The Thirteenth Annual Game Design Think Tank
Project Horseshoe 2018

Group Report:
Constructing Emergence

Report written by Mike Sellers

Participants: *A.K.A. "A Game Walks into a Bar…"*
Michael Austin, Hidden Path Entertainment
Kate Compton, Independent
James Ernest, Cheapass Games
Josh Lee, Google
Mike Sellers, Indiana University
Jerry Belich, Miami University
Joris Dormans, Ludomotion
Joel Gonzales, Firaxis
Josh Mosqueira, Bonfire Studios
Jamey Stevenson, DeepMind
Facilitator: Ray Holmes, Babeltime

Overview

Emergence can be elusive to define and discuss, much less to reliably develop in artificial systems like games. In practical game development, constructing systems that show emergence can also seem risky, since few understand what emergence *is*, and since such systems often look incomplete and broken until suddenly they're not. Unlike more monotonic methods of game design such as designing level by level, boss by boss, when building systems there's no simple metric for a producer or manager to look at to gauge progress and risk. A system being designed with emergence in mind might be on the threshold of working spectacularly well, or it may never come together; it can be nearly impossible to tell the difference.

At the same time, many (including those in our group) believe that complex systems leading to emergence make for better, more engaging, and ultimately less expensive games with longer replay value. Games and toys that have successfully created emergent systems that interact with players are evergreen favorites and commercial successes. Examples include The Sims, LEGO, Minecraft, Unexplored, and Go.

In addition to the commercial value of understanding and constructing emergence, we believe that games provide a unique way to gain insight into emergence itself, which is an important topic in its own right. Complex systems of all sorts exhibit emergent properties. Any progress toward being able to more reliably create such systems with less guesswork will be a significant step forward.

Given the importance, benefits, and difficulties of creating emergent systems, we are looking for tools and methods to more reliably construct emergence in games, rather than just hoping for it and stumbling over it.

Solution Overview

We have not fully solved the problem "how to construct emergence" in a turnkey manner; it's readily apparent that this is a convoluted problem that will require iteration to converge toward a solution. However, our findings should add to the tools and methods that can be applied by game and systems designers seeking emergence. We hope that others will continue to build on and refine these as well.

We came up with two primary and related heuristics for designing game systems that result in emergence:

1. Articulating multiple axes on which games and interactive systems can be evaluated for their potential to create emergence, allowing designers to modify their game systems to be more likely to create emergence
2. Application of existing tools and methods (e.g., the Component Interaction Matrix) to game designs in new ways, particularly hierarchically

These are both discussed in more detail below.

Detailed Discussion

Emergent gameplay is the result of creating complex gameplay systems rather than relying on expensive, one-off content to do so. Relying on content-driven game design maintains a larger degree of direct game designer control over the game, and just as importantly producer and budget control: churning out assets from a list and assuming they will result in an engaging game is seen as the *de facto* low-risk path to game development. However this also puts developers on the "content treadmill" where they are trying to supply content to players faster than the players can consume their set-pieces of art, layout, and scripting. This has several pathological consequences:

- It results in needing an army of artists, level designers, and scripters -- until they're no longer needed and are laid off, resulting in career instability and feast-and-famine business practices.
- The need to "pound out content" contributes to a culture of crunch, where new content must be sent out on a more or less constant basis, to "feed the beast" of player expectations.
- This insatiable need for content increases development budgets and thus risk of a game not providing sufficient ROI.
- This in turn reduces the opportunities for innovation in game design, since new systems and features further increase overall risk, and are typically cut in favor of lists of new content
- At the same time, the content-driven approach reduces replayability, making games more disposable.
- In an age of potential players first watching a game being played (via an online stream) those that depend on set-piece or other non-systemic content see significantly lower sales (see for example Erik Johnson's GDC 2018 "Making Indie Games that Sell" talk, especially at 16:26ff), since players have already seen what the game has to offer before they buy it.

Developing games using a systems and emergence approach avoids these issues. In effect, emergence provides more gameplay for less content -- more systems, less treadmill. Using systems yielding emergent gameplay, designers are able to create more engaging experiences in areas such as crafting, combat, trading (economics), resource usage (ecologies), and social dynamics.

