CHAPTER TWO

Game Feel and Human Perception

Understanding exactly how humans perceive the video game worlds we create is key to designing good game feel. We'll begin by examining our feedback loop model of interactivity from Chapter 1 in greater detail. By deconstructing each piece of this system and incorporating the concept of the Model Human Processor, we'll be able to define real-time control at the level of specific, measurable properties of human perception. This will tell us exactly when and how real-time control can exist and what will cause it to break down. We'll also look at the computer's side of things: what, exactly, are the parameters of machine illusion? Finally, we'll look at some of the implications of perception for game feel.

When and How Does Real-Time Control Exist?

In Chapter 1, we defined real-time control as the uninterrupted flow of command from player to game resulting in precise, continuous control over a moving avatar. It's more like driving a car than having a conversation, as we said. The part of the definition that needs clarifying is "uninterrupted." What if the player can offer new input at any time, but the game can only receive it at set intervals? Or what if the player gets locked out for a certain amount of time, unable to add new input until an animation has finished playing? In other words, what is real-time control and how do we know when it's happening and when it's not?

Let's again look at interactivity, this time with more specificity. There are two halves to this process, the player and the computer (Figure 2.1). On the player's side of things, there are unchanging properties of human perception. For example, there is a minimum amount of time in which a player can perceive the state of the game, think about how to act and pass that impulse along to his or her muscles.

On the computer's side, this creates boundaries. To sustain real-time control, the computer must display images at a rate greater than 10 per second, the lower boundary for the illusion of motion. The computer must also respond to input

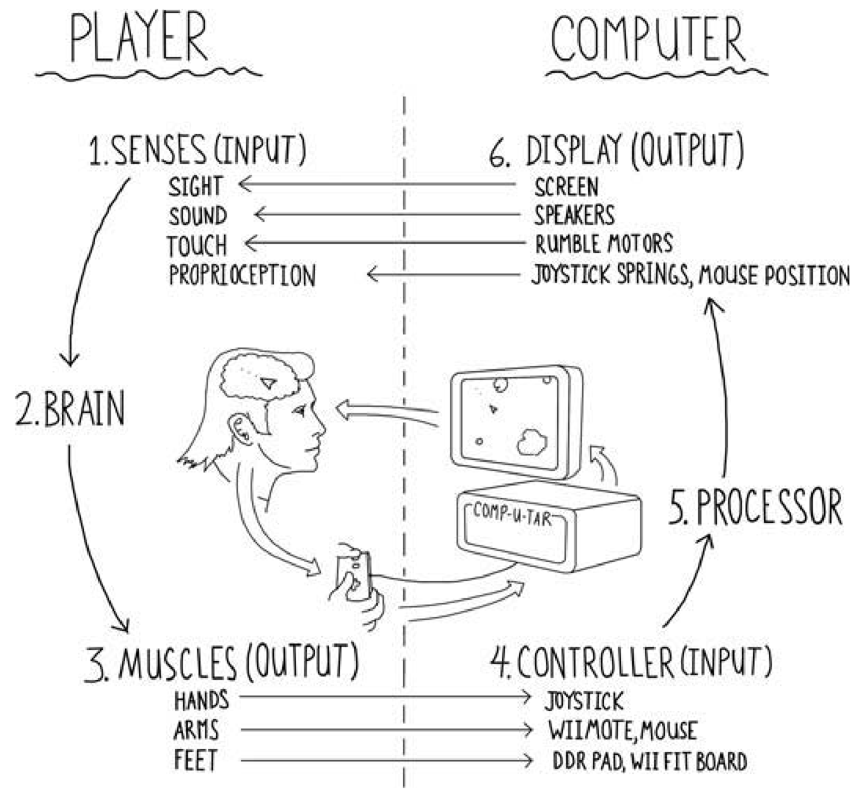

FIGURE **2.1 Interactivity in detail.**

within 240 milliseconds (ms), the upper boundary for response time. There's also a threshold for continuity; the game must be ready to accept input and provide response at a consistent, ongoing rate of 100 ms or less. If the game responds to input sporadically, the flow of control is broken. The onus for maintaining real-time control, then, is on the computer. The computer's half of the process is changeable. The player's perception is not.

On the player's side, the minimum amount of time it takes for a person to perceive the state of the world, think about what to do and act on that impulse is around 240 milliseconds. This is a very short amount of time.

This correction cycle is the increment at which people make the tiny adjustments necessary to assemble a sandwich, drive a car or exercise real-time control over objects in video games. The measure comes from Card, Moran and Newell's "Model Human Processor," the collected result of many different studies about human reaction and response time. The figure of 240 ms is an amalgam of three different measurements, one for perceiving, one for thinking and one for acting. They break down as ranges, like so:

- Perceptual Processor: ~100 ms [50–200 ms]
- Cognitive Processor: ~70 ms [30–100 ms]
- Motor Processor: ~70 ms [25–170 ms]

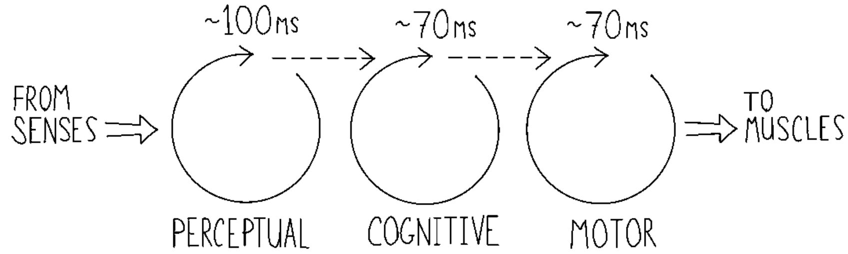

FIGURE **2.2 The three processors: perceptual, cognitive and motor.**

What's being measured here is the cycle time of each processor, the time it takes to accept one input and produce one output. The variation comes from physiology and circumstance. Some people have the capacity to process things more quickly than others and everyone tends to process things more quickly under intense circumstances, displaying a heightened sense of awareness. Likewise, processing speed goes down under relaxed or sub-optimal circumstances such as reading in the dark. In the model, each of these steps is defined as its own separate processor and has its own little cycle time (see Figure 2.2).

Try It Yourself

To appreciate the speed at which your processing functions, I highly recommend checking out humanbenchmark.com's Reaction Time Test (http://www.humanbenchmark.com/tests/reactiontime/index.php.) This will give you a clear sense of just how small the increments of time we're talking about seem when you're able to measure them against the computer. My best reaction time is around 170 ms. If you're like me, it will feel a bit weird to actually butt up against the limits of your own perception. But there it is: you can't argue with the precision of the computer measuring your reaction time. It's neat that we can measure this!

The idea is that perception, cognition and action are processed separately but feed into one another, enabling a person to see the state of things, think about how to change them and then act on that impulse. Note that this is an abstraction of human cognition—nowhere in the anatomy of the brain is there a structure called the perceptual processor—but it is a useful one because it lets us put hard numbers to components of our diagram.

The perceptual processor takes the input from the senses and makes sense of it, looking for patterns, relationships and generalizations. From all the sensory data, it creates recognizable state of the world for the cognitive processor.

The cognitive processor does the thinking. It compares intended result to the current state of things and decides what to do next.

The motor processor receives the intended action and instructs the muscles to execute it. After the impulses leave the body as muscle movements, they're out into wild, wooly reality, and the process starts again with sensory perception.

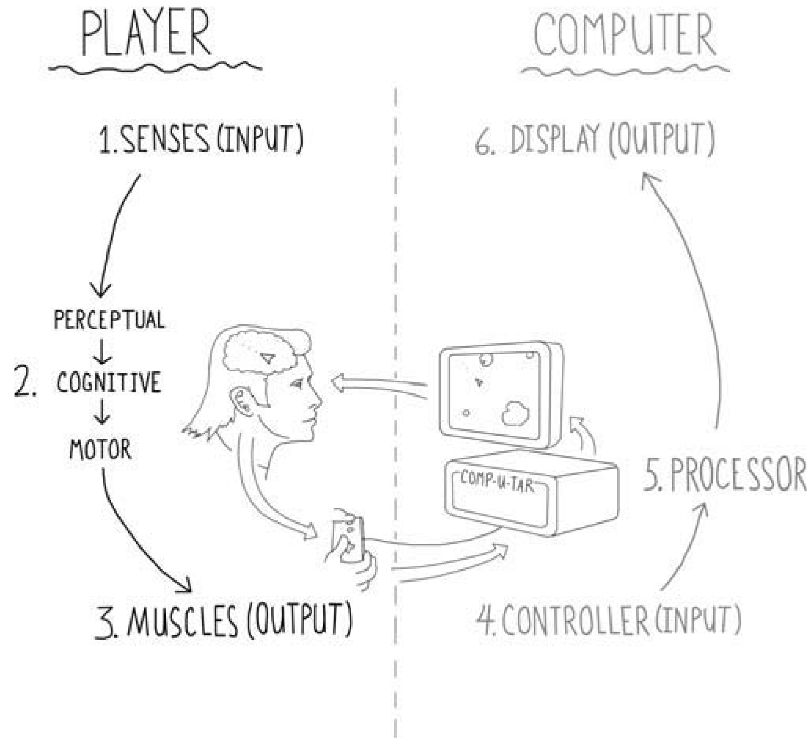

FIGURE **2.3 The three human processors in the interactivity diagram.**

All of this is happening at step 2 on our chart of interactivity, in the player's mind (Figure 2.3).

