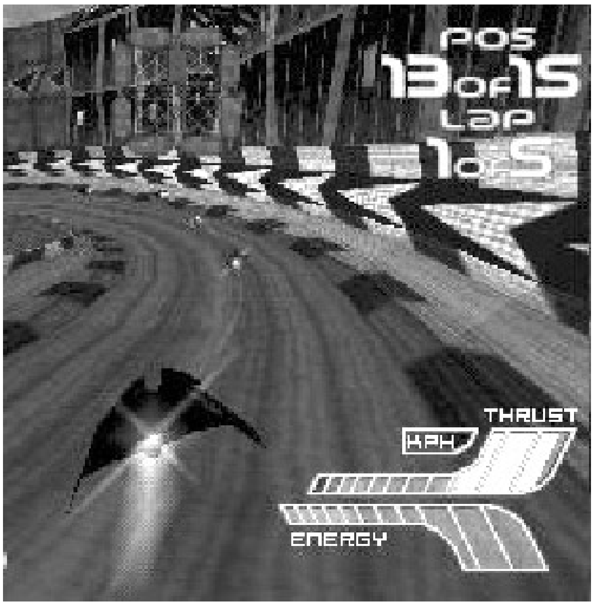
Chapter 18: Games as Cybernetic Systems



*Cybernetics enforces consistency. It permits change, but the change must be orderly and abide by the rules.—**Jeremy Campbell**, Grammatical Man*

Introducing Cybernetic Systems

Cyberspace. Cyberpunk. *A Cyborg Manifesto.* The term *cybernetic* has been appropriated by science fiction and technoculture to mean anything associated with computer technology. In point of fact, the field of cybernetics precedes the advent of digital computers. Mathematician Norbert Weiner coined the term "cybernetics" in his 1948 book *Cybernetics or Control and Communication in the Animal and the Machine.* The word is derived from the Greek word for *steersman* or *navigator,* and appropriately enough, cybernetics studies the regulation and control of systems.

Cybernetics grew out of systems theory and information theory, and like these fields, cybernetics studies a range of subjects, from mechanical and electrical systems to social and biological systems. In looking at the basic principles of cybernetics, we are touching on a field filled with great debates and a rich history, a field that greatly influenced contemporary ideas about computer technology and society.

This chapter can only offer a brief introduction to cybernetics, focusing on the ways dynamic systems change over time and the formal structures that allow these changes to occur. What are the rule structures that monitor change within a game system? How does a game system adjust to change over time? What constitutes feedback within a game? How can positive and negative feedback loops be used in the design of meaningful play? Within this schema on ***Games as Cybernetic Systems,*** we bring cybernetics to bear on these important game design questions.

< Day Day Up >

Elements of a Cybernetic System

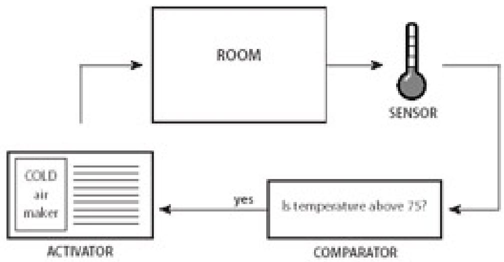
> *Cybernetics deals with the ways a system gauges its effect and makes necessary adjustments. The simplest cybernetic device consists of a sensor, a comparator, and an activator. The sensor provides feedback to the comparator, which determines whether the machine is deviating from its established norm. The comparator then provides guidance to the activator, which produces an output that affects the environment in some way. This fundamental process of output-feedback-adjust-ment is the basis of cybernetics.—**Stephen Littlejohn**, Theories of Human Communication*

As communications theorist Stephen Littlejohn makes clear, cybernetics studies particular kinds of systems. The cybernetic conception of a system is based on the interaction of inputs and outputs with the internal mechanism of a system. Inputs are how the system monitors the environment—they allow the environment to influence the system. Outputs are the ways that the system takes action—they are how the system influences the environment. Through the back-and-forth exchange between the environment and the system, the system changes over time.

A cybernetic system contains three elements: a *sensor,* a *comparator,* and an *activator.* The sensor senses something about the environment or the internal state of a system. The comparator decides whether or not a change to the system needs to be made as a result of the sensor's reading, and the activator activates that change. Together, these three elements regulate how a system operates and changes over time.

A common example of a cybernetic system is a thermostat. Imagine a hot summer day and a room with an air conditioner that is attached to a thermostat. The thermostat contains the system's *sensor,* a thermometer. The thermostat also contains a *comparator* it can use to compare the temperature of the room to a user-set temperature. If the thermostat measures the air temperature above the set amount, it activates the air conditioner, the *activator* of the system, which cools down the room.