Emergence can take place on multiple levels, such that, for example, a player character in a game might combine several of their abilities to create a novel effect in combat. This effect contributes to the overall effectiveness of her party (in combination with other characters' abilities), which in turn alters the social dynamics of their guild or clan in the higher-level social space of the game. These nested emergent effects embody John Wick's observation that "fun happens *between* the rules." That is, the players create their fun in concert with the game based on its rules, but not by being focused on them: the game isn't the content or the rules, but the experiences the player is able to create between, or in emergent terms, arising out of them. This is far more difficult to attain using content-driven play, where it is easier (and in many cases more beneficial) for players to focus on the rules and content rather than the experience, and to thus predict and minmax the game to their own ends.

The downside of creating systemic, emergent gameplay as stated before is that it can be difficult to assess during development, since such development is not linear. It's not possible to create a simple burndown chart of a new system intended to create emergent gameplay with the player. This unpredictability has made many game developers skeptical of systemic design, and has kept them from seriously investing in it. Currently some game designers hope to create emergent behavior in their games, but this often amounts to a vague desire without method, by analogy like tossing bricks into a pile and hoping they make a house.

Definitions

There are many descriptions and attempts at definitions of emergence, but they're often incompatible and ad hoc in nature. Based on prior work by those like Holland, Meadows, Lazlo, and others, our working definition of any emergent phenomenon is that it:

- is the result of the interactions between parts or agents in a system when together they interact to create a new identifiable sustained effect (or independent object) at a higher level of organization.
- is not immediately predictable at the higher level of organization based on inspection of its lower level parts, but it is often logically apparent when viewed after the fact.
- is not determined by any one constituent part at the lower level, but by the non-linearly combined actions of multiple pieces interacting together.
- may result in many levels of hierarchical emergence, with each integrative level acting as a part in the emergence in next-level-up effects if they conform to these general requirements for emergence.
- has the essential characteristic of being easier to understand and describe as an integrated thing on its own than in terms of its constituent parts.

An atom, a molecule, a cell, a person, a company, and a chess strategy are all examples of emergent effects that have have these attributes in common.

A more playful way of describing emergence is that it's like a joke: you understand the premise and the rules by which the actors and parts interact, and you know the punchline is coming, but you can't see what it is in advance. But after you hear it, it's obvious while still surprising. (Thus the name of our working group.)

Goals

Our workgroup's goal was to explore methods and tools to aid the construction of emergent gameplay. We explicitly stated that our context was games, and potentially other systems in which one or more humans are participatory agents. Games often contain multiple systems on their own, and together the game and the player(s) create a higher-level system to which both contribute. This combination can, if the game systems support it, create a set of emergent experiences that are engaging to the player. Player experiences with games like Minecraft or The Sims, or with toys like LEGO, bear this out. In this discussion we are not focused on emergence in 'pure' simulations without human interaction, though some of our methods and tools may shed light on those as well.

In effect, our goal was to find ways to say, "if your design satisfies these conditions, you will get emergent gameplay out of it," or even, "if you satisfy these conditions, you will get a particular type of emergence." We made progress on the first of these, but there is a great deal of work yet to be done to reliably approach the second.

Requirements for Emergence

The definitional aspects above led us to consider the base requirements for emergence. These included an overall systemic organization included in the above definition: any system exhibiting emergence contains multiple parts that interact in non-trivial ways. These interactions must form loops, i.e., where object A affects B, which affects C, which in turn affects A. This looping nature is inherent in any combination of objects that together create a system and emergence. In addition, no single object or small group objects in the system may overpower all the others. For example, in a combat system where one weapon always wins a contest, the results collapse to become entirely predictable, thus precluding any degree of emergence.

In addition to these requirements, our discussion led us to three distinct and, we believe, orthogonal categories or dimensions that can be considered the pillars of emergent gameplay. Combined together, these form a space within which emergent effects may be found. These three dimensions are **Agency**, **Abstraction**, and **Complexity**. Each is described in detail below.

