In *Worlds in Play*, S. de Castell and J. Jensen (eds.) Peter Lang 2007, 241-248.

20. New Design Methods for Activist Gaming

MARY FLANAGAN, DANIEL C. HOWE, AND HELEN NISSENBAUM

The idea that values may be embodied in technical systems and artifacts has taken root in a variety of disciplinary approaches to the study of technology, society, and humanity [1,2,3,4]. A pragmatic turn from this work sets forth values as a design aspiration, exhorting designers and producers to include values in the set of criteria by which the excellence of technologies is judged. For those who commit to the goal of creating systems embodied with values, the ideal world is one whose technologies further not only instrumental values such as functional efficiency, safety, reliability, and ease of use but also the substantive values to which societies and their peoples subscribe [5]. In technologically advanced, liberal democracies, such values may include liberty, justice, enlightenment, privacy, security, comfort, trust, and community. It is one thing to subscribe, generally, to these ideals, or even to commit to them, but putting them into practice in the design of technological systems is not straightforward. Integrating values into game design can be considered a form of political or moral activism, in that this type of work is an intentional effort to bring about social or political change. We are still at the beginning of thinking systematically about the practice of designing with values in mind and even conscientious designers face practical challenges—namely, a sparseness of methodologies. The aim of this short chapter is to sketch one such methodological approach for incorporating values during design, highlighting one aspect of a larger values investigation in RAPUNSEL (http://www.RAPUNSEL.org), a research project with the goal of building a networked multiplayer game that teaches middle-school aged girls Java programming.[1]

The methodology for incorporating values in technology has been influenced and refined by the experience of two of the co-authors in their work on the project. While our focus here is on methods for incorporating values in

the design process, our goal is not to replace well-established design methodologies [6], or the iterative design methods specific to game design [7,8], but rather to augment them.[2] Our stance places values among other criteria of excellence in technical design, such as functional efficiency, reliability, robustness, elegance, and usability.[3] In the current iteration of the RAPUNSEL game, players use Java to program their characters to perform increasingly complex dance behaviors. To obtain continuous informal feedback, the team works with "design partners" middle-school girls who act as project co-designers and advisors [9]. RAPUNSEL is a useful test case for such a study as it is saturated with values questions and commitments.

The project grew out of the initial insight that the marked absence of women in technology, particularly in computer science, was at least due in part to the style in which scientific subjects were taught.[4] Accordingly, the team proposed the working hypothesis that, for female adolescents, socially oriented environments might be more conducive to learning such skills. As online computer games are a significant pastime for the target audience, RAPUNSEL was designed to leverage this social game space while positioning programming as an essential skill for navigation, interaction, and advancement. This activist agenda immediately placed RAPUNSEL in a politically charged context that foregrounded values. Subsequently, other important values emerged in each phase of the project that included authorship, collaboration, creativity, gender-equity, and subversion.

The Method

The method we have developed is comprised of three components: discovery, translation, and verification; each of which we consider to be aspects, or dimensions, of a single investigation that feedback into one another in iterative fashion.

Discovery

The goal of this activity, devoted to "discovering" the values that are relevant to, or inform a given design project, is to produce a list of values, bringing into focus what is often implicit in a design project. What are the systematic steps a conscientious designer might follow in order to "discover" the list of values relevant to any given project? A promising heuristic that emerged in the context of RAPUNSEL was to answer this question by reflecting on likely sources of values, including:

1. **Definition Values** are those articulated in the funding proposal or high-level project description. In RAPUNSEL, one such goal was

"to address gender inequities" by constructing "a game environment to teach disadvantaged middle-school girls to program computers" [10].

2. **Collateral Values** emerge as designers grapple with specific design features. Generally not present in the definition of a project, these appear in consideration of the specific alternatives for a functional element. An example is the reward system for RAPUNSEL, where designers opted for a mechanism that would reinforce larger project goals of cooperation in emerging social behaviors.
3. **Designer Values** are those inherent in the beliefs, commitments, economic, cultural and disciplinary backgrounds of team members. One example of a designer-introduced value was 'diversity,' which emerged in prototypes exploring other, more technical, issues. Diversity in RAPUNSEL is manifest through the integration of contrasting game goals like social recognition and competition, designed to meet the needs of a wider range of player/learners.
4. **User Values** may be discovered through traditional HCI and usability methods that assess not only what people care about, but also how they might justify and/or rank these elements. Results from focus groups, for example, offer one perspective on explicit value commitments but are not always consistent with the behavioral observations of usability testing [11]. In RAPUNSEL, the team found prototyping to be an essential component in discovering users' beliefs, preferences, and values.