Correction Cycles and Game Feel

When all three processors (perceptual, cognitive and motor) work together in a closed feedback loop, the result is an ongoing correction cycle. A correction cycle happens any time you do something requiring precise coordination of muscles over time, whether it's picking up a book, driving a car or controlling something in a video game. Robert Miller of MIT's User Interface Design Group describes the process: "There's an implicit feedback loop here: the effect of the action (either on the position of your body or on the state of the world) can be observed by your senses and used to correct the motion in a continuous process."[1]

For example, imagine you want to reach out and grab a muffin that's sitting on your desk. You formulate intent: to grab the muffin (Figure 2.4). As soon as this intent is formulated, it is translated into action of your muscles—twist trunk in chair, activate arm muscles, open hand into "muffin claw" and so on. The moment this action starts, you perceive the position of your hand in space and see it start to move, responding to your impulses. The perceptual processor looks at where the hand is in space, passing that information on to the cognitive processor. The cognitive processor

[1] http://ocw.mit.edu/NR/rdonlyres/Electrical-Engineering-and-Computer-Science/6-170Fall-2005/8B87E671-1B67-4FEF-A655-0ABDF89F4F5A/0/lec16.pdf

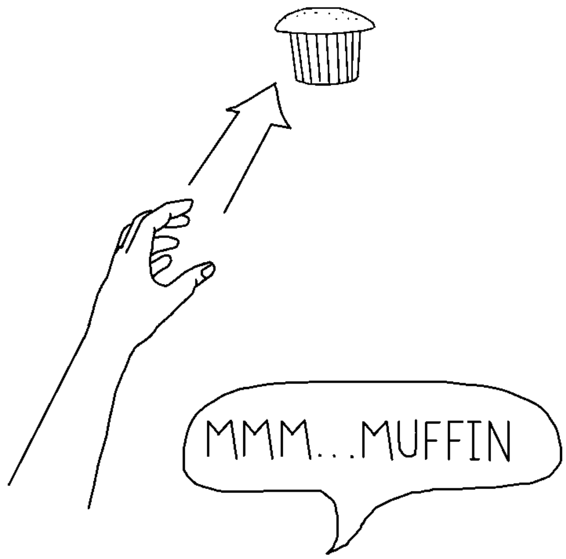

FIGURE **2.4 Intent: grab the delicious muffin.**

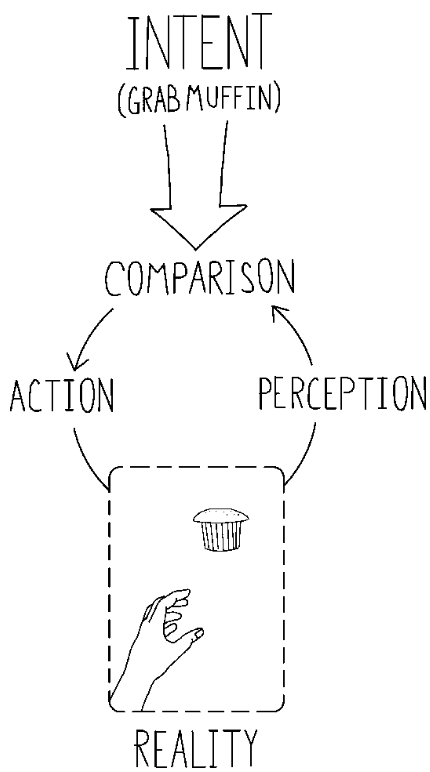

FIGURE **2.5 Grabbing a muffin: an ongoing correction cycle.**

thinks about where the hand is relative to where it should be and formulates a new plan to correct that motion. The motor processor then takes the new plan and translates it into action. From the moment movement starts to when you have the muffin in hand, you run this continuous process of action, perception and thought, increasing precision each time as a factor of distance and size of target (Figure 2.5).

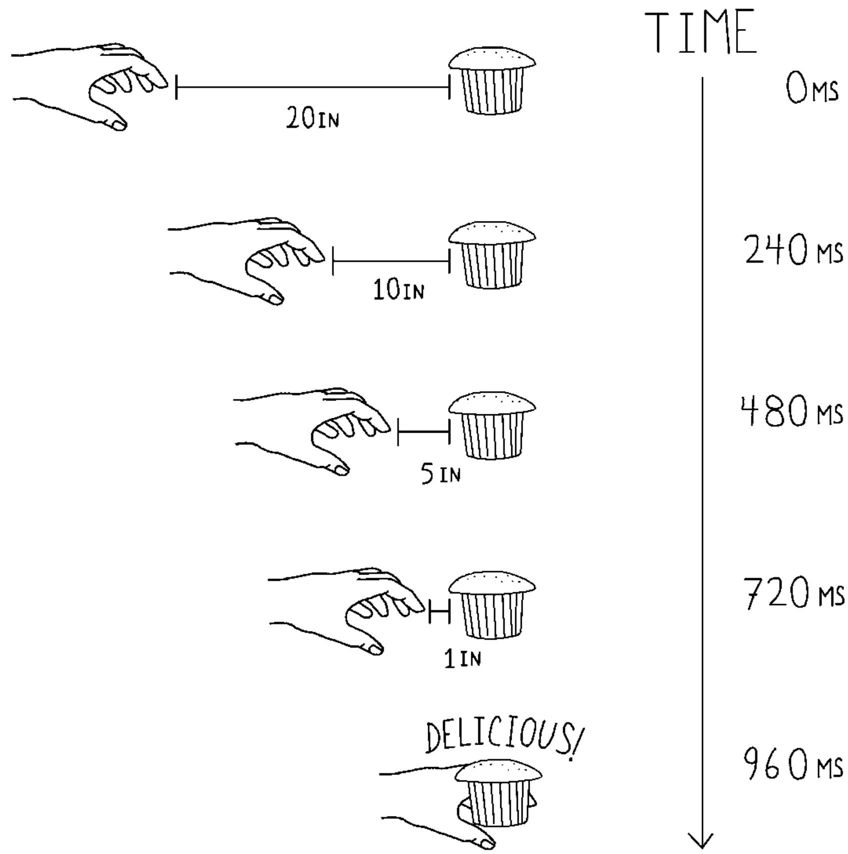

FIGURE **2.6 Correction cycle time: 240 ms.**

Because we know the cycle time of each processor (perceptual = 100 ms, cognitive = 70 ms, motor = 70 ms) we know the time for each one of these correction cycles, 240 ms (Figure 2.6).

Correction cycles are how people are able to track and hit targets with precision, steer things, point at things and navigate the physical world successfully. To experience this first hand, check out example CH02-1. Try putting your cursor directly over the dots as quickly as possible. You can see the correction cycle in action as you overshoot, undershoot and then hone in, eventually coming to rest on your target.

Now imagine you're hungry, and you're out of muffins. You get in your car and begin to drive to the muffin store. The overall goal for this trip is to get a sweet, delicious banana muffin. This goal trickles down to different layers of intention, such as "Turn right on Elk Street." At the lowest level, it's about the moment-to-moment adjustments of the motion of the car to keep it in the lane, stop it at red lights and so on. As before, you perceive the state of the world, think about what corrections you need to make to the current motion and make the adjustments happen once every 240 ms (Figure 2.7).

