4 Analyzing Games

I've said that uncertainty is a key element of games, and that uncertainty can be found in games in many ways. To gain a better understanding of how games exploit uncertainty, and how they generate it, let us examine a series of games in search of their sources of uncertainty. Once we have done so, we will perhaps be better equipped to categorize the types of uncertainty in games, to identify uncertainty in new games, and perhaps even to understand how and why some games succeed and others fail.[1]

Super Mario Bros.

Super Mario Bros. (Miyamoto, 1985) seems a good place to start, both because of its importance to the field and its huge influence on a whole generation of game designers—and because, at first glance, you might be hard put to find any source of uncertainty in the game.

When you begin the game, you see a small figure—Mario—standing under a sky. Attempting to move to the left does nothing. Moving to the right scrolls the world. There is no uncertainty about where to go; indeed, throughout the game, there is none.

You need to go right. You may at times move left to avoid enemies or the like, but completing each level requires rightward motion, and any leftward motion is purely tactical.

You quickly learn that the goal of the game is to avoid dying, and to complete each level within a time limit. The time limit is set by the game and is invariant. Completing a level requires you to face three kinds of challenges: those of navigation, enemies, and traps.

Navigational challenges are all ultimately challenges of timing; you must time a leap to jump over obstacles, or move onto or off of moving platforms at appropriate moments, and so on. All of these navigational challenges are entirely nonrandom, and simple in conception—perhaps less simple in execution. As a player, you may face some *initial* uncertainty as to what you will encounter and what you need to do to surmount these challenges—that is, you are not yet familiar with the level layout—but you quickly lose that uncertainty. Level layouts are invariant; there is no randomness or other algorithmic generation in the challenges you face. They are the same every time. They are uncertain only when first encountered.

Similarly, each type of enemy obeys simple, easily recognizable patterns of behavior. Different enemies have different behaviors—some simply move to the left; some move toward you, whichever way that requires them to move; some shoot at you—but all enemies of a particular class behave in identical ways, and you quickly learn their patterns. The specific tasks you need to accomplish to evade or defeat each enemy are uncertain only at first, but you quickly learn the drill. Once you have done so, there is no uncertainty about what enemies will do.

Nor is there any uncertainty about the effect of enemies upon you; enemies kill you. There is no combat system per se, although you can defeat many enemies by bouncing atop them. No dice are rolled, no complex algorithms applied to determine the outcome of contact with enemies: You die.

Similarly, all traps of the same type appear the same, all behave according to the same rules for traps of their type, and all can be overcome via simple rules. Traps must be avoided or evaded; otherwise, they kill you. There is no uncertainty to them.

The controls of the game are equally certain; indeed, *Super Mario Bros*. is notable for the crispness, responsiveness, and predictability of its controls. Motion is at an invariant speed—or rather, two invariant speeds: holding the B button while using the D-pad moves you faster, but at an invariant faster speed. Similarly, jumps are of predictable heights and durations, with a regular jump and a "higher" jump triggered by holding the A button. There is no randomness, no uncertainty, no variability to the controls; they are simple and intuitive.

There is almost no uncertainty as to path, either. While some levels have "secret areas," and some allow the level to be traversed in two or more ways, the branch-points at which the player may choose one path over another are few. And even when these exist, each path is always the same, every game you play.

In short, almost all of the ways that other games create uncertainty are completely or almost completely lacking in *Super Mario Bros*. The player's path through the game, with minor variations, is determined. The outcome is equally determined—with perseverance, the player will win. Moment-to-moment gameplay will vary relatively little between sessions of play.

Yet it cannot be denied that *Super Mario Bros.* is a superb game, practically the epitome of excellence in sidescroller design. Patrick Curry has gone so far as to declare: "Everything I know about game design I learned from Super Mario Bros."[2]—a statement that is remarkable, given how much an outlier this game, and sidescrollers in general, are in the universe of all games.

Playing *Super Mario Bros.*, unless you are a sort of sidescrolling Zen master, is, despite its somewhat lockstep nature, still a tense experience. An enemy is approaching, and you must jump at *just the right moment* to land atop and dispatch him. A leap to a moving platform must be timed *just so*. Enemy missiles must be ducked at exactly the right time. Success at *Super Mario Bros.* depends on hard-won interface mastery and a sense of the rhythm of the game—and combining those hard-won skills to master challenges of increasing density and complexity as the game goes on.

In other words, in *Super Mario Bros.*, the uncertainty is in *your performance*—in your ability to master the skills of hand-eye coordination demanded by the game and apply them to overcome its challenges. You can—and typically do—fail many, many times before "beating" the game. Mastering, and overcoming, the uncertainty posed by your uncertain skills, and your uncertain ability to maintain concentration and focus, is the heart of the appeal of the game.

Super Mario Bros. is practically the Platonic ideal of what game designers call a "player-skill" (as opposed to "character-skill") game. Luck is not a factor. Strategic thinking is not relevant. Puzzle solving is rarely germane. Success is virtually 100 percent dependent on your mastery of the controls, and your ability to respond to the situation unfolding on your screen with accuracy and alacrity. *Super Mario Bros.* has the same kind of elegance, simplicity, and purity we see in games like *Chess* or *Soccer* or *Diplomacy*; it is a game stripped down to a single set of challenges and carefully honed to the essential minimum. *Super*

Mario Bros. is the example par excellence of the game of performative uncertainty.

The Curse of Monkey Island

The Curse of Monkey Island (Ackley and Ahern, 1997) is one of the finest graphic adventures ever published. The third in a series featuring the protagonist Guybrush Threepwood, all set in a somewhat romanticized and fantastical Caribbean of the pirate era, its gameplay is centered on inventory puzzles.

The Curse of Monkey Island was published once CD-ROMs were widely established and PC games were capable of supplying extensive voice acting, music, and animation—but before 3D graphics became mandatory for retail-release titles. 3D is problematic for graphic adventures, which require you to identify and interact with on-screen objects; in a 3D environment, objects can readily be obscured or difficult to find given the problems of camera control in a 3D space. Thus, *Curse of Monkey Island* was published at the moment when graphic adventures were at their most appealing and before their precipitous decline.

As with adventure games of both the text and graphic variety, excellence in dialogue, language, and storytelling are central to the appeal of the game, but these are, of course, matters largely extraneous to gameplay per se. Like other adventure games, *The Curse of Monkey Island* is quite linear; you begin at a single location and must solve a set of problems to unlock others. At some times, you have access to puzzles at several locations and may solve them in variable order, but the approach is that of "beads on a string": within a bead, you have some choice of where to go and what to do, but once you have progressed to the next bead, opportunities previously available are no longer around. In short, while there may at certain points be uncertainty or choice in terms of navigation, this is extremely limited, and more illusion than reality.

Advancing in *The Curse of Monkey Island* involves three kinds of activities: minigames, "insult sword fighting," and inventory puzzles. Minigames are typically arcade-style games dependent on timing. Insult sword fighting is a system whereby you engage in a "sword fight" with another character, but success depends on countering any line of dialogue spoken by your opponent with a witty retort (e.g., "You fight like a dairy farmer," countered by "How appropriate! You fight like a cow!"). You "learn" the appropriate responses from low-level battles in order to triumph in higher-level ones. There is actually no uncertainty initially, because until you learn the correct retort, the menu does not offer it to you. Later on, the only uncertainty is in whether or not you can remember (or deduce) the correct response from those offered—a form of the game *Memory*.

As with most adventures, the main challenges are in the form of inventory puzzles. For example, at one point, you are swallowed by a snake. The snake has previously swallowed many items, which you may grab. Among them is pancake syrup. An item you will have previously picked up is an ipecac flower. By combining the ipecac flower with the pancake syrup, you create syrup of ipecac. Using it on the snake's head causes the snake to vomit you out.

Inventory problems, then, involve several features: identifying in-game objects that you can add to your inventory; identifying which can be combined with others to produce items of use; deducing a semi-logical solution to the puzzle you are posed; and, quite often, having some knowledge exterior to the game that elucidates the puzzle (syrup of ipecac is used to induce vomiting, and is often kept in the medicine cabinets of the parents of young children, in the event that the children will consume something poisonous, for which induced vomiting is an appropriate treatment).

Invariably, the number of inventory items is finite, and the potential combinations are finite, so that puzzles can often be solved through brute force (try everything with everything); but the fiero moment in games of this kind is when you figure out a solution and slap your head at the thought that you should have deduced it long ago.

Some adventure games fail because the solutions to their puzzles are obscure or illogical, or because inventory items are difficult to find (the "hunt the pixel" problem—something that the "hidden object" genre of puzzle games actually makes central to gameplay). But when an adventure game is well designed, the puzzles dovetail with the story, they are challenging but not impossible, and their solutions seem obvious and plausible, at least in retrospect.

In *Curse of Monkey Island* and other adventure games, there are no random elements, no complexity of system that makes achieving victory a challenge; there's no "player skill," no need to master physical skills to overcome obstacles. The challenge is entirely mental—solving the puzzles—and the uncertainty lies wholly in your uncertain ability to do so.

