

# Making Design Patterns Work

Joris Dormans  
Amsterdam University of Applied Sciences  
Duivendrechtsekade 36-38  
Amsterdam, The Netherlands  
j.dormans@hva.nl

## ABSTRACT

The game industry seems to be reluctant to use design patterns. This article explores the obstacles encountered in the application of many design pattern libraries. Rather than instructing the industry the proper use of design patterns, it aims identify the things academics can do to improve the practical value of design pattern libraries. It argues that design pattern libraries should depart from a clearly defined theoretical core that creates an informative lens on a particular aspect of game design. Patterns should prescribe generic solutions to common problems. Finally, in order to create libraries with a large expressive range, the number of patterns is of less importance than the number of interactions between the patterns.

## Categories and Subject Descriptors

K.8.0 [Personal Computing]: General—*Games*

## 1. INTRODUCTION

There is a problem with design patterns for games: outside the hallowed halls of academia game designers seem to be reluctant to use them. The gap between game industry and game research is still large, especially where game design is concerned. Since Doug Church's famous "formal abstract design tools" article [5] game designers such as Raph Koster [17], Daniel Cook [6], Stéphane Bura [3] and Noah Falstein [10] actively pursue similar goals to academics such as Staffan Björk, Jusi Holopainen [2], Stefan M. Grünvogel [14] and José Zagal [24]. However, their efforts are not always appreciated by the game industry at large. Raph Koster's talk on game grammar at the Game Developers Conference split the audience in half: some liked it, others really hated it [22]. Other critics voiced their opinions on personal blog posts and Gamasutra articles [11, 15]. More recent Gamasutra articles point towards a similar gap emerging between academics and indie developers [19, 16].

As an enthusiastic academic contributor to the formal lan-

guage and design patterns debate [8], I have a personal stake in advancing the theory of games. At the same time, as a professional game designer I do acknowledge some of the concerns expressed by industry and indie critics. After all, it is also my experience that the current results of game studies often seem to be of limited applicability for actual design. Simply developing theory and sending it out into the world does little to help transfer knowledge from academia to industry. As an academic I feel a strong responsibility to make sure that (at least some of) these theories are applicable. In this paper I will investigate some of the academic assumptions underlying many of the current academic design patterns projects that in my experience feed the current gap between theory and practice. It aims to provoke discussion and inspire critical introspection on the academic side. As a result it might seem somewhat opinionated. However, if we are ever going to bridge this gap, we better start building somewhere, there is little point in waiting for the party on the other side to make the first move.

## 2. 'DESIGN PATTERN', ISN'T THAT JUST ANOTHER WORD FOR TAXONOMY?

Christopher Alexander *et al.*'s original, groundbreaking work on design patterns in architecture and urban planning, frames design patterns as generic solutions to common problems [1]. In a similar vein, when the Gang of Four transferred design patterns to the domain of software development they also presented them as problem-solution pairs [12]. In addition, software design patterns depart from the theoretical notion that object-orientation is a good way to create code that is maintainable and reusable. Software design patterns prescribe how programmers best structure their code; they are inherently prescriptive. For Alexander designer patterns where a method to approach the unnamed quality that sets great architecture apart from the rest. Both design pattern libraries depart from the premise that quality in the discipline they describe can be objectified, and that by applying the patterns in the libraries, software or architecture designs can be made better.

Many game design patterns are different: instead of prescriptive patterns there is a tendency to create patterns that are "semi-formal interdependent descriptions of commonly reoccurring parts of the design of a game that concerns gameplay" [2]. The goal of these type of patterns is often to provide game designers with a "common vocabulary" and "a basic framework should be set for designers to collaborate and work off of in order to create better experiences

for the player” [21]. This puts game design patterns more inline with attempts to build elaborate design vocabularies and taxonomies for games [10, 18, 24] than they are with the design patterns of Christopher Alexander or the The Gang of Four. As I shall argue throughout this paper, design vocabularies have less practical value to designers than design patterns that offer generic, prescriptive solutions to common design problems have. In other words, the shift from design patterns to design vocabularies is unfortunate and only serves widen the gap between academia and design.

It must be said that not all design patterns proposed for games are vocabularies. Some pattern libraries do follow the problem-solution structure. However, in many of these cases the patterns are constraint to a specific game [7], a very narrowly defined aspect of games [20]. One must wonder how generic the solutions are, and for that matter, how common the identified problems are.

### 3. SO HOW DOES A TAXONOMY HELP EXACTLY?

As an academic exercise there is nothing wrong with building taxonomies. However the application of game taxonomies for actual game design remains sketchy at best. We typically supply two arguments why taxonomies should matter to game designers:

1. Taxonomies are needed to properly define games and design concepts. That way they form the corner stones of a common design language, which will advance our knowledge of games and will enable designers to pursue more advanced game concepts with greater accuracy.
2. Taxonomies can be used to explore the design space. Either by identifying relatively unexplored areas, or by randomly combining different elements from the taxonomy to generate new ideas and game concepts.