As the air begins to cool, the system continues to monitor the room temperature. When the room is sufficiently cooled so that the thermostat's sensor doesn't register the temperature as being above the set limit, the thermostat no longer sends a signal to activate the air conditioner, and so shuts off the cold air. However, the hot summer sun will begin to heat up the room again. When the temperature rises above the thermo-stat's limit, the air conditioner will again be activated. This cyclic behavior of the system is the "process of output-feed-back-adjustment" Littlejohn describes. The fact that the cybernetic system is running as a circuit, constantly monitoring itself to see whether or not conditions have been met, is the reason why cybernetic systems are sometimes called *feedback systems,* or *feedback loops.*



Negative feedback air conditioning system

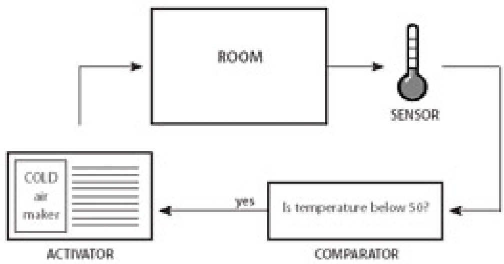
In every feedback loop, information about the result of a transformation or an action is sent back to the input of the system in the form of input data. With the thermostat, the input data is information about air temperature. If this data causes the system to continue moving in the same direction (the temperature continues to rise), then it is *positive* feedback. This means that the effect is *cumulative.* If, on the other hand, the new data produces a result in opposition to the previous result (the temperature is rising, it will now be lowered), the feedback is *negative.* The effects of negative feedback *stabilize* the system.

Positive feedback loops create an exponential growth or decline; negative feedback loops maintain an equilibrium. As cyberneticist J. de Rosnay explains,

> Positive feedback leads to divergent behavior: indefinite expansion or explosion (a running away toward infinity) or total blocking of activities (a running away toward zero). Each plus involves another plus; there is a snowball effect. The examples are numerous: chain reaction, population explosion, industrial expansion, capital invested at compound interest, inflation, proliferation of cancer cells. However, when minus leads to another minus, events come to a standstill. Typical examples are bankruptcy and economic depression.
>
> Negative feedback leads to adaptive, or goal-seeking behavior: sustaining the same level, temperature, concentration, speed, direction. In a negative loop every variation toward a plus triggers a correction toward the minus, and vice versa. There is tight control; the system oscillates around an ideal equilibrium that it never attains. A thermostat or a water tank equipped with a float are simple examples of regulation by negative feedback.[1]

The thermostat example represents a negative feedback system. The system is negative because it seeks to sustain the same temperature. Instead of letting the room get hotter and hotter from the sun, the system acts to return the room to its normative state.

A positive feedback system works in the opposite fashion. Instead of bringing the system to a steady state, a positive cybernetic circuit encourages the system to exhibit more and more extreme behavior. For example, if the thermostat were reversed so that it only activated the air conditioner when the room was *below* a certain temperature, we would have a positive feedback system. If the room temperature ever went below the comparator's threshold, it would continue to run, making the room colder and colder, so that the temperature would steadily get lower and lower. Brrr!

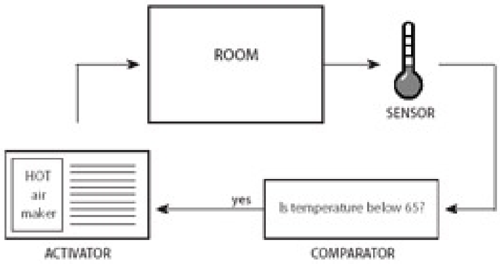


Positive feedback air conditioning system

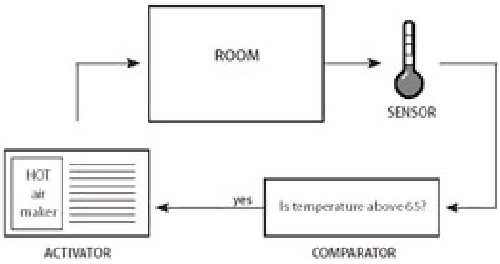
We could also construct negative and positive feedback loops with a heater. In a negative feedback loop, the heater would turn on when the temperature was *below* a certain level, raising the temperature until it reached its original state, at which point the heater would shut off. In a positive feedback loop, the heater would turn on when the temperature rose *above* a certain level, continuing to heat the room indefinitely. Hot hot hot!

Now imagine what would happen if we combined two simple cybernetic systems using an air conditioner and a heater. This dual system would have a sensor to detect the temperature, a double-comparator to compare the room temperature to a pre-established setting, and heating and cooling activators. Using a dual system allows us to control the room temperature in more subtle ways. If both sub-systems were negative feedback systems, the room temperature would be very stable, as both would seek to sustain a middle room temperature. The cooler or heater would turn on when the room became too hot or too cold, and the temperature would always be brought back to its normative position. The system would never let the temperature vary too greatly. This is, in fact, how central heating and cooling works in many homes.

Alternately, both the heating and cooling circuits could be made into positive feedback sub-systems. Whenever the temperature became too hot or too cold, one of the activators would turn on and keep pushing the temperature in that direction. If the temperature setting for the heater were above the temperature setting for the air conditioner, once the room temperature strayed from the middle range, it would never reach the center again. On the other hand, imagine that the heater's activation temperature was below the air conditioner's activation temperature. If the room started out in a middle temperature range somewhere between the two activation temperatures, when the two systems were turned on, both activators would begin battling with each other in a tug-of-war to either raise or lower the temperature.

