Uncertainty in Games

Greg Costikyan

The MIT Press
Cambridge, Massachusetts
London, England

3 Uncertainty

Uncertainty, in fact, is a primary characteristic of all sorts of play, and not of games alone; if you think like a programmer, you might say that Game is a subclass of Play, and inherits from Play the characteristic of Uncertainty.

In *Les jeux et les hommes,*[1] the sociologist Roger Caillois says: "Play is . . . uncertain activity. Doubt must remain until the end, and hinges upon the denouement. . . . Every game of skill, by definition, involves the risk for the player of missing his stroke and the threat of defeat, without which the game would no longer be pleasing. In fact, the game is no longer pleasing to one who, because he is too well trained or skilful, wins effortlessly and infallibly."

Caillois calls simple play, unencumbered by rules, *paidia*, and rules-bound play *ludus*. As I prefer to eschew obscurantism, I believe "simple play" and "game" will suffice. Even in simple play, uncertainty is necessary; if, for instance, your older brother always beats you in a footrace, you will quickly lose interest in playing with him. If your friend Jessica always wants to be the princess and insists that you must belong to the supporting cast—prince, ogre, ugly stepsister—and particularly if she never permits a reversal in the story whereby her premier status is

overturned—you will want to find another way to play. Simple play is, in the ideal, joyful and inventive; if it becomes predictable, both the inventiveness and the joy are lost.

The need for uncertainty is, if anything, even truer in games; if our expectation is of predictability, we are unlikely to enjoy the game.

Consider, for example, the game of *Tic-Tac-Toe* (or *Noughts and Crosses*, as the Brits call it). Unless you have lived in a Skinner box from an early age, you know that the outcome of the game is utterly certain. Whoever goes first will take the central square, because occupying it is advantageous, and unless one player is naïve or stupid, players will prevent each other from winning by blocking any attempt to get three in a row. It is a solved game, and a trivial one, and no one beyond a certain age can play it with enjoyment, because no uncertainty about the game's path exists.

And yet the game survives, is taught to each new generation, and is played, by children, with every evidence of enjoyment. The explanation for this is simple: the naïve player has not yet learned, or figured out, that the game has an optimal strategy. To the child, the outcome seems uncertain—as it is, since two players, both playing without an understanding of the game's strategy, produce an uncertain outcome. Thus, a naïve player may experience fiero in winning *Tic-Tac-Toe*, or the fleeting sadness of loss upon losing. In other words, *Tic-Tac-Toe* can be experienced as enjoyable only by naïve players, because only for them is its outcome uncertain.

Caillois's discussion of uncertainty, however, implies that the *outcome* of a game must be uncertain for it to be enjoyable; in this, he is incorrect. The outcome of *Space Invaders* (Nishikado, 1978) for example, is certain: The player will lose. Sooner or

later, the player will be overwhelmed by the serried ranks of invading aliens, and the game will end in a loss. *Space Invaders*, like many of the early arcade games, has, curiously, no win state. But "win or lose" is, after all, merely a binary; *Space Invaders* has a numerical score, which increases with each alien slain, and with no theoretical upper bound to the score. Moreover, a player who achieves one of the top scores on the machine with which he engages may enter his name (or a few characters, anyway), with his score thereafter recorded for everyone to see for all time to come—or until the machine is reset, of course. The goal of *Space Invaders* is not to "win," for you cannot, but to achieve a high score—perhaps bettering your own previous score, perhaps achieving a place on the high score list, perhaps outdoing a friend, perhaps achieving the top slot on the list. The uncertainty of the game lies not in its ultimate outcome, but in the final score.

Based on this, you could argue that Caillois was wrong only in failing to see that the outcome of a game can be more than a binary "win" or "loss" state—that it can be expressed numerically, with a wider range of possibilities. But actually, there's a deeper problem here; not all games have outcomes.

This is a problem not only for Caillois, but also for Salen and Zimmerman, authors of the landmark game studies volume, *Rules of Play*. They define a game as follows: "A game is a system in which players engage in an artificial conflict, defined by rules, that results in a quantifiable outcome."[2]

There's a fair bit to unpack there, and I don't propose to critique the definition word by word; I'm concerned only with "quantifiable outcome," here. Certainly, both win/loss and a score are "quantifiable outcomes"; but what is the "quantifiable outcome" of a game of *Dungeons & Dragons* (Gygax and Arneson,

1974)? *Dungeons & Dragons* has numbers, of course: experience points, player levels, hit points, and so on. It quantifies a great deal. And while the game offers players the implicit goal of improving their character and its capabilities by earning experience points and thereby increasing in level, this is not a competition among the players, who are instead expected to cooperate rather than oppose one another. Nobody "wins." A single session of *Dungeons & Dragons* may come to an outcome—a logical break point in the story is reached, or the players get tired and go home—but unless the gamemaster chooses, for his own reasons, to impose some arbitrary stopping point to the game, it can go on, in principle, forever. Indeed, some games have gone on for decades, with a degree of continuity in terms of the players, their characters, and the setting.

In short, a game of *Dungeons & Dragons* can end, and, if tied to a story, there may be some narrative outcome; and much of the game is quantified. But no outcome is necessary, and quantification is irrelevant to the outcome, if any; outcomes are narrative in nature, not imposed by the game system.

