Chapter 4

CONTEXT

Games are made of rules, and those rules allow us to create choices for the player. But those choices exist only when the player understands those rules. Context is what helps a player to internalize those otherwise-abstract rules that make up our game. Digital games have the capacity to use visual art, animation, music, and sound to shape that context and communicate with the player. These channels of communication can say a lot.
First Impressions

I played an early version of Super Crate Box (2010) before it was available online. In this game, boxes appear at random positions on the screen; the player is scored by how many boxes she can collect. Green monsters of various kinds—little and fast, big and slow, monsters that run straight ahead and ones that actively chase the player—drop from a hole in the top of the screen and move toward the bottom. If they make it all the way to the bottom of the screen without being shot, sliced, exploded, or murdered in some way by the player, they fall down a hole in the bottom and reappear from the hole in the top, faster than before and red now instead of green (see Figure 4.1).

![Figure 4.1 Monsters wrapping around the screen and becoming red.](image)

So strategy in this game is a balance between collecting crates and keeping the monster population in check using the weapons available to the player. If the player collects a box, which contains a randomly chosen weapon that she must then use, forcing her to change her strategy. This is the dynamic of the game.

Here’s the part that confused me, though: in the early version of the game I played, while the monsters re-emerge at the top more powerful (and red) after they jump into the hole, when I jump my little character into the same hole, it dies. This was, to me, a jarring lack of consistency.
As a designer, I understand why the authors would want to keep the player in a bounded area that emphasizes horizontal motion as much as vertical motion. And I understand why they would want the monsters to travel on a path between introduction and escalation. But as I played, the game failed to communicate an important rule: I can't jump in the pit like the monsters can. And the game didn't help me remember this rule, either; despite dying the first time, I jumped into that deadly hole again and again during the chaotic frenzy of the game.

The solution that the creators of Super Crate Box came up with was a simple visual effect: they set the hole on fire (see Figure 4.2). Red flames now dance in the hole at the bottom of the screen. If the player falls in, she's fried. If a monster falls in, it's cooked red hot and released, furiously speeding up as it drops from the top of the screen. I can buy this: it's just a small visual change, but it provides a justification for why the hole kills me but transforms monsters. It provides context.

![Figure 4.2](image1.png) ![Figure 4.2](image2.png)

*Figure 4.2* Both versions of the hole at the bottom of Super Crate Box are deadly, but the one on the right makes that visually obvious.

At a mechanical level, games are about rules and interactions. We introduce things that are dangerous to the player so there can be a conflict, so we can develop the player’s verbs: you’ve got to be good at moving to avoid the hazards hurtling toward you. You’ve got to understand how to direct your shots to shoot the monster. But in the abstract, our rules are just that: abstractions. Firing a gun, running around and avoiding dangers—these are complex, nuanced activities. We don’t want to make a simulation of running (caveat: sometimes, maybe we do). We’re trying to tell a story. So we abstract.

How does the player know that the blip is dangerous? How does she know not to touch it? Well, it could look like a dangerous monster, all fangs and claws. What if instead of meandering, it moves toward the player, aggressively? It could move slowly, a shamble of heavy footsteps that shake the screen. How does the player know shooting it is a good thing? Well, when the blip is shot, a melody could play suggesting relief from danger.
Digital games, running on screens, breathing through speakers, are capable of presenting visual and audio information to the player. Images, animation, sounds, music. We can use these channels to communicate the rules of our games, to emphasize and underline the important interactions.

Look at a game like *Plants vs. Zombies* (2009). In this strategy game, the player is trying to defend the left side of the screen by erecting defenses, each with a different function, that cannot be moved once positioned. The player must choose what defenses to build and where to build them. Different kinds of defenses have to be positioned to support other defenses. Enemy forces appear on the right and move slowly toward the left, giving the player time to adjust her strategy and make decisions.

The battlefield is a lawn. At the far left side of the screen is a house, ostensibly the player’s house. The defenses are plants, rooted in the earth. The enemy forces are zombies. These images suggest things about the properties of these game objects, about the rules that govern them. It makes sense to a player that when a plant is planted in the earth in a particular place, it can’t be moved. Of course, zombies move slowly. The way the game pieces are contextualized gives the player expectations about how they work.

*Super Mario Bros.* is a game that initially teaches the player a simple response to any opponent: jump on its head. It would be interesting to have a creature that Mario couldn’t just jump on top of like all the others, one he had to carefully maneuver around. But when the player is conditioned to resolve all conflicts by jumping on the creature, how do we communicate that jumping, in this one case, is dangerous? Look at the spiny beetle in *Super Mario Bros.* (see Figure 4.3).

---

**Figure 4.3** The spiny beetle looks a little more dangerous than the other enemies of *Super Mario Bros.* for a reason.

All the other creatures in the game have smooth domes, round noggins, or tortoise shells. Spiny has tall, jagged edges, gleaming and dangerous. This little guy you don’t want to jump on.
Compare this creature to another creature in Mario’s bestiary you don’t want to jump on—the piranha plant that comes in and out of pipes. Its purpose is to create platforms that are temporarily safe and sometimes dangerous. The piranha plant’s mouth, full of gnashing teeth, points upward (see Figure 4.4). Mario doesn’t want to touch this thing from above.