Crawford, in *The Art of Computer Game Design*,[3] questions whether adventure games are games at all; he holds interaction to be central to games, and static puzzles such as Sudoku or the crossword are not state machines. That is, they do not respond to player actions; they are wholly static. They have fixed solutions, but are not "interactive" in any meaningful sense. He would, by extension, categorize all puzzle-based games in the same fashion: as lacking in true interaction, static, and therefore puzzles, not games. To my mind, this is a stretch; while the solutions to *The Curse of Monkey Island*'s puzzles are invariant, the application as a whole is itself a state machine, with the options available to the player, the areas of the world open to exploration, and the position along the story arc all a function of interaction between the game and the player. It may not be as deeply interactive as *Go* or *Quake* (American McGee, Sandy Petersen, et al., 1996) but it is interactive enough to qualify as a "game," and indeed adventure games are conventionally taken as such.

Curse of Monkey Island is one example of a game for which uncertainty lies in the challenge of puzzle solving, but it—and indeed, adventure games as a whole—are far from the only sort of game that depends on this. Puzzles are the source of uncertainty in games as diverse as *Portal* (uncredited, 2008), *Lemmings* (Jones, 1991), and *Deadly Rooms of Death* (Hermansen, 2002)—games that rely on very different styles of puzzle solving from *Monkey Island*, but each of which offer one or a handful of possible solutions to a series of puzzles, with the tools necessary to solve those puzzles provided to the players, and with the solutions sometimes requiring torturous cogitation and experimentation.

If each puzzle has a single (or a handful of) solution(s), where is the uncertainty? It lies, as in *Super Mario Bros.*, in the player's performance: in this case, not in the performance of physical tasks but of mental ones. Yes, if the player is to advance, he must certainly solve the puzzle; but presented with a new and daunting one, he has some uncertainty about his ability to surmount the challenge. If the puzzle were trivial and easily solvable, if he were in no uncertainty, the game would not hold his interest. For many players it is, to be sure, a source of comfort to know that, in the vastness of the Internet, there is undoubtedly somewhere a walkthrough for even the most obscure of games, and that should they utterly fail to solve a puzzle, they can still find the means to solve it; but the challenge, and the uncertainty, still lies in their ability to do so without aid.

This is perhaps a *kind* of performative uncertainty, but I prefer to reserve the term "performative uncertainty" for games that pose challenges relating to physical rather than mental performance. For this kind of challenge, I favor the term "solver's uncertainty." In the world of crossword, logic, and other nongame puzzles, those who enjoy solving puzzles are called "solvers," and those who create puzzles are called "constructors." Or to put it another way, with games we talk of players and designers; with puzzles, we talk of solvers and constructors. Thus, "solver's uncertainty" seems an appropriate way to describe the uncertainty caused by the challenge of puzzles in a game.

I feel impelled at this point to embark on a tangent that has nothing to do with the central argument of this book, but that I feel is important and meaningful. Many students (and designers) of games are fans of the work of Csikszentmihalyi,[4] and feel that games ideally induce in player a sense of "flow," as Csikszentmihalyi defines it: an almost ecstatic feeling of action, reaction, and mastery in which time is lost and a feeling of creative impulse suffuses the person in question. Much time and effort is spent in trying to create games that induce a "flow state." I would suggest that while this may be desirable for some games, it is far from desirable for all—and that many games benefit precisely from *jarring* the player *out* of any sense of flow. Puzzle games are one example. Upon completing one puzzle and encountering the next, a player of this sort of game is not likely to feel "I am in the zone, I am the master of this, I react and do the next thing with preternatural ease"—rather, he is likely to think "Holy crap, what do I do now?"

That is, he is immediately jarred *out* of anything like a flow state and forced to grapple with new problems, to *think* about what he must do next.

I would suggest that rather than striving always to sustain flow, a designer ought to, from time to time, purposefully *break* flow—for his or her own artistic purposes, of course.

FPS Deathmatch Play

Conventionally, we view the first-person shooter (FPS) as a single genre, but gameplay is quite different in multiplayer mode than in soloplay mode. That is, both versions of the game may rely on the same set of controls, and on the same technology to display the gameworld; both may rely on a first-person view and on ranged combat as the player's main tool for overcoming challenges; but the actual concerns of the player, and his goals, are greatly different in the two gameplay modes.

In a soloplay FPS, you must traverse a series of levels, facing a sequence of monsters, traps, and physical obstacles within each level. In other words, the gameplay is quite similar to that of a conventional platformer; winning means traversing all the levels. The main difference is that instead of jumping, mostly you shoot things. You might go so far as to say that *Doom* (Green, Petersen, and Romero, 1993) is the same game as *Super Mario Bros.*—obviously, a vast oversimplification, but a core truth is there.

In deathmatch mode, however, you play within a single, defined physical space, not a linear sequence of levels. Other players also exist in the space; your goal is to shoot them, and vice versa. Death is rapid and instantaneous (none of the slow reduction in hit points typical of RPGs, though it may take several shots to kill you), but the dead quickly respawn. Typically, the game lasts until one player achieves a certain number of kills, but regardless of victory condition, deathmatch play lasts for only a few minutes.

Games generally offer a wide variety of arenas, and since FPS games are often easily moddable, many fan levels are available as well. A "level" consists not merely of a certain architectural geography, but also the placement of weapons, ammunition, health packs, and similar items within the level, which respawn at some time after being picked up. This placement, along with the architecture of the level itself, creates the strategy of the game: certain locations give good sightlines, or are most defensible; players come to learn where resources are located, and they compete to gain them and deny them, at least temporarily, to enemies.

While a game may support many arenas, there is little or no uncertainty in the design of the arena itself. That is, it may take a few play sessions for a player to get accustomed to a new arena, but players do not view arena novelty as a draw; on the contrary, they are drawn back to play in the same familiar arenas many times over. The reason for this is readily understandable; this is a game of skill, and players wish to feel that they win or lose on the basis of skill, not because of factors they cannot control—such as unfamiliarity with the level's layout. In other words, in practice, uncertainty does not lie in level design. Where then does it lie?

Not in randomness, though unbeknownst to most FPS players, their games do contain a small random factor: FPS weapon damage is typically random within a set range, an element originally introduced with *Doom*, the ur-game of the genre, by John

Romero, who specifically wanted players to be less than 100 percent sure of success with each shot.[5] But the level of uncertainty this causes is slight; it rarely alters the number of hits required to dispatch an opponent and is rarely perceptible to the players.

Clearly, there is a level of performative uncertainty: aiming precisely and timing a shot is difficult, particularly given moving and unpredictable targets. There are problems involved in obtaining sufficient ammunition and access to the best guns and in taking advantage of cover and lines of fire. And finally, there are the tricks of the trade, techniques such as the bunny hop (repeatedly jumping in order to present a more difficult target to opponents) and the rocket jump (setting off an explosive underneath yourself in order to propel yourself higher than permitted by a normal jump), which if properly performed can provide a tactical advantage.

In other words, deathmatch FPS games are very much games of player skill, so much so that online play is often a humiliating experience for new players, who will be repeatedly slagged by more experienced ones until they develop competitive chops. This is, in fact, one of the major design flaws inherent in the genre, which some games attempt to ameliorate with systems to match players by skill.

But there's an aspect to deathmatch play that we do not find in games like *Super Mario Bros.*: opposition is provided not by characteristics of the application itself—programmed enemies, traps, and obstacles—but by live players. The "performative uncertainty" involved in hitting a target is partly based on the uncertain skills of the shooter but also on the unpredictable actions of the target. That is, even with perfect skill mastery, a player may find himself outwitted by an opponent; and you are at least as likely to miss a shot because your target suddenly takes off in an unexpected direction as because your aim is poor. This is in stark contrast to the enemies of *Super Mario Bros.*, which always appear at the same locations in a level and always behave in predictable fashion.

The simple fact is that people are smarter than machines; most computer-controlled opponents are highly predictable. The industry uses the term “artificial intelligence” (AI) with gay abandon, but even “artificial stupidity” would be a stretch; the algorithms used are rarely complex and constitute nothing a genuine AI researcher would deem remotely interesting. Actual players are always far more unpredictable, tricky, dangerous, and interesting than artificial opposition.

In short, deathmatch FPS games are very much games of player skill, and thus of performative uncertainty; but there is another major source of uncertainty in games of this type—player unpredictability.

In a way, that should be obvious; any game with more than one player involves some uncertainty about what that player will do. But in actuality, some multiplayer games have relatively low levels of player uncertainty—certainly true of, say, *Tic-Tac-Toe*, at least with sophisticated players, since strategies then are predictable. *Monopoly*, which we'll discuss later, is another example of a multiplayer game with low levels of player unpredictability.

Rock/Paper/Scissors

Rock/Paper/Scissors (*R/P/S*) is, of course, a classic folk game, played by two, with trinary simultaneous moves and a rotational victory vector (Rock beats Scissors which beats Paper which beats Rock).

At first glance, it appears almost as brain-dead a game as *Tic-Tac-Toe*. A four-year-old might think there's some strategy to it, but isn't it basically random?