The process is the same as when you were reaching across the desk except that this correction cycle goes on longer. The muffin on the desk was static and represented

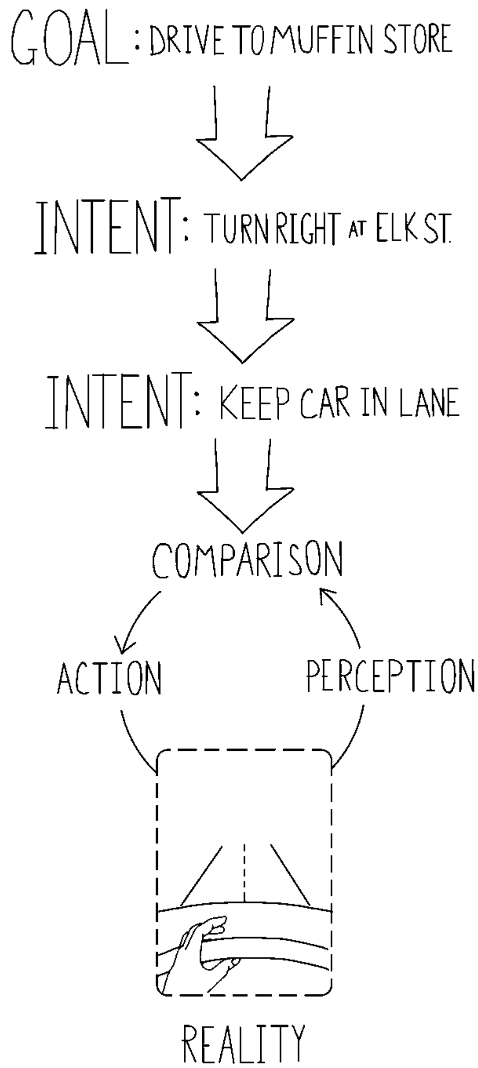

FIGURE **2.7 The correction cycle of driving.**

a single target, a single intent. Driving to the store might take 20 minutes, fulfilling 20 different sub-goals.

In a video game, real-time control is an ongoing correction cycle of this type. As with driving a car, control is an ongoing process where higher-order intentions trickle down and become individual, moment-to-moment actions. These actions are a part of an ongoing correction cycle, where the player perceives the state of the game world, contemplates it in some way, and formulates an action intended to bring the game state closer to an internalized ideal (Figure 2.8). This happens at the same cycle time of ~240 ms.

The difference is, at the point where action normally goes out to physical reality, a video game substitutes a game world for the real world. It hooks right in there. The inputs to the perceptual processor come from the screen, the speakers and the feel of the controller. The output, instead of acting on objects in the real world directly, acts on the controller, which translates to movements of objects in the game world.

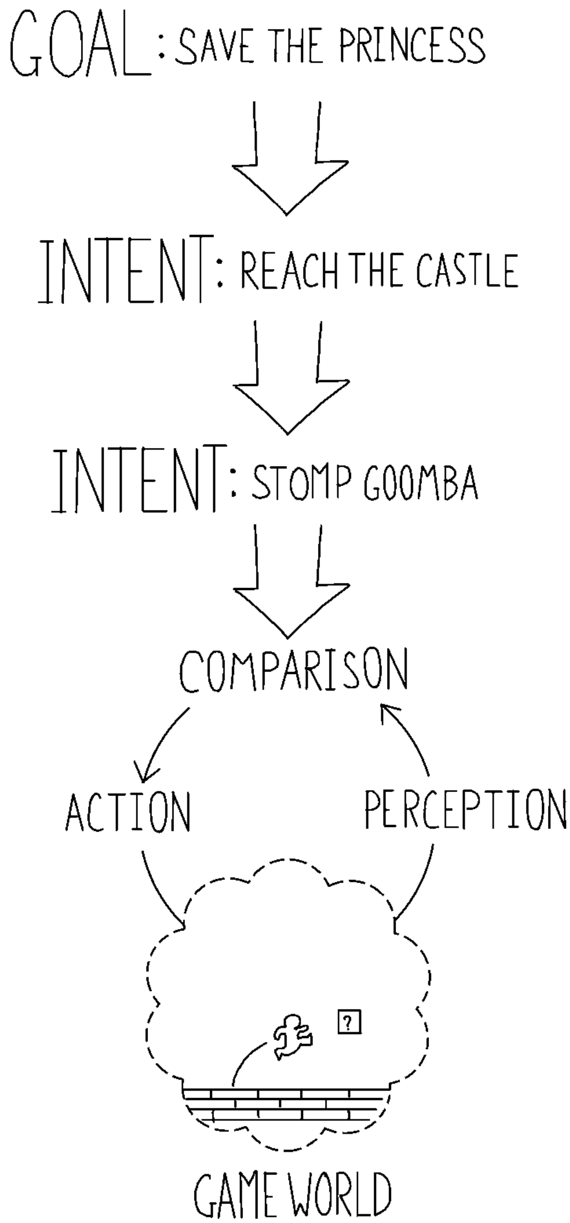

FIGURE **2.8 The correction cycle of real-time control in a video game.**

Fitt's Law

There is a well-known formula, Fitt's Law, which can accurately predict how quickly you can move your hand to a target of a particular size at a certain distance. Fitt's Law is an unusually successful and well-studied HCI model that has been reproduced and verified across numerous studies. For reference, the formula is this:

$$MT = a + b\log_2\left(\frac{D}{W} + 1\right)$$

where:
MT = movement time
a = start/stop time of the device
b = speed of the device
D = distance from starting point to target
W = width of target measured along the axis of motion

The original formula predicted how long it would take to reach out and touch something of a certain size a certain distance away, as long as it was within arm's length. It was later discovered to be equally applicable to the time it takes to move a mouse cursor to an object of a particular size and shape on a computer screen, so it is applied and studied by user interface designers. For example, the menu bar in the Macintosh OS is always present and takes up the entire top edge of the screen. This means that the "size" of the menu bar is functionally infinite, enabling the user to get the cursor onto it quickly and easily, with very few correction cycles. Compare this to a tiny checkbox button or a hierarchical submenu.

The Computer Side of Things

Real-time control relies on the computer sustaining three thresholds over time:

1. The impression of motion (display above 10 fps). The frames displayed on the screen must be above 10 per second to maintain the impression of motion. The impression will be better and smoother at 20 or 30 frames per second.
2. Instantaneous response (input to display happens in 240 ms or less). The computer's half of the process must take less than correction cycle for the player. At 50 ms, response feels instantaneous. Above 100 ms, the lag is noticeable but ignorable. At 200 ms, the response feels sluggish.
3. Continuity of response (cycle time for the computer's half of the process stays at a consistent 100 ms or fewer).

The Impression of Motion

Similar to film and animation, the way that computers create and sustain the impression of motion is well understood. Think of each cycle of the player's perceptual processor as a snapshot of reality, incorporating visual, aural, tactile and proprioceptive sensations. Each 100 ms cycle, the perceptual processor grabs a frame of all these stimuli. If two events happen in the same frame—Mario in one position, then Mario slightly to the left—they will appear fused, as a single object in motion rather than a series of static images (Figure 2.9). This is perceptual fusion.

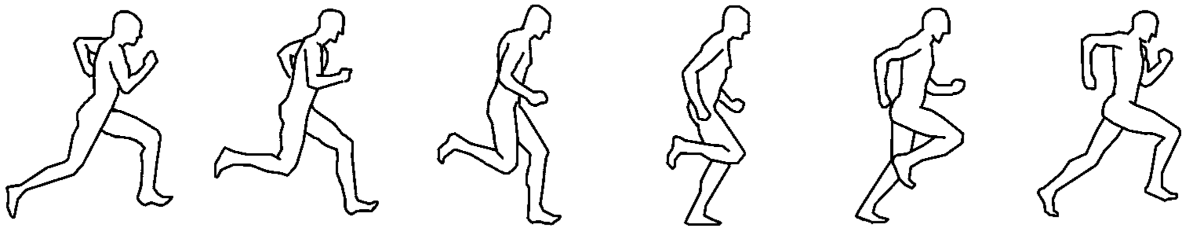

FIGURE **2.9 Ten frames per second is the threshold for the illusion of motion.**

From the computer's side of things, perceptual fusion explains how objects in a game appear to move. If the display is updated 10 times per second (100 ms cycle time = 10 frames per second) this is sufficient for the illusion of motion. This is right at the border, though, and won't feel very good—20 frames per second (fps) is better, and 30 fps is where motion begins to be pleasingly smooth. Most games run at 30 fps or higher for this reason. As game developers know, with frame rates that vary based on processing power used it's better to be safe than sorry. There's no such thing as a frame rate too high.

Instantaneous Response

Perceptual fusion also influences the impression of causality. If I flip a light switch and the light comes on within the same perceptual cycle, I will register this as a cause-and-effect relationship. My action caused the light to turn on. The same thing is true for computer response: if I move a mouse and the cursor seems to react immediately, I tend to assume that effect was caused by my action. An extension of this is the impression of responsiveness. Professor Miller describes the process: "Perceptual fusion also gives an upper bound on good computer response time. If a computer responds to a user's action within [~100 ms], its response feels instantaneous with the action itself. Systems with that kind of response time tend to feel like extensions of the user's body."

With reality, there's never a problem of lag. Response will always be instantaneous. In a game, response will never be instantaneous. Even a game running at 60 frames per second, a three-frame delay is all but inevitable. Three frames at 60 frames per second means 50 ms. (You can convert frames per second (fps) to milliseconds if you divide by 60 and multiply by 1000. So 3 frames at 60 fps is 3/6 * 1000 = 50 ms.)