Both arguments are valid points, however they also seem to miss the mark. Game developers already have a large and specialized vocabulary to talk about their trade. Their design language might not be universal, but they have little trouble understanding each other. They all know what is meant by a "Diablo-style inventory system", a "technology tree", or gameplay that is carried by "resource harvesting". Designer do not need academics to run around and put labels on these things for them. They are fully capable of doing that themselves as is clearly evidenced by the jargon already present. In fact, it is very likely in this area academics are behind on the designers. Academics study published games to find new concepts to label, while designers work on tomorrow's games right now. It is not strange that they feel that the industry is leading academia in this particular area and not the other way around.

Furthermore, designers do not really need help generating ideas. Most game designers will already have more game ideas on hold than he or she is ever going to be able to build. Those ideas might not be as spectacular or unique as the ideas generated more methodologically by using sophisticated imaginary variation techniques. However, it is not

just the designer that determines the direction of a game. Publishers often have a say in how their money is being invested, and they tend to be risk averse. If industry is going push back the frontier of the known game design space, it is going to do so taking small steps only. Even indie developers are running businesses. They take more risks, but they also need people to play and pay for their games, so even they cannot deviate too far from the norm.

What is more, most designers practically live for the initial generation ideas at the start of a new project. During that brief period in the long development cycle they enjoy the most artistic freedom. By suggesting that design patterns might be used to generate ideas, we take away something that most designers see as one of the most defining and creative tasks of their profession. It is no wonder that most designers feel that design patterns steal away the creative soul of their craft. Academic methods are directly competing with what designers feel to be the juiciest bit of game design.

If game design patterns are to be successful, it is probably wise to invest in applications of design patterns other than analysis and idea generation. Translating an initial idea into a working game is a hard task, and designers spend most of their time trying to make their designs work, fixing unexpected problems in their designs. In theory, design patterns are a good method to help them find solutions for challenges that arise during development, especially if they really describe generic solutions to common problems. For design patterns it is also important to focus on making hard things easier (instead of making easy things easier as many patterns are perceived to be doing by the games industry) and to favor utility over creativity. Only this way the design community will perceive design patterns as a valuable tool instead of a methodological competitor for their creative talent.

In order to do so, a design pattern library needs to have firm theoretical foundation: it needs to be able to describe quality in games in such a way that designers are able to replicate proven structures in order to make their games better. Sadly, most design patterns libraries lack a clear formulation of what constitutes quality in games; they typically lack an equivalent notion to the notion of object-orientation in software design patterns. In other words, in my view, design patterns should be a little more prescriptive.

### 4. WHAT IS THE PROBLEM OF BEING PRESCRIPTIVE ANYWAY?

Many game design patterns aim to be descriptive rather than being prescriptive. Even most game designers that advance the theory of game design, frequently fall into this trap [9]. I suspect that the reason for this is that many researchers active in game design patterns have a background in humanities (I am certainly no exception), and many game designers follow their lead blindly. In humanities there is a strong tradition of formulating theories in such way that they do not dictate how to create a great film, novel, or painting; there is a preference for descriptive rather than prescriptive theories. From an academic point of view this makes perfect sense: academics analyze works of art from a neutral perspective. They try to stay away from value judgments and

generally are not in the business of producing works of art themselves. However, if academics genuinely want designers to use their theory to build better games without actually formulating a theory of quality in games, they have maneuvered themselves into an awkward position.

The descriptive tradition in humanities stems from fierce debates within linguistics. Prescriptive linguistics assumes one particular way of speaking or writing is better than other ways of speaking. Language education always has had a prescriptive function: it trains students to speak 'properly'. Modern linguistics has different goal, more concerned with the way language works than what might be considered to be the way in which it works best. In this context, being descriptive makes much sense, and actually serves ethical and democratic goals I certainly support. However, in games, and any other form of artistic media, these considerations do not apply equally. Artistic expression in games is not just a matter of being understood, it is also a matter of being effective, influential and successful. In the most pragmatic sense, certain aspects of game quality can be measured as monetary value: *Angry Birds* is a good game, simply because it sold millions of copies worldwide. Critical success might be harder to measure, but is not distributed across games equally, nor randomly.