Indeed, people often turn to *Rock/Paper/Scissors* as a way of making random, arbitrary decisions—combat resolution in *Mind's Eye Theatre* (uncredited, 2005) or choosing who'll buy the first round of drinks, say. Yet there is no quantum-uncertainty collapse, no tumble of a die, no random number generator here; both players make a choice. Surely this is wholly *non*random?

All right, nonrandom it is, but perhaps it's arbitrary? There's no predictable or even statistically calculable way of figuring out what an opponent will do next, so that one choice is as good as another, and outcomes will be distributed randomly over time—one-third in victory for one player, one-third to the opponent, one-third in a tie. Yes?

Players quickly learn that this is a guessing game and that your goal is to build a mental model of your opponent, to try to predict his actions. Yet a naïve player, once having realized this, will often conclude that the game is still arbitrary; you get into a sort of infinite loop. If he thinks such-and-so, then I should do this-and-that; but, on the other hand, if he can predict that I will reason thusly, he will instead do the-other-thing, so my response should be something else; but if we go for a third loop—assuming he can reason through the two loops I just did—then . . . and so on, ad infinitum. So it is back to being a purely arbitrary game. No?

No.

The reason *Rock/Paper/Scissors* is not a purely arbitrary game, and the reason that an excellent player will win more often than chance would predict, is that human psychology is *not* random, and some behaviors are—not necessarily predictable, but likely to occur more often than chance would dictate.

One heuristic of experienced *RPS* players is "Losers lead with Rock." This is demonstrably true; naïve players will lead with Rock more often than one-third of the time. Your hand begins in the form of a rock, and it is easiest to keep it that way. The name of the game begins with "Rock," and if you are mentally sorting through the options, it is the first one that will occur to you. And the word "rock" itself has connotations of strength and immovability. These factors lead players to choose Rock on their first go more often than chance would dictate. An experienced player can take advantage of this. Against a player you know to be naïve, you play Paper.

Similarly, players rarely choose the same symbol three times in a row, and almost never four times; it *feels* wrong to human psychology. An extended streak feels nonrandom and unlikely, even though in a purely random game, each new throw is stochastic, not dependent on the outcomes of previous throws. Thus in a truly random game, no matter how many times "Paper" has come up in a row before, there is a 1 in 3 chance of it coming up again. Given the nature of human psychology, if Paper has come up twice, there is a far less than a 1 in 3 chance that the player will choose it again.

Even players who know this have to consciously try to overcome their bias against streaks—*particularly* if they lose with one gesture on the previous round. If you have played Paper twice in a row, and lost the last time you played, the human instinct is to try something different, and thus players will at that point choose Paper far less than one-third of the time.

In short, a player who has studied the game will unquestionably win more than chance would dictate against a naïve player, because he understands how human psychology is likely to affect the choices of his opponent. Of course, two players who both understand these factors are on a more even plane; but even here, there is the factor of human readability. It is hard to maintain a perfect "poker face," and some are better at it than others. Some are better at noticing subtle cues in the expressions or body language of others. These skills are not always sufficient to *ensure* triumph, but they do produce a bias in favor of those more observant—and more socially adept at reading others.

In other words, at first glance *Rock/Paper/Scissors* appears to be a guessing game, with victory going to the player who can outguess his opponent; at second glance, it appears to be purely arbitrary; and at third glance, the original supposition is justified. It *is*, in fact, a guessing game with victory going to the player who can outguess his opponent, but there are strategies to "outguessing."

Where is the uncertainty in *Rock/Paper/Scissors*? That should be obvious. It is in the unpredictability of opposing players. In fact, that is *all there is* in *Rock/Paper/Scissors*; an FPS played in deathmatch mode may rely to some degree on player unpredictability, but it also relies on player performance. *Rock/Paper/Scissors* is a game of player unpredictability in its purest form, for this single factor is the sole determinant of the game's uncertainty, its raison d'être, and its cultural continuance.

Diplomacy

Diplomacy (Calhammer, 1959) is a seven-player boardgame that ostensibly represents military conflict in Europe during the early twentieth century, but in fact is an abstract strategy game with a diplomatic/military theme. The board is of Europe, Asia Minor, and part of the Maghreb, divided into provinces. The borders are those of 1914 (though the first turn is called "Spring, 1901"), and the players represent the European great powers of the time: Austria, England, France, Germany, Italy, Russia, and Turkey. *Diplomacy* is both a beautifully designed game, with clear subtext and enormous polish and a historically important one: it is the first truly "diplomatic" game, by which I mean one in which negotiation and alliances are of vital importance during play.

About one-half of the provinces are "supply center" provinces; each of the players begins with three (four for Russia), and all of the nonplayer countries (Sweden, Spain, and so on) are supply centers as well. Each center can support one unit, either an army or a fleet, and a province may be occupied by only one unit at a time. After two moves ("Spring" and "Fall"), players add up the number of supply center provinces they now control, and either build or remove units so that they have as many in play as the number of such provinces they own, with new units built only in home supply center provinces.

Diplomacy is a simultaneous movement game; players write their orders secretly, and all are revealed at the same time, with the results then adjudicated. The key to the game (the "core mechanic" in Salen and Zimmerman's terminology[6]) is the "support order." Each unit may move only one province, in terms of distance; a unit holding still in its province prevents another unit from entering that province, and two units attempting to enter the same province "bounce." However, a unit may hold in place and "support" a move into an adjacent province, giving that move the power of 2, sufficient to dislodge an unsupported unit or to occupy the province in the face of an unsupported opponent attempting to enter it. One key element: you can support the moves of other players.

Because all the players are of roughly equal power,[7] no player can reasonably expect to win the game alone. To overcome another power, you need an ally. And because units may support moves by other players, alliances are effective.

However, the simultaneous movement nature of the game has another effect: it encourages backstabbing. You can never be certain that, on the next move, your ally will do as he has promised. In addition, the winner of the game is the player who possesses half the supply centers on the board—and if your alliance is successful, it is likely that at some point your ally will become your closest competitor. Winning not only requires you to form alliances—it also, more often than not, requires you to backstab your ally, at just the right time.

The subtext of *Diplomacy* is *realpolitik*—or more specifically, Lord Palmerston's rubric that nations have no permanent friends and no permanent enemies, only permanent interests. As such, it is an amazingly pointed, cynical, and even fascinating game—fascinating, too, for its demonstration that games, even ones relatively simple in terms of their rules, can hold ethical lessons.

The rules of *Diplomacy* are completely deterministic: there is no randomness in its system of combat resolution. It is a game of perfect information: everything on the board is visible at all times to all players. In other words, there is *no* uncertainty in *Diplomacy—except* that you never know what moves the other players are writing down, whether you can rely on the assurances of the others, whether your alliance is strong or others are weak.

While there is a strategic element to the game—planning moves takes thought, and there's a small literature devoted to the strategy of opening moves and the formation of stalemate lines—the primary determinant of victory is not mastery of strategy, but the ability to persuade and bamboozle. If you are a master of *Diplomacy*, you can unquestionably have a career in sales.

In other words, as with *Rock/Paper/Scissors*—and for reasons evoked by very different mechanics—the uncertainty in *Diplomacy* derives almost entirely from the unpredictability of the other players, and the player who best manages that unpredictability will be its winner. Hidden information, in that you do not know the moves of others until you have committed to your own, also contributes to the game's uncertainty.

Monopoly

From our discussion of *Diplomacy* and *Rock/Paper/Scissors*, you might conclude that player unpredictability plays an important role in *every* multiplayer game. In fact, this is not always the case.

Monopoly (Darrow, 1935) is a multiplayer game with a real-estate development theme that, despite high complexity (by mass-market standards), has become a perennial.

It's a familiar enough game that I will not recap its essential gameplay.

In *Monopoly*, there is no direct way for players to injure one another. It is true that when one player lands on another's property, he must pay money to the property owner, which is an injury since the game is won by bankrupting your opponents; but there is no way to force a player to land on your property, or even to influence the chances that he will. There is almost no way to benefit another player to oppose a third, either—though it is possible to trade properties for mutual benefit, this occurs only rarely.

On your turn, you roll dice, move, and suffer or benefit from the consequences of the square on which you land. You have only a few choices to make during play—whether or not to buy a property when you land on it, when to upgrade your properties with houses and hotels, whether to stay in jail or pay to get out—and other players cannot force, prevent, or affect any choice you make.

Does player unpredictability play *any* role in *Monopoly*? Yes, but only to a very slight degree; you cannot predict what choices players will make, on the few occasions when they have choices, but since you cannot influence them, their decisions have little to no impact on your own decisions. *Monopoly* is a game of near-perfect information—the only lack of perfection is that you do not know what cards will come up next in the deck, but these cards have only a glancing impact on play.

There is some strategy to *Monopoly*, but only a very little, and the optimal strategy is easily stated: always buy the first property in a block if you are able, always buy the third if you are able, and always buy if by doing so you prevent an opponent from assembling all three properties in a block. Upgrade to three houses as quickly as feasible, even if this means other properties remain without upgrades, since there is a substantial jump in rent between the second and third. Especially in the early game, always buy yourself out of jail if you can, because you will otherwise lose too many opportunities to assemble properties.