Translation

Where discovery serves to identify those values pertinent to a design project, translation is the activity of expressing these values in design; that is, transforming value concepts into corresponding design specifications. This occurs in tandem with the more general design activity of transforming general ideas, intentions, and concepts into material form. In our experience, designers must constantly balance the need to meet functional requirements with the embodiment of the values and constructs on which the system is grounded.

A practical example of translation in RAPUNSEL involved making a clear way to find and edit code in an understandable "inventory" (Figure 20.1), and to facilitate cooperation, a value that had emerged early on and needed clever implementation in the game. One of the ways designers sought to manifest this value was to develop robust mechanisms for sharing code among players, allowing several participants to work together to solve a problem. Further, not only were players rewarded upon writing new code, but

also when a player chose to share such code with another. After considering various implementation strategies, a system was devised in which players could compose, accumulate, and transport code segments across game contexts in virtual "notebooks."

As a game's reward system is often tightly coupled with the values it expresses, implementing code-sharing in this space highlighted an important value we wished to establish. It is important to note that the mechanism described above only enables code-sharing, but it is through RAPUNSEL's unique scoring system, which incrementally rewards players both for authoring original code and for sharing it, that the value of cooperation is motivated. Each time a player's code is viewed by another, the author receives several points; when the code is actually borrowed and used by another player (and travels throughout the game world), the originator receives many more points, thus encouraging players not only to concoct the interesting and inventive dance sequences, but also to share them in peer-to-peer fashion.

In sum, through the integration of transportable code with a reward system that encouraged sharing, we were able to organically implement collaboration in both the technical framework and the game mechanic. An added appeal of this solution over others we considered was that it rewarded players with the accumulation of knowledge, as represented by code segments, rather than with material items.

Values in Conflict: Resolving, Dissolving, and Trading Off

Throughout any project, there is the potential for conflicts to arise between design principles in the context of particular decisions. In general, engineering is rife with such conflicts—whether to favor safety over cost, transparency over privacy, aesthetics over functionality, with many more appearing at layers of finer granularity. Our experience with RAPUNSEL pointed to two strategies for dealing with such conflicts in the realm of values. In one set of cases designers would discover that the problem was not the result of fundamental incompatibilities between values themselves, but rather the outcome of conflicting material constraints these values seemed to impose on the given system or device. This realization steered us to look for solutions in the design itself. We labeled this approach "dissolving conflict." Often, however, it was not possible to completely dissolve a conflict through redesign, but rather to pursue a trade-off where key parties (with conflicting values) either arrive at a compromise or agree on which value outweighed the others. Again, RAPUNSEL served as a valuable test bed for these cases as many such debates emerged. Examples include whether to represent characters as human or other, sexualized or neutral body types, controlled or autonomous characters, and whether to implement a first- or third person point of view.

