A discussão sobre “Context” de Steve Swink e o *Level Design*

No livro “Game Feel”, Steve Swink discute a importância do contexto (ambientação espacial) do jogo na experiência do jogador. Os trechos a seguir apresentam a discussão sobre o que ele denomina “medium-level context”, que é diretamente relacionado ao *level design* e uma análise do contexto no jogo Super Mario 64.

Game Feel

A Game Designer's Guide to Virtual Sensation

Steve Swink

ELSEVIER

AMSTERDAM • BOSTON • HEIDELBERG • LONDON
NEW YORK • OXFORD • PARIS • SAN DIEGO
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO
Morgan Kaufmann Publishers is an imprint of Elsevier

MORGAN KAUFMANN PUBLISHERS

when your frame of reference was changed to a closer aspect. The bosses in Serious Sam and Painkiller appeared to move quite quickly, even from a distance, thus destroying the intended effect of the impression of size.

So that's one great example of how the speed of movement of objects by providing context for one another can effectively sell the size, mass and weight of an object. The sounds and particles and screen shake also speak to those properties, but in this instance the sensation was really sold by the context of one moving object relative to another.

This is the same essential principle that applies to the apparently slow motion of Zangief relative to Chun Li in Street Fighter II. Zangief's motion seems ponderously slow relative to the motion of Chun Li. His movement would probably seem somewhat slow by itself, but with the added frame of reference of twitterbug Chun Li, the effect is enhanced considerably. To contrast that, imagine an alternate universe Street Fighter II where Zangief is the fastest character. Relative to those dullards, his motion would seem zippy fast.

Medium-Level Context

The medium level of context refers to the feeling of immediate space and object avoidance. At this level, changes in context can mean the difference between sensations similar to pushing through a crowded party, wandering an empty street or a playing in a basketball game. It's not space at the low level—intimate and interpersonal—and it's not the sensation of openness you get from walking along a beach. It's the layer where, with respect to game feel, context is the "second set of knobs" for game feel tuning. The first set of knobs is in the programmed response to input; you tune the speed of motion of the character in absolute terms relative to the game. For example, the character moves at 90 meters per second forward and can turn somewhere between 0.1 and 5 degrees per second. None of these numbers has any meaning, as we've said, unless they're related to spatial context. In order for the forward speed of a car in a racing game to have meaning relative to how fast it can turn left and right, it must have a track. You have to have a track laid out that has curves of a certain sharpness and that includes objects and obstacles to avoid. It is at the intersection between the tuning of the individual response to input numbers and the spacing of the objects in the environment that the feel of a game gets primarily tuned.

So this mid-level of context is about steering and object avoidance, about navigating an interesting spatial topology with enjoyable precision and deftness. In order to compare the avoidance mid-level feel between games, what we need to examine is:

- The number of objects
- The size of the objects
- The nature of the objects

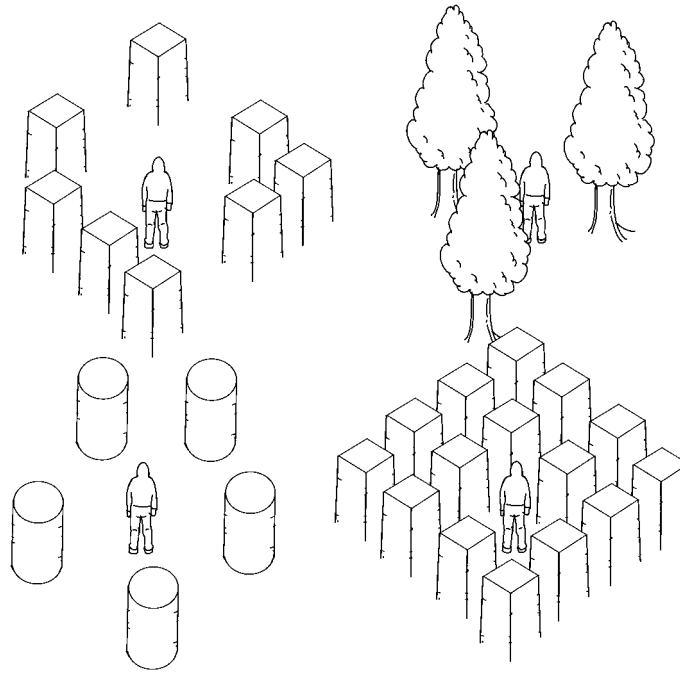

FIGURE **8.2 Different configurations and types of space yield a different feel relative to the controlled movement of the avatar.**