For each dimension, a game may have "little" or "a lot" of that quality (this is necessarily qualitative and inexact, as we currently have no global scale by which to measure them). We think of these as going from negative (cold, rigid) to positive (hot, chaotic) with the "just right" point at zero. In general, games that are closer to zero or the Goldilocks-zone (some, but not too little or too much) for each of these are more likely to yield emergent behavior. This is based in part on the work of Christopher Langton and his lambda measure for "edge of chaos" computation in cellular automata, supplemented by the work of Ben Li-Sauerwine on "cold to hot" Conway's Life simulations, and on our own experiences creating games.

Despite the Goldilocks/edge of chaos zone, there are important examples that are further out in *one* dimension more than the others. As such, we believe the "emergence probability space" is more stellate than spherical, with the extremes of any one dimension still being able to support emergence (see Figure 1).

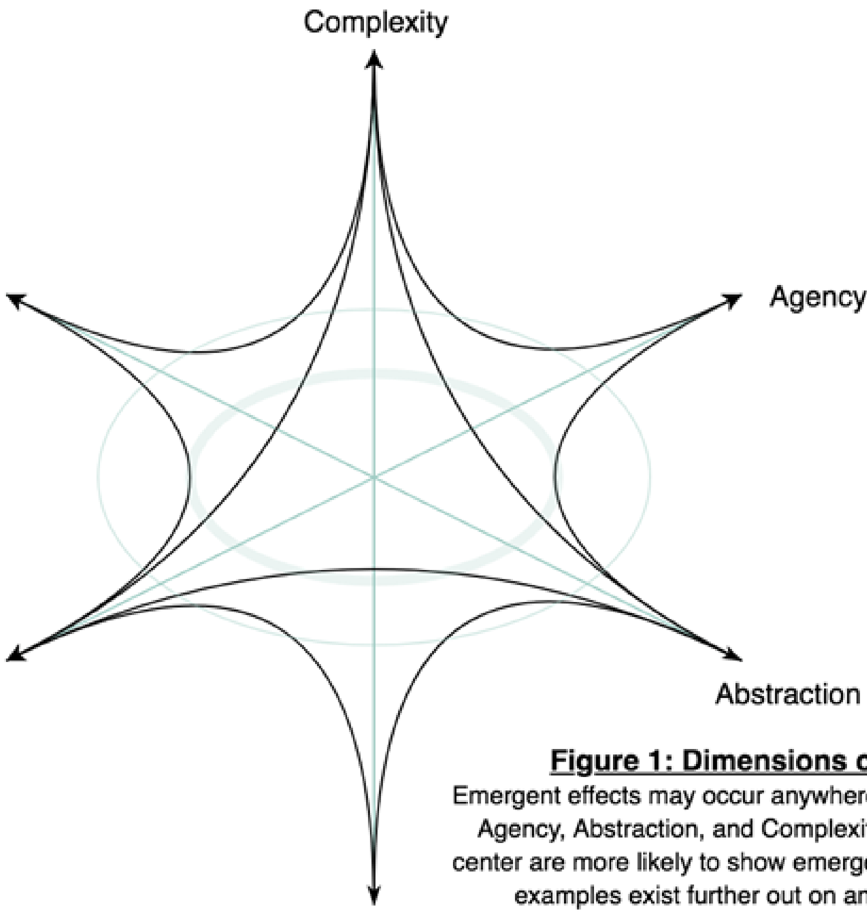

Figure 1: Dimensions of Emergence
Emergent effects may occur anywhere within the "envelope" of Agency, Abstraction, and Complexity. Games closer to the center are more likely to show emergent effects, but significant examples exist further out on any single dimension.

Agency
This is an approximate measure of the number of choices available to the player. How many verbs do they have to use in the game? In a game like Tetris, the verbs and choices are few; in Dwarf Fortress they are bewilderingly many. In the middle there are enough verbs for the player to have interesting, non-trivial choices to make, but not so many that the game overwhelms their cognitive or expressive abilities to interact with the game.

It's also important that these verbs are clear and concise. A verb that affects too many other parts of the game violates the locality rule described below under Complexity. One that has effects that are unclear to the player harm Abstraction, in that the player cannot use the game's feedback to enhance their mental model. Similarly, these verbs should be generic in their effect, in that they do not depend much if at all on their context to determine their function. That is, a healing potion will not cause a character to turn invisible if her hat is purple. Such an example is a form of exceptional specificity that violates organizational locality and throws a wrench into the player's mental model. In general, the fewer such exceptions and patches the player must apply to their mental model, the more elegant the game is perceived to be, and the more supportive of emergence it becomes.