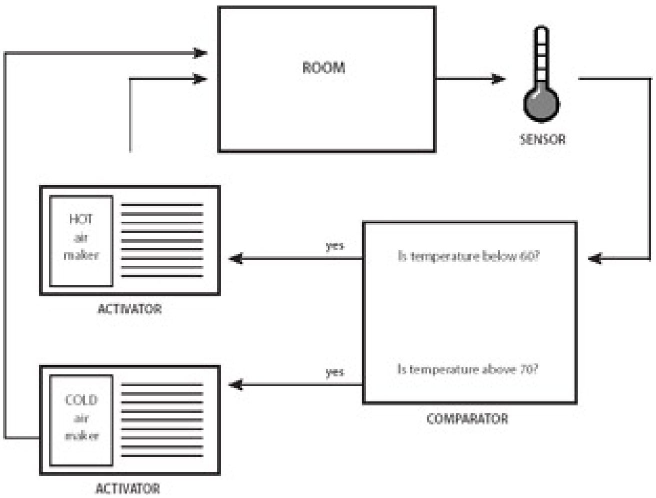


Negative feedback heating system

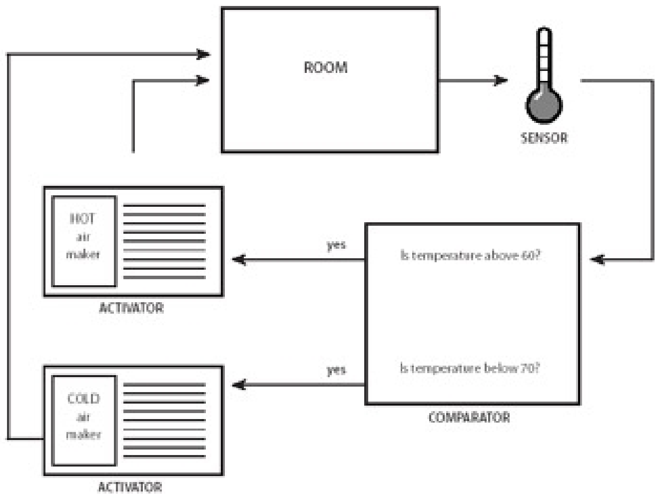


Positive feedback heating system

The important thing to notice in all of the heating and cooling examples is that cybernetic systems affect phenomena like temperature in very specific ways. When more than one cybernetic system is operating together, things get complex quite quickly.



Hot and cold negative feedback system



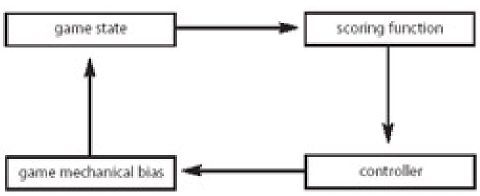
Hot and cold positive feedback system [1]<pespmc1.vub.ac.be>.

< Day Day Up >
< Day Day Up >

Feedback Systems in Games

How do feedback systems operate in games? As a cybernetic system, the rules of a game define the sensors, comparators, and activators of the game's feedback loops. Within a game, there are many sub-systems that regulate the flow of play, dynamically changing and transforming game elements. Do you want your game to move toward a balanced, steady state? Or do you want it to spin wildly toward one extreme or another? Designing feedback loops within your game can help you shape these tendencies. Feedback loops can be tricky to grasp, but they offer a crucial way of understanding how formal game systems function.

Game designer Marc LeBlanc has done a great deal of thinking about the relationship between game design and feedback systems, and this schema is indebted to LeBlanc's important work on the subject. In 1999, LeBlanc gave a presentation at the Game Developer's Conference,titled "[2] Feedback Systems and the Dramatic Structure of Competition." In this lecture, LeBlanc proposed a way of thinking about games as feedback systems, summarized in the following chart:



In this model, the *game state* represents the current condition of the game at any given moment. In a Chess game, for example, the game state is represented by the arrangement of the pieces on the board, the captured pieces, and which player is about to move next. In a console fighting game such as Virtua Fighter 4, the game state includes which two combatants were chosen, the health and other fixed and variable stats of the two fighters, their relative spatial positions, and the arena in which they are fighting. The game state is a *formal* way of understanding the current status of the game, and does not take into account the skills, emotions, and experience of the players. Of course, these player-based factors will definitely affect the game state. If you are a masterful Virtua Fighter 4 player and your opponent is not, this will be evident in the play of the game. However, the game state itself refers only to the formal, internal condition of the game.

The other elements of LeBlanc's model correspond directly to the components of a cybernetic system as we have discussed them. The *scoring function* is the system's *sensor* that measures some aspect of the game state. The *controller* is the *comparator,* which looks at the sensor's reading and makes the decision whether or not to take action. The *game mechanical bias* is the *activator,* a game event or set of events that can be turned on or off depending on the decision of the comparator.

When looking at games as cybernetic systems, it is important to note that we are not necessarily considering the entire game as a single feedback system. Instead, our emphasis is on the ways that cybernetic systems are embedded in games. Embedded cybernetic systems affect a single aspect of a larger game, such as determining which player goes first next round or the relative speed of players in a race. We know from our study of systems that all parts of a game are interrelated in some way. A cybernetic system within a game that directly affects just one component of a game will indirectly affect the game as a whole.