![Figure 4.4](image1)

**Figure 4.4** The visual context of the piranha plant’s gaping maw helps explain why Mario might not want to jump on it.

Interestingly, one of the first creatures the player meets in *Super Mario Bros. 3* (1988) is a “Venus Firetrap” plant that blows fire toward Mario’s current position. Unlike the piranha plant, this plant creature has its head turned to the side, aimed at Mario (see Figure 4.5). As a consequence of where this plant’s mouth is facing, the top of the creature is a smooth, round surface. One of the first things I thought to do when playing the game was jump on its head. Mario died, for no clear reason. This is a case of visual miscommunication.

![Figure 4.5](image2)

**Figure 4.5** In a break with visual conventions of earlier games, the firetrap of *Super Mario Bros. 3* might make you think it’s safe to jump on its head. It’s not!
Recurring Motifs

*Super Mario Bros.*' spiny beetle and piranha plant have a similar property: you can’t jump on them. They also have a shared visual property: their tops have spikes. This is the kind of symbol that can become part of a game’s ongoing visual vocabulary. If the player encounters a distinctive-looking object that obeys one game rule, she’ll expect a similar-looking object to obey the same rule later. Visual design in games is all about shaping the player’s expectations.

In *Spyro the Dragon* (1998), one of the protagonist’s primary verbs is, naturally, to breathe fire. The player uses this fire to defeat opponents and break open chests full of valuables. But not everything in the game can be defeated by fire. The game uses metal consistently to indicate things that are impervious to fire, such as armor on opponents or certain types of chests. This metal is a visually distinctive gleaming silver. If the player uses fire on it, it heats up red for a moment to acknowledge the interaction between the fire and itself, and then it cools back down to its normal silver. This is usually a signal that the player should try her other verb, a charging headbutt, on the opponent or container.

It’s not just armored opponents and metal boxes that use the visual motif of metal being heated up, however. As the game progresses, other objects appear that react to fire—containers that, when torched by Spyro’s flaming breath, expel gems that levitate above them on a geyser of hot air, and switches shaped like fans that turn when heated to open gates. These things are also metal, to indicate that they can’t be destroyed by fire, but that they will change. They heat up just like the metal chests the player first encounters, but the heated-metal motif has developed further; unlike the chests, the gem spouts and fan-switches change in a meaningful way.

Michael Brough’s *Zaga-33* (2012) is a strategy game about navigating an alien planet that is generated anew by chance every time you play. The positions of the walls, the terrain, where the monsters appear, the artifacts the players can use—all are random. In fact, the appearance of the useful artifacts is randomized as well. A four-pointed cross can be a laser weapon one play, a healing item the next, or a device that rearranges the walls of the room. Upon using an item, it’s identified for the rest of that play—the player knows that the lollipop-shaped thing freezes monsters in place this time. But out of necessity, none of the images for artifacts can convey anything about its use. They’re all weird squiggle-shapes.

But there has to be some visual consistency between them, right? The player needs to know that this thing in the corner of the room is an artifact she can pick up and use, albeit one whose purpose she hasn’t identified yet. She needs to be able to tell the difference between an abstract-shape artifact and a dangerous monster.

Brough accomplishes this by giving each set of objects a unique, consistent color palette (see Figure 4.6). Artifacts, regardless of shape, are always orange and tan. Monsters are always bright green (with small orange highlights). The walls are green and gray, and the floor is black and
purple. Each element of the game world is identified by its distinct colors. The player may not yet know what the lollipop-shaped thing does, but she knows it’s an artifact she can use. Once again, there’s a visual motif; in the abstract world of Zaga-33, the motif can be as simple as orange versus green.

![Comparison of items and monsters in Zaga-33.](image)

**Figure 4.6** Comparison of items and monsters in Zaga-33.

---

**Character Design**

The way objects in a game appear should tell the player about what they do and what their relationships to each other and to the player are. It should also differentiate objects: objects that follow different rules should be visually distinguishable from one another. An easy way to do this is to pay attention to an object’s silhouette, or the shape of the object.

As an exercise, my friend Leon Arnott created his own hacked version of *Super Mario Bros.* called *Silhouette Mario Bros.* He blacked out all the characters in the game, Mario and monsters alike, so that they appear as solid black shapes. And of course, he discovered that everything in the game is still perfectly recognizable. The enemy turtle, which becomes a projectile when Mario jumps on it, is a different shape from the enemy mushroom, which does not. The beetle that is safe to jump on looks different from the beetle that is dangerous to touch. Look at Figure 4.7. Can you recognize these characters?

![Mario characters in silhouette.](image)

**Figure 4.7** Mario characters in silhouette.
The only information the player loses in silhouette is whether a turtle will stop at a ledge or reverse course—the game uses color, green or red, to differentiate that.

*Lesbian Spider-Queens of Mars*, mentioned in Chapter 3, “Scenes,” is a fast-moving game where the queen’s renegade slaves attempt to overwhelm her with numbers. It’s important that the player can tell at a glance what kind of opposition she’s dealing with—whether a slave is an armored slave, who can only be approached from the back or side; a gladiator, who will seek out and pursue the queen; an alchemist, who leaves a trail of fire behind her; or a princess, who’s worth a lot of points if the player can catch her before she escapes.