It is important to note that prescriptive theory does necessarily not imply restrictive models; saying that quality can be objective is not the same thing saying that there is one ultimate formula for great games. In my experience, the expressive range of a model of games is a critical factor in that model's success. In natural language, expressive range is attributed less to number of words we have than it is to the number of meaningful combinations we can make from them [4]. In narrative theory the actant model [13] is a case in point. The number of typical narrative roles (the "actants") is far lower than the number of possible meaningful combinations and interrelations between them. For design patterns the interrelations are equally important. Typically, design pattern authors would point out and discuss the possible interactions. How structural these interactions and descriptions are varies considerably. In my experience, a structural treatment of the interrelations between pattern, and more importantly, a pattern library where interrelations between patterns emerge naturally from the theoretical foundation, indicate a wider expressive range and more a flexible, and therefore more valuable, application of the pattern library.

## 5. WHERE DOES QUALITY COME FROM?

The game industry, and especially indie game developers, oftentimes act as if raw creative talent is the only source of quality in games. Academics might have unwittingly fueled this sentiment by abstaining from identifying objective prescriptive qualities of games, qualities which effects can be predicted, deliberately designed into games and taught to a new generation of designers in order to advance the industry as a whole. However, if this industry sentiment were to be true it would be very difficult to explain the progress games have made during their brief history. Even though, arguably, this progress is not all that great. Progress on the technical side has been undeniable, but many critics feel that game design has progressed far less during the same time. Some would even go as far as suggesting that little

new game concepts have come up since the early eighties. The small measure of progress in game design can support the argument that game designers lack theoretical guidance from academia, especially if one considers that the gap between industry and academia is considerably less wide on the technical side. Before attributing this lack of guidance to industry reluctance, I would prefer to investigate whether our guidance has been up to par.

It is pivotal that any pattern library for game design formulates and formalizes a notion of quality in games. This might not necessarily be a grand unifying theory identifying a single source of quality in all games, but we can no longer get away with identifying what works without identifying why it works. A good example is the book *Game Feel* by game designer Steve Swink [23]. In his book Swink investigates how the controls of a game contribute to the game's visceral experience. Swink departs from idea that a game's "feel" is created from three building blocks: real-time control, simulated space and polish. These three concepts build on results in the established fields of interaction design, information theory, and animation. They lend clarity and structure to his observations, making them better transportable to games beyond his examples. Although not presented as a pattern library, I would say the clearly defined focus and firm theoretical foundation makes Swink's work invaluable to any game designer.

The point here is not to identify all sources of quality in games. I like to think that list could never be exhaustive. Depending on the particular focus of the design pattern library, many things could serve as a theoretical foundation. Ideally, the theory suggests how to deal with particular problems, how different patterns interact, how to transfer patterns to specific designs, and ultimately also how and when to abandon the patterns altogether.

When design patterns are developed around a clearly defined theoretical core the library become something more than a set of descriptions of generic solutions to common problems. They become a lens that provides designers with an informative perspective on games. Ideally, the lens reveals structures and qualities in the game that are otherwise hard to see. For this reason, I think graphical notations can be a great help. Software design patterns has UML diagrams to reveal structures software architecture. My own Machinations diagrams [8] are versatile records of game economies which would normally be hidden behind layers of mechanics. Daniel Cook's skill atoms give shape to learning curves and skill dependencies that might otherwise not be apparent [6]. I would love to see a pattern library developed on top of Cook's practical, clearly defined, and very visual lens of learning in games.

## 6. CONCLUSIONS

In order to make design patterns work it is due time we academics re-evaluate how we design pattern libraries and to what purpose we design them. Assuming the game industry is an intended audience for our design pattern libraries and following the discussion above, I would strongly suggest that any design pattern library for games adheres to the following four design criteria:

1. A design pattern library should depart from a clearly formulated notion of quality in games. This quality does not need to span all games and all aspects of game design. Rather, it is generally best if a pattern library restricts itself to particular aspects of games. Preferably, this notion of quality builds on academic or scientific results from relevant and established disciplines.
2. The patterns should not merely describe common structures games, rather they should provide a unique lens onto games that brings forward qualities of games that are otherwise difficult to perceive: they should be informative not just descriptive.
3. The patterns themselves are presented as problem-solution pairs, where the discussion and examples make clear that the problems are common, and the solutions are generic. In other words, they are (to a certain extent) prescriptive.
4. The relations and interactions between patterns are made clear, are plentiful and create a flexible and expressive language that is able to deal with a large variety of games. In other words, they are not restrictive.

Even if pattern libraries follow these criteria, success cannot be guaranteed. There are still many obstacles between the design of a pattern library and its widespread application. Many open questions remain. Most importantly, much progress can be made with better integration of design patterns into the development cycle. However, that ball is in the games industry's court. As academics it is our prime responsibility to supply the industry with pattern libraries that have enough practical value in the first place.

