

The Business Of Games

Brian Apgar

Zynga

Intro

- Sorry for the boring white slides!
- My background in the games industry
- (Largely) non-technical talk about how the game industry works
- Specifically the differences between console game development, and mobile/social game development
- Pretty significant shift from console -> mobile

Console Business

- All about risk management!
- Large teams * long times = large budgets
- Competition for shelf space, ads
- Competition for customers (especially around Christmas)
- Extremely hit-driven industry
- This leads to...

Console Business 2

- Lots of sequels and movie tie-ins
- Easier to sell, and easier to predict sales
- EA Sports titles: yearly, incremental
- Call Of Duty titles: yearly, dual teams
- Tomb Raider, Blizzard titles, etc - every couple of years, as the team can build a new one

Console Business 3

- Development (team sizes, project length, marketing budget, etc)
- Larger console games can often have a team of over 100 people for 18 months
- Plus marketing, IT, HR, translation, QA (often outsourced), audio (often outsourced on smaller titles)
- That's often not fiscally possible.

Console Business 4

- Developer / publisher relationships
- So what happens is the publisher (like EA, or Activision) will contract out to 3rd party studios to make games for them.
- Varying degrees of creative freedom, and financial rewards.
- Publisher takes the financial risk, and gets the rewards.
- Developers can pitch demos to publishers after funding some / most / all development, and get a better deal. Surprisingly rare.
- Publisher has the distribution channels, and the ability to print disks / manuals / boxes / etc.
- (Cyclic industry with buyouts and such)

Console Business 5

- Retail channels And approval process
- Great game idea, sold it to a publisher. But wait! You're not done yet!
- Need to get approvals from the console manufacturer at several points along the way (concept, final, etc)
- Need to make sure stores will actually display & sell it (WalMart examples).

Console Business 6

- Revenue models
- What I've mentioned so far is largely for boxed games.
- Steam (and other digital distribution models).
 - Great! Eliminate (essentially) the publisher! Direct to the player!
No box / manual / etc.
 - Increasing, but still need advertising, or some way to get the word out.
 - Still mostly viable for mods and smaller games (right now).
- PSN / XboxLive
 - Still have massive approval process overhead
 - Lower price points, but no cost-of-goods, so it scales better over time
 - Harder to advertise / get attention for games
 - Episodic content really hasn't caught on (hard to do! Storylines versus puzzle-packs, for example)

Mobile Business (Right Now)

- Development (team sizes, project length, etc)
 - Much lower barrier to entry - smaller teams, fewer approvals, etc
- Retail channels
 - AppStore (and equivalent)
 - Completely flooded with games.
 - Very small percentage of apps get bulk of revenue - hard to stay in front of users.
 - Consumers trained to expect games for \$1

Mobile Business 2

- Developer / publisher relationships
- Often there is none. Small / indie developers have a shot, but often production values are lower.
- Publisher brings advertising to the table, and possibly cross-promotion amongst other games in their network.
- Relationship is more equal, with the developer having more influence (because they usually fund development).
- Apple takes a big cut right off the top.

Mobile Business 3

- Revenue models
- Paid apps (often a free trial/demo available -- not free to develop / test / get approved!)
- Freemium (free to play, but buy in-game items/unlocks. Ideally consumables!)

How Are They Converging?

- More "hard-core" mobile games coming out, not just casual games
- What does that mean for engineers?
 - Performance remains critical, regardless
 - Mobile is actually very similar to a networked console game (hardware constraints, memory limits, etc)
 - Social games are similar to MMOs in many ways (databases, concurrency, etc).
 - Data pipelines are critical as games grow. Feels like we're solving some of the same problems now on mobile devices that we solved in the console space 5-10 years ago. Which is going to lead to larger teams / longer times / bigger games.
 - But now people expect the games for 99 cents?

What Do I Need To Get In?

- Plug for relevant classwork:
 - Graphics & AI are the "classic" games classes
 - Networking is big
 - Multithreaded programming is big
 - All that data structures & algorithms stuff is useful daily
 - Databases (cringe!) are a big deal now
- So everything, basically - which is one reason why I love what I do. It's always changing, and we're always solving interesting problems...
- Demo! Make a game!