The game uses a few different signifiers to differentiate the silhouettes of the different varieties of slave. First, each has a different hairstyle. The initial, random-moving slaves have Mohawks, armored slaves have short bobs, gladiators have spikes, alchemists have straight, neck-length hair, and the princess has twin ponytails that bob when she moves (see Figure 4.8). These ensure that the slaves have different silhouettes even when wrapped in a cocoon by the queen—it’s important to the player’s strategy to know which threat will return if the queen can’t collect the slave before she breaks free.

![Different slaves of Lesbian Spider-Queens of Mars.](image)

Each slave also carries a different weapon—a trivial difference, because they all signify death if the slave manages to sneak up on the queen. The different choices of weaponry just serve to help distinguish the behaviors of the different slaves.

In addition to changing the silhouettes of the characters, the different weapons help to characterize their owners. The Mohawk slave’s dagger is the most diminutive weapon in the game—the least deadly threat. The gladiator carries an axe: a larger weapon, one that perhaps
suggests that she’s a more seasoned fighter and will behave more aggressively. The alchemist carries a sword—longer than a dagger, but less brutal than the axe. She prefers to avoid direct confrontations—her real threat is her fire. And the princess’ fencing foil is a class signifier; fencing is about grace and poise before deadly force. The princess can kill but would like to avoid confrontation entirely.

The armored slave carries a long spear and a large shield: the shield provides a context for why she can only be zapped from behind or from the sides. But a shield doesn’t suggest harm, it doesn’t convince me that she’s deadly to touch. As a player, I might expect her to push the queen with her shield before I’d expect her to kill the queen. That’s why the long spear reaches past the shield, to ensure that the front of the slave’s silhouette is a sharp point, not a flat surface.

Character design also differentiates the slaves from the queen—this is more important than it sounds, because the queen’s position is the most important position on the screen. The player needs to be able to identify it immediately. This would be difficult if she wasn’t easily distinguishable from all the other moving characters on the screen—and there are usually a lot.

The queen has four legs (when viewed from the side), which makes for a different silhouette compared to the slave with two legs (see Figure 4.9). Also significant is the color. As with Zaga-33’s monsters and items, the queen is depicted with a different color palette than the slaves. Slaves are yellow, magenta, and red, sometimes white. (The princess wears white jewelry, and a cocooned slave is wrapped in white.) The queen is red and blue—blue being a much cooler color than any that the slaves wear. The combination of blue and red, in equal parts, also represents the most contrast on the screen—a natural place for the eye to home in on.

Figure 4.9  Comparing the silhouettes of the queen and slave.

A more subtle difference is that while every slave has a mouth and eyes, the queen’s face is blank. She has a nose and shadows that could suggest cheeks, but no eyes or mouth. This sets up a dichotomy that reflects the power dynamics in the game’s relationships: the primary use
of a slave's mouth, in this game, is to scream when she's cocooned by the queen. In contrast, the queen's lack of visible emotions suggests control, calm, the visage of a stone bust or some other symbol of a ruler. It's another visual reminder that this character operates very differently from the others.

Animators working outside of games have known these techniques for a long time—to make characters distinct and memorable, you can vary their silhouettes along with other visual cues like color. We have even more reason to use this idea in games, because we need to communicate and reinforce that objects in the game follow different rules.

**Animation**

Motion can be used to characterize actors, objects, and rules. The armored slave in *Lesbian Spider-Queens* moves more slowly than the others to compensate for the fact that she can only be approached from the back or sides: the player has to do more planning to capture this character. Consequently, she moves differently from the other slaves. Whereas most of the slaves in the game run, waving their weapons threateningly before them, the armored slave moves in a slow, mechanical march (see Figure 4.10). She keeps her shield directed squarely forward to remind the player that she remains unassailable from this direction.

![Figure 4.10](image)

**Figure 4.10** Walking animation of armored slave.

Players can tell what's important in *Lesbian Spider Queens* (see Figure 4.11) because these things, the living things, are moving, while the less important things, the walls and the scenery, are still. Maybe the player's character is moving even when the player's not moving her, rocking on her heels, tapping her foot, or looking at the player expectantly. Make sure the player can find her character at a glance, because her position is the most important piece of information on the screen.

In my game *Tombed*, the player's character Jane is animated even when she isn't actually moving. There's a simple, two-frame animation of Jane when she pushes against a wall (see Figure 4.12), and it serves an important purpose. As covered in Chapter 3, “Scenes,” exploiting the crushing spike wall's ongoing destruction of walls and obstacles is important in *Tombed*. There are many situations in the game that require the player to wait until an obstacle is removed by the spikes.
Figure 4.11  A screen from *Lesbian Spider Queens of Mars*; all the characters are animated, while the backgrounds and the structure of the maze aren’t.

Figure 4.12  Jane animated while pushing on wall.
I also designed *Tombed* to produce a lot of close calls between Jane and the spikes, so there are many situations where, in order to squeak by the spikes, the player will want to start moving as soon as that obstacle disappears. If the player wants to maximize her motion, she'll have to be holding the movement button before the obstacle is destroyed.