- The layout of objects
- The distance between objects

Across multiple games, we can compare how far apart objects are spaced relative to the speed and motion of the avatar and can examine how this changes game feel (Figure 8.2). A great example comes again from World of Warcraft. Moving through WoW, I began to experience highway hypnosis. Highway hypnosis happens when, while driving, you begin to zone out, and are to be lulled into a somnambulant state by the flowing uninterestingness of it all. You're sort of flying across the land and your mind begins to expand in all directions, and you have this sort of powerful alpha brain wave truncation of time. Before you know it you've driven 200 miles and are suddenly nagged by the distinct feeling that yes, perhaps I should have been paying more attention those last three hours. My driving instructor in high school told us that 20-something percent of all accidents in some place during some time period were caused by this. For me, it tends to happen when I'm driving cross country. Traversing the landscape of WoW felt for all intents and purposes like driving the long stretch of straight, level freeway between Los Angeles and San Jose.

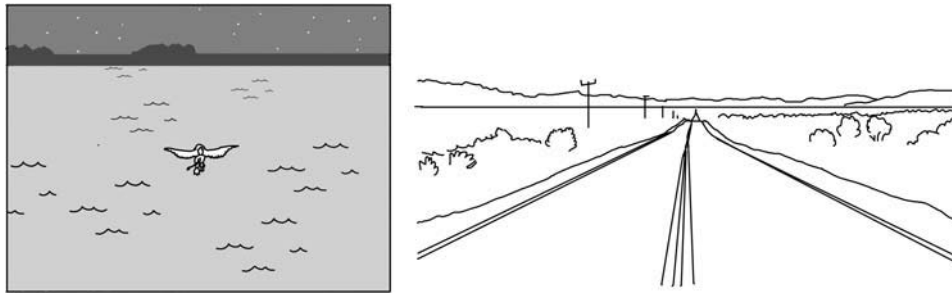

FIGURE **8.3 WoW-highway-hypnosis.**

Because the objects in World of Warcraft are spaced so far apart relative to the speed of movement of the avatar, and because the movement of the avatar across that terrain often has no gameplay function, running across this environment started to lull me into the same zoned out state (Figure 8.3).

In contrast, playing Vanishing Point is like another section of the route between San Jose and Los Angeles. After hours of mindless driving on Route 5, you arrive at Pacheco Pass. The contrast is striking. In Pacheco Pass, the wind blows right off the adjacent waters of Lexington Reservoir, buffeting your car and threatening to uncouple it from its reassuring grip on the road. The road twists at dismayingly car-commercial-like angles, and there's invariably some idiot who seems determined to get his Jetta's worth by driving the section at speeds he's witnessed in such a commercial. Playing Vanishing Point is like being the idiot in the Jetta. It's an insidiously difficult racing game from the Dreamcast era, with the most twitchy, difficult controls. The game tasked you with accomplishing the most horrendous and knuckle-whitening missions using those controls.

These two extremes, WoW and Vanishing Point, are two points on the scale of mid-level spacing. This is the primary dimension in which it's possible to create challenge. Along the gamut from WoW to Vanishing Point, you can have objects spaced farther apart or closer together. In tweaking that relationship, you also increase the challenge of navigating that space. So the alteration of the spatial context in which an avatar moves is one of the primary vehicles with which it is possible to change challenge. Again, to measure at this level, we're interested in the spacing of objects.

Low-Level Context

Finally, context affects game feel at the low level of intimate, personal space, at the level of tactile interaction between objects. At this level, what we're interested in examining is collision. Now, the maths involved in collision detection and response is a little bit scary. At least, it is me (being a lowly brain-dead

game designer). Regardless, there are several excellent books and online resources dealing with different kinds of collision and how to implement them. For our purposes, we want to compare different ways it is possible to model collision and the resulting feel is to simply draw comparisons to the interactions of physical objects in our everyday lives. Again, this is something of a soft metric and requires some conceptual leaps in terms of metaphorical relationships between objects moving in a game and objects moving in the real world. But the analogies are usually sound, and this is a useful tool for categorization and comparison.