Because the strategy of the game is so simple—and invariant, there are no alternative strategies that become better under some circumstances—there is no real uncertainty in terms of the players' ability to master that strategy.

The uncertainty of *Monopoly* lies almost entirely in the dice. The player who is lucky enough to land on and purchase the better group of properties will almost certainly win. Luck overrides all other considerations.

This is fairly jejune in itself, but the reason *Monopoly* is ultimately dull is its excruciatingly extended endgame. By the end of the first hour, or at most two, of play, you can predict, on the basis of properties owned, who will ultimately win the game, with only a small margin of error. Yet to actually play the game to conclusion, you must play on for hours on end, slowly whittling away at the holdings of inferior players until they, at great length, are forced into bankruptcy. This is not a clever design.

Our point remains: *Monopoly* is multiplayer, but only peripherally affected by player unpredictability. Almost all of the game's uncertainty lies in the unpredictability of dice and, to a lesser degree, card draws—random elements. Only the theme and the paraphernalia of properties and houses and hotels and money hides from the players that they might as well be playing *Roulette*, and without the real-money risk and reward that makes *Roulette* of interest to those who play it. The appeal of *Monopoly* lives in its color, and not in its gameplay.

Chess

Chess is the abstract strategy game par excellence, a direct conflict of mind to mind, devoid of randomness, with an extremely simple rules set that nevertheless evokes enormous strategic depth.

Clearly, as a game that pits two players directly against one another, with gameplay dependent on capturing each other's pieces, player unpredictability is a main source of uncertainty. Given the lack of any kind of hidden information, you might expect that it was the only one; but in fact, there is another and deeper form of uncertainty.

To an external observer, a *Chess* game is an arrangement of pieces of different shapes on a square grid. *Chess* players themselves see it differently; each piece projects force across the squares to which it may move. *Chess* players are conscious of the interplay of forces, viewing the current gamestate, metaphorically, as rays of light and darkness extending across the board. In moving, they seek to alter the disposition of forces to their benefit, with an intermediate goal of forcing the opponent to sacrifice pieces and an ultimate goal of checkmate. In considering the next move, a player considers how it shifts the projection of force, how the opponent is likely to respond, and what the best response to that response would be.

To look at it another way, the game of *Chess* is an extremely complicated decision tree. In a given gamestate, the next player to move has a limited number of options, and each potential move can be seen as a branch from this node. Each move leads to a new gamestate, with each potential move of the opponent a branch from that gamestate. It's computationally possible, in principle, to parse the entire decision tree to find the optimal move (or a set of moves of equal optimality)—but impossible in practice, at least so far, because of the bushiness and complexity of the tree. *Chess*-playing programs typically combine an attempt to look ahead a few moves with a set of heuristics.

Human minds, of course, work differently from computers, and experienced *Chess* players don't make any effort to explore every potential branch of the decision tree; they have a gut understanding, fueled by study and the play of many games, that allows them to quickly dismiss most options and spend time exploring the more likely ones. In other words, the sort of pattern recognition that humans excel at, and which computers do not, is what allows humans to compete against computers at *Chess*—not always successfully, of course—despite computers' superior speed in calculation.

The point here, however, is that no *Chess* player—human or computer—can truly cope with the strategic depth of *Chess*, at least until the endgame when the reduction in the number of pieces prunes the decision tree drastically.

The uncertainty in *Chess* lies not solely in the unpredictability of the opponent's move, but also in the player's uncertain ability to cope with the sheer strategic complexity of the game. The difficulty of analyzing the strategic options is so high that the player is rarely certain of the consequences of his next move.

Or, in other words, *analytical complexity* is the primary source of uncertainty in the game of *Chess*. There is perfect information, there is no uncertainty in the ability to perform any particular move, and there are no random elements; in a game of lesser depth, the combination of these factors would limit uncertainty greatly, perhaps leaving as a residuum nothing but the uncertainty caused by player unpredictability, as in *Rock/Paper/Scissors*. But because grappling with the explosive branching of the decision tree in *Chess* is so hard, uncertainty survives despite the simplicity and clarity of the game in other regards.

Analytic complexity is a source of uncertainty in many other games as well—but *Chess* is perhaps the purest example.

Roller Coaster Tycoon

Roller Coaster Tycoon (Sawyer, 1999) is a sim/tycoon-style game in which a single player builds and operates an amusement park. As with most games of the type, the player's main activity is building things—rides, pathways, decorations, concessions, and the like. Income derived from the player's operations can be used to build more things, but initial capital is borrowed at an interest cost, so interest payments must be met and debts repaid.

While *Roller Coaster Tycoon* has a "sandbox" mode, the main game is a set of levels, each of which poses some challenge to the player—achieve some level of revenues, or some number of concurrent guests, and so on.

Patrons of the park are represented by small animations wandering the paths, waiting in lines, riding on the rides. Clicking on one shows something of the complexity of the underlying algorithms. Each patron's current state is represented by several values—happiness, hunger, nausea, energy, and money in pocket among them. Every few tics, each patron gets updated—happiness perhaps increased if on a ride, but nausea increased if the ride is intense, energy decreasing over time but refreshed by a rest at a bench, and so on.

Cynics have sometimes described sim/tycoon games as "animated spreadsheets," and there's some truth to this; under the hood, a set of fairly simple algebraic algorithms relates the statistics of rides to the statistics of patrons, moving patrons through the park, altering the player's revenues over time.

But if *Roller Coaster Tycoon* is a spreadsheet, it is not a simple one. As an example, intense rides cause some patrons to vomit. Unsightly vomit on the paths reduces the happiness of other patrons who encounter them. You may hire janitors, who move semirandomly about the park and remove vomit (as well as litter from concession patrons), but they require pay.

Clearly, unsightly vomit reduces the pleasure of guests, causing them to spend less time at the park and therefore spending less money; this also reduces the reputation of the park, meaning that the influx of guests will lessen, with a secondary negative impact on income. In other words, cleaning up the vomit costs money, but failing to clean up the vomit also costs money. There is some optimal number of janitors for any given park configuration, but figuring this out is difficult. The algorithms are not directly exposed to the players, but even if they were, they could not be solved directly, since this is, after all, a computer program, executed as a sequence of operations rather than a static spreadsheet whereby the change to one parameter causes an immediate and straightforward change to other values. For instance, whether a patron encounters vomit depends on where he is, where the vomit is, his velocity of motion, potential changes to the direction of his motion, and so on; these factors are not easily predictable ahead of time, though one might model the rate at which guests encounter vomit statistically.

Consequently, even though *Roller Coaster Tycoon* is not anywhere as strategically deep as *Chess*, and is real-time rather than turn-based, it shares with *Chess* a fundamental attribute: both games are of sufficient algorithmic complexity that the consequences of a player's actions are uncertain, because the player cannot grasp all potential ramifications of an action.

The problem of analysis is compounded by the fact that the underlying algorithms are not exposed to the player. For instance, the farther a patron must walk before encountering the next activity, the more tired he will become, and the more likely he is to leave the theme park before spending all his money. Some arrangements of park features will therefore be more optimal than others, because they will offer opportunities to rest at the right moments, and new opportunities for fun on a schedule that dovetails with the patron's energy budget. But there is no way to know which of two options is preferable a priori, because the algorithm for decrease of energy with distance is not known, and it is not known how this character stat affects other aspects of the game. Consequently, the best a player can do is to gradually obtain an intuitive sense of the workings of the system, and therefore make informed decisions based on gut feeling. This is a common factor of digital games: with tabletop ones, all aspects of the game are explicit and knowable, while with digital ones, many are hidden in code and not even in principle knowable to the player (short of decompilation and careful study of the code itself).[8]

Algorithmic complexity is not the only cause of uncertainty in the game; the game relies on randomness as well. Random factors govern the motion and choices of the patrons, the breakdown of rides, and the rate of guest influx, among other things. But the fact that the underlying algorithms harness randomness is not necessarily perceptible to the player; he simply knows that rides break down from time to time, and that it is necessary to employ some repairmen. Because of the number of guests, the random element of their behavior is also not perceptible to the player; while he may follow one guest closely at times, by and large he simply perceives a vast mass of patrons milling about and, antlike, engaged in behavior best viewed as a sort of statistical aggregate. To put it another way, as the number of random tests in a system approaches infinity, outcomes regress to a mean, and there are so many random tests here that players almost never see behavior that seems unusual. Randomness is being harnessed here for greater simulation fidelity, that is, as a way of representing factors over which the player does not have direct control, such as the behavior of his customers.

The *perceptible* uncertainty, therefore, and the cause of the challenge to the player's ability to master the system, derives almost entirely from algorithmic complexity and not from the random elements involved in those algorithms.

Poker

Poker is, of course, a gambling game, and there is a tendency to assume that all gambling games are games of chance—indeed, the legal definition of gambling assumes as much. In a game of chance, outcomes are largely or exclusively dictated by random factors—as with, say, *Roulette*. And certainly, there is a strong random element to *Poker*; the deck is randomized with each hand, and no player is permitted to know the order in which cards are arranged within the deck.

But from the perspective of uncertainty, *Poker* is actually quite a complicated game. Despite its relative rules simplicity, it possesses three distinct sources of uncertainty—moreover, those sources of uncertainty interact in quite interesting ways.