If Jane is standing next to a wall, and the player presses the key that moves Jane toward the wall and nothing happens, what does this lack of motion communicate to the player? What she might take from the lack of animation is that the game has not received her input, or that holding the movement key has no effect here, when in fact the time the player gains by holding the key early might be very important.

The pushing animation tells the player that her input has been received and is having an effect. It communicates.

Motion can be aggressive, cautious, panicked, or controlled. Look at any cartoon and pay attention to the way characters are moving when they’re scared, when they’re sneaking up on other characters, and when they’re giddy or gleeful. Motion can characterize the relationship between objects. In *Berzerk* (1980), the player and a bunch of hulking, broad-shouldered robots try to shoot each other in a maze of electrified walls (see Figure 4.13). Touching the walls means death to either player or robots.

![Figure 4.13](image)

*Figure 4.13  Berzerk maze, with player and robots.*

The mechanism that keeps the player moving through the maze is an invulnerable, pursuing robot named Evil Otto. When the player lingers in a scene too long, Evil Otto enters from the same direction the player did and begins to chase her (see Figure 4.14). Evil Otto is represented in the game by a simple smiley face. But the way Otto moves—a fast, high bounce like a rubber ball as it moves steadily toward the player—tells us a lot about its relationship to the player and to the maze.
Otto’s bouncing motion suggests it is above the maze, outranking the robots and the player alike, maybe connected in some way to the maze. More tangibly, its bounce provides an explanation for why the electric walls, fatal to all the game’s other characters, don’t affect Otto. It is literally above the maze. Otto’s slow (though faster as the game accelerates), unwavering pursuit of the player characterizes it as a threat, despite its smiling face.

Simple animations are often enough to convey important changes in state. A common signifier is blinking—making a character invisible every other frame—to convey that the character is temporarily invulnerable, as in a grace period after being hurt. A powerful opponent might flash red when struck to indicate that the player’s attack has been successful, but that the opponent hasn’t been fully overcome. A weapon glancing off an opponent harmlessly conveys the player’s attack has been ineffective. A character meandering randomly might suggest that it’s harmless. We have a rich vocabulary of animation to exploit in our games.

Scene Composition

Large images can tell us as much as small images. The composition of a scene can direct our attention to the most important part of a scene. Figure 4.15 shows the final scene of a game called Labyrinth of Zeux (1993) by Alexis Jansen. The object of the player’s quest, the Silver Staff of Zeux, is present in this scene, and you can see how the entire scene is designed, visually, to push the player toward it.

The T-shaped objects below the Silver Staff are poles the player can ride on. They form an arrow pointing at the Staff, and the poles nearest to the Staff are brighter. The Staff is positioned in the center of an open space, framed by a wide rectangle. The rainbow colors of the walls, the corners of every concentric layer, all point to the Staff. Everything in the screen urges the player upward toward the game’s ultimate goal.

The visually striking scene from l’Abbey des Mortes (2010) shown in Figure 4.16 is almost entirely portrait: there’s nothing for the game’s protagonist to do on this screen beyond stand at the window and look out. This tranquil scene, almost empty but for the tops of trees and a sprinkling of stars, represents a moment of peace in the middle of a larger, more hectic adventure.
The protagonist is a Cathar that has been pursued by Crusaders to an abandoned church—death is close at hand. For both the player and the protagonist, this scene represents a break from the rest of the game. The composition suggests just that: a view, a visual reward.

Figure 4.15  Drawing the player’s eye to the Silver Staff of Zeux.

Figure 4.16  Taking a break from danger in l’Abbey des Mortes.

The composition of a scene can suggest a lot about its nature. Figure 4.17 shows the first four scenes in Todd Replogle’s Monuments of Mars (1991). These scenes represent a trip across a barren Martian landscape to the entrance to a strange alien structure, shown at the right side. What marks the transition between Martian crags and industrial construct?
Figure 4.17 First four screens of Monuments of Mars show a change in landscape.

Where the Martian landscape ends and the structure begins, there’s a shift in the appearance of the shapes, from a rough, pebbly texture to clean, symmetrical girders. But there’s also the symmetry of the fourth scene itself, a marked contrast from the irregular surfaces of the previous scenes. Symmetry, here, speaks to the artificiality of the structure: here is something designed.

The robots move back and forth and hurt the player if she comes into contact with them. In the wasteland scenes, they move through open air, through ditches and pits, often more incidental than threatening. At the Martian structure, they move back and forth along the top like soldiers patrolling the walls of a fortress. They’re incorporated into the symmetry, moving inside those octagons like pieces of a clock. This is the place they come from, to which they belong.

The horizontal symmetry of the first three scenes points to the entrance to the Martian base, the hole in the center of the fourth scene. The vertical layout of the fourth scene, including two streams of green electricity, guides the player downward into the base. The flat desert scenes suggest lateral motion. This industrial scene, with its symmetry, shouts vertical.

Variations in composition can also say something about a scene. In Loren Schmidt and Mickey Alexander Mouse’s collaboration, Murder Simulator, the terrain alternates between smooth, straight, symmetrical hallways and rocky, uneven dirt. This suggests an incomplete artificial structure built into the earth, giving a reason for the scene’s irregular geometry.
Visual Shape

In the late Fukio Mitsuji’s *Bubble Bobble* (1986), players control bubble-spitting dinosaurs (see Figure 4.18). Their goal is to capture every scene’s dangerous monsters within bubbles and then to pop the bubbles, destroying the monsters. The dinosaurs can also ride the bubbles, hopping on them as they float through the scene. Bubbles can be popped individually or en masse in one big clump for lots of points.

