A Conversation with Will Harvey

PHOTOGRAPHY BY TOM UPTON

In many ways online games are on the bleeding edge of software development. That puts Will Harvey, founder and executive vice president of Menlo Park-based There, right at the front of the pack. There, which just launched its product in October, is a virtual 3D world designed for online socializing.

Harvey developed his first game at age 15, after teaching himself assembly language on his Apple II. He has never stopped. While earning his bachelor's, master's and

An online-gaming

INNOVATOR TALKS ABOUT

TECHNICAL CHALLENGES

IN HIS VIRTUAL WORLD.

Ph.D. in computer science from Stanford, he founded several game companies that have produced platinum and gold game titles

including Zany Golf, Immortal, and Music Construction Set. He has filed five patents related to networking, graphics, and automated scheduling.

Before founding There in 1998, Harvey was at Adobe

Systems where he ran dynamic media products, including AfterEffects and Adobe Premier. He came to Adobe when Adobe acquired his previous company, Sandcastle, which Harvey had founded to develop network technology to enable low latency interaction over the Internet. Prior to Sandcastle, Harvey served as vice president of engineering at Rocket Science Games in San Francisco, where he led its transition from full-motion video-based games to games focused on interactivity.

Harvey shares his thoughts on gaming technology with Chris DiBona, a fellow game developer and cofounder of Damage Studios. Damage is developing a massively multiplayer game called Rekonstruction.

Prior to his work with Damage Studios, Chris DiBona was best known as a strong advocate of the open source movement. After working in information management and computer security at the U.S. State Department and Tandem Computers, DiBona became the third executive hired at VA Software. He ran developer and Linux community relations and played an important role in creating OSDN (Open Source Development Network). He wrote *Open Sources: Voices from the Open Source Revolution* (O'Reilly and Associates, 1999) and is a frequent commentator on open source and development issues. He has also spoken on game development at the Gaming Developers Conference.

CHRIS DIBONA Why don't you tell us how you got started with computers?

WILL HARVEY I'll start with how I got started with computer *games*, because that was pretty much the beginning. In the early '80s, there was no barrier of entry for people

writing video games. I taught myself assembly language from Roger Wagner's "Assembly Lines" column in the back of *Softalk* magazine for the Apple II and managed to put together a game that was as good as the games—at least in my opinion—that people were selling. The coding style was horrible. I was just a 15-year-old high schooler. I didn't know anything. But the next step was to find a publisher.

I didn't even drive at the time, so I took the bus to the computer store and looked for a computer game that had good package artwork. How else would you choose a publisher other than the packaging?

CD There's still a lot of truth to that.

WH I found one that had great package artwork and then looked on the back and found the phone number. It was Sirius Software in Sacramento. I called them up, said, "Hi, my name is Will. Can I talk to the president, please?"

What was great was—and this shows you how low the barrier to entry was—I actually got to talk to the president. He said, "Show me your game," so I took the bus to Sacramento. They were moderately impressed. They had a few changes they wanted me to make. I ended up shopping it to a few other publishers.

CD Which game was this?

WH It was a space invaders type of game called Lancaster, and it was for the Apple II. Meanwhile, I needed to have music in the game. I didn't know too much about it, so I wrote a music editor, kind of like a word processor, that would allow you to copy sheet music onto the screen, then convert it to MIDI [musical instrument digital interface], which was just beginning at the time, and play it back for you. It would also save it as a MIDI file so I could then have the game play it in the background.

As it turned out, the music editor, called Music Construction Set, became an enormous hit. Electronic Arts picked it up in about 1983 or '84. It was educational, it was creative, it was fun. It was a new category of software that was perfect for the time. I was able to take the proceeds from it and build a little video-game company. I've been doing it ever since.

CD When did you start There?

WH I started There in 1998. I funded it myself for the first year and hired a brilliant prototyper to work with me in creating what the There world would be like. When you're building something like There, you want to get everyone on the same page for exactly what you're build-

ing. It's very easy to describe something and then have four different people have four completely different ideas about what it is.

CD Did you license an engine technology or did you write your own?

WH We wrote our own engine. You can buy an engine to do the graphics or you can buy one to do the networking. Often, one of these engines assumes it's sort of the center of the world and all the other programs plug into it.

I don't think that the engines were mature enough at the time. I'm not sure that they are even now to support a new product like There.

CD Is your engine written in C++?

WH It's entirely C++, maybe with bits of assembly language here and there. Networking is the center of There. The graphics—the secondary problem—is built on top of OpenGL.

CD Have you ported it to the Macintosh or other platforms?