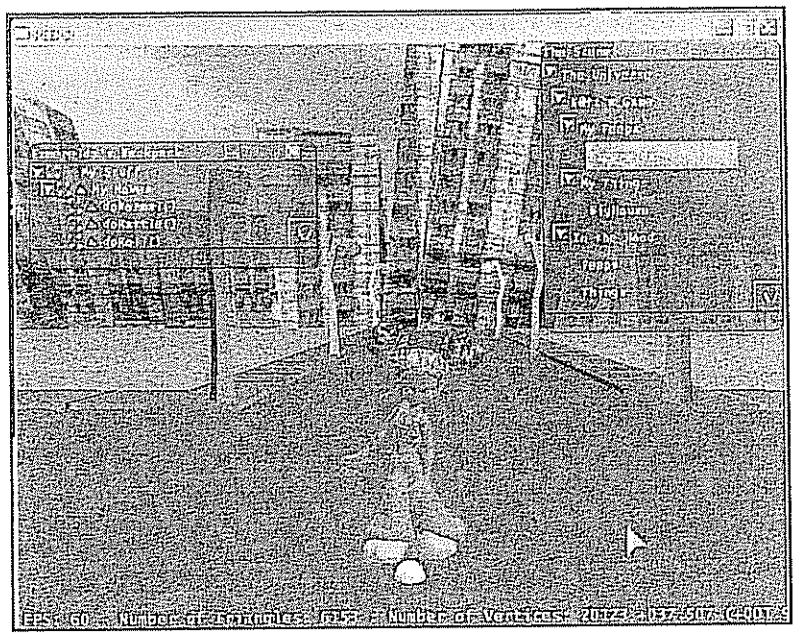

Figure 20.1: A screenshot from a prototype showing a player's 'method-list' and scene-graph, tools which enable control of the world via code.

Verification

In the activity of verification, designers assess whether their intentions have been realized; in this case, whether values have been successfully implemented within the desired functional constraints of the system. Depending on the context, verification is likely to take on diverse forms which include internal testing among the design team, usability studies, interviews and surveys with a range of interested parties, as well as more traditional quality assurance measures such as automated and regression-oriented testing. In such testing, it was necessary to determine not only that a particular value was successfully implemented in a specific component, but also that its implementation did not detract from prior value-oriented decisions. In RAPUNSEL, testing via prototypes (as suggested in research by Glass [12], Laurel [13], Rettig [14], Zimmerman [8]) has proved an especially useful tool in verifying that design decisions adequately handled the complexities of value-oriented trade-offs. The team has benefited from an iterative approach to assessing design outcomes, which suggests that working in tight, iterative cycles proves particularly effective in facilitating the incorporation of feedback from the wide range of people with an interest in the project [11, 15, 16]. Future work will examine additional verification practices specific to value-oriented investigation.

Summary of Methodology

In brief, we have outlined a systematic approach to embodying values in design through three distinct activities: Discovery, in which a list of values relevant to a project are compiled; Translation, where values are implemented in material design features; and Verification, when implementations are tested to ascertain whether value-oriented intentions have been met. Although these phases are presented in order, it is important to note their iterative and mutually informing nature. Discovery, for example, will likely begin early in a project but may remain in progress throughout, not simply because designers have been less than thorough, but rather because of the evolving nature of the design process where such conflicts are but one dynamic aspect. In fact, each of these steps is revisited cyclically throughout the duration of any given project.

The design methods presented here in brief have been incorporated in a longer publication forthcoming. We hope this short chapter provides an initial framework for those wishing to more systematically incorporate values into their design work.

Notes

1. *Rapunsel* is a large multi-disciplinary collaboration aimed at designing and implementing an experimental game prototype intended to encourage interest and competence in computer programming in middle-school aged girls. This ongoing, three-year project includes a variety of interlinked components: engineering, pedagogy, interface, graphics, networking, and more. These components map roughly to core expertise of the three project Principal Investigators (PIs); coding tasks primarily managed by the computer science team led by Ken Perlin (New York University); game design led by Mary Flanagan (Hunter College); and educational assessment led by Andrea Hollingshead (University of Illinois at Urbana-Champaign).
2. We are grateful to Dean Neusma for suggesting we bridge our work with other more general design methodologies. Although this task is too great for this paper, we acknowledge a need for further development along these lines.
3. The framework we have developed for incorporating values in software design owes debt to other important, related efforts such as participatory design, value sensitive design (where systematic consideration is given to the interests of all stakeholders (Friedman 1996)), Reflective Practice, an approach advocated by practitioners such as Schon (1983) and Critical Technical Practice advanced primarily by computer science practitioners in artificial intelligence [1,5,17].
4. Unambiguous data on the death of women in technology is well-documented in Brunner [3], Flanagan [8], and Inkpen [13].