![Figure 4.18](image1.png) Dinosaurs using bubbles in *Bubble Bobble*.

In a later scene, a new way to destroy monsters is introduced: a bubble of fire. It looks like the other bubbles the dinosaurs spit, but it has a flame trapped inside it. When one of the players pops this bubble, the flame falls vertically until it touches a surface. Then it spreads horizontally, covering that surface with fire that, while it lasts, fries monsters that touch it (see Figure 4.19).

![Figure 4.19](image2.png) The fire bubble spreading flames across a surface.

What is the most effective way to communicate the rules for *Bubble Bobble*’s particular brand of fire? *Bubble Bobble* accomplishes it using an entire scene as a visual metaphor. Figure 4.20 shows a scene of *Bubble Bobble* that has what looks like a giant frying pan. When a player pops a fire bubble, the fire travels down and the horizontal surface of the pan becomes covered in flames—heated, as though the stove had suddenly been turned on.

The monsters that inhabit this scene move diagonally in straight lines, changing direction when they touch a wall. These creatures bounce up and down until they touch the lit surface of the frying pan. Then they pop into the air, as does any monster when it’s defeated. That’s the source of the scene’s inscription, “POPCORN.” The visual shape of this scene, composed of the same block objects found in the rest of the scenes, tells the player a lot about the interactions they should expect.
We've already considered one way to think about a game's shape: in Chapter 3, we discussed how the experience of playing a game changes over time and how each scene gives the player a different range of choices, sometimes wide open, sometimes more narrow. Of course, it's just as important to think about the visual shape of a scene—and what the placement of objects in a scene communicates to the player. These two senses of “shape” can play off of and affect each other in all sorts of ways.

The shapes into which we arrange game objects can determine the way the player thinks about them. In Chip's Challenge (1989), Chip the protagonist collects computer chips before finding his way to every scene’s exit. She manipulates switches, sliding boxes and moving objects and other pieces to do so.

The “Castle Moat” scene in Chip's Challenge doesn't involve searching for chips, but it does have a river that Chip must cross to reach the exit. Crossing the river involves carefully pushing boxes found on the left in the maze. They float when pushed into the water, becoming platforms Chip can walk on. Or Chip can find a pair of flippers hidden in the upper right, allowing him to swim the moat. But how does the scene motivate the player to do all that work?

The important areas of this scene are revealed by the way they are shaped. The area of the scene that contains the exit is shaped like a castle, complete with gate, windows, and battlements. It's a simple arrangement of three objects—a wall, a gate, and an exit tile. But this simple piece of design transforms the body of water around the exit into a moat, an obstacle to overcome.
Chip actually starts this scene on the periphery of the map. To get to the moat and the maze, Chip must walk a path that takes him around the outside of the castle—ensuring that the player sees the exit. While playing the game, the player is only shown a 9x9-square area around Chip at any given time. It’s important to have Chip pass the outside of the castle to make sure the player knows where to go. This relates to the idea of camera, which is discussed in the next section.

**Camera**

Camera in this context refers to what the player can see at any given time. Does she see just part of the word? The whole world? Does the camera move, or stay still? It might move in periodic transitions, or slide over the world freely, or it might follow the protagonist. If the game consists mostly of text, perhaps there is no camera, only the narrator’s voice.

The way that the player sees the game characterizes the game world and the player’s relationship to it. If she is looking at the game from above, the scenes of the game might feel like a map—maybe one that’s being slowly uncovered, as in *Desktop Dungeons* (see Figure 4.21). This camera suggests that strategy will be important. If the camera moves with the protagonist, that provides a closer relationship between player and protagonist. If it doesn’t, the player has a closer connection to the world than the protagonist. A camera that lets the player see the entire game from a remote distance or high vantage can suggest that the player is like a god, considering each and every object in the world; in this case, even a protagonist character might be just another small piece that the player can manipulate.

If the camera shows us only what the protagonist sees, from a “first-person” perspective, the protagonist is the camera. In this kind of game, the player has a very different relationship to the world. Now everything is not of equal value; what the player’s looking at is what’s important. And the player cannot look at the protagonist. The protagonist is no longer just a part of the game world; she’s the way that the player can perceive the world, the lens she must look through.

Santa Ragione’s *Fotonica* (2011) is a game with a first-person camera, but it’s a fixed camera (see Figure 4.22). It’s a game about running and leaping, and the camera is always fixed on the horizon, even as mountains and hills and other shapes glide by on the periphery of the player’s vision. The game provides beautiful things to look at but never lets the player turn her head to look at them directly. The purpose of the game is to run and jump forward, following a track that consistently steers the player’s focus toward the center of the screen.
Figure 4.21  Desktop Dungeons gives the player an overhead view of each scene in the game.

Figure 4.22  Fotonica’s fixed camera moving the player forward.
A common mistake designers make when using a moving camera, most often in a three-dimensional game world, is taking the camera away from the player. The camera flies through the room to show us something the creator has deemed important: the exit, an important thing to collect. This isn’t design; it’s a failure of design. Stepping in and forcing the camera to look at something breaks the relationship between player, camera, and world.