[2]Marc LeBlanc, presentation at Game Developer's Conference, 1999.

< Day Day Up >
< Day Day Up >

Positive and Negative Basketball

To bring this abstract discussion closer to game design, let's look at several game examples. In his talk "Feedback Systems and the Dramatic Structure of Competition," LeBlanc invented two variations on the formal structure of Basketball: Positive Feedback Basketball and Negative Feedback Basketball. Each variation adds just a single rule on top of the existing formal structure of the game:

> ***Negative Feedback Basketball:*** For every *N* points of difference in the two teams' scores, the losing team may have an extra player in play.
>
> ***Positive Feedback Basketball****:* For every *N* points of difference in the two teams' scores, the winning team may have an extra player in play.

How do the addition of these rules change the game? Say, for example, that *N* is 5. In a game of Negative Feedback Basketball, when Team A fell behind by 5 points, it would gain a player on the court and begin to play with a team of 6. As soon as Team A scored points that put it behind by less than 5, it would drop its extra player. On the other hand, if Team A continued to do poorly, when its score was 10 points behind the other team, it would gain a second extra player. Why is this an example of negative feedback? Because the adjustments in the system (gaining and losing players) encourage the system to move toward a stable, steady state. A losing team gets extra players, which helps it catch up to the winning team; when it moves to within 5 points, the two teams are evenly matched. The steady state of this system is not that the total points tend towards zero, but that the *difference* between the two teams' scores stays near zero. The end result is that Negative Feedback Basketball games would tend to be very close games.

Positive Feedback Basketball creates the opposite situation. As soon as one team increased its lead, it would gain additional players. These new players would help the team do even better against the opposing team, which would increase the winning team's lead even more, which would result in yet more players for that team. Eventually, the court would be absurdly crowded with members of one team, who would completely overwhelm and defeat the team with only five players. Positive Feedback Basketball encourages a large difference between the two teams' scores, so that there is a runaway, devastating victory instead of a closely matched game.

As in the examples of heating and cooling, there are many ways to transform the game system. We could, for example, change the rules to remove players instead of adding them. In this case, in Negative Feedback Basketball, when one team pulls ahead by *N* points, it would lose a player, making it easier for the other team to catch up. In Positive Feedback Basketball, the team that was behind by *N* points would lose a player, encouraging them to fall further behind, which would result in the loss of even more players. Eventually, one team would fall so far behind that none of its players would be left on the court. In both games, even though players are removed rather than added, the end results remain the same: Negative Feedback Basketball tends toward stable, close matches and Positive Feedback Basketball tends toward unstable, unbalanced matches. Each variation on the game of Basketball would result in vastly different player and spectator experiences. Yet all we did was add one rule that affected the behavior of the system. Feedback systems offer game designers a powerful tool to affect a game's formal structure and the way that structure manifests in play.

< Day Day Up >
< Day Day Up >

Racing Loops

Positive Feedback Basketball and Negative Feedback Basketball were variations on the game of Basketball. But many existing games already make use of feedback systems in their designs. Here, we look at the use of cybernetic systems in two digital racing games.

Wipeout is a science-fiction racing game originally released for the Playstation, in which the player pilots a fast-moving hover vehicle around a track, trying to beat the computer-controlled vehicles and come in first place. It is common in racing games such as Wipeout for the program to employ feedback mechanisms. Obviously, a computer program can drive a vehicle as poorly or as skillfully as the game designer wants. It would be simple to program the computer-driven cars so that they drove in a mathematically optimal fashion and always beat the player. However, that would simply not be fun. Instead, in racing games the computer vehicles are programmed to drive in a less than "perfect" manner, sometimes not steering or accelerating efficiently, in order to provide a challenge that a human player can overcome.

Wipeout

One way to create a scaled challenge for the player would be to program different skill levels for the computer-controlled vehicles. Some vehicles would be easy for a beginner to beat whereas others could only be bested by experienced players. Programming a static skill level for each opponent vehicle, however, is not yet a cybernetic feedback loop. Why would we want to add a feedback system to a racing game? In order to keep the flow of play exciting, of course. Part of the fun of a racing game such as Wipeout is jockeying for position among a dense cluster of hover vehicles, battling for first place with another racer who is hot on your tail or dead ahead in your sights.

Without a feedback loop, these moments are unlikely to occur. What if a player crashes early in a race—will she ever catch up to the computer-controlled vehicles? Or what if a player's skill far outmatches the pre-programmed computer opponents? Once she gains a lead early in the race, she might as well be racing alone, because the computer opponents will never catch up to her.

This is precisely why Wipeout (and many other digital racing games) make use of cybernetic feedback systems to control the speed of the computer opponents. There are two general rules we can abstract from the behavior of the computer-con-trolled vehicles in the game. Although these are not the only factors determining their speed, they do have a clear impact on the experience of the game:

- If the human player is in first place, the vehicle in second place will accelerate and catch up to the human player's vehicle.
- If the human player is in last place, the last few vehicles will slow down to let the player catch up to them.