To raise or lower the amount of agency, the number and variety of verbs can be increased or reduced. Reusing interfaces to allow for more verbs while maintaining similarity and low cognitive load (with respect to "how do I carry out this intention?" on the player's part) may allow for greater agency and more verbs without breeding chaos. And in some cases, if the player is willing to put in the work needed to build a coherent mental model of the game, their agency remains in the sweet spot even with a large number of available verbs. This is the case with many games acknowledged to be difficult, but where devotees believe the results are worth it (e.g., EVE Online, Dwarf Fortress, many "grand strategy" games). This far end of what might be called costly agency can enable a great deal of emergence -- but it is perceivable only to players who have the expertise in having built their mental model to recognize it.

Abstraction

Where Agency is about what the player can do in the game, Abstraction is about how well the player can "read" the game in terms of its universally applicable forms. We are using Abstraction here in the sense used by Scott McCloud in Understanding Comics (Figure 2) to arrive at the essential, generalizable, abstracted level of detail. A game too low on Abstraction is too specific to allow for emergent interpretation; one that is too high is so abstract that it carries no symbolic content. On either end, the player is left without much scaffolding for building meaning in their mental model, and thus recognizing emergent effects from the game.

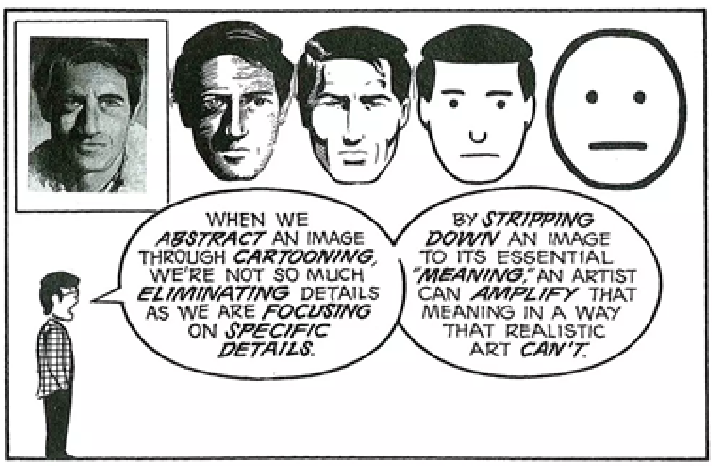

Figure 2: Visual abstraction in comics, as explained by Scott McCloud

However, there is at least one important case of emergence in games that is on the high end of the Abstraction dimension: Go. This game consists of nothing but black stones, white stones, and a gridded board. These are so abstract as to be stripped of all symbolic meaning. However in this case (in combination with being in the "sweet spot" in Agency and Complexity), the player uses these plain components to build emergent patterns at a higher level. As with games providing "costly Agency," such costly Abstraction games put intense requirements on the player to form their mental model without the help of significant symbolism within the game. Nevertheless, as Go shows, for those who devote themselves to the game, the depths of hierarchical emergence that they experience are endlessly rewarding.

Abstraction goes far beyond the visual aspect of the game: it includes incorporating resonant themes and archetypes, but without tipping over into idiosyncrasy, cliche, or overused tropes. In many ways, the sweet spot for abstraction in a game that enables emergence the most is evident in what are often called "deep games" as used by Doris Rusch. Games that speak to the breadth of human experience (well beyond shooting things or matching three of a kind) tap into universally recognized human themes, but without becoming so specific that the player and game cannot create an emergent experience.

Complexity and the Component Interaction Matrix
The final dimension, and the one that is most fully embedded in the game itself (rather than being in both the player and the game), is that of Complexity. Complexity has to do with the ways and the degree to which components within a system interact. As with the “cold to hot” metaphor mentioned earlier, in systems where components interact, on the cold end they may do so too little to create complexity, creating only stability or repetitive content, or on the hot end they may interact so much that all that is created is chaos -- all noise and no signal. This hearkens back to cellular automata (CA) again, particular the Class I, II, III,and IV one-dimensional automata that have been studied extensively.