Design has other ways to get a player to look at something. The space should draw the player’s attention to the important area. Think about the composition of the player’s view when she enters the scene. Is the important thing framed in that view? Does it look important?

There’s a story from the “Developer Commentary” to Half-Life 2: Episode 1 (2006) about a smart solution to the challenge of drawing the player’s attention without snatching control of the camera. The scene in question involves the protagonist and his friend trying to escape from a collapsing building. As the characters hurry across a bridge, a helicopter full of enemy troops zooms by below the bridge.

Naturally, the game’s creators wanted the player to see this. It develops the game world: the enemy is evacuating its troops because their former headquarters has been shattered. But it comes from a weird direction—to the side of where the player is moving. How did the designers get her to look?

To get the player’s attention, the designers put an enemy soldier in front of the take-off site. The game already contains signals to show the player where she’s being shot from—the straight, lingering streak of a shot going by or a red glare in the direction from which she’s hit. Knowing from which direction one is being shot at is important in a game with so many gunfights. When the player turns to retaliate, she has a great view of the helicopter taking off and zooming away.

Whatever camera our game uses, whatever window the player looks through into our game, design must decide what the player sees. This isn’t a place where we’re allowed to give up on design.

**Sound**

So far in this chapter, we’ve talked exclusively about visual elements. Digital games also have the capacity for audio expression, and it’s a powerful tool. We can use sounds to communicate, to underscore the important interactions in our game. A metal **tink** could tell the player her weapon has glanced harmlessly off an opponent. A melody could tell the player the coin she just touched is valuable. Sound is a very different channel than video. It can support visual information, oppose it, clarify it, or confuse it.
Sound as Emphasis

Jeff Minter’s *Space Giraffe* (2007) continually bombards the player with visual information, with melting lights and distorting lines. As the player becomes oversaturated with visual chaos, she is forced to depend on the audio. Each enemy makes a different noise when shot, allowing the player to construct a map of the scene and determine what kind of enemy behaviors she’ll have to plan for, using a kind of sonar.

*Lesbian Spider-Queens of Mars* is a fast-moving game. It signals the arrival of each new slave with a different noise. The slow, shielded slave that can only be zapped from the back has a lower-pitched kind of laughing noise, for example. This gives the player an idea of the threats that exist without her having to look away from the danger she’s currently concentrating on. *DOOM* is similar: because of its first-person perspective, the player might not be facing a monster when it becomes aware of her presence and begins to move toward her to attack. So each different species of monster has a unique “roar” when it first sees the protagonist and becomes aggressive.

Sound can influence a player’s choices. In *Super Mario Bros.*, every successive sound made when Mario jumps on an enemy increases in pitch until the player achieves a reward (an extra Mario). A rising pitch creates an expectation: it encourages the player to complete the progression.

Mike Meyer’s game *Horse vs Planes* (2012) pulls a similar trick. The player is given a score bonus every time she collects a fruit in quick succession. The first is worth 100 points, the next is worth 200, then 500, to a maximum of 10,000 (see Figure 4.23). There’s an element of time pressure, though: if the player takes too long between collecting fruit—two seconds—the bonus resets. The pressure to collect fruit rapidly and maintain the bonus complicates the protagonist’s— a horse’s—relationship with the antagonists—the planes—that are zooming dangerously through the play area. As in *Super Mario*, the fruit collection noise increases in pitch along the C scale with every increase in bonus points, up to the maximum, at which point the pitch (and the bonus) remain the same until the timer resets (and the pitch along with it).

The pitch of these sounds communicates three things. Most importantly, it tells the player that her actions are having a cumulative positive effect, that she should continue doing what she’s doing. It also tells her when the effect has plateaued. Finally, when the bonus times out and resets, so does the pitch of the sound, so when the player hears the pitch drop, she knows the chain has been broken. The sound that accompanies any fruit tells the player, concisely, where in the game’s reward structure she currently is. And the expectation of successive notes combined with the disappointment of the pitch resetting creates a strong drive to perform well and not break the chain.
Sound as Texture

Sound can also be used as texture. In *Dig-Dug* (1982), a little tune plays whenever the player is moving, digging through the dirt. The tune stops as soon as the player stops moving, giving way to silence and the sound of prowling monsters. In *Dig-Dug*, the player is hunting monsters, building passageways to try to goad them into the range of her weapon. If the player isn’t moving, it’s because she’s waiting for a monster to come into range so that she can attack it. But the proximity of a monster is also a dangerous opportunity for that monster to catch the player. The sudden silence, the interruption of the little digging tune, builds the tension. Something similar is done in *Lesbian Spider-Queens of Mars*, also a game where the player alternates between chasing antagonists and lying in wait for them. A little melody plays when the Spider-Queen is moving and goes silent when she’s not.

