CMS.922/CMS.610 Media Industries and Systems, Spring 2006
Prof. Christopher Weaver

Lecture 3: Concept Development and the Proposal Process

Readings
Game Design-Bates Chapter 1, Getting to Yes (pp 1-20)
Game Design-Bates Chapter 10, Project Lifecycle (pp 206-218)
On Game Design-Crawford Chapters 6-8 (pp. 71-106)

A word on the papers submitted:
The papers ranged from very non-analytical, short papers to many tomes of constructive, highly detailed analysis in singular planes on the aspects of a game. This is indicative of a typical MIT class because some of the students have all the tools necessary to design and create games, while others need to understand what the tools are. In terms of the pitch, there's a big difference between building a game and pitching a game. Pitching is a little easier than actually building a concept, and for this course, what we shall do is pitch.

Crawford Reading: *On Game Design.* Indianapolis: New Riders, 2003. Chapters 6-8 (pp. 71-106). ISBN: 0131460994.

What we'll talk about today is the nature of interactivity.

In the reading, Chris Crawford talks about an army, and the *Schwerpunkt* of that army. Soldiers have guns, but where did the guns come from? The guns have to be made. They have to be shipped from the manufacturing plant. The soldiers have to eat. There have to be cooks, and what about food that the cooks use? Crawford drives it back to mission central. He argues that the cook is just as important as the soldier. But wait, let's go back to *Schwerpunkt*. The core of the army is that the soldier kills another guy on the other side so that one side wins. There is always a core to which these related support items can be reduced.

Crawford goes on to explain that a computer is simply a typewriter without the interactivity. Similarly, the word processor and the typewriter offer us about the same thing. The typewriter types text. The word processor does the same (ignore fonts, styles, etc.). So what's the difference?

> *Student: the word processor allows you to easily alter the text.*

Good point. You type things in because you know you can erase them or move an entire paragraph. To a certain extent, this tool was supposed to require less paper, but have any of those arguments come to pass? This is an interesting aside.

Don't you find that the word processor and the ability to write something knowing that it can easily be revised has facilitated less prosaic expertise on your part? The great authors like Chaucer, etc lived in a time where there was more preparation for prose. Likewise, Crawford forces us to look at games from the core and question what really is interactive.

Crawford: the computer can do graphics and sound, but that's not the primary design…there's a way of measuring interactivity and he gives three points: how much of what the player might desire to say does the game permit the player to actually say?

> *Student: it's an interesting question, how do you identify what the player wants to say. The player sometimes doesn't know what s/he wants to say.*
>
> *Student: I think a well designed game guides the player to make most of the desires in sync with most players.*
>
> *Student: at one point in Redding's short fiction, Damon Night argues that there are two possible extremes as to what research can be. On one side, you have complete freedom and no constraints, the other one is where you're closed-in in the bottom of a well. Clearly what we want is a balance between these extremes.*

When you came to the strength vs. weakness issue in your papers, many of the weaknesses were the flip side of the strength. Many of you acknowledged that there are substantial flaws in the game and you pointed them out. Nevertheless, the conjunction is followed by, but because of these things, it's my favorite game. It all comes down to game balance. No one can tell you what made the game enjoyable for you. Everyone is a game creator/writer/film director in their head and have their own opinions. We do it because we have a comfort level in our personal environment. Most of us are aware of our limitations, are embarrassed, and don't have it in us to go into a public space and belt out, "I have rhythm." Students only have a week to think creatively of their favorite games and the students are taken out of singing in the shower so to speak and placed on a stage. Students are forced to look at themselves and do something great in a very short amount of time. Everyone in class will read each other's game proposals.

In the fourth week of a 14 wk class, we will have a stripped down core of everyone's game. The student games will be voted on, and the 4 top concepts will remain. Each concept will be pitched by 3 to 4 students per group. Initially, the idea creators will create their own team, but the professor has the final say. The professor needs to have the final say because students need a diverse group to succeed with their project. It's not enough to have 3 or 4 literature specialists any more than programmers alone on a team. Not that there's anything wrong with literature scholars or programmers in moderation, but the pitch must be held together in all aspects of the design and execution. The professor's associates who work in the games industry will be the judges of the final projects and they will be very straightforward in their critique of the pitches.