The result of these two rules is a negative feedback system. Like Negative Feedback Basketball, together these two rules operate to reduce the distance between vehicles in the game, eliminating the "extremes" of the player being very far ahead or very far behind the computer opponents.

In this system, there are three states that the comparator needs to monitor: when the player is in first place, when the player is in last place, or when the player is in neither first nor last place. If the player is somewhere in the middle of the pack, then no special activator event comes into play. But if the player is in first or last place, vehicle behavior adjusts accordingly. The outcome of this feedback system is that racing in Wipeout tends to offer exciting and satisfying play. Significantly, Wipeout only affects the computer-opponent vehicles, not the hovercraft that the player is driving. In essence, the program carefully adjusts the competitive backdrop, rather than boosting or handicapping the player directly. However, there are games that apply a negative feedback system more directly to a player's abilities.

One example of such a game is Super Monkey Ball for the Nintendo GameCube. Super Monkey Ball contains several different game modes; one of them is a racing game in which up to four players simultaneously race monkey characters through a series of tracks.When players drive through a power-up object on the track, they gain a special power that can be used one time. These powers range from forward-firing attacks (shoot a bomb at another player ahead of you) to rear-based attacks (drop a banana peel, hoping a player behind you will run over it and slip) to non-attack powers (a speed-up that temporarily boosts a player's velocity).

Super Monkey Ball

Whereas many racing games use this power-up convention (including Wipeout), Super Monkey Ball uses a feedback system to determine which power-up a player will receive, depending on whether the player is ahead or behind other players. If a player is in last place, the player is much more likely to receive the speed-up power, which will help that player catch up to the other competitors. On the other hand, a player in first place is more likely to get forward-firing attacks, rather than speed-ups or rear-based attacks.The lead player thus receives the least useful kind of power-up: a player in first place can't use a forward-fir-ing attack to better his position, because no one is ahead of him. These rules add up to a negative feedback system. As with Wipeout, Super Monkey Ball's feedback loops encourage a close race, in which no player is too far ahead of or behind the others.

Powerstone

In Super Monkey Ball and Wipeout, negative feedback loops are used to engender meaningful play. As we know from *Games as Systems of Uncertainty,* the outcome of a game needs to be uncertain for meaningful play to occur. If, as a player, you fall so far behind or ahead of the other players that the outcome is a foregone conclusion, meaningful play is diminished, because decisions you make won't have an impact on the outcome of the game.This does not mean that feedback systems guarantee a close race every time: skill plays an important role in racing games, and it is possible for a player in Super Monkey Ball to fall so far behind that there is very little chance of victory. There is no universal strategy for crafting meaningful play. But in Wipeout and Super Monkey Ball, feedback systems support meaningful play by making the game responsive to the ongoing state of the game.

< Day Day Up >
< Day Day Up >

Positive Feedback in a Game

Not all games use negative feedback systems. Some make good use of positive feedback systems as well. Powerstone is a console fighting game for up to four players that features cartoony, fast-paced brawling action. In Powerstone, when a player is successfully hit by a powerful attack, the target of the attack will be stunned for a short time, during which the target player cannot move his or her character, and the attacker can continue to strike the stunned character. A stunned character cannot move out of the way, defend, or counterattack, and is much easier to hit. This is, in effect, a positive feedback system: because a player successfully launches a powerful attack, the player has a better chance of launching yet more attacks as the target remains stunned and easier to hit. The new attacks continue to stun the target, increasing the ability of the attacker to deliver more damage. Positive feedback creates dramatic results, in which a player can be devastated by a rapidly delivered series of attacks. This kind of slapstick action makes sense in a lighthearted and humorous game such as Powerstone.

However, if the positive feedback system were permitted to play itself out until the end of a match, then the game as a whole wouldn't work. Once your character was hit for the first time, you would remain stunned while an opponent continued to attack you. In effect, the first blow landed would determine your fate and the game would lose the back-and-forth struggle that is an important ingredient of fighting games. Powerstone gets out of this feedback trap by adding a different behavior to the game system. After receiving a certain number of attacks, a stunned character will be hurled across the playfield, making it impossible for the attacker to indefinitely continue rapid-fire strikes. The attacker can pursue the character that flew across the playfield, but the far-flung character is usually no longer stunned by the time the attacker gets there.

Because positive feedback systems are inherently unstable and push a game system toward an inevitable outcome, they are usually dampened by other game factors that limit the acceleration of the feedback loop. In real-time multiplayer strategy games such as Warcraft II, players gather resources, which allow them to build more units that can gather yet more resources, increasing the acceleration of resource-gathering. In this way, all of the players are building their own positive feedback loops, joined together in an arms race to see who will gather enough resources and be the first to front an army capable of winning the game.

