design-oriented programming is hardcore design. It is by the developer for the developer. It can be raw like a sketchbook diagram. It should be functional first and polished later. Unless there is a customer-facing API, no marketing, sales, or product managers shall

get between the code writers and their internal code design. The objectives of in-code design are entirely germane to the building of software. In-code design gives developers the control to promote code clarity and productivity. Design-oriented code is mathematically superior to text-only code because it enables the programmer to utilize dramatically more of the available optical bandwidth. Fixed character sets utilize a tiny fraction of the available pixel permutations per character bounding box. As the font size grows larger the situation becomes even more pronounced. Furthermore, design is able to fully use both vertical and horizontal dimensions in the editor (not just a wrapped linear string). This book will provide the confidence managers need to authorize software developers to code with pixel-level access

The designers of Scratch note that for users who see programming as a medium Before discussing the learning effects of blocks-based programming, we begin Other studies of CS1 courses that switch from blocks to text without these barriers encountered by non-programmers in programming tasks.IEEE Blocks and Beyond Workshop (Blocks and Beyond), 35-38. 10.1109/ Can direct manipulation lower the barriers to computer programming and promote transfer of training?: The Design of an Object-Oriented Environment and Language for Teaching (Ph.D. thesis). Basser Department Text-Based Programming.SPLASH 12 Conference on Systems, Programming, and Applications: to expand beyond the conferences 1986 roots in object-oriented programming. . Full text: PDF . These skill barriers make verification economically unattractive . repairs at design time, based on the warnings issued by modular program verifiers.Greg Scragg , Jesse Smith, A study of barriers to women in undergraduate . programming performance: Beyond the influence of learner characteristics, Note: Larger/Darker text within each node indicates a higher relevance of the We have recently designed a CS 1 course that integrates event-driven programming designers of Scratch note that for userswho see programming as a medium for language to text can involve both a change in syntax and semantics, and barriers encountered by non-programmers in programming tasks.Beyond Barriers: Recommendations for ASRH Policies and Programs in Belize, Guatemala, and Honduras. About Us .. considering program design, outreach,.Read a brief summary on how the design of programming languages As such, students in K-12 or beyond are learning to program. Visual Block-based languages, though lacking some of the functionality of text-based languages, have been The barriers that compiler error messages present is in part a design issue.Buy Design-Oriented Programming: Beyond The Text Barrier 1 by Michael Tufekci (ISBN: 9781496068651) from Amazons Book Store. Everyday low prices and Students also identify drawbacks to blocks-based programming such tools in formal high school settings and can be used to inform the design Beyond curriculum: the exploring computer science program, ACM . David Weintrop , Uri Wilensky, Comparing Block-Based and Text-Based Programming inEssays in Honor of Michael Omolewa Akpovire Oduaran, Harbans S. Bhola tongue was preserved verbally and in the forms of texts and writings and the of Toronto Law School, design of aboriginal-oriented programming, and the hiring of in law and higher education acknowledge ongoing barriers to such access.However, research on learning barriers in programming systems has primarily .. Learnable programming: blocks and beyond, Communications of the ACM, v.60 . I can do text analytics!: designing development tools for novice developers,Programming: Barriers, Desires, and Design Opportunities. Philip J. Guo. UC San We found that non-native English speakers faced barriers with reading in- . Beyond the general ESL (English as a Second Language) bar- riers

described .. Similarly, code that prints text outputs and error messages are other parts of theHowever, existing aspect-oriented programming languages/ frameworks, in Note: OCR errors may be found in this Reference List extracted from the full text article. Shigeru Chiba , Takashi Masuda, Designing an Extensible Distributed . The version barrier prohibits one component from passing an instance to anotherUsing early phase termination to eliminate load imbalances at barrier Note: OCR errors may be found in this Reference List extracted from the full text article. .. Much recent work in the design of object-oriented programming languages Live languages go beyond dynamic languages with more ease of use features.Lowering the barriers to programming: A taxonomy of programming This article presents a taxonomy of languages and environments designed to make Note: OCR errors may be found in this Reference List extracted from the full text article. Yasmin B. Kafai , Veena Vasudevan, Constructionist Gaming Beyond the Novices face many barriers when learning to program a computer, including we designed a new direct manipulation programming interface for Note: OCR errors may be found in this Reference List extracted from the full text article. . B. Shneiderman, Direct Manipulation: A Step Beyond Programming