WH We haven't yet, but our general plans are to get it to work on everyone's computer and, ultimately, on devices that aren't computers. There is a virtual world that exists as electrons on a bank of servers in our data center. So you can imagine logging into that virtual world through many kinds of devices that have different display capabilities—a cellphone, television set, game console, or computer. Your manner of logging in, your portal to the world, obviously would be different on each of those different devices.

CD I've always been a big believer that if the game is online, you should be able to access it, not just through a 3D client, but through any device. It's nice to hear that from other people in the business. Have you done any outreach to the PDA world?

WH Not yet. It's very tempting to port to all kinds of computers and to add all kinds of features. But we've focused on the PC as our first platform, and will concentrate on that until we have traction and know we're good on that platform.

CD For your back end, you mentioned your data center. Do you have multiple clusters or just one?

WH We have one cluster right now. The clusters are designed to handle hundreds of thousands of simultaneous users. It's all on Linux machines, a massively

distributed set of servers. The There world is the size of the earth. As you can imagine, you can't simulate something of that magnitude on a single computer. It's necessary to share that load over multiple computers. One of the common ways of doing that—and the way we're doing it—is to break down the world into sectors and to have a sector correspond to a geographical region that is served by one computer.

The sectors actually can split and merge according to load dynamically. But what's basically happening is when you're walking around the world, your client is connected to one or more servers, and it's hopping from one to the other. In the cellphone business, it's called *handovers*. If you think of talking on a cellphone when you're driving down the freeway, sometimes you lose a call when you hop from one cell to the other. Usually it works; sometimes it doesn't.

In There, the problem is so much harder than with cellphones. The cellphone is just a voice stream of data. In There, imagine that you're driving a dune buggy, with a passenger who is shooting a paintgun at another person with a paintgun. And a dog is heeling behind the dune buggy. Now, that whole collection of relationships—you driving the vehicle, the passenger sitting in the vehicle, the paintgun being held by the passenger sitting in the vehicle, the person being shot by the paintgun, and so on—has to transfer from one server to another in a way that is transparent to the user. That's an enormously difficult problem—much more difficult, in fact, than a cellphone handover because you're handing over all these relationships.

What's actually happening is that your front end initially is logged into the sector that you are in, and as you drive from that geographical region to the neighboring geographical region, in advance of hitting that line in between, your client will log into the neighboring server as well, so now it's logged into two servers and beginning simultaneous streams from both servers.

Meanwhile, in the back end, the action is being simulated on both computers for a certain overlapping period of time, and then once you're safely in the new region, you can drop the earlier connection.

CD Did you write your own database software?

WH No, we use Oracle, and we've used MySQL. We've used a number of databases. The key point is that we actually use many instances of those databases. So every simulation server has a data server that has an instance of the database. Unlike having many simulation serv-

ers and one database to keep track of credit cards and whatever, we'll have many simulation servers and many databases. By *many* I mean in the thousands. I think our cluster architecture was designed to have approximately 1,000 simulation servers and 1,000 data servers and 1,000 databases. The economics are much better for us to have 1,000 small databases than one database that is 1,000 times bigger. The price curve on databases is non-linear.

CD I've seen that a lot of people are starting to adopt scripting languages in their games. Are you doing that?

WH Yes, we use variants of Lua. We call it "ThereScript." It's Lua plus whatever functions we've added to it.

CD Do you have a submission pipeline? What can users submit?

WH Right now, users can submit clothes. I'll describe that pipeline for you. The user signs up for our developer program, downloads a pattern, and uses Photoshop to edit the texture within that pattern. The user then submits it, and There reviews it for any flagrant copyright violations, then adds it to the auction system, which is like eBay's. When anyone buys that shirt, say, the user who made the shirt gets the proceeds from that sale, minus some cost for manufacturing, which goes to us. That's how we make money to cover our costs for doing the review.

Users can also submit textures for objects, like the dune buggy or HoverBoards. They can also submit models that they've made in 3D Studio Max.

CD How does There's economy differ from other games?

WH Many of the massively multiplayer online games appeal to a small market of extremely passionate people. Dungeons and Dragons, for example, is a small market relative to the number of people who chat. But the people are passionate about it.

CD Passionate as defined as "willing to pay for it?"

WH Willingness to pay for it. Number of hours. Significance in their lives.

Those differences translate to different architectural demands on the product.

There, from the beginning, has included online socializing. There are 30 million people who have chatted in IM [instant messaging] within the last 30 days. Thirty million people is a huge number. So scalability has been very important to us.

CD What is the biggest technological change that has

impacted gaming over the last 10 years, in your opinion?