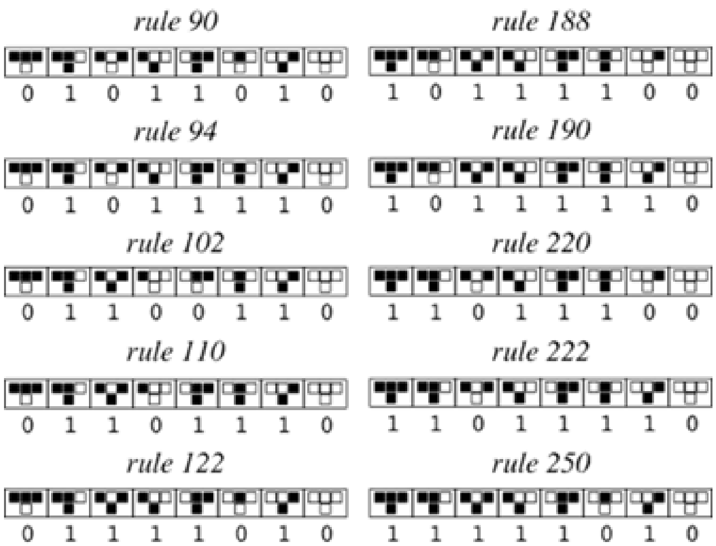

Figure 3: Examples of 1-dimensional cellular automata.

Briefly, these CA are ones in which any adjacent group of three bits determine what the center bit will be at the next time step. Because each bit can be in one of two states (on or off), and it takes three bits to determine the state of the next one, there are 8 states per rule (23) and 256 total rules (28). Examples are shown in Figure 3. Drawing the result of each rule out with each step taking place on the next line down reveals that some create static results (for example, all bits turn off in Rule 0); some create merely repetitive line patterns (Rule 108), some create only static-like chaotic noise (Rule 30), and a few create similar but not quite repetitive, meta-stable patterns (Rule 110). See Figure 4.

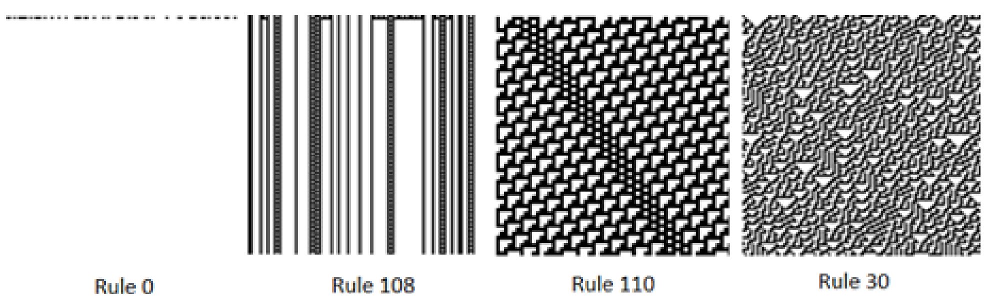

Figure 4: Class I, II, IV, and III cellular automata followed over time

Class I and II have too few interactions between cells to produce anything interesting; they are too “cold” to create complexity. Class III has too many connections, too much activity -- too hot -- and so results in chaos. Class IV is in between these extremes in the Goldilocks or “edge of chaos” zone. In a 1-dimensional CA like this, the edge of chaos zone appears to center around a probability of just over ½ - about 0.55 - of a cell turning on. This appears to be a local optimum, a peak where complexity can emerge.

As you can see in the depiction of Rule 110 in Figure 4, there is a degree of stability that isn’t simply static.

The diagonal grouping moves its position with every generation (every new line drawn), maintaining its stability as a whole, as an emergent phenomenon, even though the underlying bits within it change in each generation. In other words it is meta-stable based on changing components one level down.

One other important point here is that each object, each bit in the CA, interacts only locally. That is, a bit does not affect the state of another bit far away (spatial locality), nor does it affect that behavior of a large group of bits (organizational locality). This attribute appears to be important to creating sufficient complexity for emergence without creating either too much rigidity or chaos.