In Warcraft II, potentially unstable positive feedback loops are balanced by the fact that each player is creating his or her own feedback loop in parallel. Furthermore, these feedback loops help bring the game to conclusion. Because of their complexity, real-time multiplayer strategy games can sometimes drag on interminably, with players evenly matched and unable to get an upper hand. Because of the way that positive feedback systems can quickly grow out of control, a player that can gain a slight advantage (such as capturing a resource-rich gold mine from another player) can use the advantage to overwhelm an opponent. Obviously, there are many strategic factors other than the resource-feedback loops that determine the outcome of Warcraft II. (For example, skillful battle tactics can help defeat a more resource-powerful opponent.) However, positive feedback systems are clearly a key element of the game design and contribute to the successful play of the game.

< Day Day Up >
< Day Day Up >

Dynamic Difficulty Adjustment

Increasingly, digital game designers are incorporating more sophisticated feedback techniques into their game designs.The game developer Naughty Dog Entertainment is known in the game industry for what it calls "Dynamic Difficulty Adjustment," a technique it has used in the Crash Bandicoot series of games, as well as the more recent Jak and Daxter.

Dynamic Difficult Adjustment, or DDA, uses feedback loops to adjust the difficulty of play. For example, in the original Crash Bandicoot game, the player is generally maneuvering the character Crash through a series of jumping and dodging obstacles, trying to overcome damaging hazards and reach objectives to finish the level. When a player dies, the game restarts at the beginning of the level or at the most recent "save point" reached in the level.

The danger in designing this kind of game is that players possess widely varying skill levels. An experienced gamer might breeze through a level, whereas a beginner might become frustrated after dying several times without making any progress.

The DDA operations in Crash Bandicoot evaluate the number of times that a player is dying at a particular location in a level, and make the game easier as a result. A player having trouble might suddenly find that there are more helpful objects nearby, or fewer enemies to avoid. This kind of attention to the balancing of player experience is evident in the play of Crash Bandicoot games, and it helps explain the fact that a wide audience of both hardcore and less experienced players enjoys them.

Using DDA and other feedback mechanisms in games raises some fascinating game design issues. If we consider the millen-nia-old tradition of pre-computer play, games are traditionally about a player or players competing within a formal system that does not adjust itself automatically to player performance. As you play a game such as Baseball or Othello, your fluency with the system and your ability to manipulate it grows. The game itself and the other players provide the challenge for you. As your play deepens, you find new forms of play, new ways of expressing yourself within the system of the game.

DDA points to a different kind of game, a game that constantly anticipates the abilities of the player, reads the player's behavior, and makes adjustments accordingly. Playing a game becomes less like learning an expressive language and more like being the sole audience member for a participatory, improvisational performance, where the performers adjust their actions according to how you interact with them. Are you then playing the game, or is it playing you? Is a game "cheating" if it constantly adjusts its own rules? Could such a scheme be designed into a multiplayer experience and still feel "fair" for everyone involved? These questions have no definitive answers, as there are always many solutions for any given game design problem. Dynamic Difficulty Adjustment could be considered a heavy-handed design tool that takes agency away from the player, or it could be considered an elegant way of invisibly shaping game play so that every player has an optimal experience. Regardless of your opinion on the matter, DDA is an important tool, and as digital games rely more and more on their ability to automate complex processes, this kind of design strategy will become more common.

< Day Day Up >

A Simple Die Roll

Because most of the examples used so far have come from complex digital games, we wanted to finish by looking at a cybernetic feedback system within a more minimal game context. Sometimes in games, there is no game AI or referee to sense and activate the changes in the game state. However, elegant feedback systems can still emerge directly from the game rules.

Let us take a look at one of our favorite examples, Chutes and Ladders. Chutes and Ladders is an extremely simple game of pure chance. But can you spot the feedback system in it? It is not the actual chutes and ladders. Yes, those seem like they regulate the positions of the players, but they do not act in a cybernetic way. They merely randomly shift the position of the players on the board. The chutes and ladders do not constitute a dynamic feedback loop.

The feedback loop in Chutes and Ladders occurs at the very end of the game, when players must land exactly on the final square in order to win (rather than being able to overshoot the final space and land there anyway). This rule creates a kind of negative feedback system.The exact landing rule serves as negative feedback on the distance between players. In a game of Chutes and Ladders, the player that is farthest ahead will eventually be within six spaces of the finish square and will usually end up spending a few more turns trying to make the exact roll, or possibly inch ahead by rolling small numbers. During this time, the other players often catch up. The overall effect is to level out the playing field by reducing the difference between the positions of the players. The result of stretching out the end of the game in this way is a closer and more dramatic finish.

Think about the game without this rule. If players can overshoot the final space and still win, imagine that you are playing against someone who is just three spaces away from the last square. Even if that player has very bad luck (rolling three 1s in a row), that player is no more than three turns from winning the game. If you are more than 18 spaces away (the total of rolling three 6s in a row), there is no way you can win. On the other hand, if your opponent has to make an exact roll, then he has a 50 percent chance of rolling too high so that he has to stay put, as you keep getting closer. The game is prolonged, the outcome remains uncertain, and in general, the game is more satisfying to play. Those last few die rolls become dramatic, nail-biting game events.

We should point out that this is not a true example of a cybernetic feedback system. An orthodox systems theorist would point out that there is no sensor, comparator, and activator in actual operation. As a counter-example, if there were a rule requiring that the player in first place subtract 1 from his die roll, we would have a true feedback loop, in which a procedural change is enacted when certain conditions are met. Here the player is the sensor, the rule itself the comparator, and the activator is the action of subtracting one from the die roll.

