

# Hardware Matters – Multiplatform Game Development

Brian Apgar  
Technical Director  
Buzz Monkey Software

[brian.apgar@buzzmonkey.com](mailto:brian.apgar@buzzmonkey.com)

## Game Development Is Hard!

- Technically difficult
- Schedule pressure & team conflicts
- But... it's challenging work
- It's creative
- It's always changing
- And it uses more CS than any other discipline I've come across...

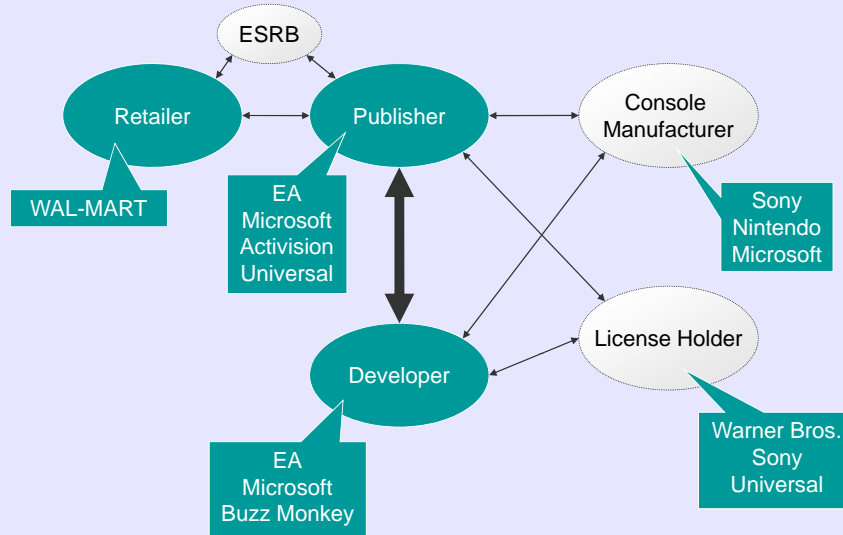
## How Do I Get In?

- What do I look for in entry-level candidates?
- Solid CS background
  - Data structures, algorithms
  - Architecture, operating systems
  - Graphics, AI, networking, multi-core, etc.
- Solid C++ knowledge
- Working with a team in a meaningful way
  - Communications with non-engineers
- Passion for games

## Passion For Games?

- How do you judge THAT?
- Not just “I play games a LOT”
- Demo is great (obviously)
- Awareness of issues in game development is good, too (although harder to see)
- That's what we're talking about today...

## Aside #1 – How The Industry Works



## Aside #2 – Critical Issues

- Performance!
- Feature creep based on competition
- Performance!

## Back To The Technical Side...

- So you're working on a game, and the publisher says it has to be multi-platform...
- Why would they do that?
- What does that mean to an engineer?

## How Do Platforms Differ?

- High-end versus low-end PCs
  - Dramatic differences in CPU, GPU, memory
- PC versus consoles
- Console differences
- Bigger differences
  - Multi-core
  - Shaders
- Input systems

## Lowest Common Denominator

- Approach generally doesn't work.
  - People with high-end hardware want to take advantage of it...
- Enabling features based on hardware
  - Common in PC games
- Means that the code support needs to be present at all times.
- What about consoles?

## More Low-Level Hardware Differences

- Cache. Both d-cache and i-cache.
  - High-end PC has 2-8MB L2 cache.
  - PS2 has 16kB data & 8kB instruction
  - PS3 has 32kB data & 32kB instruction (PPU)
- Bus bandwidth
  - Bus contention becomes a big deal as resources become bigger
- Fill-rate

## What Can We Do About It?

- Different content per-platform
  - Business issue
  - Development time
  - Team sizes
- Tool pipeline to massage data into platform-specific formats
- Algorithms that can scale
  - AI?
  - Physics?
- Performance / memory tradeoffs

## What We Can Do, Part 2

- Abstraction layers
  - Performance issues?
- C++ versus C code
  - Performance issues?
- Data planes
  - Effect on cache
  - Effect on code & maintenance

## Conclusion, Sort Of...

- Design code for flexibility and performance at all levels.
- Yes, that's really hard.

Any questions, about this, or game development in general?