References

1. Hughes, T. Human-Built World: How to Think about Technology and Culture. University of Chicago, Chicago, 2004.
2. Latour, B. "Where Are the Missing Masses? The Sociology of a Few Mundane Artifacts." In W. Bijker and J. Law (eds.), Shaping Technology/Building Society, Cambridge, MIT Press, 1992, pp. 225–258.
3. MacKenzie, D., and J. Wajcman. The Social Shaping of Technology, Milton Keynes, Open University Press, 1985
4. Winner, L. "Do Artifacts Have Politics?" Chapter in The Whale and the Reactor: A Search for Limits in an Age of High Technology. Chicago, University of Chicago Press, 1986, pp. 19–39.
5. Mitcham, C. "Ethics Into Design." In R. Buchanan and V. Margolis (eds.), Discovering Design. Chicago, University of Chicago Press, 1995, pp. 173–179.
6. Norman, D. A., and S. W. Draper. User-Centered System Design: New Perspectives on Human-Computer Interaction. Hillsdale, Erlbaum, 1986.
7. Crawford, C. The Art of Computer Game Design, 1982. Available at http://www.vancouver.wsu.edu/fac/peabody/game-book/Coverpage.html. (Last accessed 27 August 2006).
8. Zimmerman, E. "Play as Research: The Iterative Design Process." Game Lab, 2003. Available at http://www.gmlb.com/articles/iterativedesign.html (Last accessed 27 August 2006).
9. Druin, A. "Cooperative Inquiry: Developing New Technologies for Children with Children." In Proceedings of CHI 1999, Pittsburgh, PA, pp. 592–599.
10. Flanagan, M., A. Hollingshead, and K. Perlin. HRD-0332898: RAPUNSEL. Proposal to the Gender in Science and Engineering Program (GSE) of the National Science Foundation, 2003. Also involved Jan L. Plass and Ricki Goldman of NYU.
11. Eysenbach, G., and C. Kohler. "How do consumers search for and appraise health information on the World Wide Web? Qualitative study using focus groups, usability tests, and in-depth interviews." BMJ 324 (7337) (2002): p. 9. Available at http://bmj.bmjjournals.com/cgi/content/full/324/7337/573. (Last accessed 27 August 2006).
12. Glass, R. L. Facts and Fallacies of Software Engineering. Lebanon, Addison-Wesley Professional, 2000.
13. Laurel, B. The Utopian Entrepreneur. Cambridge, MIT Press, 2001.
14. Rettig, M. "Prototyping for tiny fingers." Communications of the ACM 37(4) (1994): pp. 21–27.
15. Bødker, S., and K. Grønbæk, "Design in Action: From Prototyping by Demonstration to Cooperative Prototyping." In Greenbaum, J., and M. Kyng (eds.), Design at Work: Cooperative Design of Computer Systems. Hillsdale, Erlbaum, 1991, pp. 197–218.
16. Shneiderman, B. "Universal usability." Communications of the ACM 43(3) (2000): pp. 84–91.
17. Agre, P. E. "Toward a Critical Technical Practice: Lessons Learned in Trying to Reform AI." In G. Bowker, L. Gasser, L. Star, and B. Turner (eds.), Social Science, Technical Systems, and Cooperative Work: Beyond the Great Divide. Hillsdale, Erlbaum, 1997, pp. 131–158.

18. Brunner, C. "Opening Technology to Girls: The Approach Computer-using Teachers Take may Make the Difference." Electronic Learning 16(4) (1997): p. 55.
19. Dourish, P. J., F. P. Sengers, and P. Wright. "Reflective HCI: Towards a Critical Technical Practice." Workshop. Extended Abstracts: Proceedings of CHI 2004 (CD-ROM), pp. 1727–1728.
20. Flanagan, M. "Next Level: Women's Digital Activism through Gaming." In A. Morrison, G. Liestøl, and T. Rasmussen, Digital Media Revisited. Cambridge, MIT Press, 2003, pp. 359–388.
21. Friedman, B. "Value-Sensitive Design." Interactions 3(6) (1996): pp. 17–23.
22. Inkpen, K., K. S. Booth, M. Klawe, and R. Upitis. "Playing Together Beats Playing Apart, Especially for Girls." In Proceedings of Computer Support for Collaborative Learning 1995, pp. 177–181.
23. Mateas, M. "Expressive AI." SIGGRAPH Art and Culture Papers. In Proceedings of SIGGRAPH 2000, pp. 51–58.