The first of the three sources, to which we have already alluded, is, of course, randomness. The second is hidden information—or its converse, partially revealed information. Consider the variety of *Poker* now most familiar to the world,[9] *Texas Hold'em*. At the inception of play, each player receives two "down" cards, that is, cards that only he is entitled to examine. Over the course of multiple rounds of betting, five additional "community" cards are revealed. The goal is to end with the best hand, by *Poker*'s somewhat complicated scoring system, using any five cards from among his own down cards and the common community cards.

Each player therefore has two data that the others do not: what cards he holds. Until the final round of betting, however, the player cannot know the true value of his hand, because any single new card may improve it. Indeed, in some cases, players remain in the game in the hope that, for example, the next card displayed will allow them to fill out a straight or flush.

Secondarily, the cards also tell the player that the others cannot hold the two that he knows he does; thus, if a player has a pair of Aces, he can be sure that no one else can have three, and that the likelihood of someone filling out an Ace-high straight is half the norm.

The player has no direct way of learning anything about the other players' cards; his only recourse to learn anything about them depends on his ability to read the expressions and body language of his opponents, or to learn from the manner in which they bet. Thus, if a particular player is betting heavily, one must conclude either that he believes himself to have a good hand or else is bluffing—and possibly his ability to read people may allow him to determine which is more likely. Similarly, over time, players may notice patterns of behavior in the actions of others, for example, that a particular player bluffs frequently or rarely does.

Players examine both their own and the community cards, taking into account the pattern of other players' bets, to attempt to model the thoughts of the others, in a fashion not utterly dissimilar from *Rock/Paper/Scissors*. That is, a player asks himself what the other players *might* possess. For example, if the community cards contain a pair, it is quite possible for another player to have a third card of the same type, and therefore trips—not a bad hand, particularly if the cards are high in value. Furthermore, players know which cards are in the deck, and can, on the basis of known information, calculate the likelihood that any particular hand is possessed. Some players, being fast calculators, in fact do so routinely, though others rely more on a gut intuition of probabilities, based on long experience with the game and many hands of play. Players combine an understanding of the odds, therefore, with theories about the mentalities of the other players to try to understand what is going on, and determine their own likelihood of winning.

The third uncertain factor of *Poker* is thus player unpredictability—a factor particularly brought out by the bluff. The bluff is possible because players may fold during any round of betting, so it is possible to win by getting another player to conclude that he will lose, and should therefore fold rather than risk any additional stakes—even if he would have won had he stayed in the game. You bluff by betting aggressively at some point, hoping that players will conclude you have a stronger hand than in fact you do. Contrariwise, you can never be entirely certain that a player's pattern of betting is motivated solely by the cards he possesses; he can, in effect, as in *Diplomacy*, be lying.

Because card distribution in *Poker* is purely random and, in *Texas Hold'em* at least, cannot be affected by the players in any fashion, the strategy of the game does not lie in card play as such, but in betting. The ultimate goal is not to win or lose any particular hand, but to wind up with the largest amount of money over time—thus losing frequently is acceptable, so long as monetary losses on lost hands are more than compensated for by gains on rarer ones. Even the best player can never be certain of achieving this goal, however; he might err in his assessment of the odds, or in his ability to accurately assess the other players—and, of course, he might simply be foiled by ill luck.

Games of pure chance, like *Roulette*, rely for their appeal on the tension of winning and losing something of real and tangible value—that is, money. No rational person, after all, would play *Roulette* for the fascination of the game itself—watching a ball rolling around a wheel might be fascinating to a cat, but not to a human being—and it is hard to believe anyone would play it for long for artificial rewards, for example, points or matchsticks. *Poker* is considered a gambling game because it relies partially on chance for its uncertainty—and also because it relies on the tension of monetary reward and loss.

But *Poker* is a far deeper and richer game than pure-chance games because randomness is not its sole source of uncertainty; hidden information alters the equation of the odds, and player unpredictability further complicates the situation. Mastering a pure-chance game, from a strategic point of view, is simple; the odds are calculable, and you always know what you are risking and what your reward may be. But mastering *Poker* requires far more—not just knowledge of the odds, but of how the odds mutate in the face of hidden information gradually revealed, and, more important, the ability to cope with and manage the complexity of unpredictable human behavior.

Rogue-Likes

Rogue-likes are named for the original game of the form, *Rogue* (Toy and Wichman, 1980), a text-based dungeon-crawling game originally available mainly on academic computers. The player controls a single character, with *Dungeons & Dragons*-like race, character class, level, and stats; as with conventional RPGs, of both the tabletop and digital variety, characters rise in level as they gain experience points, engage in combat with a variety of weapons (dying if hit points are reduced to zero), and gain equipment by slaying monsters and taking their treasures. The relationship to *D&D* is reinforced by the fact that players are exploring a literal 'dungeon,' a series of underground levels populated by monsters whose sole reason for existence is, seemingly, to be slain by adventurers who want their treasure.

The original *Dungeons & Dragons* contained a series of tables for use by gamemasters who did not wish to think too deeply about creating their dungeons; by rolling on these tables, you could create a (fairly lame) dungeon, generating monster types, number appearing, and treasure possessed. Another set of tables allowed you to generate "wandering monsters" you might encounter in the corridors between rooms.

D&D's random generation system is a poor substitute for an engaged and intelligent gamemaster, but Rogue-likes are a better substitute. They are experiences quite different from, and more engaging than, randomly-generated *D&D* levels. Rogue-likes take the notion of randomly generated content and run with it, creating algorithmically generated content of depth and character far beyond the simple patterns of *D&D*.

Rogue-likes are pure text games, with keyboard input; some more recent examples may have small illustrative tiles to replace ASCII characters, and recognize the occasional mouse click, but as a genre, Rogue-likes are the virtual antithesis of the conventional videogame, the creators of which are seemingly obsessed with ever more detailed and lifelike images.

The main tool Rogue-likes bring to creating interesting gameplay through randomness is algorithmic complexity far beyond that provided by a few 1D100 tables in the back of the *D&D* rulebook. In *NetHack* (Stephenson, 1987)—a particularly interesting example because of a multi-decade history of open source development by innumerable contributors—when you toss a ring down a sink (which is not a good idea, in general), the *sound* it produces (reported to you in text) indicates something about the type of ring you just threw away. That is, when you start a game of *NetHack*, it associates each type of magic ring that exists in the game (of healing, of fire resistance, etc.) with a random adjective (garnet, pearl, gold, etc.). You find a "gold ring" and have no idea what it does, and you are reluctant to put it on, lest it be one of the bad rings (cursed ring of levitation, say, which lifts you up against the ceiling and cannot be taken off). If you toss a ring down the sink, you have lost it, but may, if you have done this many times before—or read the source code, since this is an open source game—be able to identify what a gold ring does by the sound it makes.

This is an obsessive level of detail, possible only in a game of long development by many hands—but it's typical of Rogue-likes. Most of the content is algorithmically generated, meaning via a system using a random seed and lookup tables, though *NetHack* does have occasional "designed" levels with preestablished content. But there is so much variety in what these lookup tables can create, and so much attention has been paid to tiny nuances of gameplay, that even after many repeated plays, these games remain capable of surprising the player.

Randomness at this level has one strong advantage, as well as one strong disadvantage. The advantage is that no two Rogue-like games follow the same path. Indeed, paths are likely to diverge quickly and to a high degree, even if starting from identical conditions (that is, with characters of the same class and race). One character may quickly obtain an item of great use to an early character, such as a ring of healing or invisibility, while another may read a scroll and be cursed to lug around a ball and chain. Randomness, in other words, is harnessed to create enormous variety of encounter; while the universe of all encounterable things in the game is by nature limited, there is such variety, encountered in so random a way, that the player is always uncertain what they will encounter next.

The disadvantage is the flip side of the coin: players often find themselves in impossible situations. Bad luck with magic items or monster generation can often doom a player, regardless of how optimally the game is played. In part, this is a reflection of the permadeath aesthetic of the game; Rogue-likes are specifically designed so that a player will typically need to attempt several hundred dungeon descents before winning. And indeed, the humorous nature of the deaths they create—"killed by a bugbear on level 32 while paralyzed"—are part of the genre's charm.

Randomness is not, of course, chaos. Within a particular level range, only a certain set of monsters will be encountered; they become tougher the deeper a player delves. But exactly what the player will encounter around the next corner is always uncertain, because of the random nature of the game.

Videogame players generally claim to dislike "randomness," rarely being aware of how deeply many games rely on random factors for either the generation of variety or simulation value. Despite this, Rogue-likes have ardent fans—and, indeed, *NetHack* is one of two games (along with *Civilization*) that have been on the hard drive of every computer I've owned since I first encountered it.

Like *Poker*, Rogue-likes depend on randomness as the main source of uncertainty; but here randomness is harnessed for variety, not as a source of strategy.

CityVille

CityVille is, as of this writing, the single most successful "social" game, meaning a game played on social networks, with almost 100 million monthly players—making it one of the most widely played games in global history. It is also fairly typical of what we might call a “sim/tycoon lite” game—that is, of the same general style as richer sim/tycoon games like *Roller Coaster Tycoon* or *SimCity* (Wright, 1989), but with simpler user interface (UI) and gameplay more suitable to the “casual” audience that social games attract.