Coming back to our exact landing rule, if we frame the rule in the following fashion, we might consider it to have a feedback loop: "If a player is fewer than 6 spaces from the final space, then rolling higher than *N*, where *N* is the number of spaces between the player and the final space, has no effect."

The rule now feels more like a feedback loop, where the player senses proximity to the finish and the rule acts to limit the effectiveness of the die roll. Ultimately, it does not really matter whether an orthodox systems theorist would approve of this example or not. As designers, the value of a schema is its ability to solve design problems. The rule that requires players to land by exact count on the final space does create more meaningful play. Understanding the rule as a cybernetic feedback loop, or even a pseudo-cybernetic feedback loop, can only enhance our appreciation for the game's design.

< Day Day Up >
< Day Day Up >

Putting Feedback to Use

As a game design schema, ***Games as Cybernetic Systems*** is one of the most practically applicable frameworks presented in this book. Cybernetic feedback systems can be wonderful ways of balancing your game to arrive at a particular result. What is wrong with your game: Is it ending too soon? Running on for too long? Is it too uncertain? Not uncertain enough? Is it too easy or too difficult for players to gain an advantage? You can address all these fundamental questions by looking for feedback loops existing within the formal structure of your game's design, or by adding additional loops of your own.

In his lecture, Marc LeBlanc boiled down the relationship between game design and feedback systems to a set of design "rules." These rules offer a useful set of guidelines for integrating feedback systems into your design. Here are a number of LeBlanc's "rules" and some of our comments on each of them:

- **Negative feedback stabilizes the game.**
- **Positive feedback destabilizes the game.**

These two observations form perhaps the most fundamental cybernetics insight for game design. As a designer, you should be aware of the ways that your game creates stabilities and instabilities. If your two-player card game lets the most powerful player take cards from the weaker player, then you have created a positive feedback system where the most powerful player will quickly dominate. The game is unstable, and will rapidly fall out of balance. Perhaps the solution is to add more players to the game and allow them to team-up on the player that is ahead. This would be adding a negative feedback system to re-balance the game and make it less likely that a player who gains a small advantage will end up winning.

Although our examples have emphasized negative feedback as a useful game design tool, too much negative feedback can make a game too stable. Imagine a variation on Chutes and Ladders in which, whenever a player is ahead of another player, all of the players go back to the start. Although this rule would certainly add negative feedback to the game, ensuring that no player would get ahead of the others, it stabilizes the game to the point of stasis, so that the game doesn't move forward at all. Finding a balance of negative and positive factors for your game is crucial in designing meaningful play.

- **Negative feedback can prolong the game.**
- **Positive feedback can end it.**

LeBlanc's next two "rules" should follow intuitively from our many examples of feedback systems. Positive feedback can rush a game to conclusion, rewarding a player that is already ahead, as in Warcraft II. Negative feedback, as in the Chutes and Ladders exact landing rule, makes it easier for a losing player to catch up, prolonging the game by reducing the winning player's lead.

- **Positive feedback magnifies early successes.**
- **Negative feedback magnifies late ones.**

These two "rules" follow closely from the last pair. In Warcraft II, an early advantage in establishing positive feedback resource loops can put a player too far ahead of the other players. In Chutes and Ladders, on the other hand, negative feedback at the very end can allow a player that has been behind the whole game to catch up.

How can you apply these ideas to your own game design? It depends on the kind of game experience you want to create. There are no universal guidelines for the proper length of a game. Wargamers might play a game for weeks, whereas less hardcore gamers might think an hour is a long time to be playing a single game. Similarly, there are no fixed rules that tell you to make the opening moves or the ending moves the most important ones in the game.

Your guide to making these kinds of decisions should be the core principles of meaningful play. Regardless of the length of your particular game, you should strive to create meaningful play at *all* moments, where the game outcome is uncertain until the end and every action a player takes can help determine that outcome in an integrated way. In general, players that play well should be rewarded with victory. But perhaps there is always a chance for a dramatic turn of events at the end, where the first becomes the last and the last becomes the first.

- **Feedback systems can *emerge* from your game systems "by accident." Be sure to identify them.**
- **Feedback systems can take control away from players.**

LeBlanc's final few "rules" are crucial. Game systems are complex and unpredictable and you can never be sure what feedback systems might be hiding out in the space of possibility you are constructing. Feedback systems can be great ways of shaping player experience, but as LeBlanc warns, as you incorporate systems into your game that actively reshape the experience, you run the danger of removing player agency, leaving your players feeling powerless. Some feedback systems, such as the last space rule of Chutes and Ladders, are relatively innocuous. But many game players will feel "cheated" if they can detect a game adjusting itself to their play. If that second place car is *always* on your tail, does it really matter how well you perform? Perhaps there should be limits on the speed of the second place car, so that a truly masterful player can have the satisfaction of driving far ahead of the rest of the pack.