A number of the students want to get jobs in the games industry. Other than being a beta tester, sooner or later students will want to develop a sense of where they want to go. Remember, everyone wants to be a film director, but not everyone is cut out to be one.

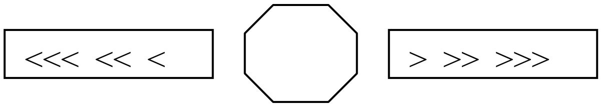
What sounds great in the shower may not sound so great on stage. These professionals at the end of the term will hopefully find very few weaknesses in your pitches. That is the ideal.

Crawford's 2nd Question: How well does the game think about the player's inputs? How well did the designers get out of the way of the player? What makes a game intuitive? In terms of your quick comeback, sometimes the thing that makes a game your favorite is that it simply makes sense to the player. And that's a completely valid reason. What makes a game intuitive?

> *Student: I could hand the controller to a person who has never played the game, and he/she could figure out what's going on in less than a minute.*
>
> *Student: I think it may be more about the controls than the game. You can have a very complicated game, but if they can't even master navigating through the world, that's something completely different.*
>
> *Student: when you sit down and look at the game, it should be obvious in 10-15 seconds what you should do. Depending on how complicated the game is.*

The first question had a lot of different interpretations. *What might the player expect?* Everyone applied their sense of creativity, etc. When we talk about things like interface, and how well the game thinks about what the player wants; we are addressing the question in a much more coherent way.

Spatial Data Management System. This design from twenty five years ago made sense through symbology. This is what they eventually came up with.



You press the first left arrow to go backward, the three left arrows to go backward really fast. To play push the single right arrow and to fast forward press the 2x or 3x arrow. The stop sign does exactly what you think it should do.

9 times out of 10, when you come to a door, you don't have to examine it and figure out how this particular door works. This applied methodology is intuited through experience. Interface should ideally take advantage of that learned ability and allow your experience to make the complex seem comfortable and understandable. This is the whole concept behind "front end" software—where complex systems are controlled by easy to understand graphic controls.

Crawford's 3rd Question: How well does the game itself express reactions?

Student: it has to do with the feedback.

When you put the three together, what do you get? Crawford is forcing you into his version of the perfect game.

When you're creating your game, think about the platform. You wouldn't believe the number of people that try to shoehorn Civilization (complicated game) into a cell phone, and then wonder why the cell phone companies won't buy the game. You don't have to pitch on every single platform. It's possible to pitch to one because you think it works best on that particular platform.

Managing teams is like the Chinese proverb that "ruling a large country is like cooking a small fish"-- it has to be done very carefully. Building a game is a very demanding process. Most people would not walk up to a neurosurgeon and direct him/her as to where to put the gamma radiator relative to a particular finite tumor. People inherently understand they lack the qualifications to do that. But everyone has their own opinion on games (and movies).

Some aspects of Crawford's book are convoluted and make things more complex than they actually are. But that's OK. It is worth wading through his prose to find the occasional jewels he leaves for us.

Crawford says that in the 90s, the game industry stomped out all of the creativity that existed in the 80s. There's a kernel of truth to what he says. Here's the problem, on the one hand, you do get better at something like an engine if you get enough tries. But, along the way it may cost you in time not to mention money. You are rarely rewarded for a long period of inactive time unless you do something very novel. So it may be better to purchase expertise (such as engines) when costs rise as dramatically as they have in the games industry over the past twenty years. Bottom line—even one's own child doesn't do everything right.

The last point to make before class ends is that Crawford says an inner tyrannosaurus rex needs to be developed in every person. The weaknesses need to be pounded out as best you can on your own time so that when your pitch is presented, people will say "wow that's a great concept". Hone your inner critic. It will stand you in good stead down the road.