Mick West, programmer-designer of the original Tony Hawk mechanic, defines this as response lag. "Response lag is the delay between the player triggering an event and the player getting feedback (usually visual) that the event has occurred. If the delay is too long, then the game will feel unresponsive."[2]

[2]http://cowboyprogramming.com/2008/05/27/programming-responsiveness/

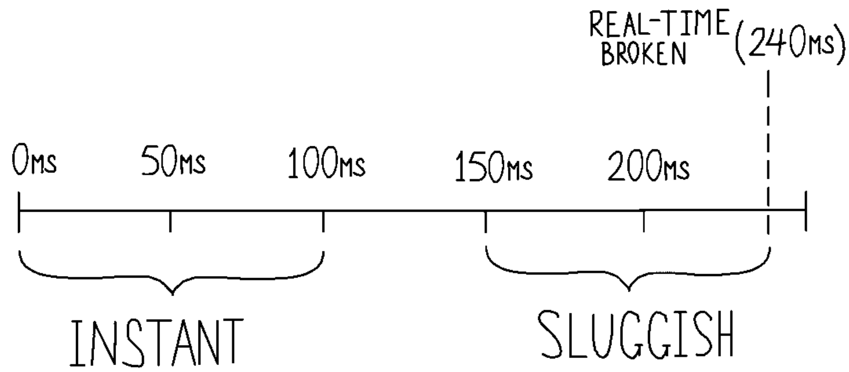

FIGURE **2.10 Response time and player perception.**

Mick notes that games with a response time of 50 to 100 ms typically feel "tight" and "responsive" to players. This is because 50 to 100 ms is within one cycle of the human perceptual processor. Above this level, a game's controls begin to feel sluggish. The progression, from responsive to unresponsive, is gradual (Figure 2.10).

Mick West on Responsiveness

Check out Mick's articles about programming for responsiveness at www.cowboyprogramming.com. He offers an awesome technical grounding for avoiding response lag as well as an engagingly practical way to measure response time in any game using an inexpensive digital camera (recording both screen and controller at 60 fps).

There's no exact point at which a game's response lag can be said definitively to have gone from tight to sluggish, because other factors, such as mapping and polish effects, can shape this impression of responsiveness. But there is a threshold above which the sensation of real-time control is broken: 240 ms. Past this response time, the player can perceive, think and act before the computer is ready to accept a new input.

Continuity

If it takes a player 240 ms to perceive, think and act, how is it that the computer has to finish its tasks and offer feedback within 100 ms for the response to seem instantaneous? This is because all the human processors run concurrently (Figure 2.11).

The perceptual processor passes information along to the cognitive processor, then starts cycling again. By the time the instructions from that original cycle have been sent out into the world as movements of the muscles, three perceptual frames have passed. In real life, this never matters because the response is always

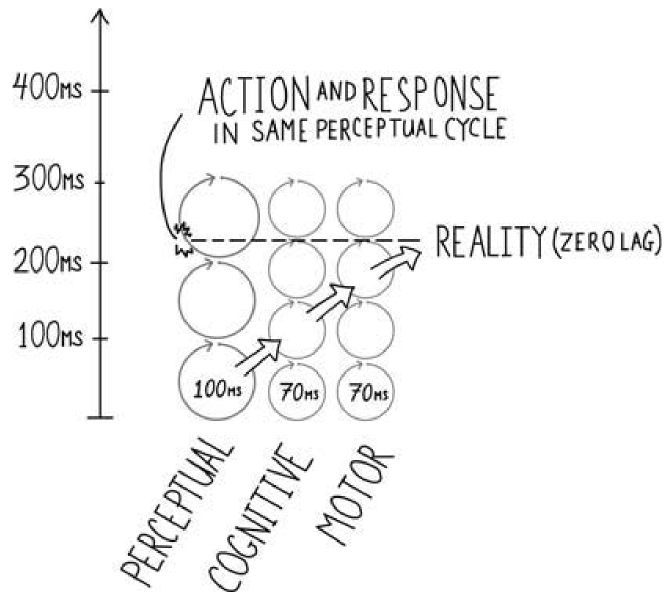

FIGURE **2.11 The three processors running concurrently, passing into reality and back with zero lag.**

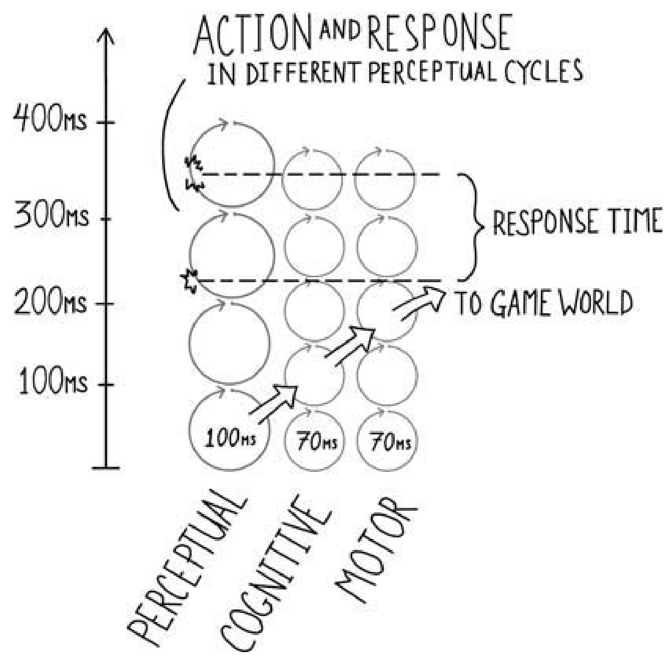

FIGURE **2.12 The three processors running concurrently, passing into a game and back. This time, there is some delay.**

instantaneous. When the motor impulse passes to a game, though, there will always be some lag (Figure 2.12).

What this means is that the game needs to update faster than the player's perceptual processor is running. If it doesn't, the feedback from the game will skip perceptual frames, and the player will be aware of the lag. The game has to enable input at any time even though the player won't be giving it all the time. This enables the player to modulate an input over time, adjusting it incrementally. The steering wheel will remain pulled to the left and the car will continue steering to the left even though the player adjusts how much it's turning at 240 ms intervals. This is the threshold of continuity: the computer half of the process must update consistently at or below 100 ms for real-time control to be sustained.

Continuity is tricky to measure. Many games respond to input at some times and not at others. This can be a frame rate drop, a delay between the computer and the display or a problem with the processor being overloaded by too much awesomeness in the game (not being able to pass frames along to the display in a timely manner). Any time the computer's half of the process takes more than a cycle of the perceptual processor, the player will notice. If it takes more than a correction cycle, real-time control is broken.

Other times, however, the delay is part of the game. For example, if I execute a fierce punch with Zangief in Street Fighter II, I lose control of the character for about 750 ms. This is longer than the threshold needed for an ongoing correction cycle by 510 ms. This temporary loss of control is ignorable, though, because I chose when to trigger that move. It becomes a risk-reward tradeoff rather than a disruption. It also makes sense in the context of the game's unique world. Big dude, heavy punch, does lots of damage. It should take a long time to execute. I knew what I was getting into. This shows that the continuity of real-time control can be broken without interrupting the player's feeling of being in control.

This makes interruption more than simply a response time too long or broken continuity. It means interruption can be smoothed over by other factors, such as metaphorical representation. Real-time control, then, is a gestalt. It primarily relies on the computer sustaining three thresholds over time, the impression of motion, instantaneous response, and continuity of response.

But the ultimate judge is the player. The designer may have done something clever with input mappings or animations to mask any interruption to the flow of control. The end result is an impression in the player's mind. If it feels like real-time control, it is.

Now let's explore perception and game feel implications for game design from a more general point of view.

Some Implications of Perception for Game Feel

To examine game feel is to see perception in a particular way. First, game feel includes many senses. Visual, tactile, aural, proprioceptive—experiencing game feel, these senses combine into one sensation. Second, the notion of proxied embodiment should be addressed. Game feel enables objects external to the body to be subsumed into body image and feel like extensions of the body. Third, game feel is an ongoing process of skill building and practice. It is part of the larger realm of human skill building and relates to skills like driving and playing tennis because it requires the same kind of repetitive practice to master. From n00b to l33t, as it were. Lastly, a model for perception for game feel needs to encompass the physical nature of game feel. Experiencing game feel is like interacting with a surrogate reality which obeys its own rules and which must be understood through interaction and observation.

Expanding on the experiences in Chapter 1, here are five interesting ideas about perception that support our definition of game feel:

1. Perception requires action.
2. Perception is skill.
3. Perception includes thoughts, dreams, generalizations and misconceptions.
4. Perception is a whole-body experience.
5. Tools become extensions of our bodies.