In game terms, emergence requires *some* connections between objects in the game (and between the player and the game). Too few, and the game is unremarkable; too many, and the game may have lots of activity, but no meaning. The latter condition is known in game AI and narrative circles as the “Tale-Spin problem,” after the early (1976!) story generator. There was a great deal going on within the program, based on the many interactions between characters in Aesop’s Fables. However, there was so much going on, so many characters pursuing their own goals, that no sense of narrative or coherence emerged. The player was often left wondering what happened and why, with no way to build a mental model of what was going on. There was a great deal of activity, in other words, but no meta-stability to hold it all together and create higher-level emergent structures.

To sum this up: to create usable, perceivable emergence, a game must have sufficient behavioral connections between components so as to not be trivially predictable; some degree of surprise must be possible. However, if the system has too much going on, it becomes too chaotic for the player to build a mental model from which they can make useful predictions about the game’s overall, emergent behavior.

In an earlier AI application, one of our group came up with the term “retrospective predictability” to describe the quality of a player being surprised by the AI’s behavior, but then having a sufficient mental model to be able to integrate an event or behavior into the model in a way that ultimately confirms and strengthens the model itself. It’s the experience of saying, “Why did it do that? Ohhh, I see!” This was perceived by users of the system as a mark of satisfying authenticity in the AI, as well as of complexity and emergent results -- not predictable in advance, but understandable and even obvious in retrospect.

Component Interactions

The question then becomes, how many connections are enough but not too much? While we do not have a particular number or ratio we can definitively state, we do have some useful, directional heuristics and a potentially useful tool.

Consider using a Component Interaction Matrix (CIM) in defining game objects (Figure 5). This is a matrix that lists every object in the game -- or alternatively, every attribute, or every grouped-object -- in both rows and columns. The interactions between one object and all others are marked in each row of the matrix. The goal in game design terms is not to have every object interact with all others; that would in effect over-specify the matrix, resulting in too many connections and chaos. On the other hand, if the matrix is too sparse, and there are too few connections between objects, then there is insufficient complexity in the matrix and the results of all interactions are trivially predictable. In this case no interesting behavior or degree of complexity is likely to result. In particular, if the interactions between objects do not form loops (as described above, A > B > C > A), then no emergence will result (and in effect, no system has been created).

		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
Radiator	A		X														
Engine Fan	B	X				X											
Heater Core	C																X
Heater Hoses	D																
Condenser	E						X		X								
Compressor	F					X			X	X							
Evaporator Case	G																X
Evaporator Core	H					X	X			X							X
Accumulator	I						X		X								
Refrigeration Controls	J																
Air Controls	K																
Sensors	L																
Command Distribution	M																
Actuators	N																
Blower Controls	O																X
Blower Monitor	P			X				X	X							X	

Figure 5: A Component Interaction Matrix

How many connections is “enough” to create complexity sufficient for emergence? While we do not have firm numbers, it seems that if N is the number of rows in the matrix, then just about half (N/2) per row may be optimal for enabling emergence. This number is clearly a heuristic and should be treated tentatively. However, it’s interesting that this is not only consistent with experience, but it is close to the “edge of chaos” lambda value of 0.55 mentioned earlier. It’s also reminiscent of the general case of intransitive systems such as rock-paper-scissors or, with a bit of expansion, rock-paper-scissors-lizard-spock (Figure 6). In games such as this, every object beats (affects) two others and is beaten (affected) by two others. That is, each interacts with (N-1)/2 others in unidirectional interactions. Games such as this can be extended indefinitely; examples exist with up to at least 101 individual components (always an odd number so that the (N-1)/2 formula yields an integer).

Creating “enough” effects between objects in the game performs several other important functions as well. This ensures enough connections for loops to form, as required to make a complex system. As such it creates not only explicit links (from A to B) but implicit ones (from A to D by going through B and C). This maintains the locality of effect described earlier -- every object does not “touch” or directly affect all others -- while creating a spread of effect throughout the system sufficient to create emergence. Finally, this many connections reduces the probability of both an “I win” object, one object that interacts with or beats all the others, or an orphan object that interacts with few if any other objects in the game. Both of these are common problems in unbalanced games that preclude any degree of emergence.

