

Construct a familiar symbol to open the stairs into the floor in one of the game's cooler sequences.

deserve special mention. Halfway through the project, we picked up Jessica Tams as our content lead, who took over the content and put it in order. She wrote nearly all the scripts and logic for the entire game, completed them on schedule, and somehow made them all work despite the problems with the engine (more on these problems in a moment). Lead animator Ray Bornstein came onto the project with a year left to go, put the animations in order, created a realistic schedule, and made the animators stick to it.

## What Went Wrong

**1. Team casting problems.** When someone is placed in a role in which they don't belong, I call this being "badly cast." Many of the problems with GK3 resulted from developers being badly cast in their roles, usually because the project requirements were so severely underestimated. To give you an idea of the casting problems we had, consider this: we went through a total of two producers, three art directors (we spent the last year of the project without one), and three project leads (the producer was forced to take over as project lead towards the end).

This was an ambitious, massive project that required experienced engineers and the original team was simply not up to this task. GK3 was initially built from members of the SHIVAS 2 team (one of the last games built with SC1) and they had practically no 3D experience. Engineers under the venerable SC1 engine were basically scripters and putting them in charge of building a game engine from scratch was like feeding them into a furnace. To make things worse, the developers that were in over their heads didn't ask for help, which gave management a false sense of progress.

**2. Severe morale problems.** Hundreds of books and articles have been written about this and here we have yet another Postmortem listing it under "what went wrong." It's time for me to get on my soapbox. To managers everywhere: morale is

one of those icky personal political things that many of you avoid dealing with. But you need to understand that your development team is not a factory churning out content and code. To paraphrase Peter Sellers in *Being There*, "The team is a garden of creativity that requires regular watering and sunshine in order to build strong roots." Loyalty is not something that comes easily. The job market is very competitive — your best developers will simply leave and work for somebody else if they aren't treated well and maintained properly. On GK3, there was a serious lack of love and appreciation throughout the project. Recognition of work (other than relief upon its completion) was very rare, lacked sincerity, and was always too little, too late.

Internally, a lot of the team believed that the game was of poor quality. And of course, the many web sites and magazines that proclaimed "adventure games are dead" only made things worse. Tim Schaffer's GRIM FANDANGO, although a fabulous game and critically acclaimed, was supposedly (we heard) performing poorly in the marketplace. Rumors circulated among our team that GK3 was going to lose money, due largely to our high burn rate.

The low morale resulted in a lot of send-off lunches for developers seeking greener pastures. GK3 had a ridiculous amount of turnover that never would have been necessary had these people been properly cast or well treated in the first place. More than 45 developers worked on GK3 (the average standing team size was 15 to 20), and now, just a few months after it shipped, only seven remain at Sierra. Strangely, the opposite also hap-

pened — several of our developers were included in Sierra's mid-1999 housecleaning layoffs but these individuals were allowed to stay on for a couple of months, postponing their last day until we shipped GK3. I believe this was done in good faith out of respect for the developers' hard work up to that point but it ended up being a prolonged drain on morale. Having a small group of people who are (understandably) upset with your company for laying them off and actively looking for a job while still trying to be productive and contribute to a project is a tough situation that should be avoided.

After a certain amount of time on a project like this, morale can sink so low that the team develops an incredible amount of passive resistance to any kind of change. Developers can get so tired of the project and build up such hatred for it that they avoid doing anything that could possibly make it ship later. This was a terrible problem during the last half of the GK3 development cycle and as a result there are many aspects of the game that we aren't proud of. These were problems that should have been fixed but nobody wanted to take the time to correct them because we were so focused on trying to get the game out. I don't think anyone on the team is directly at fault for this and I don't know what we could have done to correct this problem.

**3. Schedule problems.** Our engineers never had an accurate development schedule — the schedules we had were so obviously wrong that everybody on the team knew there was no way to meet them. Our leads often lied to management