Perception Requires Action

In order to perceive something, you have to see it in action. This has been verified experimentally with kittens and blind people.

The kitten study (Held and Hein, 1963) involved two groups of kittens, each raised in the dark. The first group was allowed to roam freely, while the second group was "kept passive" which we can only reasonably assume meant a Clockwork Orange style tie-down. The experimenters controlled the conditions such that both groups of kittens were exposed to the same limited stimulus: flickering lights, sounds and so on. Then, they released the kittens into a normal, lit environment. The ones who had been allowed to move around in the dark were able to function just fine, while the ones who were tied down helplessly staggered around as though blind. What this seems to indicate is that perception is an active rather than passive process. The lights and images used to stimulate the kittens didn't make much difference; being able to explore their surroundings and perceive things in motion relative to their own bodies did. To perceive, you need to interact.

Another study (Bach y. Rita, 1972) did something similar with a bunch of blind folks. The researchers created a special video-camera-driven matrix of stimulation points, shown in Figure 2.13. A TV camera (mounted on spectacle frames) sends signals through electronic circuitry to an array of small vibrators (left hand) strapped to the subject's skin. The pattern of tactile stimulation corresponds roughly to an enlarged visual image.

Each vibrator was mapped to a particular pixel of the image that the video camera was receiving, giving a sort of tactile image of what the video camera saw. When the participants were allowed to move the camera themselves, they were able to learn to "see" in a limited way. If they were not allowed to control what they were "looking" at, the image stimulus device was just a gentle, if unskilled, masseuse.

The concept that perception requires action has relevance to game feel because it accurately describes the sensation of exploring and learning your way around an unfamiliar game space. And it correlates physical reality with virtual reality in a meaningful way: the thing you're controlling in the game becomes your surrogate body, your hands. Humans are adept at learning the physical properties of a new and unfamiliar

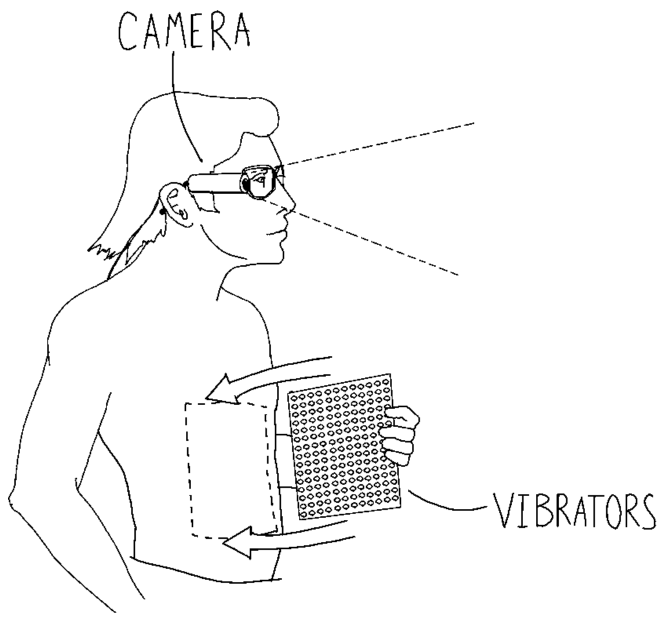

FIGURE **2.13 A blind subject with a tactile visual substitution system.**

object and do it very quickly. Noodle it around in your hands and you soak up a wealth of detail: weight, density, material, texture, color and so on. This same ability extends to virtual objects and, interestingly, to virtual worlds governed by different rules, laws and physics. For some reason, it's immensely pleasurable to suss out a new and unfamiliar world by probing around in it using a virtual instrument. The thing being controlled becomes both expressive and perceptual: as you control it and move it around, feedback flows back through it to your eyes, ears and fingers.

Perception Is Skill

If perception requires action, that action must be learned. We don't usually think of it this way, but perception is in some ways just a set of skills, honed across one's lifetime. From the moment we're born, we learn new distinctions, forge new neural pathways and generally undertake an ongoing process of becoming better at perceiving. Grab keys. Insert keys in mouth. Rinse, repeat and learn; so it goes until a fully functional adult emerges. "Perception is to a large extent an acquired bodily skill that is shaped by all our interactions with the world," as Dag Svanaes says.

Part of this process is making generalizations and learning abstract concepts like justice, freedom and cheesecake. Another part is bringing with us all experiences that came before. There's a concept from two venerable psychologists, Donald Snygg and Arthur Combs, that vividly portrays the role past experience, ideas, generalizations and fantasies play in perception: the perceptual field.[3] People have a great deal

[3]Near as I can tell, the idea started with Snygg and Combs in 1951 as the "phenomenal field." Combs changed phenomenal field to perceptual field in a later work, so we'll stick with that. I prefer that anyway, as it's more descriptive of the concept as it applies to game feel.

of practice navigating the space around them and developing relationships with the things in that space. Snygg and Combs capture the phenomenon of memory, perception and skill building with their concept of the perceptual field.

The idea of the perceptual field is that perception is carried out against the backdrop of all previous experience, including our attitudes, thoughts, ideas, fantasies and even misconceptions. That is, we don't perceive things separately from what's come before. Rather, we experience everything through the filter, against the background, and within the structure of our own personal vision of the world. Another way to put it is this: "one's constructed representation of objective reality; the meaning given to the profusion of stimuli that bombard the brain and are organized and conceptualized on the basis of individual and personal prior experiences."[4] And still one more way: "The perceptual field is our subjective reality, the world we are aware of, including physical objects and people, and our behaviors, thoughts, images, fantasies, feelings, and ideas like justice, freedom, equality, and so on."[5]

So your perceptual field is your world; your structural understanding of everything you perceive around you and its meaning.

This is a cool idea because it goes a bit beyond the notion of a simple mental model, which is more about the dry, clinical details of how a person thinks a thing functions (usually compared to the system image, the way the thing actually functions).

In the case of a video game, the brain recognizes that the "world" of the game is a subdomain, a microcosm of the larger perceptual field of reality that it understands. It's apportioned, separate and obeys its own rules. At the same time, the mind brings all its past stored experiences to the table to help it understand this new place.

The difference is that a game world is not necessarily governed by the rules of physical reality or bound by them. This is a useful way to think about the creation of game feel—as the creation of a separate but related physical world. Simplified, but whole, cohesive and self-contained. Many times, creating and tweaking a game feel system means literally the construction of a set of generalized laws and rules that govern all action within the system. It's like writing your own universe from scratch: you have your own gravity, your own simple momentum and friction, your own simplified definition for what collision between two objects means and how it should be resolved.

The world around you is objective and immutable. You're not going to wake up one morning and discover that gravity has suddenly stopped working as expected. I throw a grapefruit at a wall, it's going to hit the wall with a thud and fall to the ground. This kind of consistency can be frustratingly difficult to achieve in a game world.

Because the way we cope with and understand events in a game world is so similar to the way we interact with the real world, we expect the same consistency. The tiniest thing can break this perceptual immersion.

---

[4]http://phenomenalfield.blogspot.com/
[5]Dr. C. George Boeree, http://webspace.ship.edu/cgboer/snygg&combs.html

Unlike a film, which presents one framed perspective on a world and has only to maintain visual and aural consistency within that frame, a game has to stand up to active perception. The player can freely explore every possible permutation of every action and response in the entire world. This is the skill of perception, the one every human being has been constantly honing since birth.

Because people are extremely skillful at perceiving the world around them, any tiny inconsistency becomes glaring and obvious. A character's foot clipping through a stair, an invisible wall—these things never happen in the world around us. So this process of discovery is much closer to our experience of being in the world than the experience of passively viewing a film or reading a book.

The same mechanisms we use to cope with the world around us, to understand it and function within it, come into play when exploring a game world. It is the same process of expanding the perceptual field into this new world and probing around to make the generalizations and distinctions needed to interact with and succeed in the world.

This is why a consistent abstraction is so much more important than a detailed one. It's fine to create a simplified version of reality for players to interact with; they'll figure out the parameters of the world—its constraints, rules and laws of physics—within minutes of picking up the controller. Just don't violate the rules you yourself have set. If an object is portrayed as being large, heavy and massive, don't let it go flying off into space with the slightest touch, or pass through another object without interaction. Easier said than done, of course, but this is at some level a designer's decision. Better to have a simple, tight, cohesive world like Dig Dug than a weird, inconsistent world like Jurassic Park: Trespasser. People are going to figure out everything about your world either way—our physical reality is much more complex and nuanced than any game world, and we've got years of experience at perceiving it—better to make it simple and self-consistent than a broad inchoate mess.