Dungeons & Dragons is far from unique in this regard; *World of Warcraft* (Metzen, Pardo, and Adham, 2004) is the same. There are lots of numbers, and characters work to increase them, but there is no leaderboard, no end of game, no wins or losses or competitive ranking. If *World of Warcraft* ever has an "outcome," it will be because Blizzard tires of the game, or its player base erodes over time to render it unprofitable, and someday the operators close it down. It has no outcome in any meaningful sense.

World of Warcraft is, of course, ultimately derivative of *Dungeons & Dragons*; but the same characteristic pervades today's most popular and commercially successful game form, the

so-called social game. *CityVille* (uncredited, 2010) and *Mobsters* (uncredited, 2008) have no "outcomes"; like *Dungeons & Dragons* and *World of Warcraft*, they are "games neverending."

Certainly these games contain uncertainty; if they were entirely predictable, people would long ago have stopped playing them. The uncertainty is not in the *outcome*, however, because there *is* no outcome. The uncertainty is in the path the game follows, in how players manage problems, in the surprises they hold.

Caillois is correct, therefore, in his assertion that uncertainty is a key element of play, and by extension all games, and incorrect only in his suggestion that uncertainty of *outcome* is essential; uncertainty can be found almost anywhere, as we will see when we begin to analyze individual games.

What Caillois and I call uncertainty, the cultural anthropologist Thomas Malaby[3] calls "contingency." Interestingly, he claims that the main reason games are compelling is that our experience of the real world is "contingent"—the world is unpredictable—and that grappling with the same kind of unpredictability in the more constrained context of the game appeals to our fundamental nature. In other words, he's making essentially the same claim I made at the beginning of this book; that part of the reason games appeal is because they allow us to explore uncertainty, a fundamental problem we grapple with every day, in a nonthreatening way.

I don't have any greater use for the term "contingency" than I do for Caillois's "paidia" and "ludus," however; it obscures rather than reveals. Contingency merely implies that one thing depends on another. The statement "If A, then B" is contingent; the truth of B is contingent on the truth, or falsity, of A. But it

is also perfectly certain; if we know the state of A, we know with certainty the state of B.

Indeed, the distinction between contingency and uncertainty is illustrative of the distinction between games and puzzles. Puzzles are full of contingencies; the solution to one clue in the crossword is contingent on the letters revealed by a cross. The solution to a logic puzzle is contingent on the clues provided. The solution to Sudoku is contingent on the arrangement of the prefilled squares. The only uncertainty involved is in the solver's ability to sort through the contingencies; or to put it another way, a puzzle is static. It is not a state machine. It does not respond to input. It is not uncertain; and it is not interactive.

All games are interactive—nondigital games just as much as digital ones. To be "interactive" means that there are two (or more) parties to a phenomenon, and the actions of one meaningfully affects the state of the other, and vice versa. Conversation is a form of interaction. So, for that matter, is using a light switch; the user's flick causes a change in the state machine that is your house's electrical system, which produces a stream of electricity to a light bulb, which casts illumination on you.

Consider the game of *Chess* as an interaction between two players. The game itself is a state machine whose state is recorded in the positions of the pieces on the board. The players impose a culturally agreed-upon set of algorithms to determine how and under what circumstances the state of the game may be modified, which involves each player responding to the actions of the other sequentially, until a particular state, known as "checkmate," is reached. The fact that the gamestate is represented in physical form, and that the algorithms used to modify its state

are applied by live people rather than a computing device, does not alter the fact that, at its core, the game is interactive.

What would a "noninteractive game" be like? Games by nature either involve multiple players, who interact with each other in some fashion—or a single player attempting to deal with a system that poses some kind of challenge, whether that be 'beating' a level-based videogame or applying the rules of *Klondike Solitaire* to move all cards legally from the tableau to piles sorted by value and suit. In short, even soloplay games are "interactive," albeit in this case the interaction is between a single player and some algorithmic system that responds to the player's actions.

If you took the pieces of a *Chess* set and nailed them to the board, you might have a "noninteractive game," in some sense, but it would no longer be playable.

So all games are interactive. Of course, many other things are interactive as well—the light switch we alluded to, the word processor on which I am composing this book, Google, eBay, and the American political system, for instance. None of these things are games.

To say *why* these things are not games would require us to define "the game"; while trying to do so is an enjoyable pastime in its own right, one in which I have indulged elsewhere, it could produce a book in its own right, and not this one. But it's worth noting one major distinction between games and just about every other form of interaction; games thrive on uncertainty, whereas other interactive entities do their best to minimize it.

Indeed, in the realm of interactive applications, whole disciplines—information architecture, human–computer interaction (HCI), and user-centered design (UCD)—have been invented

precisely to help people create *less* uncertain interactions. If we are shopping online or operating an air conditioner, or for that matter electing a congressman, uncertainty and challenge are the *last* things we want. Rather, we prefer simplicity, surety, and consistency.

You often hear people saying that they want to make their applications or websites more "gamelike." They do not, in fact, mean it. I could make Microsoft Word more gamelike; let us say that in order to make text boldface consistently, I need to be a level 12 Word user. Before I get to that level, every time I try to boldface something, the application does a check, rolling against my level, in effect. If I fail the check, it applies a random font effect instead of boldface. This would not be "more entertaining"; it would be infuriating.

In short, in designing most interactive products, the elimination of uncertainty is desirable. In designing games, a degree of uncertainty is essential. This is why people who try to apply, say, the theories of HCI expert Jakob Nielsen to games often err; interface clarity may still be desirable, but eliminating challenge and uncertainty is not. Games are *supposed* to be, in some sense, "hard to use," or at least, nontrivial to win.

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 non-game 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