In *CityVille*, you build and place three basic types of buildings: municipal buildings, which increase the maximum population of your city; houses, which increase the actual population when built and from which money can be collected periodically; and businesses, which need to be supplied with “goods” and also produce money from “sales” to the population (though the game uses a statistical model rather than the simulation approach of classic sim/tycoon games).

Goods are obtained in a number of ways, including farming and via ships. Both systems require the player to select a period of time, from minutes to days, after which goods are collected; if not collected shortly after the appointed time, they “wither,” a system to encourage player return.

At first glance, there seems to be very little uncertainty in the basic game model. Stats are exposed explicitly to the players; they know how much population a building grants, how much income it produces, and on what schedule. The number of experience points required to level up is open, and even such data as the effect of decorations on nearby buildings (they increase monetary yield by a small percentage) is explicitly reported to the player, via a fairly intuitive UI.

In other words, you can treat the game’s economic model simply and algebraically; no math beyond the middle-school level is required, and there’s no randomness or other source of uncertainty apparent in the model. Unlike *Roller Coaster Tycoon*, you can represent the economic model with a fairly simple spreadsheet, and doubtless many players have.

And yet, those who denigrate this style of game often refer to them as "slot machines," implying that randomness rules. The reason for this is that when a player clicks on almost any entity to collect goods or money, reward icons (known in the industry as "doobers") spill out, slot machine-like, showing the number of experience points, money, and/or goods received. And there *is* a random element here: there's a chance of a slight bonus in terms of experience points, money, or goods—and also a chance that a "collectible" is received.

Collectibles, which players may view in a separate window, are grouped into "collections," and when all items in a collection are obtained, the collection may be "turned in" for a minor game reward. Those who view social games as slot machines maintain that the intermittent nature of these rewards is addictive, in a Pavlovian sense. That is, experiments have shown that an animal who always receives food when it pulls a lever will pull it only when hungry; an animal that receives a reward only intermittently will learn to pull it obsessively. Supposedly, the intermittent and semirandom nature of doober drops trains players to play obsessively.

Needless to say, game players are not chimpanzees, and the claim that this is a source of addiction is absurd; collectibles are a minor, charming draw—and in fact, their contribution to the game's uncertainty is equally minor. Collections *will* be completed over time and (at least in *Cityville)* completing a collection or obtaining a particular collectible is not critical to gameplay. (Some other social games make them more important.) After a time, players hardly notice which doobers are dropping. In other words, we need to look elsewhere for uncertainty in this type of game.

Doobers do produce another form of uncertainty, however; if you mouse over them rapidly and in succession to collect them, a "bonus bar" appears on the screen, with text beneath it becoming increasingly enthusiastic as you collect more. When you stop collecting, after a few seconds, you receive a monetary reward. If you collect only a few doobers, this reward is small, but if you collect many, it can amount to 3,000 or more in game money (soft currency), an appreciable reward. Generating sufficient doobers to get there requires you to plan your resource collection clicks and time them so the bonus bar does not expire until the maximal reward is obtained; there's a degree of performative uncertainty in doing so. This is, in a sense, a minigame of doober collection within the overarching builder game, and a surprisingly engaging one, given its simplicity. Thus, there is some performative uncertainty in the game, albeit to a mild degree. The main sources of uncertainty still lie elsewhere.

Where do they lie? In four places: in the player's own schedule; in dependence on other players; in the hidden information of the quest narrative; and in the expectation of future development.

Let us take them in turn. When farming or shipping, a player must anticipate the likely date of his return to the game, and suffers consequences if he fails to do so. Given the exigencies of life, this is uncertain. Quite often, real life intrudes, and you return to find a substantial investment in crops or shipping ruined.

While *CityVille* is largely a soloplay game, players affect each other at the interstices—and, in fact, progress in the game often requires *either* the cooperation of other players, or else the expenditure of real money.

Often, completing the construction of a building requires other players to respond to "requests"; often, completing a quest requires other players to "help" you by sending a free gift (that is, one that does not diminish what they possess in the game, but benefits you). Furthermore, access to some content is gated by the requirement that you must have some number of "neighbors" in the game—social network friends who have agreed to this status. Similarly, you can obtain extra energy (a required resource) by asking friends for it, and withered crops or ship deliveries can be "unwithered" by friends.

In short, the game frequently prompts you to send requests to Facebook friends to provide things you need; and indeed progress requires them to respond positively, and within a particular period of time. Of course, the game always allows you to buy your way out of these requirements, with actual money; but even players who don't mind spending to play would prefer to advance without doing so.

The response of your friends is uncertain; while you are likely to be able to obtain what you need over time, it can be days, or possibly even weeks, with repeated requests over the network, before you do. Repeated requests spam your friends with network messages, and you may get sick of bothering them, motivating you to forgo a particular route of advancement or spend money.

On a day-to-day, moment-to-moment basis, these are the two main factors of perceptible uncertainty to the player: Will your friends respond, and how quickly? And will you manage to return on the right schedule or not? The appearance or lack of appearance of a collectible or two is hardly noticeable, and neither a cause of worry nor a triumph.

There's a degree of narrative uncertainty in the quest system, as well. At any given time, a player has a set of "goals" to accomplish, many part of a set of chained goals, with completion of one unlocking another. The goals serve several purposes for the game's operator: Some are "tutorial," encouraging the player to use features of the game to ensure that they understand how to do so. Some are "narrative," with a small story arc that players may enjoy in its own right. And many are "grind" missions, ensuring that the player has a set of tasks to accomplish even if unsure of how to progress or what to do in the overall builder game at that point in time.

As a player, you do not know in advance what goals are important in terms of unlocking new and interesting content; nor do you know what the rewards for accomplishing one will be. The goal structure, in other words, is an example of hidden information, revealed only by play; players feel motivated to accomplish goals to see what comes next, the same kind of narrative anticipation aroused by fiction. Surprisingly, the rewards for goals are remarkably small; it's interesting that players feel motivated to accomplish them regardless. Narrative anticipation—another form of uncertainty—is a sufficient draw.

The final source of uncertainty derives from the fact that social games, unlike conventional PC and console games, undergo continuous development. The developers add new content and features on an ongoing basis, so long as the revenue generated by the game is sufficient to justify employing a "live team" to create new material. Even if a player has advanced far, and essentially consumed the available content, there's always an interest in seeing what new things the developers will create next. There's no way of predicting that, of course, unless you happen to sit in on staff meetings at the developer's offices.

What's interesting about the sources of uncertainty in *CityVille* is that, with the exception of the performative aspect of doober collection, *every one of them* ties directly into the game operator's business objectives. Those objectives are trifold.

First, the operator wants players to use the messaging channels provided by the social network, because at least some of these messages are visible to nonplayers and help encourage new players to join (virality).

Second, the operator wants players to message even those who have already joined the game, in order to persuade those players to return to the game and play further (retention).

Finally, the game operator wants to tempt players to purchase things in the game with the game's "hard currency," which unlike the soft currency is sparsely doled out and beyond a certain point must be purchased with real-world money (monetization).

The reason the operator has these objectives should be obvious: the greater the number of players, the longer those players continue to play, and the greater the number of players who are persuaded to part with actual money, the more money the operator makes. Virality, retention, and monetization are the watchwords of social gaming, just as distribution and marketing are for retail games.

Schedule uncertainty plays directly into retention: if you do not return to the game, your goods will wither. And if they do, you will be incentivized to message your friends to come unwither them, retaining them for the game operator as well.

Gating so much game progress by player messaging serves all three purposes at once: messages to existing players serve a retention purpose, messages to friends who are not yet players serve a viral purpose, and you may always buy your way out of any player assistance requirement.

Similarly, accomplishing quests often involves some player messaging, and sometimes expending scare energy, with additional energy gained by spamming or visiting friends, or via purchase. In all cases, each requirement for completing a quest can be bought out with real money. Advancing in the quest narrative thus requires actions to benefit the game operators.

And, of course, the desire to see newly introduced content requires players to keep playing, directly playing into player retention.

In other words, the creators of *CityVille* know exactly where uncertainty lies in their game, though they may not think of it in those terms; it is the game's uncertain elements that they rely on most strongly for financial success. The essential nature of uncertainty in games could not be more clearly revealed.

Memoir '44

Memoir '44 (Borg, 2004) is a board wargame that simulates the D-Day landings and the battles in Normandy prior to Operation Cobra. Its game system is unusually simple by the standards of board wargaming; other wargames are among the most complex of nondigital games. Its system is also highly original, and because of its popularity, the game has spawned a huge number of expansions and follow-on titles adapting the system to small-unit conflicts in virtually every theater of the Second World War. Despite its originality, however, the game's players face the same basic concerns—and sources of uncertainty—as players of other board wargames.

The game comes with a number of different maps representing different battlefields, some drawn from real-world conflicts (e.g., the Normandy beaches), and others more representative of typical conditions during the war (e.g., a map depicting the hedgerows of Normandy). Each scenario starts with a set number and distribution of units on one of the maps; there is no uncertainty about the order of battle or the reinforcement schedule.