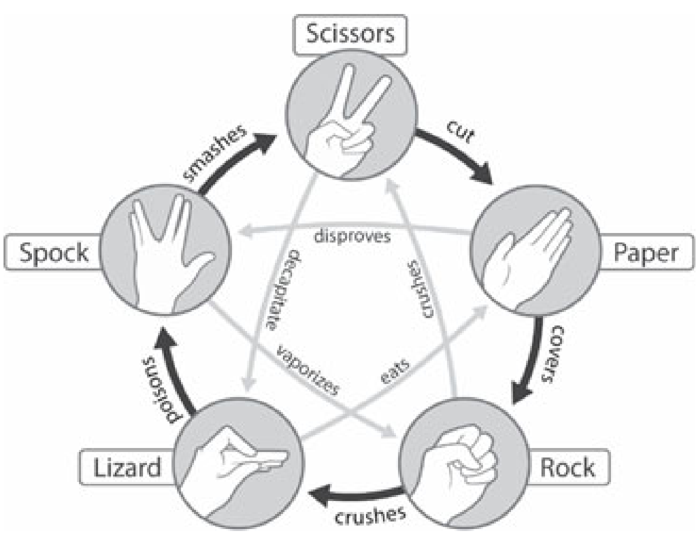

Figure 6: Rock Paper Scissors Lizard Spock

Creating Hierarchical Emergence
The Component Interaction Matrix and the specification of interactions between objects is also useful in another way. As mentioned earlier, emergence may appear at multiple levels in a game. The more levels

the player can perceive and incorporate into their mental model, the more depth the game will appear to have. And, as each level of emergence creates an integrative whole, a new object at that level, this reduces the cognitive load on the player, allowing them to maintain a larger, deeper, more complex model without overtaxing their mental resources.

Using a CIM to define the lowest-level interactions, such as between different kinds of resources, different kinds of attacks and defense, or other attribute-level game objects, is a good place to start. But it is then possible to take aggregate objects -- different kinds of potions, weapons, business strategies, emotions -- each of which is represented in attribute form in the lower-level CIM, and create a next-level-up CIM mapping the interactions between these aggregate components. There is no inherent limit to this: types of attack and defense aggregate into types of characters in an RPG, which aggregate into types of parties, and then into types of guilds, types of nations, etc., with wholly new interactions (verbs) mapping in their own CIM at each organizational level.

That may be difficult to contemplate in terms of explicit game design. It may also be, however, that by creating an explicit lowest-level CIM with just the right objects, verbs, and edge-of-chaos number of interactions, that layers and layers of implicit CIMs naturally emerge. This is what appears to happen, for example, in the game of Go: the game's explicit interactions are entirely simple and abstract. But they are so finely balanced that they result in the emergence of new implicit interaction matrices at ascending levels of complexity, where the complexity is represented in terms of the stones and the voids between the stones on the board. The depth of these matrices appears to exceed that which the human mind can handle, which is part of the lifelong fascination for many (and part of the surprise and fascination at the best AI Go player showing human players new levels of emergence and in effect entire new ways to play this game at its highest levels). This is an edge case of course, but a powerful one nonetheless.

Examples

Finally, here are some examples of games within our three dimensions of Agency, Abstraction, and Complexity. As a group we scored a variety of games on each of these dimensions as shown below in Figure 7. The diagram is a sort of 3D star field diagram, with Complexity shown above and below the zero plane. The placement of these games are illustrative, and serve as examples of games that tend to create emergent experiences, or not, based on their position within the emergent space envelope.