Perception as skill is also correlated to the way people get better at things over time. You practice something, you get better at it. Your perceptual field includes an ever-growing body of experience about the task, which makes each new try at the task easier. It's not just a bank of information to draw on, but affects what happens at the moment-to-moment level of perception and action. Chris Crawford would say that the neural pathways are getting pushed farther and farther down, away from conscious processing and into subconscious, automatic processing.

Merleau-Ponty defines this distinction as abstract versus concrete movements. If an action is unpracticed and requires conscious thought and effort, it is an abstract movement. If it's so practiced that it happens automatically and without conscious thought—a pure translation of intent into action—then it's considered a concrete movement. By this process, people learn things incrementally over time, turning what was once difficult and requiring deep, conscious involvement, into an easy and subconscious action. Clearly, this is what happens with game feel, though the process is almost always much quicker when learning a game skill.

However we conceptualize it, it is obvious that humans get better at skills they practice. And if we consider all perception to be a skill, this handily explains why game feel seems like such a skill-driven activity, why skill learning is the price of admission to experience it. Perception in a game world is just a simplified, modified version of perception in the real world. The rules are different, but the process is the same.

Perception Includes Thoughts, Dreams, Generalizations and Misconceptions

Another interesting offshoot of the concept of the perceptual field is that it encompasses not only physical reality, but attitudes and ideas. Perception of something is heavily influenced by biases, ideas, generalizations and worldview, all of which have become incorporated into the perceptual field through a lifetime of experiences. Of course, generalizations about a particular thing may not accurately reflect the objective reality of that thing.

For example, I have a central heating and air conditioning system in my apartment. One day I was cold and wanted to adjust the temperature. The wall mounted control unit displays the current temperature based on a thermometer which is … somewhere in the apartment. I hope. I think. The current temperature is compared to a sliding blue control representing the desired temperature. I assume that if I move the sliding control to a temperature different from the actual temperature, the air will turn on and will adjust the temperature. The current temperature reads 61°. I move the control from 63 to 75. The cold air turns on briefly and then turns off. A few minutes later, the air turns on for a longer period, blowing warmer air, then turns off. Eventually I get fed up and turn the temperature to 90°. Again, the air turns on for a few minutes, warm, and then turns off. I get a blanket. An hour and a half later, I walk into my office, the door of which has been closed and I'm suddenly sweltering hot and realize I've left the temperature on at 90°. I toss the blanket, strip down to a T-shirt, and check the thermostat, which now reads 77°. I put the thermostat back down to 72. The air turns on for a few minutes, cold, and turns off. Gah!

What's going on here? In my conceptual model of the heating system, there is a certain threshold between the actual and desired temperature. If that threshold is exceeded—if the desired temperature is 3° above or below the actual, for example, the system turns on and applies an amount of heat or cold equal to the change in temperature that's necessary to bring actual back in line with desired.

The reality of the system is that it turns on and blows air, either hot or cold, for a certain duration. Five minutes of hot or five minutes of cold, depending which way the temperature is supposed to go. The hot isn't particularly hot, and the cold isn't particularly cold, but applied across a long enough time scale, they get the job done. In addition, the timer that governs how long each on state should last functions independent of the controls. That is, it's always counting down five minutes at a time. If you switch the heat on with 30 seconds remaining on its five minute cycle, you get 30 seconds of cold air because it takes a while for the heater to warm up and there are only 30 seconds of "on" remaining before the system goes back to the off state. (Oh, and the thermostat, evidently, is in the office.)

Donald Norman would say that my mental model is out of sync with the "system image" of the heating system. I hold in my mind a logical construct, a picture of how the system works, but that picture is wrong. This construct frames my interactions with the system and dictates my expectations about what the result will be for a particular input. If my mental model differs from the system image, the way the system actually functions, errors occur. Norman would say that this is the designer's fault and go on to point out that the "design model"—the way the designer imagined the user would interpret the system—is out of sync with the user's mental model. This is definitely applicable to game feel as it helps greatly in designing what players refer to as intuitive controls. Norman suggests seeking out and exploiting "natural mapping" between input device and system.

For example, in Figure 2.14, Stove C makes more sense than stoves A and B and is far easier to operate because there is a clear, spatial correlation between the position of the knobs and the position of the burners they operate. Finding natural mappings in games is a similar process, though not always an obvious one. I like the

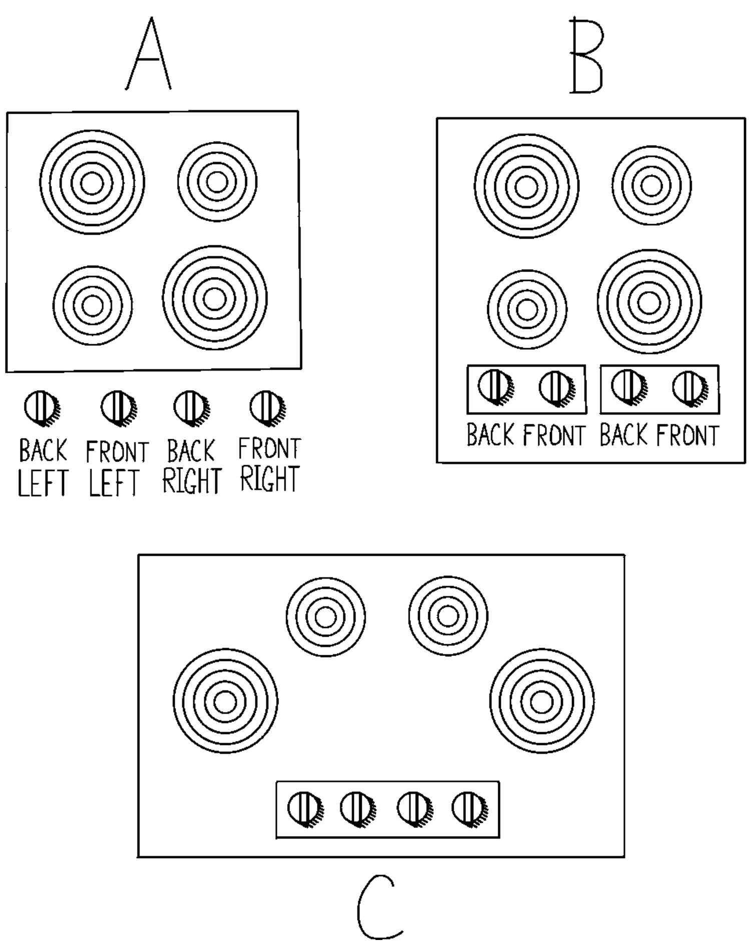

FIGURE **2.14 Three configurations offer three different correlations between stove burners and controls.**

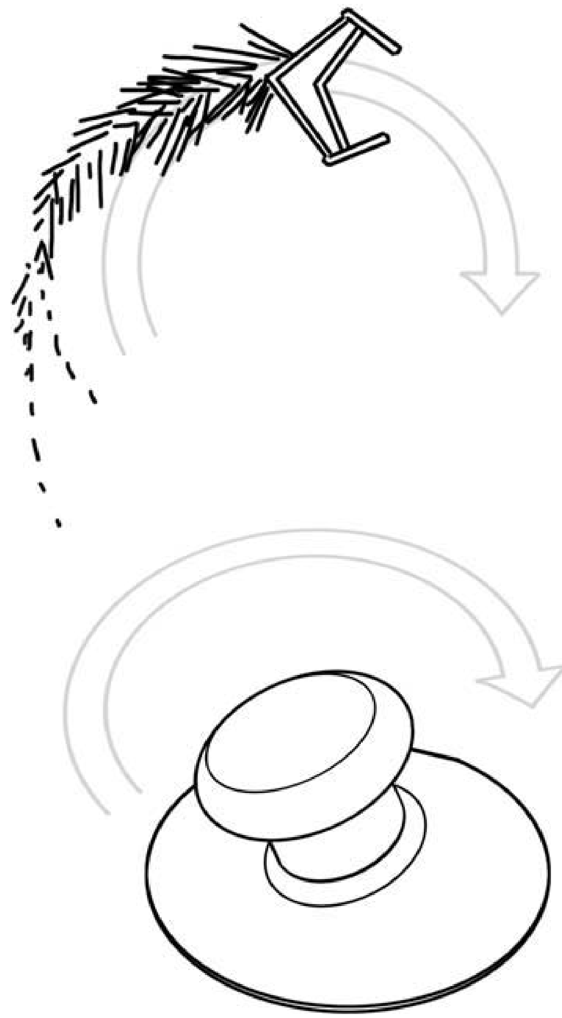

FIGURE **2.15 The motion of the thumbstick in Geometry Wars offers a simple and clear connection to the action taken by the ship.**

example of the game Geometry Wars, which tends to be easy for people to pick up and play effectively—the movement of the thumbstick corresponds very obviously to the motion of the little ship in the game making this something of a natural mapping, albeit one in a virtual space (see Figure 2.15).