In *Knytt* (2006), the predecessor to *Knytt Stories* (discussed in the next section), a soft scampering sound accompanies any of the protagonist’s motions—her running, jumping, and climbing across the strange caverns and crevices of an alien world. This little sound—a gentle rustle in scenes that are otherwise silent save for the sound of wind—does a lot to emphasize the huge-ness of the world, the smallness of the protagonist’s intrusion upon it. When the player is still, the sound goes away, and the austerity of the planet’s near-silence is restored. This sound helps to characterize the relationship between the player and the world.
Fotonica also uses a sudden change in the texture of a scene’s sound to great effect. It uses sound to strengthen the player’s empathy with the protagonist, to help her better inhabit the game rules. When the player is performing well enough, maintaining a fast enough speed as she runs, a golden haze falls over the screen and the scene’s musical score becomes muffled. The running and jumping sounds that accompany the player’s movement, always present though normally quiet, now come to the forefront. The effect is something like a runner’s high—a feeling that’s sometimes called “flow,” a concept we’ll return to in Chapter 6, “Resistance.” In this condition, the player’s speed is greatly increased, and those elements of the game’s audio not important to maintaining that speed recede. This change communicates something: the player wants to be in this state for as long as possible, to maintain her speed and to attain this high.

Certain scenes of Shaun McGrath and David Kanaga’s game Dyad (2012) also use sound to convey the game’s rules. A siren begins, faintly at first, and becomes louder as the player’s energy approaches exhaustion. Ultimately, it becomes overwhelming, drowning out the original musical score of the scene entirely. The goal is to communicate to the player not only how close the player is to “death,” but a feeling of increasing nearness to death, to cause panic. When the player performs the correct action—lancing a target with a special attack, for example—the siren fades somewhat.

Real Talk

Nicklas Nygren’s Knytt Stories (2008), discussed in Chapter 3’s end-of-chapter group activity, is a game-making tool and spiritual sequel to Knytt, mentioned in the previous section. I’ve engaged with Knytt Stories both as an author and as a player. Having for several years followed the small community that has arisen around Knytt Stories, I’ve noticed a lot of similar mistakes committed by amateur authors. A lot of these mistakes are of design—scenes that are simply too hard, or built around objects whose behavior is unpredictable—or they’re technical—a surface is too jagged, and the protagonist, Juni, gets snagged while trying to climb it. But there are also mistakes of communication: the appearance of a scene or piece of a scene gives the player unclear expectations about the game.

A surface in Knytt Stories can look like anything: an author can import her own images into the editor, and those can become walls, floors, ceilings, and surfaces. But the editor comes equipped with 256 sets of graphical tiles, and many amateur authors choose to use images from those tile sets when constructing their worlds. (Handily, a bunch of those tile sets are consistent in style, making it easy to construct a whole world of different terrains that look like they could occupy the same planet.)

Miscommunication can arise when the author uses a tile in a way the artist didn’t intend. Often tilesets contain both tiles that are intended as foreground tiles—as walls and floors and surfaces that the player can touch and interact with—and ones that are intended as background
tiles—the texture on the wall of the room the player is in, or a mountain in the distance. In many cases the artist has drawn the tiles in such a way to distinguish foreground images from background. Foreground tiles are bold with solid black outlines, while background tiles use more muted colors, are partially transparent, or have softer outlines (see Figure 4.24).

![Figure 4.24 Comparing foreground and background tiles in Knytt Stories.](image)

A common problem found in games made with *Knytt Stories* is background tiles being repurposed as foreground tiles, as things for Juni to climb on and over. With their muted colors and soft borders, they don’t stand out from the background, so the player doesn’t always read them as solid, climbable objects. In one particular Knytt Story, I was unable to proceed in a game because a scene required climbing up what I misread as a piece of background scenery. I ran back and searched the entire game world another time, certain that the screen was the back entrance of a one-way passage: you could jump down from the top of the cliff, obviously, but there seemed to be no way to climb up.

Here’s another communication problem many authors invent different solutions for: Juni can climb on any surface, so long as she possesses the climbing ability. (The author can give it to her or take it from her at any game transition.) Often it’s untenable to allow the player to scale any surface to the very top—it’s significantly more work to make lots of extra scenes above the current, for the event that the player chooses to climb up there, and it makes it much harder to constrain the player’s movement, to build a path for her, if she can just climb up and over any wall.

The obvious solution is to make each wall meet a ceiling, but it’s boring to have each area capped by an artificial ceiling, and it’s inconvenient in the case of many scenes. If we want the player to experience this as an outdoor area, how will that come across when there’s a ceiling?

The bank of objects in the *Knytt Stories* editor offers another solution: “no climb” blocks that can be placed over any existing game block, which prevents Juni from scaling that block’s sides. These blocks are visible in the editor but invisible to the player (see Figure 4.25), meaning that the wall beneath the block can look like anything the author wants. It can look like the rest of the wall, for example.
Figure 4.25  “No climb” blocks prevent Juni from going up a wall; they’re invisible to the player (top) but can be seen in the Knytt Stories editor (bottom).

But it’s jarring for a player, scampering up a wall, to suddenly encounter a piece of wall that won’t hold her. It can seem every bit as artificial as a sudden ceiling, and, what’s more, the player can only discover that a wall is not climbable by trying to climb it. That leads to wasted player time, as the player tries every wall to discover which are climbable and which, seemingly arbitrarily, are not.