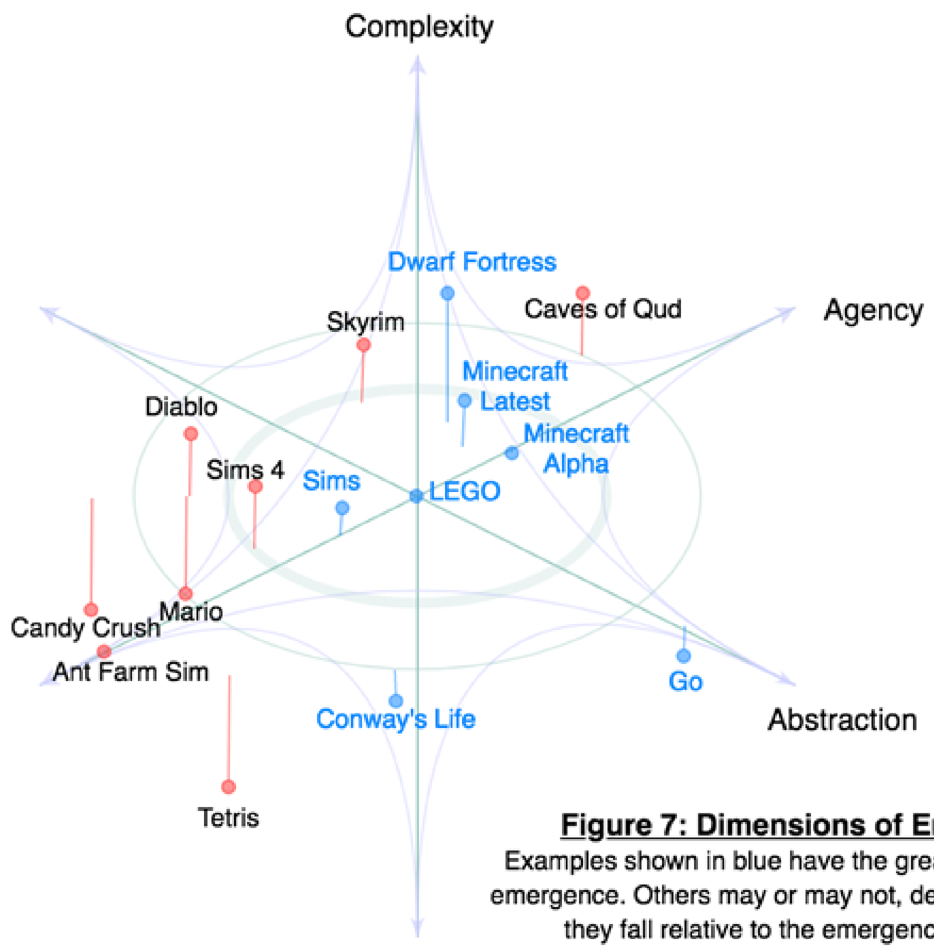

Figure 7: Dimensions of Emergence
Examples shown in blue have the greatest potential for emergence. Others may or may not, depending on where they fall relative to the emergence envelope.

For those interested, here are the values we assigned for each game: (A = Agency, B = aBstraction, C = Complexity)

- Game of Life (A-8, B8, C-1)
- Ant Farm Sim (A-10, B0, C0)
- Go (A0, B10, C-1)
- Mario (A-5, B-5, C-6)
- Minecraft Alpha (A4, B0, C0)
- Minecraft Latest (A4, B-2, C3)
- Caves of Qud (A10, B-4, C4)
- Dwarf Fortress (A4, B-3, C6)
- The Sims (A-2, B0, C1)
- The Sims 4 (A-4, B-2, C4)
- Skyrim (A4, B-5, C3)
- Legos physical product (A0, B0, C0)
- Diablo (A-5, B-5, C4)
- Candy Crush (A-8, B-5, C-7)
- Tetris (A-8, B2, C-7)

Conclusion

Emergence remains an intriguing and conceptually difficult topic in games and in general. We believe that viewing games within the space defined by the three axes of Agency, Abstraction, and Complexity, and specifying object interactions via a Component Interaction Matrix (or multiple hierarchical ones) will help demystify emergence and sharpen the designer's ability to create it.

There is a great deal more to be done in terms of being able to reliably construct emergent gameplay. Finely tuning each of these dimensions, in particular the underpinnings of complexity via a game's CIM, will help game designers more reliably create emergence. There is also work to be done in mapping this more effectively and more certainly to the player's mental model of the game, and especially to the narrative of their journey through the game that they construct as they go. As we come to understand the subtleties of constructing emergence, we may be better able to create emergent narratives that include engaging moments that are emergent rather than static, and that thus increase the player's sense of meaning in their gameplay experience.