Midterm Tips

1. **don't do all-nighters: they make you stupid**

2. start studying early

3. make questions and ask each other

4. do old midterms

Curriculum Preview: SE380

Rollen will talk about one of your core courses, SE 380 (Feedback Control).

Game Development

Lots of SE students are interested in game development. So I did some research about the industry. Here's my take on it. I talked to two friends in the industry, both based in Montreal. Their paths:

1. education in theatre, on to web development, novel writing, and currently a career focussing on indie game design (art track);

2. Waterloo computer engineering degree including most co-ops in the game industry; currently works at Behaviour Interactive (`ww.bhvr.com`) (tech track).

Workload. The game industry is notoriously time-sensitive: demos absolutely need to be ready by the big deadline. This results in unhealthy work habits ("crunch time"). Crunch time works because people really want to be in the game industry. It can be frustrating when poor management results in a crunch not actually having been needed. I got a report that it's not as bad as it used to be, with the maturation of the industry; however, it still does depend on whether the company is working on contract with another company (better specs) or if it has its own publishing arm.

Work situations. There are indie game devs, and there are huge game companies. Formal engineering education is perhaps best suited/designed for being a cog in a huge company, although Waterloo's entrepreneurship initiatives aim to help equip people for working on their own. If you are thinking of being closer to the indie side of the spectrum, you have a lot more responsibility for yourself. It's worthwhile to learn from artists about how to be an artist and pay the bills. Talk to people in East Campus Hall. Take the Computational Fine Arts option.

Diversity. There's a lot to say about diversity. Perhaps I can summarize with "don't be an awful person, and realize that we all have biases."

Your education. I was surprised that my friend cited some of the more hardware-related computer engineering material as being helpful. In particular, even things like high-pass and low-pass filters (ECE 140) were helpful. Of course, if you are writing code, then the fundamental CS knowledge in courses like CS 137 through CS 240 etc will help. Graphics (CS 488, requires CS 370) is going to help, obviously. And AI is helpful also (CS 486/ECE 457A,B, plus feedback control).

A higher-level statement that I can make is that a lot of the things that we teach you are not immediately applicable. You need to figure out how they relate. So when you see a physics principle, go ahead and also try to program a simulation. There are lots of toolkits on the Internet which let you do that. Almost everything you learn can be applied to game design, but it is up to you to develop the connections.