© *1997−2009, Millennium Mathematics Project, University of Cambridge.*
Permission is granted to print and copy this page on paper for non−commercial use. For other uses, including electronic redistribution, please contact us.

September 2002
Features

Career interview: Games developer

by Helen Joyce

BOOKMARK

Nick at Havok's offices in Dublin

When you're playing a computer game, have you ever wondered how come, when your characters move around their virtual world, they interact realistically with their environment? If you think about it, you'll quickly realise that there is no way games developers could have predicted the exact sequence of events that occurred in that precise game session. Something more general must be happening − there must be some

physics in the game's virtual world.

Mathematician Nick Gray works for Havok, a company specialising in providing the physics behind computer games and animations. His job allows him to play God and play games − creating the rules behind virtual worlds and enjoying the end product.

Cool maths and school maths

Nick's interest in maths started early. "I always knew I loved maths", he says. "My dad was a maths teacher, and he had the whole series of Martin Gardner books. That meant that I got a lot of cool maths as well as school maths. There would be chapters I didn't understand but as I learnt more and more maths I knew that I'd understand more later." He was always certain that he would do a maths degree when he left school, although he had no concrete plans about what would follow. Further research seemed a possibility − "although I couldn't really understand how anybody could actually research! How you could find something that somebody else hadn't looked at? Of course now I know that there are so many areas that there are still directions you can go, but it seemed impossible at the time."

Nick started his B.A. at Trinity College, Dublin, in 1992. After the first two years of his four−year degree, there was the chance to specialize. As well as focusing on algebra, Nick took practically all the programming options. "I had always liked computers. I played a lot of games as a teenager − probably too many! I had considered doing a computer science degree, but I didn't have enough programming experience. I liked playing but I didn't really want to know how they were wired together."

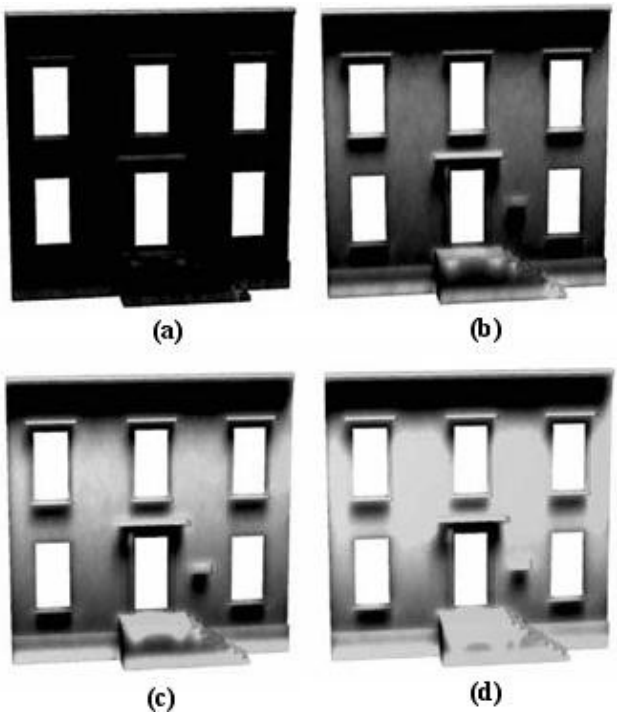
Getting the picture

Nick found the courses on image synthesis (computer graphics) particularly appealing. They were taught by Stephen Collins, who later went on to be one of the two co−founders of Havok. "These courses were really cool because you got to program and at the end you got a picture. That's much better than other programming where you can't show anybody anything! You have to intersect lines and planes and spheres and do calculations with rays of light, but then you program it and you know it works, because instead of getting a big blank screen you get an image. And the better you do it and the more things you model, the better the image looks."

As graduation approached, Nick was sure he wanted to go further into mathematics. He thought about a PhD in algebra, and sent off to lots of universities in the U.S. for information. But he found the enormous packs he received in response overwhelming. "I didn't know which one to pick, and didn't like the idea that I might pick the wrong one. It was a strange time and I didn't really know what I was going to do."

But in the end things just worked out − which seems to have been a recurring theme through Nick's career to date! "I was enquiring around the computer science department [in Trinity College, Dublin] about anything to do with computer graphics and one of the professors there said 'I have some money, I'll take you on,' So I thought 'that's great, I'll just stay here, I don't have to go anywhere!'"

Although Nick enjoyed his M.Sc., it took interdisciplinarity to its limits and beyond. Partly because of the wide range of material he needed to master, but also because he got involved in other projects, it took him over four years to complete − from 1996 to 2000.



The evolution of a building facade during a thunderstorm. Black is dry, white is wet. Image copyright Nick Gray

"The first year was brilliant", he says. "The proposal was an extension of a paper that had just been published on modelling weathering for building facades. You model the outside of the building, you model all the weather and you model the erosion, dirt, algae growth, and you get really good textures to put onto a building. This is useful for anyone who wants to create a realistic environment. The computer games industry need it for games environments, and the TV industry and the film industry need it for faked environments."

Nick explains that the games developer or computer graphic designer will create a virtual world as a geometric structure, described in terms of simple shapes, like triangles and cuboids. This basic structure looks unrealistic because it is pristine and shiny − it needs a bit of virtual dirt. Realistic dirt patterns are pretty complicated, and designing them by hand would be very timeconsuming and boring − and, as Nick says, "who cares about a stain?"

Who cares about a stain? I do!

Nick had to learn about physical, chemical and biological processes in order to create realistic models. "I had to model the sort of brick being used, its porosity, the weathering process − ice expanding and things cracking − how it dissolved and got brought around and transported over the surface, how the acidity of the rain coming down would change the rate of the process, the effects of algae growing on the surface...

