Hardware Matters

But Less Than It Used To?

Quick Introduction...

Technical Director at Buzz Monkey Software
Making games professionally for 16 years (6 at Buzz)
Shipped 15+ games on 9+ platforms

3rd or 4th year speaking in this class
Always give a little into on how the game industry works
Then go off on issues related to multi-platform console games
Finally talk about what I look for when hiring engineers

But... I wanted to mix it up a little this time around

First, Some Evangelism...

Game development is hard!

- Technically challenging
- Schedule pressure & team conflicts
- Doesn't pay top dollar, unfortunately.

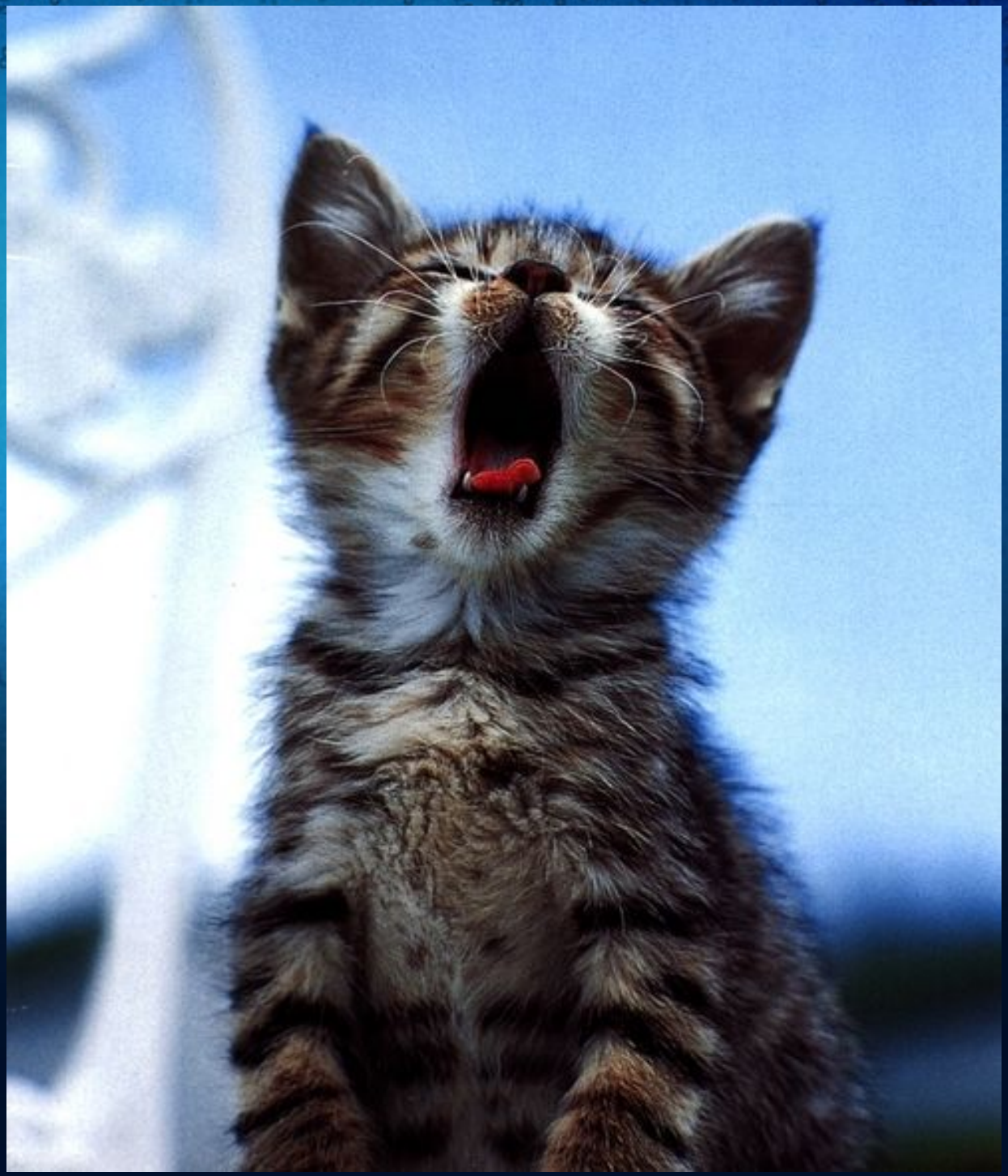
But...