## 7. REFERENCES

- [1] C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King, and S. Angel. *Pattern Language: Towns - Buildings - Construction*. Oxford University Press, New York, NY, 1977.
- [2] S. Björk and J. Holopainen. *Patterns in Game Design*. Charles River Media, Boston, MA, 2005.
- [3] S. Bura. A game grammar. <http://www.stephanebura.com/diagrams/>, 2006.
- [4] N. Chomsky. *Language and Mind, Enlarged Edition*. Harcourt Brace Jovanovich Inc, New York, NY, 1972.
- [5] D. Church. Formal abstract design tools. Gamasutra, available online at [http://www.gamasutra.com/features/19990716/design\\_tools\\_01.htm](http://www.gamasutra.com/features/19990716/design_tools_01.htm), 1999.
- [6] D. Cook. The chemistry of game design. Gamasutra, available online at [http://www.gamasutra.com/view/feature/1524/the\\_chemistry\\_of\\_game\\_design.php](http://www.gamasutra.com/view/feature/1524/the_chemistry_of_game_design.php), 2007.
- [7] S. Dahlskog and J. Togelius. Patterns and procedural content generation: Revisiting mario in world 1 level 1. In *Proceedings of the First Workshop on Design Patterns in Games, Raleigh, NC*, 2012.
- [8] J. Dormans. *Engineering Emergence: Applied Theory for Game Design*. PhD thesis, University of Amsterdam, The Netherlands, 2012.
- [9] G. S. Elias, R. Garfield, and K. R. Gutschera. *Characteristics of Games*. The MIT Press, Cambridge, MA, 2012.
- [10] N. Falstein. Better by design: The 400 project. *Game Developer Magazine*, 9(3):26, 2002.
- [11] M. Fitch. Against a formal language for game design. Available online at <http://www.micrysworld.com/office/formallanguage.html>, 2004.
- [12] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison Wesley, Boston, 1995.
- [13] A. J. Greimas. Actants, actors, and figures. In P. J. Perron and C. Frank H, editors, *On Meaning: Selected Writings in Semiotic Theory*, pages 106–120. University of Minnesota Press, Minneapolis, 1973.
- [14] S. M. Grünvogel. Formal models and game design. *Game Studies*, available online at <http://gamestudies.org/0501/gruenvogel/>, 2005.
- [15] D. Guttenberg. An academic approach to game design: Is it worth it? Gamasutra, available online at [http://www.gamasutra.com/view/news/8908/Student\\_Feature\\_An\\_Academic\\_Approach\\_to\\_Game\\_Design\\_Is\\_It\\_Worth\\_It.php](http://www.gamasutra.com/view/news/8908/Student_Feature_An_Academic_Approach_to_Game_Design_Is_It_Worth_It.php), 2006.
- [16] R. Ismail. On the church of reason and punk game development. Gamasutra, available online at [http://www.gamasutra.com/blogs/RamiIsmail/20130226/187384/On\\_the\\_Church\\_of\\_Reason\\_and\\_Punk\\_Game\\_Development.php](http://www.gamasutra.com/blogs/RamiIsmail/20130226/187384/On_the_Church_of_Reason_and_Punk_Game_Development.php), 2013.
- [17] R. Koster. A grammar of gameplay: game atoms: can games be diagrammed? Presentation at the Game Developers Conference, 2005.
- [18] C. Lindley. Game taxonomies: A high level framework for game analysis and design. Gamasutra, available online at [http://www.gamasutra.com/view/feature/2796/game\\_taxonomies\\_a\\_high\\_level\\_.php](http://www.gamasutra.com/view/feature/2796/game_taxonomies_a_high_level_.php), 2003.
- [19] A. Martin. Do game designers actually need to go to school? Available online at <http://killscreendaily.com/articles/essays/do-indie-game-designers-actually-need-go-school/>, 2013.
- [20] D. Milam, L. Bartram, and M. S. El-Nasr. Design patterns of focused attention. In *Proceedings of the First Workshop on Design Patterns in Games, Raleigh, NC*, 2012.
- [21] G. Rivera, K. Hullett, and J. Whitehead. Enemy npc design patterns in shooter games. In *Proceedings of the First Workshop on Design Patterns in Games, Raleigh, NC*, 2012.
- [22] B. Sheffield. Defining games: Raph koster's game grammar. Gamasutra, available online at [http://www.gamasutra.com/view/feature/1979/defining\\_games\\_raph\\_kosters\\_game\\_.php](http://www.gamasutra.com/view/feature/1979/defining_games_raph_kosters_game_.php), 2007.
- [23] S. Swink. *Game Feel: A Game Designer's Guide to Virtual Sensation*. Morgan Kaufman, 2008.
- [24] J. P. Zagal, M. Mateas, C. Fernández-Vara, B. Hochhalter, and N. Lichti. Towards an ontological language for game analysis. In *Proceedings of DiGRA 2005 Conference: Changing Views - Worlds in Play*, 2005.