The terrain is public information; no fog of war exists. In addition to physical terrain, there are often preestablished defensive fortifications, such as barbed wire or entrenchments. This information, too, is not hidden. Terrain and fortifications affect rate of movement, provide defensive advantages, and sometimes block lines of sight.

Each player has a hand of cards; each card permits a player to perform actions with a limited number and/or type of units, in a limited portion of the battlefield. For instance, a player might have a card allowing him to activate three units on the left flank, or a card allowing him to activate all his tanks, in whatever portion of the battlefield. During a player's turn, he plays one of his cards, moves and attacks with the permitted units, and draws a replacement card.

There are three basic types of units: infantry, armor, and artillery. Infantry may take four hits before elimination (represented by grouping four soldier pieces together and removing them as hits are taken). Infantry moves slowly, may not advance after combat, and is maximally effective in attack only when attacking adjacent units. Armor takes three hits, moves quickly, may overrun enemy positions, and attacks at full strength up to a distance of three hexes. Artillery takes only two hits, but may attack at a distance of up to six hexes, although its effectiveness declines with distance.

When a unit attacks, the owner rolls between one and three of the special dice provided with the game; the number rolled depends on the firing unit's type, range to target, and terrain. Each die is printed with two infantry symbols, one armor symbol, one grenade, a star, and a flag. The flag and star symbols are normally misses, though some cards change their effect. A grenade is always a hit; an armor symbol is a hit only against armor; and an infantry symbol only against infantry. Thus, in a single die roll, there is a 3 in 6 chance of inflicting a hit on infantry, a 2 in 6 chance on armor, and a 1 in 6 chance on artillery.

To win, a player must earn some number of "badges." Each time an enemy unit is eliminated, one badge is earned; most scenarios provide additional methods for earning badges, such as taking key positions on the map. Since both players are earning badges in play, this usually makes for a tense game, because it's rare for a large lead to open up in the early game.

Where does the uncertainty lie? Clearly, between the dice and cards, randomness is an important factor. As I've argued elsewhere,[10] the fact that board wargames involve many, many die rolls means that the random nature this introduces is somewhat illusory. Given many random tests, the odds regress to the mean, so that winning by pure luck becomes less likely. In *Memoir '44*, however, luck with the dice can play a strong role, particularly in the endgame, when units have taken casualties and the players may be closely matched in terms of badges; at that point, a lucky hit can decide the game.

However, luck in the card draw can be more important; there is nothing so frustrating as having a hand of cards that permits activation of only a small number of units, in a sector of the board where they can accomplish little. Contrariwise, having powerful cards at the right moment can easily tip the game your way.

As with all wargames, there is also a strong strategic element. On any given turn, you examine your cards and try to figure out which one will benefit you most; once you have done so, you try to determine which unit activations will be most beneficial. After that, you must determine precisely which moves and attacks will advance your objectives most strongly. Although wargamers rarely think in these terms, this is, in essence, a form of puzzle solving: figuring out how to optimally use the limited resources you have available, given the complexities of the system. Your ability to determine the optimal move is uncertain, because of the strategic depth of the game. In other words, in addition to randomness, *Memoir '44* depends both on solver's uncertainty and analytic complexity.

Furthermore, there is a smidgen of hidden information, since you do not know what cards your opponent possesses and hence have difficulty anticipating their likely response to your actions. And, of course, as with any multiplayer game in which players can injure or benefit each other directly, there is a degree of player uncertainty.

Board wargamers are, in general, more tolerant of randomness than gamers of other types; they understand that warfare is complicated, and that even the most capable commander must deal with high levels of uncertainty, with elements they cannot hope to control directly. While the universe itself may be nonrandom, at least at a macroscopic scale, the easiest way to simulate elements beyond a commander's control is to treat them as random factors. Thus, randomness may increase the chance that a game is won or lost on the basis of luck, rather than excellent play; but it also increases the verisimilitude of the simulation, and "realism" is an important value for wargamers.

In short, while *Memoir '44*'s primary source of uncertainty, at least at first glance, is randomness, it is a far deeper game than this suggests. There are elements also of puzzle solving, analytic complexity, hidden information, and player uncertainty. Indeed, there are few games so rich in sources of uncertainty, and consequently of such replayability and tenseness.

Civilization V

Sid Meir's Civilization V (Shafer, 2010) is the most recent installment of the series originally created by Sid Meier. As in all versions of *Civ*, you begin with a single settler, a single military unit, and neolithic technology, eventually guiding your civilization into the space age. Though it can be played multiplayer (and version *V* works somewhat better in this mode than previous iterations), it is a game that takes hours to complete, and is therefore not particularly well suited to online, multiplayer play. Most games are played solo, with other civilizations controlled by AI routines (though, as with most game "AI," to call it such is faintly risible; opponents' actions are controlled by a fairly simple set of heuristics).

Version *V*, in particular, has been criticized by some as a lesser game than version *IV*, because some elements (such as religion) have been omitted and others simplified considerably; the developers doubtless wanted to try to sustain the appeal of the game for new players and perhaps felt that the series was accumulating more complexity over time to please its most devoted players at the expense of accessibility to n00bs, a phenomenon I've elsewhere dubbed "grognard capture."

Nevertheless, it remains a rich, deep game with many dimensions: military, diplomatic, economic, and technological. The tech tree of the game means that every game tends to move through the same succession of eras, and thus there is a sameness once you have played many times; but there are also elements that ensure that games take unique paths. Each of the civilizations (French, Japanese, Aztec, etc.) has a different character, with bonuses in some places that encourage different styles of play; the Aztecs, for instance, gain culture points for killing enemy units and begin the game with Jaguar warriors, the strongest early-game military unit. Playing the Aztecs, you are encouraged to be militarily aggressive, particularly in the early stages. As Siam, by contrast, you gain additional food and culture from city-states (small nonplayer powers) with which you ally, and have an early-game building, the Wat, which increases your culture; you are encouraged to take a more pacific path.

Similarly, culture allows you to select national “policies” that differentiate your civilization from others; for example, in the early game, you may concentrate on policies under the Liberty, Tradition, or Honor categories.

Finally, the game can be won in several different ways: through world conquest, by having much more culture than the other civilizations, by building a United Nations and gaining a diplomatic victory, or by being the first to build a starship and reach Alpha Centauri (the closest star system to Sol). Particularly toward the endgame, your play will depend heavily on which style of victory you are aiming for.

The first element of uncertainty in *Civilization* is evident from the moment you start to play: you can see only a small area immediately around your two primitive starting units. The vast bulk of the world is unknown, shrouded in black. The world is generated algorithmically with each new game start; you have no idea whether you are on a small continent or a large one, what civilizations and city-states exist in the world and which are near you, what resources abound nearby and which are lacking, or how the geography of the world will shape military and strategic considerations. You know, at least if you have played before, what types of terrain can exist, and a “minimap” reveals your latitude; you know what types of resources are out there.

But beyond that, the information is hidden. Much of the early game is spent exploring.

Resources are critically important to your play; a city founded near important luxury production sites will find it easier to maintain happiness (important for maximal production from your citizens), and some resources will provide a major boost to production, population growth, or taxes in the early game. In addition to that, other resources become critical as the game progresses: You cannot build units such as swordsmen without iron, or aircraft without aluminum—and the locations of these resources are not revealed until you develop the corresponding technology. Sometimes you find yourself moving along nicely, only to discover that all the aluminum is outside your reach, and your goal of getting to Alpha Centauri is therefore blocked—unless you initiate a war to conquer the resources you need. Moments such as these are among the most interesting of the game, and they work to knock you out of the normal flow of play, sit back, and ponder what to do next.

The AI opponents, too, are somewhat unpredictable. Each has a personality, and you learn to gauge these over time: the Aztecs and Mongols are militarily aggressive, the Siamese and Indians will be peaceful, the Americans will expand quickly, the Germans will drive up the tech tree quickly, and so on. Yet you can never be sure if and when they will declare war on you, try to take control of a city-state you counted as yours, or decide to end a trade deal. Because you can never quite be certain, you are forced to plan for contingencies—perhaps maintaining a larger military than you would desire against the risk of attack, or working to found a new city quickly to close off an avenue of expansion for a competitor.

In the endgame, it is often difficult to judge what style of victory each of the remaining civilizations is going for, and therefore what you need to do to thwart them, or how close each is to obtaining their goals.

In short, even though the opposing civilizations are controlled by AI routines, they are unpredictable enough that there is, in essence, a form of player uncertainty to the game.

Another source of uncertainty lies in the game's system of "wonders of the world." During the game, each city you control is always working on producing something—often, new units, but equally often, on new buildings to improve the city's capabilities. Wonders are a type of building, but unlike other buildings, are unique to the world. Thus, there can only be one Hanging Gardens and one Three Gorges Dam. If an AI opponent builds a wonder before you, you may not then build it yourself (and if a city has been building toward it, that city must change its production to something else). Since you do not know what opposing cities are doing, you do not know what wonders others are working toward, or how close to completion they are, until the wonder is revealed; thus, the competition over wonder construction is an aspect of both hidden information and player uncertainty.

