Games with(out) rules

Sotirios D. Kotopoulos  
Massachusetts Institute of Technology, USA  
skots@mit.edu

Abstract
Fifty years after the first attempts to introduce algorithmic methods in design we have reached a point where we might ask if design has become not a game in which the designers play with the notion of rule, but a game where they play according to rules.

1. Introduction
Computational design deals with the construction of systems that use algorithms to produce designs with certain properties and with the construction of theories that give account for the way these designs are produced. The paper examines algorithmic and digital representation, randomness and innovation within the twofold context of design production and design explanation. It is pointed out that digital design corresponds to a specific sub-domain of computational design. Digital design confines design on specific conceptual and productive boundaries, which require accurate planning and sophisticated machinery of reproduction. When design is approached via digital means, doing away with ambiguity becomes a primary concern. This is especially true for the designers who feel that what they want to do must be determined by what a digital machine can do. Fifty years after the first attempts to introduce algorithmic methods in design we have reached a point where we might ask if design has become not a game in which designers play "with" the notion of rule, but a game where they play "according to" rules.

Pre-school children explore the playground by interpreting the scattered toys in unexpected ways. Pleasure does not depend on knowing where the play is leading. Rules are made up and abandoned effortlessly. Grown-ups, on the contrary, determine rules with sincerity. They participate in games in which they have to outwit each other. Sometimes it is pleasant to think of design as a playground, where we make up games without rigid rules and forget the rivalries between winners and losers. The Greeks had placed players under the protection of Mercury (in Latin), or Hermes (Ερμής in Greek). Hermes was also the god of interpretation or herme-neia (ερμηνεία, in Greek). This implied that the interpretation of rules is involved in playing, but also that the interpretation of a course of action leads to the formation of rules. Hermes was widely held responsible for encouraging his followers to cheat by diverging from the established rules. Sincerity would never help a player to win. In the arts, cheating was often titled poetic license. The presence of the "deus ex machina" in Greek drama was acknowledged as poetic license. Today, it is generally accepted that the task of the creative person is not to legislate, but to unsettle. Writers and poets risk the reinterpretation of the rules of language to produce original interplays of words, while painters, sculptors, and architects violate the conventions of visual expression, to be championed by people who strive for visual novelty. Nevertheless, the premise of organizing games without rigid rules does not preclude the possibility that in the name of poetic license one may invent games dedicated to rule setting, like the construction of algorithms or the making of computer programs.
2. Background

Ever since Plato and Moses denounced the making of representations as illusions, the artists of the Western world have been treated with distrust as propagators of errors. Even when modern art reconfigured the relationship among art, imitation and abstraction, artists have been repeatedly blamed for deceiving “truth” and “beauty”. In the modern world, the artists were accused of betraying the canons of beauty much like in the Middle Ages the prodigal sons of the church were accused of betraying the rules of God. The folly of the prodigal son consisted in his having used his own rules to please himself rather than following the rules of the church and serve God.

At the beginning of the 20th century, Henry Poincare paved the way for artists and mathematicians to communicate through their common admiration of form. Poincare claimed that the foundations of mathematical creation are extra-logical. He proposed that mathematical and artistic creation has aesthetic origin: it implies an aptitude to discern and select among constructions the ones that can be of potential use. This does not consist in employing new combinations of existing forms, Poincare said, because calculations made on this basis are of limited interest. Ludwig Wittgenstein treated art as a language able to reflect a state of the artist's mind. Initially, the young Wittgenstein attempted to construct a logical language, defined to reveal unambiguously the facts to which it refers. He presented these ideas in Tractatus Logico-Philosophicus claiming to have established “definitive truths”. Soon, he abandoned this certainty, which he came to regard as neither possible, nor desirable, for what he called language games. Around the same time, Rudolph Carnap in the preface of the book Der logische Aufbau der Welt was attempting to place his logistic approach to philosophy, in context with art and architecture. Carnap took strong interest in the ideas developed at the Bauhaus, where he used to give lectures to young designers. At the Bauhaus, Paul Klee and Wassily Kandinsky were attempting to realize a formal framework for painters, sculptors and architects. Key aspect of their approach was the effort to “formulate the laws of art as simple rules”. The objective was not to reduce design into prescriptive formulas but to arrive at conventions that propel creativity. Klee suggested that one should build one’s own system of rules with precision: “We shall try to be exact but not one-sided. What we are after is not form, but function”. But Klee also endorsed artists’ tension to diverge from the established order: “It is clear that different functions operating in different elements will lead to sharp divergences. And yet some people would like to deny the artist the very deviations that his art demands”.