"I had to look up all these things and then model all of them with some sort of plausible formulae, because I couldn't implement the really good formulae in any reasonable amount of time. I had to do a bit of mathematics to justify the models − integrals for example, because I had a reasonable amount of probability. The rain is coming in at this angle − what's the distribution over the surface? The rain is bouncing off the ground − how far up does it go and what's the distribution?"

The test of how well Nick's models worked was simple. "It doesn't matter how you do it as long as it looks right. And 'looks right' is the ultimate test, because the human visual system is so good at saying what looks

wrong. At the end of all of this modelling, you come up with a result and if it looks right you say 'yeah, that's a pretty good model'."

The unweathered facade of a virtual museum. Image copyright Nick Gray

Nick showed us one of the images he produced, which used the text "MUSEUM" as a height field, and then aged the surface by running through 20,000 simulation steps. The result is impressively realistic.

The facade after weathering. Image copyright Nick Gray

Nick liked the fact that he could show people a picture and say "I did that". But there was a downside. He spent a lot of his time "looking up... Swedish cement erosion reports! and things like this. And even when I was into it and enjoying it, I knew nobody would enjoy me telling them about it."

While working for his Master's degree, Nick shared an office in the computer science department at Trinity College with students from the image synthesis group. They were doing a research project on physically−based animations. The two supervisors − Steve Collins and Hugh Reynolds − decided to set up a company based on their ideas in this field. They started up in 1999, at which time Nick was finishing up his MSc. From watching what the students in that group had been doing, and liking the look of it, he started to work for the new company, Havok − for free at first, because there wasn't money to pay him. "So again I slid, without any effort at all, from one thing to another, which was great!"

Starting out with a start−up

Animating a pile of boxes falling over Image copyright Havok

During the summer of 1999 the small group wrote a physics engine, which is what other programmers use to provide the animation for virtual objects.

"We worked for the games industry initially, and there were several reasons for this. One is that it was a market that hadn't yet been developed. Also we knew we couldn't do something sophisticated enough to handle an industrial system. A massively important factor was that it was a fun industry, and it's so easy to get in there and just start tackling problems. The sort of skills you need to write games are the sort of things you pick up by writing your own games."

Nick uses examples to explain what a physics engine does. Follow the links to the shockwave demos. "If you want to animate a pile of boxes falling over, you really need physics to do it properly. But a lot of the time it would just be a side−effect of something else being knocked out of the way. Or, you have a piece of cloth on a character − the artist might animate the character, but they won't be concerned with animating the cloth, they just want the cloth to follow the character and bend and crease. If you've got physics you can attach physical objects to other things which the artist or gameplayer has control over and they will react and give a nice secondary motion and save the artist a lot of effort."

Attaching physical objects to each other. Image copyright Havok

The overwhelming pressure on Nick and his colleagues is to find fast ways of doing things. Physics engines are used both by artists creating (non−realtime) animations, and by games developers, creating games or interactive things, which must be real−time, because they react to what the gamer does there and then. For realtime animation the computer must calculate how everything reacts to the current set of forces in the system as the user sees it. So it has to regenerate the position of objects 60 times a second. "Every sixtieth of a second, the engine looks at the positions of all the objects in the world − their mass properties, which are integrated over the volume to find how heavy the object is; their inertia tensors, every force in the system, maybe some object attached to a spring, there might also be some constraint such as a hinge − and it applies the differential equation to move everything forward by a sixtieth of a second. It's got to do all this in *under* a sixtieth of a second!"

It's a mathematician's world

Most of Nick's colleagues have degrees in computer science, but he fits into the company as a mathematician. "This means I'm really comfortable problem−solving and manipulating things, and not phased by reading a paper full of equations. A lot of the basic data structures that we use, like vectors and three−space and rotations and translations, are mathematical. In order to be comfortable with them and to know you've done things correctly you have to think mathematically.

"For example, how can you parametrise rotations? Well, you can use an orthonormal matrix − but that requires nine variables. So there's redundancy, and if you implement it on a computer you lose precision, and start to creep towards something that isn't orthonormal. So how do you reorthonormalise it?

"Or, how do you average two rotations? It isn't obvious − it depends on the underlying mathematics. Quaternions are really good for rotations. They are on the surface of a unit hypersphere, so if you have one point here and another point there then the average is just halfway along the arc from one to the other and you just work out what halfway along the arc is to get your answer."

The future for physics

Things change very rapidly in computer games − processors get faster every year, and games get more sophisticated. Companies like Havok can only stay in the lead by innovating constantly. They have a symbiotic relationship with games developers, supporting them as users, and relying on their feedback to incorporate the right new features. "We write the tools", Nick says, "but we spend a lot of time helping them to use the tools and they help us fix our own design of the tools. We are constantly redesigning the engine and thinking of new features to put in, based on what users want.

"In this area there are so many unsolved problems. In mathematics, if you solve something, that's pretty much it. It doesn't matter how complicated the proof is. But we're limited by how fast an algorithm runs. We might have a way of doing something that's just too slow, and maybe we can't wait until next year, or five years' time. Maybe we have to develop a new algorithm. And there's new hardware coming out, for the PC, Xbox, Gamecube, PlayStation2, so we have to redesign our algorithms to get the most out of them. We can't just rest on our laurels. We've constantly got to be improving things, getting them faster and better−looking. This time next year, what problems are going to be unsolved? It's going to be great."

About this article

Image copyright Havok

Helen Joyce is editor of *Plus*.

Nick Gray works for Havok at their offices in Dublin. The company was founded in 1998 and also has offices in Redwood City in California, and Guildford in England.

Plus is part of the family of activities in the Millennium Mathematics Project, which also includes the NRICH and MOTIVATE sites.