For example, most racing games subscribe to the "waterslide" method of collision and response. The reason is that it really sucks if you're playing a driving game which requires a lot of precision and the car reacts like it's coated in glue when it comes into contact with another object. Or, you know, do something akin to what it would do if you were to scrape a real car against a barrier at 200 miles per hour. Instead of this quagmire, most modern racing games use collision schemes which essentially feel like a waterslide. There's almost no friction at all when the cars run into something. When you bang into a barrier, instead of crumpling or exploding, the car ricochets off and keeps on going. This is a very different feel from a collision system with a huge amount of friction, where if a car runs into a wall it might roll or get stuck or crumple sideways.

So essentially what we can do is look at the feel of collisions between objects in a game and compare them to the feel of everyday things and so to one another. For example, the collisions in Loco Roco feel like a big bowl of Jell-o or a bunch of water balloons banging into one another. It feels very soft, very jiggly, very spring-loaded. Very different from Gran Turismo, which has a very solid feeling to its collisions. It's a rigid, unyielding solidity, and an interesting comparison can be drawn between the crisp smooth solidity of the collisions in Gran Turismo versus the dirty, broken, mushy collisions in Burnout Revenge. In Burnout, they still employ the waterslide model at some level, but they're doing some rather sophisticated damage modeling. This causes the car to compress and mush up even if there's nearly zero friction applied to the collision so the car can continue driving apace.

As a final example, consider the low-level feel of World of Warcraft. While the high-level feel of WoW was open and boundless, it felt barren and empty. It was tactilely sterile. When I climb Superstition, I can reach down and feel the rocks beneath my hands. Indeed, doing so is mandatory—to make it to the top requires a short section of light rock climbing. In WoW, I never really interacted with anything. It felt very sterile. There was no reason to pay attention to what was nearby or whether I was running into a building, running through a desert or running off a cliff. The collisions felt smooth but dead in the sense that there was no energy coming back out of them. You can't smack into something and rebound or indeed see any interesting interaction. It's a bit like playing only with Nerf toys. Everything is soft and mushy and safe to play with indoors. You're never going to put your eye out with them.

Apart from the default behavior described above, the camera avatar has many special case solutions that change its motions. First, it collides with walls and other surfaces. When it does so, its motion in that direction stops, the same essential effect as when the character is moving only side to side under normal circumstances. In some areas, such as the main castle foyer, the camera switches to a fixed perspective from a specific, preset vantage point.

This blending of direct and indirect control over the camera is admittedly clumsy and would probably be labeled as irretrievably broken by modern standards. The problems with the system, though, are mostly mitigated by specific case hole-plugging (as with predefined security camera vantage points) and through judicious level design, which emphasized mostly large, open areas with towering central landmarks.

In general, the approach of camera motion in Super Mario 64 attempts to avoid superfluous motion and to show players what's ahead in the direction they're traveling as much as possible. Though quite sophisticated for the time, experienced today it can seem jarring, frenetic and inadequate.

Control Ambiguities

Up to this point, the feel of Mario 64 has been characterized as unambiguously wonderful. But it is not all sunshine and mushrooms. Here's the rub: there are some rather glaring control ambiguities in Mario 64's setup that should have been resolved. As Mick West points out in his most excellent "Pushing Buttons" article,[3] there is a troubling crossover between the Ground Pound, Back Somersault (here called backflip) and Long Jump moves in Super Mario 64:

> In Nintendo's Super Mario 64, when playing as Mario, pressing A to jump then R1will trigger a ground pound. Pressing R1 then A will trigger a backflip. Pressing them both at the same time will cause one of: a ground pound, a backflip or a normal jump, seemingly at random. This is bad because the user has no control; they are doing the same thing over and over, yet getting different results.
>
> This problem also shows up in Mario when you try to do a long jump, which is done by running, then jumping by pressing A+R1. Sometimes while attempting this you will do a Ground Pound by accident. This is not the fault of the player. To the player it appeared they did everything right, but the results were not what they expected.