Randomness plays a role: random tests are used in the algorithmic generation scheme for world generation, of course, but in addition, combat is resolved in a semirandom fashion, with each attack by a unit doing a variable amount of damage, within a particular range. *Civilization V*, in contrast to previous versions, alerts you to your likely degree of success or failure before you commit to an attack, but the randomness inherent in the system makes it uncertain: you might still win against the odds, or lose despite them. Consequently, warfare, except when there is a major technological disparity between the opponents, is always tense.

Another form of uncertainty is fog of war. Although the map is revealed as it is explored, only units within a short distance of your own cities and units are visible; hence, you do not know what military units your opponents possess. While their technological advances are reported, you can sometimes be subject to a nasty surprise at the size and advanced state of an opposing army.

Finally, the depth and variety of the game system make achieving your objectives uncertain. While information about the economy is exposed, it is often difficult to judge what to build in a city next, what technological development will benefit you most, or what national policy is the right thing to adopt in the current circumstances. As players gain experience, they learn the ins and outs of the system, but it is still sufficiently complex to be hard to master.

Civilization, in short, has many different sources of uncertainty—more than most games, in fact. This makes it a highly repeatable and compelling experience.

Magic: The Gathering

Magic: The Gathering (Garfield, 1993) is the original trading card game; it swept the hobby game market by storm when first published and retains enduring popularity. It was hugely innovative, both in terms of gameplay and business model. Players purchase cards, either in the form of “starter decks” or “booster packs”; starter decks are designed to be playable out of the box, while booster packs are a random assemblage of cards you may use in play.

To play, each of two players assembles a deck; that is, they select a suite of cards from among all they own. *Magic* is an exceptions game; that is, the basic rules are fairly simple, but many cards have special rules printed on them that must be learned and applied only when the card comes into play. Thus, though the basic structure is easy to learn, the entire universe of possible rules is huge.

The basic structure is that players play “land” cards in front of themselves that provide a certain amount of magic each turn (in five different colors). These magic points may be used to introduce other cards into play, each having a cost in terms of magic points, often requiring points of a specific color. Cards include creatures, which can be used to attack an opponent or defend against their attacks; spells that have some immediate effect; and spells that can be attached to creatures to modify their capabilities. Each player has twenty hit points, which erode as they are subject to successful attacks. The objective is to reduce your opponent to zero hit points.

Interestingly, the strategy of the game lies primarily in deck construction, rather than decision making in the course of play. There are a huge number of unique cards in the *Magic* universe—several hundred when the game launched, and more than ten thousand today (counting all cards published over the course of the game’s history—not all are in print today, but can still appear in a player’s deck). In building a deck, your objective is to assemble cards that work well together, reinforcing the capabilities of other cards; sometimes you have a particular opposing deck in mind, the deck of a friend who beat you and whom you want to beat in return by finding its vulnerabilities. Other times, you want something that you think will be strong against many opponents. Because of the huge variety of cards, decks have strong personalities: one might be built around powerful attack spells and a screen of defending creatures; another around cheaply cast creatures and spells to make them more powerful; a third designed simply to survive as long as possible, with the maximum permitted number of cards in the deck, and with cards to make your opponent cycle through his deck quickly. (One way to win is to have your opponent exhaust his deck before you do.)

Consequently, one source of uncertainty in play is caused by the huge variety of cards; generally, you do not know what cards your opponent has in his deck, what strategy they are aiming for, and often, you will encounter cards you have never seen before. This is, indeed, part of the fascination of the game: the sheer variety it offers, and the cleverness of players in combining cards in effective ways. The game exhibits a combination of player unpredictability, analytic complexity, and hidden information.

Before playing, you shuffle your deck. One rule of *Magic* is that while you may have multiple instances of a single card in your deck, no card may be present more than four times, and decks are typically on the order of sixty cards. Consequently, you are not assured of getting a card you may consider vital to your strategy out of the deck in the early game, or possibly even throughout the game. You can, in other words, be screwed by the randomness of the draw; you may find yourself with many lands but not enough creatures to provide an adequate defensive screen, or conversely with hugely powerful spells and not enough lands to provide the magic power to bring them into play. In other words, your own strategy can be thwarted by the randomness of the game; as in *Poker*, even a very powerful deck can sometimes be beaten by a weaker one, through luck of the draw. Clearly, randomness is part of what makes the game uncertain.

The huge rules set of the game as a whole, coupled with the huge variety of cards, makes deck construction an exercise in parsing through your available cards to create an interesting and powerful deck; there's an element of analytic uncertainty here, since the options are manifold. Despite this, algorithmic complexity is rarely an element of uncertainty in actual face-to-face play. When it is your turn, you have a limited number of cards in your hand, and limited options in terms of which you may play. The cards you and your opponent have already brought into play are now known factors. Frequently, the right move to make is obvious from these factors; *Magic* is not the sort of game that provokes analysis paralysis, with players spending long minutes pondering their options. There is, in other words, little uncertainty in moment-to-moment gameplay, except for the uncertainty of knowing what cards will appear next—and, of course, the hidden information inherent in the unknown nature of the cards your opponent holds.

Randomness is a factor in *Magic* not merely during face-to-face play. Players obtain most of their cards through the purchase of booster packs; the cards in a booster pack are assembled at random (subject to rules that make some rarer than others). Consequently, when you purchase and open a booster pack, you are always uncertain what you will obtain—and may experience delight at finding a new card that works well with others you have, or disappointment at receiving cards that duplicate ones you already have, or worse, quintuplicate them—meaning you already have the maximum of this card you can use in a single deck. This is, of course, one reason *Magic*'s business model is so effective: there's always a temptation to buy more cards, and players can be induced, in essence, to spend the maximum amount they are comfortable spending on their game, whether that be a few dollars or a few thousand.

Finally, *Magic: The Gathering*, almost uniquely among tabletop games, shares with social games like *CityVille* what you might term development uncertainty. Wizards of the Coast, *Magic*'s publisher, releases new cards on a quarterly basis. Consequently, even if you have grown bored with the cards available to you and the options they permit, there's always new content to be explored, a new set of permutations to be mastered. As we saw with *CityVille*, this is an important aspect of player retention for social games, and equally so for *Magic*.

Gardens of Time

Gardens of Time (uncredited, 2011) is, like *CityVille*, a social game, developed by Playdom, a Disney subsidiary, and my employer. (I had no hand in its development.) I discuss it here, because its core appeal derives from a form of uncertainty we have not previously addressed.

Like many social games, *Gardens of Time* has a "builder" aspect, the garden of its name; players purchase decorations for their garden with game currency, and expanding the total decorative value of their garden unlocks other aspects of the game. There is very little uncertainty here.

The core of the game, however, is hidden object gameplay. In this, it is not original; hidden object games were previously among the most popular of genres in the casual-downloadable market, and are also popular in print form, with books such as *Where's Waldo*. *Garden of Time*'s originality is in marrying such gameplay to a social network frame.

At game start, a handful of hidden object scenes are available to the player, and game money is earned mainly by "playing" them. Spending this money on garden decoration unlocks additional scenes.

Each scene is supposedly in the past (or future) and is a busy illustration: a room containing many objects, for instance. Though there are a number of different modes of gameplay, the most common is this: along the bottom of the screen, the names of six objects appear—statue, gold coin, feather, and so on. Each of these items appears somewhere in the illustration. When you locate one, you click on it; this scores you points and removes the item from the list, replacing it with another—at least until six have been found, because one session requires locating twelve objects in total. If you identify a second object within a short period of time, you gain additional points; a maximal score is gained by identifying all objects in sequence without any "time-outs" between them. In the first play-through of a scene, this is virtually impossible, because the busy-ness of the image makes it hard to pick out these items.

However, players are encouraged to play a single scene multiple times. The items called out for identification with each session are varied; essentially, the twelve you are asked to find in a session are drawn at random from a longer list of perhaps twenty-four, so while on your second play, some of the items you found previously may be included, there are likely to be new ones. You will be likely to find them more quickly, even the new ones, because you have spent some time examining the scene, and even if you were not previously asked to find the dog collar, say, you may have noted its presence.

The game contains a "mastery level" system, awarding a player a new level within a scene for achieving a certain cumulative score; new items to find are added to the list for random selection with each level of mastery, ensuring some variety for players over time. Even so, by the time a player achieves the highest level of mastery in a scene, he will likely be finding the objects quickly and may well be able to find them without timeouts.

Where is the uncertainty in this? Initially, the uncertainty lies in your ability to spot items quickly; later on, it lies in your ability to recall where they are and to keep moving ahead of the timeout clock.

This isn't performative uncertainty, in the sense I use it, because this is not a physical task; it isn't solver's uncertainty, since there's no puzzle involved. You could call it a form of hidden information, except that it isn't, really; it's all out there, in the image, hidden only by the difficulty of parsing a busy picture.

The uncertainty lies in your ability to perceive, to use your eyes to translate a word of text to some artists' rendering of that item—sometimes, in fact, different words refer to the same item (die, or cube, for instance). Call it uncertainty of perception.