Current computational design research concerns the construction of algorithmic processes that generate designs with specific properties. Computational design was introduced in the 60’s and 70’s in an effort to use of formal devices like set theory (Alexander 1964), graph theory (Steadman 1973), Boolean algebra (March 1970), programming languages (Eastman 1972; Mitchell 1974), formal syntax (Hillier et al. 1976), and shape grammars (Stiny and Gips 1972) in design. Computation was employed as a prescriptive or as a descriptive instrument. In prescription, computational rules were applied as prescriptive instructions to

4 see Carnap R, Der Logische Aufbau der Welt, translated as The Logical Structure of the World, Chicago and La Salle Illinois: Open Court, 1928 (2003), xviii
provide a norm for the production of designs. In description, the rules were used to map the behavior of designers and to affirm that the claims of some general hypothesis produce the desired results.

Computational design aims to provide minimum principles by the means of which we can practice and explain design, in three ways: First, by describing the design process explicitly; second, by leading to the implementation of devices with strong generative capacity; and third, by making a design process available for future reference. The descriptive task involves the mapping of the actions of a designer with the aid of rules. The generative task involves their implementation in grammars or computer programs. The reference task involves the assemblage of data structures that can be retrieved by future users. Computational design systems include a calculating and a syntactic-interpretive part. The calculating part provides an environment where calculations of some kind take place. And, the syntactic-interpretive part consists of statements assigning practical meaning to the computations. The application of a thought is expressed computationally by the application of a rule. In this way, the thought becomes a step in a calculation. New steps can be added by inserting new rules. A rule specifies that given some condition \( x \), a conclusion \( y \) is produced, that is, an objective is accomplished provided that some conditions are satisfied. For example, the dissection of a quadrilateral "room" by a "wall" can be expressed by a rule. The anticipation of this action consists in the ability to know that a specific result can be produced whenever such a "room" is found in a description.

Table 1. Rule-expression, rule-condition, rule-conclusion

| rule | + | \[ | + |
|------|---|---|
| condition |  + | |
| conclusion | + |

3. Games without rigid rules

Kant observes (§ 49) that all exemplary artistic creation seems to be a product of rules. These are self-imposed by the artist without conscious attention. Further, Kant notices that an important work of art seems unprecedented because “it discloses a new rule”.\(^6\) In both cases a kind of free play, similar to that of children, is involved.

A common criticism to the computational approach to design is that computational models are too rigid and fail to capture the freedom with which designers act and think. Computational models provide a formalized environment for calculation, but actions have to be reduced to become expressions of the model language in order to be treated computationally. Computational models – the criticism continues – can represent, at best, only moments in a process characterized by constant flux. Winograd and Flores contrast formal representation with actual human thought, by arguing that the projection of human capacities onto computational devices is misleading. Classifications or distinctions caused by formal procedures eliminate certain possibilities thus causing a specific kind of blindness: “Blindness is not something that can be avoided, but it is something that we can be aware. The designer [of a system] is engaged in a conversation of possibilities. Attention to the possibilities being eliminated must be in constant interplay with expectations for the new possibilities being created”.

Like most researchers in the area of computational design, I see the existence of a formal component necessary for productive and explanatory purposes. The selection of the appropriate formal component can have great qualitative and quantitative implications for the modeling of a specific domain. One cannot simply pick a computational apparatus and squeeze the empirical content in. The philosopher Nelson Goodman discloses some of the theoretical implications of descriptions belonging to diverse modeling systems, or “worlds”: “In other cases, worlds differ to theoretical rather than practical needs. A world with points as elements cannot be a world having points as certain classes of nesting volumes or having points as certain pairs of intersecting lines or as certain triples of intersecting planes. That the points of our everyday world can be equally well defined in any of these ways does not mean that a point can be identified in any one world with a nest of volumes and a pair of lines and a triple of planes; because all these are different from each other. Again the world of a system taking minimal concrete phenomena as atomic cannot admit qualities as atomic parts of these concreta”. Theoretical considerations like the above obtain practical significance in design, where designers use points, lines, surfaces and solids to construct design descriptions. The process of construction becomes problematic whenever computational systems restrict the use of these elements in unintuitive ways. For example, digital software requires descriptions to be registered in the memory of the computer with a permanent structure. The preservation of the structure becomes a barrier: it does not allow one to diverge from one’s initial design approach. The creator of the first CAD system Ivan Sutherland explains: “An ordinary draftsman is unconcerned with the structure of his drawing material. Pen and ink or pencil and paper have no inherent structure. They only make dirty marks on paper. The draftsman is concerned principally with the drawings as a representation of the evolving design. The behavior of the computer-produced drawing, on the other hand, is critically depended upon the topological and geometric structure built up in the computer memory as a result of drawing operations. The drawing itself has properties quite independent of the properties of the project it is describing.”