WH I think the ubiquity of at least 56K modems has enabled massively multiplayer online games. I think the shift from single-player and multiplayer games to massively multiplayer games is a fundamental shift in the medium, for which we are just seeing the tip of the iceberg. The degree of innovation and change for online entertainment over the next 10 years, I think, is going to be strongly in the area of the massively multiplayer online games.

■
CD What were some of the most difficult technical challenges that you ran into while developing and in maintaining There?

WH In terms of product development, the hardest part was focus. A number of companies have tried and failed in the last 10 years to build the Metaverse [the virtual reality Internet world from Neal Stephenson's 1992 science fiction novel, *Snow Crash*]. One reason is that it was too early; computers weren't fast enough, there wasn't a large enough market, etc. But I think the major reason that many companies haven't succeeded is that they have been unable to focus on what version their product is.

The Metaverse is something like version 25 of what we're doing now. It's 10 years out in the future that people will be doing things approaching the Metaverse. If you don't focus on version 1—what you can accomplish and what value you are delivering to users at the very beginning—then you just get lost.

For There, our answer was to create a platform that allows online socializing. That's it. That's the value that we are delivering with version 1 of our product.

Technically, the difficulty for There is that we want to build a platform to enable a virtual world in which many developers, not just us, are able to create the content of the world. So we need to invent that platform. At the same time, we have commercial demands for being a viable business to have the content for that platform.

So our challenge is double. We are building a platform and the content for that platform. The engineering problems have been that you can't really build the content until you have the platform to build the content on. But you can't really know what the platform is until you build, or try to build, the content. So there's some iteration back and forth between building the system and building the application. Through that iteration, the content gets better each time and the system gets better each time.

CD Over the last several years, what do you consider the technology that got the most attention but never materialized? For instance, people have been talking about broadband now for six years. Only now is it really getting any traction, in my opinion. How would you answer that question?

WH High-end graphics on low-end machines—that's a problem that many a video-game developer has fallen on his sword for since 1993 or 1994. There has always been a promise of high-end graphics available on all machines, and the promise still is not met.

When I began There as a startup company in 1998, it was economically and technically infeasible.

But, of course, I knew that there were going to be four or more years of building it before we would be ready to ship our commercial product. So you have to project the curves: the rendering curve; the CPU speed curve; the money-spent-on-the-Internet-on-online-games curve; the number-of-people-who-play-online-games curve. I think we guessed right on almost everything, but we underestimated Moore's Law and we overestimated the low-end graphics capability.

A lot of machines are shipped today with 2.4-gigahertz processors, with under-powered hardware graphics accelerators where the balance of load between the processor and the graphics accelerator is different from what we anticipated. The processor is able to do a lot more on a low-end machine than we thought it would be able to. The graphics accelerator is not able to do as much.

■
CD Especially on laptops—and if you're really going to keep people consumed by these environments, you need to make sure it's available no matter where they are.

WH Absolutely. That's especially true for There, which is a social product. So unlike a passion niche game—such as Dungeons and Dragons—we have to make sure that a

person X and all of person X's friends can use it on their computers, which may be laptops.

CD What about the memory space? Do you see that your target client has enough memory—both hard-drive and RAM?

WH That hasn't been the constraining factor with us. It's been mostly the combination of graphics and CPU. I'm disappointed that we don't have higher-end graphics on the lower-end machines.

CD What latency problems have you been seeing?

WH Latency isn't actually as much of a problem as packet loss. If you were using TCP, for example, then packet loss manifests itself in very jittery latency. If you lose a packet, then you aren't going to get any information until it gets re-sent.

So the problem isn't so much the consistent latency of a couple hundred milliseconds for modem users. The problem is the inconsistent gaps in information. We use UDP [User Datagram Protocol] to get around that problem. The software is designed to make sure that even in the cases of packet loss in UDP that the user experience remains good.

If your client freezes, when information stops coming in, that's bad. On the other hand, it's not so bad if everything on your client continues to work exactly smoothly and responsively to you, even when somebody disconnects your network chord, except you're slowly becoming inconsistent with the real world during that time in which you're not connected because you're not getting real information and the other people aren't getting real information about you. Then you plug back in the network chord and the things that are inconsistent just snap back into place to become consistent when they were getting patches. What's happening is things are getting pulled back into their true positions when those true positions are again on your machine. That's an example of adapting the software to accommodate the network characteristics rather than trying to change the network characteristics.

CD When you released *There*, I think some confusion may have arisen because two other similar games came out at about the same time: *Second Life* [Linden Lab] and *The Sims Online* [Electronic Arts]. How did you approach that problem?

WH All three games have virtually identical press

announcements as platforms for online socializing, or not too far from that. But if you look at the products, they are extremely different. *Sims* is a game, driven by game objectives, and it's actually not a very good platform for online socializing. It's not all that easy to have a conversation. It's great as a game, but that's different from the objective of *There*.