- It's challenging work
- It's creative
- It's always changing (!)
- And it uses more CS than any other job I've come across...

Aside #1 - Relevant Computer Science

- Obviously graphics & AI
- Data structures & algorithms
- Computer architecture
- Compilers
- Operating systems
- Multithreading
- Networking
- Even databases!

Still Awake?



How The Game Industry Works - Follow The Money (Definitions)

Interactive Entertainment industry is large (more $$ than movies)

Publishers have that money.
They buy *licenses* for intellectual property (IP).
They hire *developers* to make the games.
They work with *retailers* to sell the games.
Console Manufacturers license titles for various platforms.
They're risk-averse when competing for shelf space -> sequels.

Examples:
Sony (publisher, license holder, console manufacturer, developer...)
Microsoft, Nintendo, Activision, EA
WalMart

But What About Digital Distribution?

Steam, XBLA, Sony, etc.

Cuts the retailer out of the equation, right?
No shelf space issues. No physical cost-of-goods (boxes, manuals).
Should be glorious!

But practically speaking, most of these titles aren't successful.
The digital shelf space is too crowded
There's a lot of price-pressure

12-18 Months Ago…

Console game market tanked. Wii sales especially.
Big studios shut down (Harmonix, etc.)
The WiiU is coming. Yay.
The PS Vita is coming. Yay.

Still some huge console projects out there: Call Of Duty, anyone?

But the big money is chasing...

Social & Mobile Games

Facebook games
Zynga is the big dog (Farmville, Cityville, etc). Playdom, Kabam, etc.
"Freemium" model - free to play, with microtransactions

Games are different
Designed for smaller chunks of time (like when I'm boring up here)
Need to appeal to a broader audience

Development cycle is shorter & iterative. Release features to 4+ million people every 2 weeks, instead of 12-18 months.

What Does That Mean For Developers?

Casual game devs were enterprise software developers, not gamers.
Console devs were gamers, not scalable network software people.

They're coming together...

The games are getting more sophisticated.
More gameplay elements from 'traditional' games.
The developers have a broader skill set, technically.

Tools pipelines, automated builds/tests, workflows in social games are becoming more like console projects - better able to handle bigger games, and bigger teams.

Modifying 'live' games. Testing cycle is different.

Hardware Still Matters!

Performance in games is still a big key, regardless of the game.
You get really annoyed if Angry Birds gets choppy!

In order to get good performance, you need to understand what's going on at a low level.

The performance characteristics of the PS3 and the PSP are very different, right? So you need to account for that.
Similarly, a gaming-PC and a netbook are very different.
And an iPad and an iPad2 are very different.
Touch input, gamepads, the WiiMote... very different code!

Sometimes you can handle that with code design & optimization
Sometimes you need to modify the design for different hardware

Evangelism, Part 2: Entry-Level Engineers?

Used to be easy:
C++. 3D Math. Graphics, or AI.
Passion for games - show me a demo.

In the last 18 months, my engineering team has worked in:
C, C++, Assembly, C#, SQL, Java, JavaScript, PHP, ActionScript, Objective-C (and several scripting languages)...
How do you hire for THAT?

What We Look For

Do you have a solid, broad-based CS education?
Are you smart?
Do you get things done?

Problem-solving ability
Quick learning
Communications with non-engineers
Still want a passion for games

Lots of people still want to make games, so it's competitive.
Interviews are actually pretty brutal, unfortunately.

Questions?

Please?