Context

As has been alluded to, the spacing of the objects in the levels of Super Mario 64 has a hugely positive effect on the overall feel of the game. Relative to the avatar's

[3]http://cowboyprogramming.com/2007/01/02/pushhing-buttons/

movement, levels were constructed with three specific spatial relationships in mind: vertical height, horizontal distance and the size of platforms.

It's fine and well to say that the spatial context of Mario 64 matches perfectly with the tuning of its mechanic, but what does this actually mean? From the standpoint of the pragmatic level designer, what did this mean in terms of actually placing polygons? And how was the mechanic tuned, relative to the movement of the character? What was the mechanic designer's role in this process? First, I believe that in the case of Mario 64, as in most high profile Nintendo games, these two were one and the same. At least, initially. Anecdotally, the prototype form of Mario 64 was a "gameplay garden," a test level which included a near-final version of Mario, complete with animations and moves, and a wealth of different things for him to interact with. As the jump heights and trajectories were tuned, so were the distances between objects. The Wall Kick and walls spaced the right amount apart were created simultaneously. This meant that as the mechanics were evolving, so too were the general rules about how far apart objects should be spaced, how big or small they should be, and what kinds of environments would be built around them and out of them. Simply put, the size, nature and spacing of objects were part of the same system as the height of Mario's jumps and the speed of his running and turning. These guidelines seem to consist of four primary spatial relationships: vertical height, horizontal distance, the X/Z dimensions of each walkable platform and the angle of incline of each piece of terrain.

By vertical height I mean the distance between a given current position of the character and some other, higher position. The vertical height of objects relative to one another comes in three distinct and premeditated flavors. First there are objects which can be scaled by a basic jump. Many blocks are spaced at just the right height for the basic jump. They're lower than the apex of the jump to enable a wide range of jumps to land on them, as were the blocks in Super Mario Brothers. Other jumps are clearly just right for the Back Somersault, Side Somersault or Triple Jump. For example, at the beginning of Whomp's Fortress, there's a wall that is the perfect height for a Side Somersault. On the Shifting Sand Land level, there's a platform with a Flying Cap block on top of it that is perfectly spaced for the Triple Jump. Throughout the levels in Mario 64, these relationships are maintained. As you play, you quickly become accustomed to not only the predictable height of the various jumps at your disposal, but the fact that the environment seems tailor-made for the heights of these jumps. It becomes easy to walk around a level to see which ledges are basic jump height, which are Triple Jump or Back Somersault height, and which are too high to reach by jumping. I also note that there are many unforced opportunities to use higher jumps. Especially in the earlier levels such as Whomp's Fortress; Cool, Cool Mountain; and Bomb-Omb Battlefield, there always seems to be a way to circumvent the normal path—which emphasizes jumps at the basic height—by using a Side Somersault or Back Somersault to get higher earlier.

When I say horizontal distance, I mean the distance from one point to another along the same plane. Rather than trying to ascend to a higher platform, the horizontal distance of a jump dictates how wide a chasm Mario can cross. Can I make

it across this gorge or patch of lava in one Long Jump or Triple Jump? Or do I need to use a basic jump and pull back slightly on the stick because I need to land on the small portion of a moving platform that isn't currently covered with scalding lava? As with the relationships between vertical objects, there are various specific relationships between the position of horizontal objects in space that are maintained throughout the levels in Mario 64. Some platforms are clearly spaced to be just the right distance relative to the basic jump, where others can only be accomplished with the Long Jump. In each case, it becomes easier and easier for players to eyeball these relationships as they play through the game. If a jump looks to be just the right distance to clear with a Long Jump, it almost always is. And, as was the case with earlier Mario games, increasing challenge usually means longer, more precise horizontal jumps.