The only problem with Norman's concept of the mental model is its rigidity. To say I have a mental model of the world of The Legend of Zelda: The Wind Waker could be useful to isolate and tune certain mechanics, to eliminate ambiguities and make the controls more intuitive, but it ignores the fact that I have a relationship to that world. I feel a certain way about it and think a certain way about it, and breaking that down into a dry, clinical, logic diagram seems to ignore some of the most important parts of that relationship, some of the parts the designers understood and worked really hard to develop.

For example, it's extremely important to the success of the game that I feel a sense of freedom and adventure while sailing the open seas in the game. I could distill that to a system image of my possible actions while sailing, but that would miss much of the point; it's not as important to track exactly how far objects are spaced apart or how far I can sail in one direction before hitting something or how long it takes to get from object to object as it is to make the space of that world feel open, free, but full of possibility. I can sail aimlessly in a particular direction and be certain of two things: 1) I'll be free to sail as far as I want in that direction and 2) eventually I'll find something new and interesting. The system, speed of the ship, sharpness of turning, distance between islands and so on, would be interesting to track, as would a player's mental model of that system. But it definitely misses one of the essential experiential qualities of the game. A few small changes to the system, making the ship move 20 percent slower for example, could make the oceans of Wind Waker feel imposingly large, tedious or lonely. The idea of a perceptual field that incorporates not only the system image, but the thoughts, ideas, feelings and generalizations about the system that players have brought with them, and is constantly forming and reforming, is a much more effective for understanding that experience.

Another important thing that Norman's mental model concept ignores is paradigm shifts. "Aha!" moments, as master puzzle designer Scott Kim would say. For example, another Zelda game I've been playing recently is The Legend of Zelda: Phantom Hourglass for the Nintendo DS. In that game, there is a particular puzzle that requires players to expand their perceptual field. In one particular part of one temple, players are told to step up to an altar and "stamp" their "map" with the location of a new area to be explored later (see Figure 2.15). At first, this puzzle threw me for a loop.

To access new areas in The Legend of Zelda: Phantom Hourglass, you first need a "sea chart" for that area. The new location to be stamped is outside of your current available areas, but clearly corresponds to the map you've had access to and used throughout the game. I systematically exhausted each possible action I had ever used in the game, and then some I'd only ever used in other games. I pressed buttons, I scrubbed the stylus back and forth in every conceivable pattern. I drew X's and O's, I traced the pattern of the Triforce over and over again. Nothing in my perceptual field equipped me to deal with this puzzle. Clearly, my current understanding of this system was inadequate and flawed somehow. I had to take a step back and reflect on my perceptual field. So I started thinking about other possible interpretations of stamping, and other possible ways I might stamp this stupid map, short of putting the DS on the floor and stamping it with my foot. The solution eventually occurred to me: close and open the DS quickly. It's sort of a stamping motion, as you can see in Figure 2.16.

The problem was my perceptual field: in my combined understanding of the DS, how the DS functioned and the possible actions in the game, there was no reference point for using the functional opening and closing of the DS as an action in the game. Across all my experiences with playing DS games—which are numerous—I had never come across a game that used the functional closing and opening of the DS' lid as a button. New Super Mario Brothers says "Goodbye!" in Mario's voice if

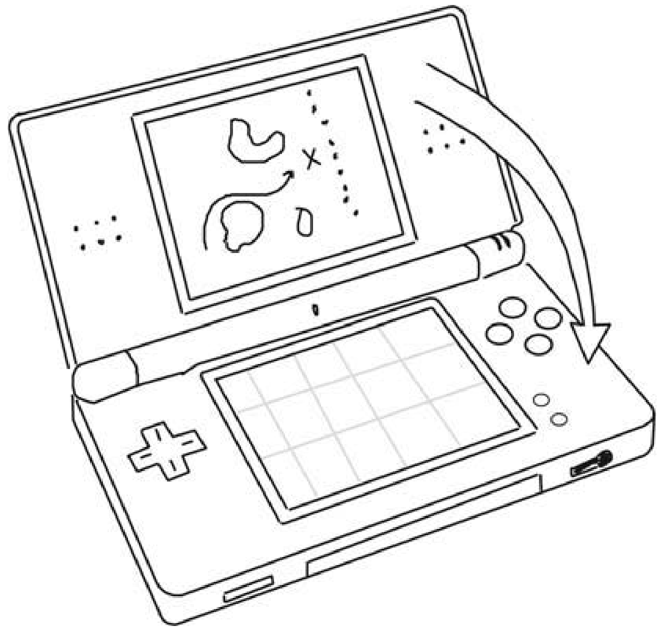

FIGURE **2.16 Closing the DS as an input = unexpected.**

you close the lid of the DS while the game is still on, but that was my only point of reference, my only hint in the entirety of my perceptual field. I solved the puzzle through a paradigm shift, by seeing the system in a new and different way. I felt a rush of pleasure and enjoyment as I incorporated this new information about the very nature of this game's universe into my perceptual field. I had amended and changed my mental model not only of this particular game, but of all games I'll ever play on my DS in the future. This was not an error or a breakdown in my ability to interact with this system, but the point of the puzzle and, in some sense, the whole game.

So what's going on here? I was rewarded with pleasure by expanding my perceptual field. Norman's mental model would call this an error, blaming the designer for misleading the player. Since this is one of the fundamental pleasures of playing this game, clearly something's missing. The perceptual field gives us a way to understand that game feel is as much about how players feel about a particular space and their relationship with it, as it is about the dry clinical details of the mental model of that space they keep in their minds to help them deal with and understand events in that game world. The dry details are important—they represent the player's understanding of the physics and rules of the game world and are a great way to find dissonance that causes player confusion—but they are not everything.

Perception Is a Whole-Body Experience

Eyes, ears, tactile sense, proprioception—there is no separation when a person perceives something.

For example, a fork. It's shiny and it has a pointy end. It's cold, hard and solid, but easy to pick up. I can eat food with it. It sinks in water. My ideas and perceptions about what that object is, how it behaves and what it means, combine into the concept of "fork." This is my attitude toward forks and generalizations I can apply later to other fork-like objects. The subconscious nature of perception masks a complex process, one which involves all the senses and which constantly and rapidly brings us closer to the world we inhabit.

The takeaway here is not to think of each kind of stimuli as somehow separate, but as an integrated part of perception. This is how a combination of visuals, sounds, proprioceptive sensations (from the position of the fingers on the controller or whatever) and tactile sensations (from controller rumble or haptic feedback) become a single experience in a game. A game world substitutes its own stimuli for those normally created by interacting with the real world, but the experience of perception is much the same. This also indicates why we're so sensitive to inconsistencies between stimuli. If a large hulking mass of a character steps through a staircase or has their arm pass through a wall, the brain says, "Hey, that's not right!"

The experience of perception of real-life phenomenon never has inconsistencies across stimuli so the brain has a hard time ignoring them when they happen in a video game.

Tools Become Extensions of Our Bodies

As we said in Chapter 1, a tool, once picked up, is an extension of the senses. It is used for both action and perception. Intent and action can be expressed through the tool as though it were a part of the body, and feedback flows back through it.

Another way to visualize this is to think about a blind man's cane. When the blind man first starts using a cane, it is unfamiliar and requires a lot of reflection. The tapping movements are unpracticed, abstract movements to him. As he gradually builds skill at perceiving the world through the cane, he is able to more accurately and effortlessly tap around, getting a clearer read on his surroundings. His intent begins to flow effortlessly from the cane, any barrier between himself and the cane is removed, and the cane becomes an integrated part of his perceptual field. The cane now acts just as his hand would; it probes around, touching things, interacting with the world around him and returning him crucial orienting feedback. This helps him establish a much larger personal space (also sometimes called the "perceptual self"). Quite literally, his perceptual field has been extended to a much larger physical area around him. Effectively, he's made his arm's reach much greater. We might say that by integrating the tool of the cane, he's changed his world.

So what does the blind man and his Cane of Reaching +3 teach us about game feel? One interesting distinction is between body space and external space. When we interact with the world, we perceive our bodies in two ways, as part of our self and as one object among the many objects of the external, objective world. As Dag Svanaes puts it: "The bodily space is different from the external space in that it exists only as long as there are degrees of freedom and a skillful use of this freedom. The bodily space is mainly given by the subject's specific potentials for action. For a totally paralyzed body with no kinesthetic experiences, there is no bodily space. Different bodies give rise to different spaces, and so do external factors such as clothing, tool use and different kinds of prosthesis. It is important to notice that learning a new skill also changes the body space."