As this last example clearly demonstrates, the most important thing about players and control is not their actual control in a game, but their *feeling* of control in the experience of play. We explored this phenomenon in the schema on ***Uncertainty***, and it is just as valid here. Meaningful play is, after all, measured by what a player experiences, not by the underlying rules of a game.

< Day Day Up >
< Day Day Up >

Afterword: Don't Forget the Participant

Before departing a discussion of cybernetic systems entirely, we would like to make a few critical comments on the field. Cybernetics is clearly a formal way of understanding systems, which is why the schema of ***Games as Cybernetic Systems*** belongs within our **RULES** primary schema. However, as with all formal schemas, there are many things that cybernetics fails to address.

As a field, cybernetics initially considered a system as a completely self-contained entity. Cybernetics played into the classical scientific idea that the observer of a system had no effect on the operation of the system.This initial model of a cybernetic system was rocked by the introduction of second-order cybernetics into the field. Second-order cybernetics took the observer into account as a part of the system itself, undermining the "objective" stance of classical cybernetics. The insight of second-order cybernetics is that to observe a system in operation is to be part of that system. Although many thinkers, such as Katherine Hayles in her book *How We Became Post-Human,* have since criticized second-order cybernetics for falling into many of the same objectivist traps as its predecessor, second-order cybernetics went far in attempting to understand systems within a larger context.

What does all of this mean for game design? For the purposes of this schema, we made use of the more "classical" first-order cybernetics. We looked at games as self-contained systems, ensconced entirely within the magic circle demarcated by the rules. Occasionally we peeked a bit at the way formal changes play out in the experience of a game, but by and large we kept to the formal mechanics of game systems. This formal emphasis, of course, is what the **RULES** schemas are all about. The conceit of looking at games as formal systems is to leave out all of the emotional, psychological, social, cultural, and contextual factors that influence the experience of the game for the players. In the **PLAY** and **CULTURE** sections of this book, we do in fact look at games as much more than self-contained systems. For the time being, however, we continue our rules-based investigations. Even considered as purely formal structures, there are still many layers to the complex phenomena of games for us to uncover.

< Day Day Up >
< Day Day Up >

Further Reading

How We Become Post-Human, by Katherine Hayles

Hayles' book is less an explication of cybernetic theory and more an ideological critique of the field. However, her detailed research on the development and evolution of cybernetics and second-order cybernetics provides a great deal of insight to how these movements intersect with cultural beliefs about technology. In this complex book, Hayles also relates these subjects to literary theory and contemporary ideas about computer technology, virtuality, and identity.

Recommended:

Chapter 3: Contesting for the Body of Information: The Macy Conferences on Cybernetics
Chapter 4: Liberal Subjectivity Imperiled: Norbert Wiener and Cybernetic Anxiety
Chapter 6: The Second Wave of Cybernetics: From Reflexivity to Self-Organization

Theories of Human Communication, by Stephen W. Littlejohn (*see page 200*)

Recommended:

Chapter 3: System Theory

< Day Day Up >
< Day Day Up >

Summary

- Cybernetics studies the behavior of self-regulating systems. A cybernetic systems consists of three elements:
    - A **sensor** that measures some aspect of the system or its environment
    - A **comparator** that compares this measure to a set value and decides whether or not to take action
    - An **activator** that creates a change in the state of the system

For example, in an air conditioner, the sensor and comparator are in the thermostat, which activates the air conditioner activator to cool down a room when the temperature gets too high.

- **Cybernetic feedback systems** can be positive or negative:

  A **negative** feedback system is **stabilizing** and brings a system to a fixed, steady state. The air conditioner example, which keeps a room from getting too hot, but shuts off when the room cools down, is a negative feedback system. The temperature remains within a narrow range.

  A **positive** feedback system is **cumulative** and makes a system unstable. If the air conditioner turned on when the temperature was **below** a certain number, then the room would become colder and colder, moving away from a stable state.
- A game can contain many feedback systems that interact with each other within the larger system of the game.
- Many game feedback systems are negative, reducing the advantage or disadvantage of a player or a team. This phenomenon is common in digital racing games.
- Games also make use of positive feedback systems for dramatic effect or to bring a game to conclusion. Often, a positive feedback system is countered by a negative feedback system in a game. Powerstone's stunning and hurling features demonstrate positive and negative feedback systems working together.
- **Dynamic Difficulty Adjustment**, or **DDA**, is the modification of a game's challenge according to player performance. It is most often used in complex single-player digital games.
- Game Designer Marc LeBlanc outlines a number of design "rules" that apply cybernetics to game design. These "rules" include the following:
    - Negative feedback stabilizes the game.
    - Positive feedback destabilizes the game.
    - Negative feedback can prolong the game.
    - Positive feedback can end it.
    - Positive feedback magnifies early successes.
    - Negative feedback magnifies late ones.
    - Feedback systems can emerge from your game systems "by accident." Be sure to identify them.
    - Feedback systems can take control away from the players.
- In the field of cybernetics, the more classical **first-order cybernetics**, which considers a system as a self-contained entity, was challenged by **second-order cybernetics**, which includes the observer of a system as an element of the system. Within this formal schema, we have not made use of second-order cybernetic thinking.

< Day Day Up >