Of course, each jump represents a trajectory, including both horizontal and vertical movement. To land on a platform, whether it's a tiny shelf of rock far above and behind the character or whether it's a wide platform over a gorge directly ahead, requires movement in both the vertical and horizontal. What Mario 64 does wonderfully is to present the player with consistent vertical and horizontal relationships throughout the game, regardless of what else is going on in the level. As a result, the complex, imprecise motions of Mario through 3D space become manageable, predictable skills that can be learned and mastered. This feels great; the player almost always gets the result he or she was after. The onus, then, is on the player to plan and execute maneuvers more accurately and skillfully.

A platform's dimension refers to how much landing or maneuvering space it provides.

Relative to the speed at which the character runs when on the ground, the levels are very open, without much obstruction. The running is a very precise, responsive motion with no floatiness or looseness, so there is little emphasis in the design of most levels on running very precise patterns. Releasing the thumbstick brings the character to a halt immediately, so there's no real risk of unintentionally running into or falling off of something. The game says: wait until you're ready. It's supposed to feel easy and safe just to move around the world by running, and it does. The levels in which it is not so safe—Lethal Lava Land and the three Bowser stages—are genuinely unnerving by comparison, requiring an unaccustomed amount of focus on the character's exact position on the ground. Keeping the player scrambling forward for extended periods of time and taking away the safety net makes the game feel very different. That both feels exist in the same game speaks to the fact that the designers had a deep understanding of what they were doing in constructing each type of level. More than that, the different sensations create an excellent and rich contrast which enhances each.

As we noted when talking about collision and response, Mario 64 models friction, especially with respect to the angle of incline of the terrain beneath him. The character has a certain coefficient of friction, which can be overcome and send him slipping and sliding. In this way, incline is used throughout the levels as soft boundary and soft punishment. If you're not supposed to go somewhere, there will be a

steep incline to turn you back. It's a gentle, negative reinforcement with a clear, logical physical relationship. You can't climb up and over the wall in Bomb-Omb Battlefield because you start to slip and slide back down if you try. It feels futile, quickly getting the message across without painfully overt constraints such as an invisible wall or other contrived boundary. In this way, what's been accomplished is a victimless blame shift, a hallmark of good level design. Player don't feel the direct intervention of the designer like some deus ex machina dipping in to wag a disapproving finger and tell them where they can and can't go. The physical relationship between incline and slide is consistent throughout the game, so the limit feels like a logical consequence rather than an overt constraint.

Finally, it's worth noting the overall spatial layouts of most Mario 64 levels and the effect that has on the high-level spatial feel of traversing them. For the most part, the spatial layouts of Mario 64 levels are like zones of a theme park, with the important features poking prominently above the landscape, visible from any vantage point. The tower in Whomp's Fortress, the central spire in Bomb-Omb Battlefield, and the giant snowman central to Snowman's land are all designed to provide an instant point of reference and include many of the level's important star-giving interactions. Two benefits of this landmark-focused approach are improved camera behavior and a delightful sense of vastness and exploration. The camera motion in Mario 64 was a sore point for many players and critics, but one of the instances in which it always works well is in following the Mario avatar around a spire or pillar. Once at the top of a huge structure, the camera is free to look around and down, surveying the ant-like surroundings far below. This feels great, like hiking Superstition Mountain, El Capitan in Yosemite or the Space Needle and peering down on all the places you've just been. This doesn't affect the moment-to-moment feel of interaction, but it certainly lends a highly positive high-level sense of space to the proceedings.

Polish

The primary area of emphasis for all polish effects in Mario 64 is the interaction between Mario's body and the ground beneath him. Generally speaking, the pertinent polish effects are the animations, which are mostly in sync with the speed at which the avatar moves, jumps and otherwise interacts with the environment and which show the character leaning into turns, planting his feet and otherwise being a believable physical being. In harmony with these detailed and excellent animations, the sounds and visual effects adhere to a three-tiered structure, corresponding across senses. Impacts come in three varieties: light, medium and hard, and each type of interaction has a special animation, visual effect and sound effect. Combined with the ubiquitous footstep sounds and sliding noise, these effects serve to convince us of a Mario who exists in a believable, physical world of his own and who interacts with it in a logical, law-driven way.

Because it made the most sense for this particular game, I have pointed out many of the important polish effects as they occurred, while discussing the simulation.