Crossroads
The ACM Student Magazine

Preparing for a Job in the Games Industry: An Interview with Ensemble Studio's Colt McAnlis

by *James Stewart*

In 2004, Colt McAnlis worked as an intern for Petroglyph Games while completing a BS in Computer Science at Texas Christian University. In the first quarter of 2005, he became a graphics programmer at TKO Software. In the fourth quarter of the same year, he accepted a position at Microsoft's Ensemble Studios. In addition to his programming duties, McAnlis is a lecturer at the Guildhall at SMU, a graduate program that specializes in video game development.

In an interview conducted in May 2006, McAnlis offered advice—often frank and irreverent—to students pursuing jobs in the games industry:

Let's say you're a freshman undergrad who wants to work in the games industry. How do you start? I'll say this: You generally have to be one of the best game programmers to get into the industry right now. If you're a freshman in college, and you're just now getting aboard, you've got a lot to catch up with. I won't say it's too late, but there are people out there who have been doing it since they were ten. The people who push themselves first are going to be better. If you're driven enough, and you're just starting as a freshman in college, you're going to do fine. But you can't approach it lazily. We're not an industry that has positions waiting around for random generalist programmers.

What course of study is most valuable for a would-be game programmer? If you're starting college, and you want to get into the game industry as a programmer, my suggestion is to become a math major. Pure and simple. If you become a computer science major, you're not going to get the math background that you really, really need to work in the industry. It's not that you need to know Calculus III, it's that you need to be able to adapt to those sorts of systems and solve complex problems. Become a math major and take a large amount of computer science electives on the side. Do you understand how Newton's laws of motion apply to a rigid body in a viscous fluid? That's what's going to make the next generation of games—not that you know how to use a deque in STL.

What's your opinion of specialized programs for game development? I think that game development schools are a great idea. They give us better programmers. I expect the next generation of game developers to be better than I am. I don't care if I have twice as much time in the industry; I expect them to come out of these schools with twice as much experience. The same thing that took me five years to learn because I had to teach it to myself, I expect these guys to learn in one semester—and I expect them to be damn good at it, too. If someone comes from a game development school and they're not that good, I think, "What were you doing for four years?"

Any drawbacks to attending a specialized program? One problem is that they're trying to offer a prestigious academic education for a chaotic industry full of people who drink nothing but Mountain Dew and stay up till three in the morning. The schools that adapt and constantly change will be the schools that are best for the students.

Is there a litmus test to distinguish worthwhile schools from fly-by-night schools? How cutting-edge are the professors who work there? Are they doing research? If you've got a professor teaching graphics, and the last engine he lectures about is four years old, he's not worth his weight. Now, if he can tell you everything that's been published about an engine that hasn't even been released yet, and he can pick it apart and understand it, that guy has some good stuff. That's going to be someone you want to learn from, because he's driven enough to stay on the cutting edge even as he's teaching.

What sort of programmers is the industry looking for? We're in the age of the specialist. Companies aren't looking for programmers in the abstract. They're looking for network programmers, AI programmers, tools programmers, graphics programmers. Specialization is the new way into the industry. If you're just a regular programmer, there's not going to be a spot for you.

Which specializations are most in demand? Look on any job boards—everyone's looking for graphics and tools.

What does a tools programmer do? If you're coming on the team as a tools programmer, your job is to know the art content pipeline from when the artist boots up Max or Maya or Photoshop all the way to getting that data into the game. Every hand that touches it, you need to be there. So you need to write exporters; you need to write converters. When someone says they're looking for a tools programmer, what they really mean is that they're looking for a content tools programmer: someone who knows a little about MaxScript, a little about Maya, someone who knows the .obj and . ase formats and how to parse and load these files. You have to understand 3D transformations and how to calculate normals.

How do you prepare to be a tools programmer? Become a C# god. Hunker down and just do it. C# will allow you to get tools up and running faster than anything I've ever worked with. Not just from a graphics standpoint, but also in terms of content generation. It's definitely the tools language for the future, or at least the next five to ten years. If you don't jump on board with C# right now, you're going to be left in the dust real quick. I still think C++ will remain the core of game development for quite a number of years, but C# allows you to produce tools faster.

Let's talk about work samples. Do you have any advice on preparing a demo reel? You definitely have to put your due diligence in. You have to have demos that show you know the basics, and then you have to have some demos that show you can branch out—that you're not just recreating DirectX samples. You have to research. You have to be looking at different things. That's what will make you stand out. I can get a thousand resumes on my desk from people who just graduated from game development schools, and they all say the same thing. The only difference is what their final project was. Not one of them has extracurricular stuff. What do you do when you're at home? Do you work on the crazy stuff? You should.

How important is it to know people already working in the industry? The more people you know, the better off you are. It really is like Hollywood. It's always great to have other people in the industry to shoot the breeze with. It's important to spend your time wisely at conferences like GDC (Game Developer's Conference). There's good information to be had, and there's a great deal of potential contacts there.

Let's say you don't know anyone in the industry. How do you develop contacts? A great way to find people on the inside is just to talk to them. Not everyone can afford a pass to GDC, but everyone has an email address. Before I got into the industry, I would download presentations from the previous SIGGRAPH and other conferences, find two or three bullet points that concerned me, or that I had thoughts about, and I would email the authors. If we had a correspondence that went on for more than two or three months, I'd say, "Hey, I'm going to be at this next show. I'd like to talk to you about this and this" and then I'd go and meet them. If you don't know people, you've got to go find them.

Last question: How does a student know when he or she is ready to work in the industry? For any specialty, there's a universal test: If you can look at a game, and you can produce the same effect—whether it's rendering or networking or AI or physics or anything—if you can look at a game and say, "I can duplicate that" then you are ready to start working in the industry doing that. Until you reach that point, you're not ready.