Digital design is a specific sub-domain of computational design, where rules and procedures are converted into machine language to be executed by digital machines. The interest in the use of digital machines in design is a cultural phenomenon that characterizes the post-
industrial era. The objects that digital design exhibits are conceived and produced digitally, as opposed to analogue-ly, or mechanically. Digital design production rests on the assumption that man is greatly impressed by the results of calculations occurring in a specific domain, namely, the domain of numerically defined representations. This domain involves calculations with zero-dimensional elements (points and symbols) as opposed to elements of higher dimension (lines, planes, solids).

When design is approached via digital means, doing away with ambiguity becomes a primary concern. The program of a digital machine has to assimilate a sequence of execution tasks free of ambiguity and with an unambiguous overall objective. The language, in which this meticulous description is assembled, does not provide the best means for inducing doubt. But, it is within the territory of doubt that a designer conducts the most fruitful experiments. The programmer of a digital machine has to determine in advance, with precision, possible spaces of configuration, which underlie what we see on the screen. In this way, many possibilities are being created, but also many others are being categorically eliminated. The aim of the programmer is not to depict the visual “how” but the measurable “why”. In order to build and use this type of knowledge, the programmer has to be an iconoclast: one who has more faith in the realism of the quantities preserved in the memory of the machine than in what is visible. George Stiny who has written extensively on the properties of visual calculation, observes: “When memory counts more than what I see, it isn’t visual calculating. There’s a conservation law of some sort to uphold the decisions I’ve made in the past – to recognize (remember) what I did before and act on it heedless of anything else that might come up”.10

Digital descriptions of observable things need to be conserved in the memory of the computer. Therefore, form is expressed as an n-dimensional manifold, which can be numerically generated. Also, form, color and other attributes can be detached from objects. And because, all observed attributes obtain numerical expression, arbitrary combinations of their values can be inserted in random processes to generate unforeseen results. Geometry can also become indeterminate. The probability of obtaining unexpected formulations can be increased by allowing chance among the elements of a numeric calculation.

Since process has become part of the design content, chance has become indispensable element of the composition. Players, artists and children share the same fascination with chance, but their motivation differs. Players seek fulfillment by risking in a pre-determined process, which can bring them wealth. Artists and children engage themselves into a non-deterministic process that enriches their experience. Nevertheless, there are no available algorithms for producing masterpieces, as there aren’t any for winning the roulette. In art, chance can be introduced by a lapse, a slip, or by accident, or it can be invoked by loosening control over a process, or a material. The inclusion of chance in a sequence of machine instructions should not be confused with non-determinism. In random machine processing, chance makes its involvement present as a surprise. If we were to evaluate the worth of a surprise, we would have to view it as a ‘not-easy-to foresee-result’ occurring in the course of a sequence of predefined (deterministic) events. When a work of art is conceived as a surprise, it is called a happening. If we narrow our view to the point at which all events are reduced to happenings, changes become meaningless, since – like in flipping a coin – they are no longer placed within any context.

Chance makes its involvement present in more interesting ways, as emergence. Emergence is a feature of systems that display qualities not directly reducible to the system’s constituent parts. In such systems ‘the whole is greater than the sum of the parts’. Knight observes that systems based on elements of higher than zero dimensions (lines, planes, solids) exhibit strong emergent behaviors. Further, Stiny shows that within such systems, descriptions are characterized by the absence of permanent subdivisions, and rules can apply unrestricted. The structure of a description ceases to be rigid and becomes “an artifact of computation” guided by the changing priorities of a viewer.

Figure 1. A rule that rotates a square by 180°

Figure 2. Each time the rule applies on the interior small square, a new square emerges

4. Epilogue

If design is to be comprehended as a process, it cannot be dissociated from a set of rules, a method, or a language, thanks to which conclusions and calculations can be verified. The originality of a design lies in the uniqueness achieved through the incomplete adherence to the method (μέθοδος in Greek means “following the road”). The reasons for departing from the road can vary. First and foremost, designs are made unique so that others may appreciate the designer’s point of view. The departure from the road becomes problematic in systems that restrict the free application of rules by imposing inflexible structures to descriptions. The preservation of structure for a description operates as a barrier: it does not allow one to diverge from one’s initial course. An example can be found in the use of digital software packages: they require descriptions to be structured in specific ways. After a description is complete, one can only employ combinations of the prescribed parts. But this – as Poincare points out – leads to calculations of limited creative potential.

11 See Knight T, "Computing with emergence" Environment and Planning B: Planning and Design 30(1), 2003, pp. 125 – 155
Digital design extends design beyond visual description to abstract spaces of configuration. In order to use this type of knowledge the designer must become a programmer: one who has more faith in the realism of quantification than in the presence of the visible. Fifty years after the first attempts to introduce algorithmic methods in design we have reached a point where we might ask ourselves if design has become not a game in which the designers play “with” the notion of rule, but a game where they play “according to” rules. The risks taken in games played according to rigidly determined rules are limited when compared to the risks that one can take in games that are approached as fields of unrestricted experimentation. Design still remains an open game.

5. Endnotes