Second Life is an environment where people can build things. It's technically demanding. It's a different market from *There's* market, which is people who want to hang out with their friends. Some of them want to build stuff that will be interesting to other people, but the appeal of *There* is to people who want to socialize, not specifically to people who want to build things.

CD User-submitted content brings with it some trouble. It means you have to build your pipeline to keep people from putting prurient things on a T-shirt or inviting lawsuits with their creativity. Have you seen that kind of thing yet?

WH The first user-created content we had in *There* was a T-shirt. We thought that it would be great if users could also make their own T-shirt designs. Several weeks into that program, users realized if they painted part of the T-shirt using skin tone or the color of the flesh of their avatar, they could make nude avatars or avatars with extremely racy clothes. After this discovery, of course, we had to create submission guidelines because *There* is a PG-13 world.

An online game or a community on the Web develops a culture that is set by the initial group of people in the community. The way people interact in *There* is much different from an online game. You will often find somebody who will be very happy to show you around or loan you his or her dune buggy or HoverBoard so you can fly around and decide if you want one for yourself. *There's* a community spirit. I think that is probably the largest influence on user-created content. It is actually the community of the people. It's not something technical.

CD Can you speak about population levels?

WH We had 27,000 people participate in the beta test. We had a hardcore couple of thousand people who were logging in many times a week; we had in the neighborhood of 1,000 simultaneous users. Since we launched, a lot of our numbers have come out much higher than we expected.

CD How many T-shirts are being created?

WH We have more than 1,000 new auctions submitted every day. One of the top designers of clothes in the game is earning the equivalent of \$3,000 a month.

When you sell clothes, you're dealing in the currency of ThereBucks. You can buy ThereBucks from us. Some users run banks where you can cash ThereBucks back for U.S. dollars. There are 130 Web sites of users who have opened up their own businesses advertising their own clothes. Those Web sites now also include user-run newspapers. They sell advertising on their sites in ThereBucks to other users who are advertising their clothes. They employ reporters who they pay in ThereBucks.

CD Have you seen any growing pains inside the company as the company and technology have grown?

WH One of the objectives of There has been to create a platform that can support FPS [first-person shooter] games, fast-action games, races, all of the kinds of games that you see in single-player or multiplayer versions.

To allow a racing game, say, you must have responsive, simulated physics, so cars can run into each other, collide, go over jumps. If your car doesn't feel responsive when you're steering it, it's not going to be a very fun game. Initially, parts of the system didn't support that instant responsiveness in simulated physics because we thought that it wasn't necessary. About three years ago (two years into the project), we realized that if we really wanted to build There the right way and support instant responsiveness for everything, we had to make that decision then; it would be impossible to retrofit later on when we had mounds and mounds of code written on top of it.

We sucked it up and shifted There to an architecture that supports instant responsiveness across-the-board for all types of interactions between objects and people. Now the platform is pure in that respect. We're happy to have that, but that was definitely a growing pain.

CD How would you describe the technical difference between There and an FPS game?

WH A first-person shooter is a game, and There is a platform. There can have games in it. First-person shooter can be one of those games, but there is a big difference between a platform that supports multiple games and people interacting between those games, and a game itself. That's the biggest difference.

By building a platform that allows for instant responsiveness and fast-action game play, yet still supports hundreds of thousands of people simultaneously we've had to

make a few compromises. Our frame rate is 30 frames per second. For a good first-person shooter, the frame rate will be 60 or above. The graphics for There are not as high end as the graphics on the high-end first-person shooter because we want There to be accessible to people with lower-end PCs.

CD How many employees do you have, and how is your team divided up?

WH There has more than 100 employees right now. We have 30 to 40 engineers working on the product, and a team about half that size working on the graphics.

We are still managed centrally as far as the content and the platform itself. We've built an interface called Animob for third parties to create new content for the world. If you want to program a game to be in There or program a new object like a dune buggy or an airplane or helicopter, then you can program that entirely on top of the Animob interface. Everything you need to know is in that one small interface. We've been working on that now for two years because third-party development is going to be key to our platform success.

I took a team of five people, split off in a separate office, to develop house building and other things that we can't talk about yet, on top of this third-party interface. We are vetting the interface, proving that it really works for somebody who doesn't work within the same walls. We think that sometime next year, we'll be able to start having legitimate third parties developing for There on top of this interface. Q

LOVE IT, HATE IT? LET US KNOW

feedback@acmqueue.com or www.acmqueue.com/forums

© 2004 ACM 1542-7730/04/0200 \$5.00