



# Developer Development

Session I

Chris DeLeon

Sept. 8, 2006

[www.chrisdeleon.com](http://www.chrisdeleon.com)

I love video games.



# Introduction

- **18 years playing games**  
Beat 300+ commercial titles
- **10 years making games**  
Finished 40+ games
- **GCS founder & former manager**  
Involved in 15+ GCS projects
- **Past 2 summers with Electronic Arts**  
Medal of Honor Airborne team



# Why I'm up here

I'm graduating in May.

I want you to know everything I do.

# Topics this semester

1. Console History Paradigm
2. Effective Project Management in GCS
3. Engineer vs. Designer vs. Producer
4. Feature Complexity and Interrelations
5. IP Universe Construction and Analysis
6. Fighting for a Spot in the Industry
7. Classical Game Design Study
8. Summary of Recommended Reading

# Today's Topic

1. **Console History Paradigm**
2. Effective Project Management in GCS
3. Engineer vs. Designer vs. Producer
4. Feature Complexity and Interrelations
5. IP Universe Construction and Analysis
6. Fighting for a Spot in the Industry
7. Classical Game Design Study
8. Summary of Recommended Reading

# Problem #1

**The first game(s) you make will be bad.  
It's ok. Get it out of your system.**



# Problem #2

Having no project is better than  
Having a failed project

An unfinished project is *not* a game.

But it *does* waste a ton of people's hard work.



# Problem #3

**“Finishing a game” is its own skill**

Like programming, modeling, composing, managing...

It takes time, work, and experience to master it.



# The solution (kinda)



TT

# Console History Paradigm

Demonstrate and achieve

- **Atari** complexity, before
- **NES** complexity, before
- **SNES/Genesis** complexity, before
- **N64/Playstation** complexity, before
- **Xbox/PS2/NGC** complexity

# Console History Paradigm

Demonstrate and achieve

- Atari complexity, before
- NES complexity, before
- SNES/Genesis complexity, before
- N64/Playstation complexity, before
- Xbox/PS2/NGC complexity

***Note: nothing says you MUST take on more complexity if you don't want to!***

# Why?

- Earlier projects have less stable teams
- Prove **work ethic, leadership, and communication** before thinking “talent”
- Chance to focus on making a game FUN
- If the game doesn't turn out awesome, less time and work went into it
- The finishing experience is equally valuable regardless of project scale

**I'm serious.**



**It'll make your next games better.**

**Whatever you're thinking...**

...is probably more complicated  
than you're thinking.

# Think Simpler

How can it require less complicated assets?

How can it be finished in less time?

# The trick

You'll have modern APIs, programming languages,  
and hardware

Virtually all games before Playstation were programmed  
entirely in assembly (incl. SNES & Arcade!)

The N64 clock speed is  $< 100$  MHz

The SNES runs at 3 MHz with 1 MB of RAM

The Atari 2600 had 4KB game size, 128 bytes of RAM

# Meaning

You should be able to do  
**alone in 6 days**  
what took a  
**team of engineers 6 months**  
in 1977.

But “can” isn’t the same as “did.” Try it.

**Any Questions?**

