WHAT MATTERS: APRIL 2009

Apply Game Design as a Science for Public Policy to Rescue Economy, Planet

Screen shot of the action role-playing game *Titan Quest*Ã‚Â© (THQ Inc., 2006) that McGuire helped develop.

What do the Obama vs. McCain presidential election, the Sox playing at Fenway, government spending bills, and the video game *LittleBigPlanet* have in common? They are all subject to analysis as games: scenarios where intelligent agents (players) seek to maximize their payoff (win) under a set of rules. Although some seem like fun and others like work, an understanding of each instance informs the others, and insights for any improve how we both work and play.

Games are about decisions, and decisions matter. Computer science, mathematics, psychology, economics, and political science have explored decisions in formal games for several decades. Their classic problem is to find the best strategy under a set of rules. A new, more important problem reverses this: design a rule system that drives players to desirable behaviors. In the real world, we want the rules in our tax code and laws to be fair and encourage strategies that benefit society as well as the individuals. In virtual worlds, choices must also entertain the agents. Sometimes the line between these is blurry: Ebay's auction rules intentionally trade market efficiency with the thrill of last-minute deals. Swoopo's "entertainment shopping" takes this one step further.

Rules are designed by lawyers, politicians, and game designers. Science and engineering can help improve their process. This is because rules are mathematical and mechanical. In fact, machines, mathematics, and rule systems are equivalent-computer programs demonstrate this [1]. Any design field benefits from rigorous analysis techniques and a solid theoretical base. Unfortunately, most rule systems are not designed or tested in any engineering sense. Instead, they are crafted by gurus using black-box intuition and defended with subjective arguments. We see bad policies every day, such as Boston closing its public transportation 90 minutes before its bars and Citizens Bank requiring a cash withdrawl and

deposit to move money between account holders. These show that even simple rules can easily go awry. Informal design of a complex and critical system, such as a tax code, can be disastrous.

For example, a 2005 U.S. transportation law gives tax credits for combining alternative fuel with diesel. Paper mills are some of the agents subject to the U.S. tax code. Their efficient, 80-year-old kraft process for making paper uses no diesel because it draws energy from pulp waste. But after the tax rules changed, mills immediately started burning unnecessary fossil fuel to receive tax credits. Calling this "exploiting a loophole" sways emotions against the mills. However, the objective fact is that a rule design error is encouraging pollution and may cost taxpayers up to $8 billion dollars this year [2]. Furthermore, we've recently seen failures in SEC regulations and bailout rule systems that cost the U.S. a decade of economic growth, with political scapegoats everywhere. Blame is missing the point. Architects and mechanical engineers don't blame wind or friction for exploiting their design errors. They fix their designs and the processes behind them. Rule makers should, too.

So, how can we improve rule design? And how can we become better players within the rules that others impose?

Classical research on game theory assumes clear choices and objective goals. The methods developed can motivate tactics like price discrimination, proscribe efficient auction strategies, and solve some board games perfectly. Unfortunately, these methods are limited; one of the largest successes was solving checkers (it is a draw if both players are perfect) [3]. That was a major contribution, but it won't get us to modern video games or real-world scenarios.

Perhaps solutions to design and play lie in more rigorous exploration of entertainment games. In the last ten years, board games have exploded in complexity and popularity. Recent titles like *Dominion*and *Agricola*are orders of magnitude beyond Monopoly or chess, to the point where precise analysis is probably intractable. That is, they approach the complexity of real-world situations. Designing one is comparable to designing a car engine or a large set of company bylaws. Playing such a game trains you to recognize and exploit patterns in complex systems; you learn most when playing a game that you're bad at, and should perhaps focus on those to train for serious analysis tasks at work [4].

Modern video games augment their rules with significant technology (i.e. more rules) for 3D graphics, physics, and networking. These are arguably the most complex systems engineered in any discipline, ever. As a coarse comparison of complexity, the entire U.S. federal tax code is about 14k of printed pages, and a video game contains about three times as much source code, plus extensive data. Both contain errors, yet most games seem to avoid serious loopholes. Furthermore, the best game designers seem able to create stable rule structures and balance them for fairness-properties that the SEC and tax code evidently lack.

Bringing rigorous analysis to how their designers create, we see some principles emerge. Game designers work with recursively nested systems, each balanced separately. There are tens of established patterns for these systems that provide a basic framework. Many of these ideas will resound with engineers across disciplines. Games also veer away from classic engineering in some ways. Designers consider a subjective notion of engagement and weigh it against the mechanical nature of their systems. They accept that players have a meta-choice of not to play the game at all, and adjust their systems to present actively attractive options rather than least-negative ones. Recall that these are products that ask a group of people to learn complex rules and solve hard problems in multihour sessions, and yet which the customers love and voluntarily seek. There are many places outside games that would benefit from this success, from education to product design.

Unlike most other synthesis disciplines, games lack a clearly articulated design theory. To leverage the insights of game development for solving real-world game-like problems and complex engineering tasks, much more work is needed. The first step is to cast the still largely undocumented and ad hoc methods of entertainment game designers into a rigorous theory. We can then reconcile this theory with that of other design fields, benefiting from the complementary pieces. This will reduce some of the design risk that plagues the games industry and export its abilities to stabilize massively complex systems and engage its players.

I invite you to explore these ideas and some specific design strategies with my coauthor Chad Jenkins and me in our **new book, *Creating Games*, and online (http://graphics.cs.williams.edu/creatinggames)** . The text explains the design, art, and technology of entertainment games. It relies heavily on computer science and economics for rule design but touches disciplines from management science to graphic arts for describing the other aspects of games. The centerpiece is a discussion of how rules interact with the broader problems discussed in this article. The Web site also contains electronic versions of worksheets and presentations for hobbyists and students interested in designing entertainment games and my blog, where I weekly highlight new articles and research developments that I found most relevant to computer graphics, games, and rule design.

Notes

[1]The formal **Church-Turing Thesis (http://en.wikipedia.org/wiki/Church%E2%80%93Turing_thesis)** statement of this correspondence is the foundational idea of computer science.

[2]Hayes, " **Pulp Nonfiction (http://www.thenation.com/doc/20090420/hayes)** ", *The Nation*, Apr 2, 2009

[3]See **http://www.cs.ualberta.ca/~jonathan/OJA/oja.html (http://www.cs.ualberta.ca/%7Ejonathan/OJA/oja.html)** .

[4]Koster, *A Theory of Fun for Game Design*, Paraglyph Press, 2004

About the Author

Morgan McGuire is an Assistant Professor of Computer Science at Williams College. He is the coauthor of the new book, ***Creating Games: Mechanics, Content, and Technology*** **(http://graphics.cs.williams.edu/creatinggames)** and many scientific publications on computer graphics and games. He has contributed to commercial games including *Titan Quest*, *Marvel Ultimate Alliance 2: Fusion*, *ROBLOX*, and *Nimby*. He earned his BS and MEng degrees at MIT in Course VI, and MS and PhD degrees at Brown University.

What Matters is a guest opinion column written by a different MIT alumnus or alumna. The views expressed are entirely those of the author and do not necessarily represent the views of the Alumni Association or MIT. Interested in writing a column? Email **whatmatters@mit.edu**.