This idea, that bodily space is defined by the potential actions of one's body in the world, correlates very clearly to the way players interact with game worlds. Players tend to think in terms of abilities and constraints within a game world. It is possible, and often desirable, to create a game where abilities change across time, where the avatar itself has different tools that change across time. For example, if I'm playing as Samus Aran my abilities, my "bodily space" as Samus within the game world of Metroid, are defined by the abilities currently available to me. I may or may not have the Morph Ball. If I do, I can transform into a small rolling sphere and explore tiny corridors. The very nature of the world has changed, as has my potential to interact with it. The objective world of Metroid is the same as it ever was—every block is still in the same position it was before. My abilities, my verbs, my virtual bodily space have changed the world.

The interesting thing about this thinking is that it does not consider Samus Aran a tool. To say that we can incorporate a tool into our bodies, that the tool can become an organ of expression and perception, and that our identity and perceptual field subsume it is useful, doesn't quite define what happens when a player takes control of an avatar in a game. I wouldn't call Samus Aran a tool. Not just because she has an "identity" as a freestanding character that I'm temporarily inhabiting and controlling, but also because she has her own bodily space. She has her own tools which can become a part of her body and extend her own perceptual space. In this way, a video game world is truly a microworld, and perception within this microworld is a surrogate for real-world perception. It's an interesting idea—it seems to explain why identity in a video game is so very malleable. You can go from "being" Gordon Freeman one moment to cursing his vile clumsiness the next. This is because the constructed subdomain of a video game reality provides two other kinds of spaces: virtual bodily space and virtual external space.

A video game has its own model of reality, internal to itself and separate from the player's external reality, the player's bodily space and the avatar's bodily space. The avatar's bodily space, the potential actions of the avatar in the game world, is the only way in which the reality of the external reality of the game world can be perceived. As in the real world, perception requires action. The difference is that the action in the game world can only be explored through the virtual bodily space of the avatar. Players extend their perceptual field into the game, encompassing the available actions of the avatar. The feedback loop of perception and action that enables you to navigate the world around you is now one step removed: instead of perceiving primarily through interaction of your own body with the external world, you're perceiving the game world through interaction of the avatar. The entire perceptual apparatus has been extended into the game world.

To wrap back to our earlier discussions of identity and game feel, how does this concept of avatar as perceptual substitute, rather than extending tool, relate to proxied embodiment? Because a game world represents its own reality external to its avatar's bodily space in the same way that the physical world is external to our own bodily space, it seems much more like a substitution than an extension. The same might be said for identity. We said that objects outside ourselves—and objects in a game world—can become extensions of identity. Vessels for identity might be more accurate. The view of tool as extension of body defines the "self" is in terms of perception. The perceptual self is the immediate surrounding environment and your ability to interact with it, your potential for action. To say "he hit me!" instead of "he hit my car" or "his car hit my car" is an artifact of the way we perceive the immediate environment around us and the fact that an inanimate object can become a part of the perceptual self, part of the perceptual field. You literally perceive the world through the car as you actively control it. Again, though, the way we perceive game feel seems to be much more of a substitution than an extension. I perceive the world of Hyrule as Link, via his virtual body space. My identity intermingles with Link's as I take over and make my own his skills and abilities, his bodily space.

Summary

Where and when does real-time control exist? On the human side of the equation, we categorized three types of processors (perceptual, cognitive and motor), which work together in a closed feedback loop. This feedback loop results in an ongoing correction cycle. In a video game, at the point where action normally goes out to physical reality, the designer substitutes a game world for the real world. This reinforces the idea from Chapter 1 that game feel is an experience of a unique physical reality.

On the computer side, real-time control relies on sustaining three thresholds: the impression of motion, perceived instantaneous response and continuity of response. Knowing the duration of the human correction cycle and the relationships between the three human information processors, we can say with certainty that a particular game has or does not have real-time control. The unknown variable here is the player's perception. In the end, game feel is an impression in a player's mind. Examining the frame rate, response time and continuity of response in a game and comparing this to the thresholds of 10 fps for motion, 240 ms for control and 100 ms for continuity gives us a baseline and is useful for classification. But motion at 10 fps feels stilted and a 200 ms response time feels sluggish. These impressions can be smoothed over by the use of gestural input or by playing back animations. In the end, the player's perception is what's important. The ultimate goal for game feel is to create an impression in the player's mind.

Finally, we looked at some of the other implications of human perception:

1. Perception requires action.
2. Perception is a whole-body phenomenon.

3. Perception is an effortless fusion of visual, aural, tactile and proprioceptive stimulus.
4. Perception is an ongoing process of skill-building.
5. Perception can be extended to tools.

These explain, in terms of human perception, the experiences outlined in Chapter 1. Understanding how human perception works offers us insight about the imperfect apparatus of human perception that we're designing for. Knowing this will help us to develop a palette of game feel that's separate from the emulation of reality and doesn't borrow from film or animation except where applicable. If we understand how perception works, we can build games that feel good instead of trying to build games that feel like things that feel good.

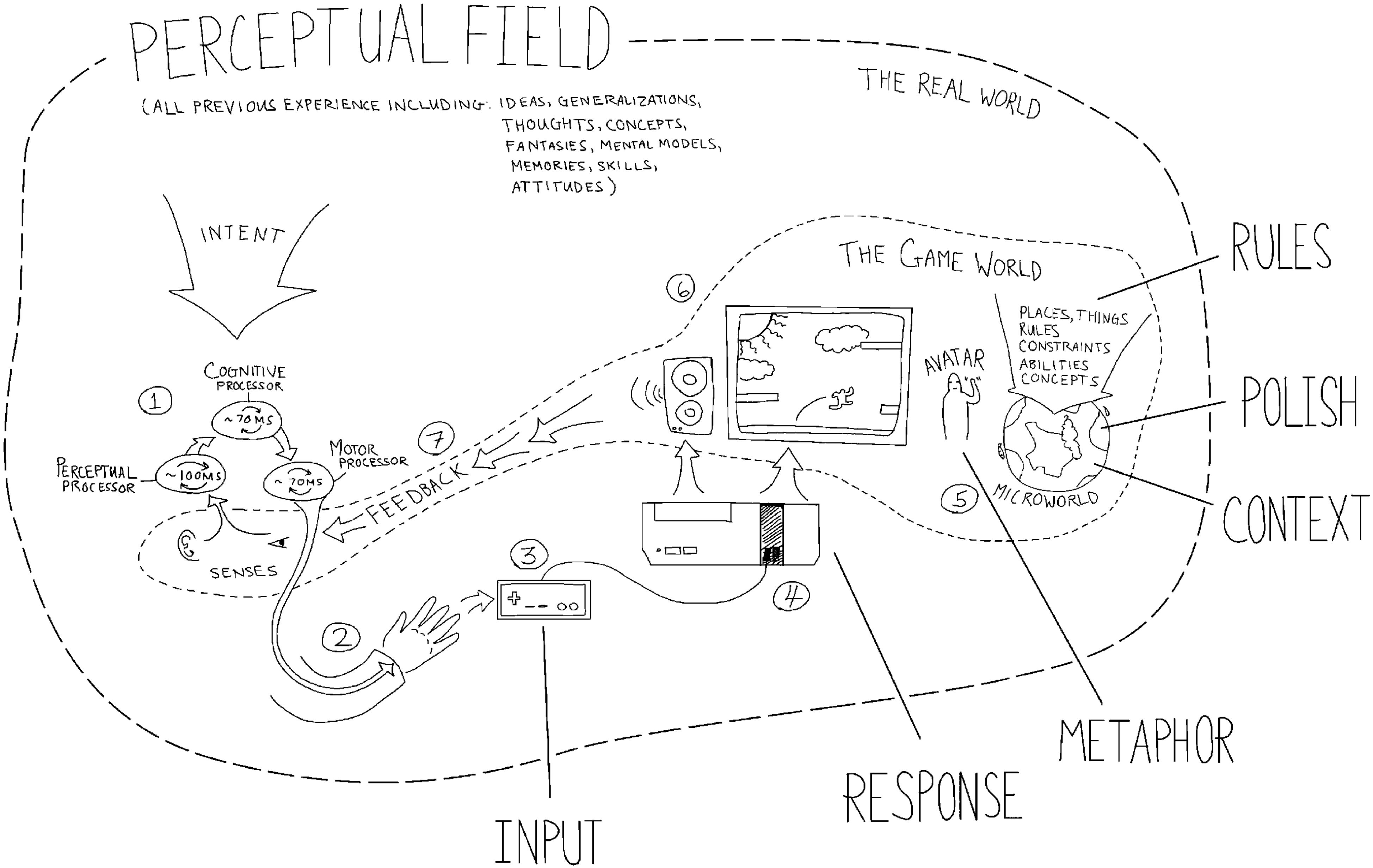

FIGURE **3.1 The model of interactivity brings together all the elements of the the gamer, the game and the world around him or her.**