What smart authors have done is find a way to visually distinguish climbable from nonclimbable walls. The simplest way involves using another object from the object bank: waterfalls. They come in blocks the size of any other game block and are entirely superficial. They don’t affect the rules of the game, but authors have used them to communicate rules. Draw a waterfall over a wall, or over the side that the player might otherwise have expected to climb, and it becomes clear the wall is not climbable: it’s too wet (see Figure 4.26). Once the player gets a sense of this logic, she’ll be able to recognize, from then on, which walls Juni can climb and which she can’t.

Here’s a different solution: simply draw different walls, having a wall version that’s climbable and a version that’s not climbable. One way to do this is by drawing walls that have visible handholds in them and other walls that are smooth. If the player sees handholds, she knows that wall is climbable. If there are no handholds, she expects that the wall is not climbable (see Figure 4.27).
There’s a similar object in *Knytt Stories*’ object bank: the sticky block. If Juni tries to run across a sticky block she’ll be stopped, unable to continue. The only way to move again is to jump off it. Like the no-climb block, this block is invisible to the player, so authors have to use their own methods to communicate to the player which blocks can be walked on and which are sticky. A few authors smear these floors with sticky-looking pink goop (see Figure 4.28). This seems like the best way to describe the rules of these blocks.
Review

- The appearance of game objects can tell the player about their function, their purpose, and their relationship to the player. For example, something covered in spikes is dangerous to touch.

- Recurring visual motifs allow us to develop an ongoing vocabulary of game rules that the player understands. If the player learns that one spiked thing is dangerous, she’ll expect other spiked things to be dangerous.

- Character design is useful not just for communicating the rules that govern objects, but also for making them distinct and recognizable. Giving characters unique silhouettes helps the player distinguish them visually.

- The way that things move can be used to differentiate the important from the less important objects in a game. Animation can characterize an object as aggressive, timid, friendly, or dangerous.

- The visual composition of a scene can direct the player’s attention to what’s important in that scene. Symmetry can suggest importance or artificiality; irregularity can suggest a natural landscape.

- A scene can be an image as well as space. It can be a pretty view or a foreboding visage.

- The way a space is shaped can provide a context for what the player is doing there. For example, battlements create a castle; a circle of water makes a moat. A castle on the other side of a moat can give the player an incentive to cross the moat.

- How the player looks at the game world—the camera—does a lot to characterize the player’s relationship to the world and what’s in it. Seen from above, the game may look like a map on which the player uses strategy. Seen from the side, it might look like a diorama a character has to navigate.

- A common mistake is to take control of the camera to force the player to see the things you’ve deemed important. Good design leads the player—and the camera—to the things that are important.

- Sound can communicate information about the game, such as changes in game state (perhaps the presence of characters or monsters). It can indicate success or failure or being on the right track.

- Sound can be a texture, a layer that can be added or taken away to change the player’s immediate relationship to the world.
Discussion Activities

1. Think about games you’ve played recently and find an example of a visual motif: a recurring use of shape, color, or representational imagery that relates to some aspect of how the game works. What did this motif represent, and what did you learn from seeing it appear repeatedly? Were there times when this motif—or a visual that resembled it—was confusing or misused?

2. Can you imagine a scene in this game—or any game—where it might be useful or challenging to the player to disrupt the conventions established by a visual motif? Discuss some scenarios and how they might shape gameplay. As the creator of a game, why might you want to deliberately confuse the player?

3. Name one of your favorite characters from a game you’ve played or one that you found particularly memorable. This character doesn’t necessarily need to be the protagonist of the game; it could be an enemy or another character not controlled by the player. Describe the visual elements of this character’s design that stood out for you: the silhouette of the character, use of color, and what various aspects of the visual design represent. How do those elements relate to the role that this character plays—their behavior and the way they affect gameplay?

4. Describe the soundtrack or sound effects of a game that were especially memorable for you. What do those sounds represent, emotionally? What does hearing that sound make you remember, and how do those associations relate to the way the game works or events that happened in the game?

5. Choose a game that you’ve already discussed, that you used for an exercise from a previous chapter, or that we’ve talked about in this book. What role does the camera play? Is it fixed or moving? From what point of view does the camera let you see the action? How does this point of view affect your perception of what’s going on in the game as a player? What would this game be like if it used a different camera with another point of view? If the game’s camera shows scenes in the game from an overhead perspective, like the camera in Pac-Man, describe what the game would be like if the player saw everything from Pac-Man’s point of view. How does the game change?

Group Activity

As a group, come up with an existing game that you’re all familiar with. This could be a board game as well as a digital game, but should be something with a recognizable visual theme. Monopoly, for example, suggests the blocks of a city, with railroads criss-crossing it.

Design a very different visual theme for this game. Don’t change how the game works, but come up with new visual representations for all the elements of the game. Sketch what these new elements might look like. What if Monopoly took place in a cemetery or an entire solar
system? You'll probably want to rename the elements and characters in the game as well. When you're done, discuss how the experience of the game has changed. Are there new visual motifs? Are they communicating the same things to the player as they did in the original game?

Now discuss what kinds of changes you might make to the game’s system and how the game works. Do you need new rules to help your visual theme make sense? For example, if you modified *Monopoly* to be set in outer space and changed the railroad stations into “spaceports,” would it make sense to let players fly to another location from a spaceport? How do you think your rules changes would affect the way the game would be played?