Big Games in Small Packages

Lessons learned in bringing a PC MMO to Mobile.

John Bergman

Founder, CEO – Guild Software, Inc.

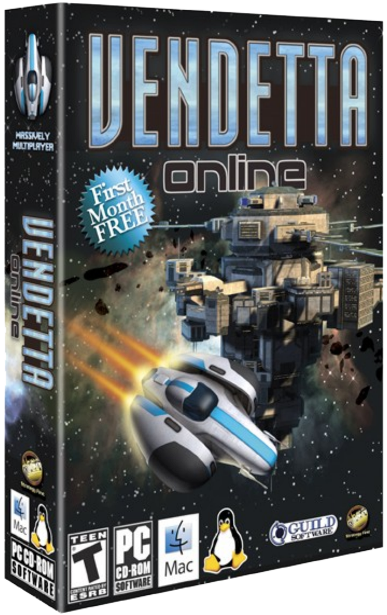
GAME DEVELOPERS CONFERENCE

SAN FRANCISCO, CA
MARCH 5-9, 2012
EXPO DATES: MARCH 7-9

2012

Who We Are

- Guild Software – Small independent studio, based in Milwaukee.
- Became a full-time startup in 1998, building MMO game engine from scratch.
- In 2004, shipped our first retail product, multi-platform space MMO "Vendetta Online".
- Since then, we have continually expanded upon the live game, supported by our paying subscriber base.



Why did we go Mobile?

- Huge emerging market.
- Good timing: Early 2010 - "real" performance had just arrived, chance to be one of the first out of the gate (free marketing, preloads, etc).
- "*Social Spiderweb*" game design approach. All platforms are "portals" into the same game world, from different vantage points. Enabling users to connect through as many mediums as possible.
- "NAOS Engine" middleware.

Helpful Strengths

- We use our own engine.
- *Vendetta Online* has always been multi-platform.
- We have maintained a high level of client scalability.
- Game has always made use of procedural universe generation. Space is Very Big, we re-use art assets as much as possible.
- As a result of asset re-use and compression, game is under a 300MB total download.
- Same art assets used across all platforms, both a strength and a liability.

Specific Challenges

- Game must be self-updating, or able to *reliably* push out patches on very short notice (hours). Lengthy “validation” is not an option.
- As an MMO, there is no guarantee of what content is onscreen, making performance profiling more challenging than a single-player game.
- Fast-paced game with intense combat and “twitch” flight model.
- Very complex interface, difficult to miniaturize.
- Which leads to..

"VENDETTA ONLINE" LEARNING CURVE

Mobile Platform Trends

- Android generally growing faster on phones.
- iOS growing faster on tablets (Kindle Fire may change that?).
- Android is strong in Asia (85% of South Korean market?) and emerging markets (China, Brazil).
- Intel is still “out there”. Somewhere.
- Nokia with Windows should not be counted out, but:
  - *NO* native apps on Windows (XNA or Silverlight only)
  - Windows 8 may be MS's longer-term mobile play.

What's in a Platform Choice?

- As an (MMO) game developer, I care about:
  - Does a platform have a "*Whole Lot*" of users?
  - Is there solid monetization infrastructure in place?
  - How is the platform trending? Will the investment still be of value in 3 years? Longer?
  - (For non-MMO developers) is there a fairly strong copy protection solution?

For all of the above, either iOS or Android are likely to be fine. Much investment is applicable to either..

Which Mobile Platform?

- Apple Pros:
  - Consistent devices, known specifications.
  - High quality level of components (IPS, 10 touch pts)
  - Larger tablet market share. Great phone market share.
  - Users have strong purchasing power.
- Apple cons:
  - No self-updating apps *officially* allowed.
  - Idiosyncratic and opaque app/update approval.
  - No way to self-distribute app if not approved.

Android Pros & Cons

- Android Pros:
  - Self-distribution of APKs is at least *possible*.
  - Publishing and updating Market apps is nearly instant.
  - Google is friendly to self-updating apps.
  - Many high-performance devices, huge phone market share.
  - Many large (competing) companies as potential partners.
- Android Cons:
  - Many different device designs, specs, qualities.
  - Not as cohesive an experience as iOS, evolving fast.

Other Platform Tradeoffs

- IOS supports Subscriptions; not available (yet) on Android.
- Control and Input Methods:
  - ***Android openly supports Gamepads, Mice*** and Joysticks in recent OS versions. USB and Bluetooth. Raw HID support available too. Mouselook still a problem.
  - ***iOS is notoriously unfriendly to control accessories*** or allowing HID access. Most "controllers" are keyboard emulators. New 60beat controller may be different, but not Apple-supported.

Kindle Fire - “The Other Android”

- Amazon Pros
  - Fast growing market share. 5+ million units out there?
  - Kindle Fire has good performance and content.
  - Coming from Android, the “port” is trivial.
  - Amazon is self-update friendly. So far?
- Amazon Cons
  - Self-update friendly, but app publishing has opaque and idiosyncratic (lengthy) validation, in common with iOS.
  - Amazon may suddenly discount your app.
  - No In-App Purchase solution yet. (Sudden discount risk?)

Nook Tab - “The Other Kindle”

- Nook Tablet Pros
  - Growing market share. Shipped 1 million units in Dec.
  - Performance is identical to Kindle Fire (dual core, etc).
  - “Captive” audience to B&N Marketplace, no sideloading.
  - Seemingly self-update friendly.
  - Ownership skews strongly female (pro or con depending on app).
- Nook Tablet Cons
  - Non-standard data layout, only 1GB available to user.
  - No in-app-purchase solution.

Commonalities of Android & iOS

- Both have a GCC-based toolchain.
- Both are largely for ARM (Intel making noises about Android on Atom, we'll see).
- Both support OpenGL ES 2.0.
- Both have physically similar interfaces (touchscreens, accelerometers, gyros, compass, etc).
- More in common than, say, DS vs PSP.

Downsides to both Android & iOS

- App discovery is poor. Glu and others build “app networks”. Social discovery systems help too.
- Mobile marketing is also a very nascent industry.
- Industry moves very quickly, agility is needed to stay ahead of the curve.
- Neither iOS or Android are unassailable. All this has been built in a few scant years, and new challengers could arise just as rapidly.

Primary costs in a Mobile Port

- The user interface. Design, testing, trial & error.
- Code porting, APIs, performance profiling, shaders.
- Installation and self-update mechanics.
- For those needing copy protection, your *specific implementation* is the most critical component of success or failure.
- Asset conversion (texture compression, we'll come back to this).
- Of the above development time, probably 60% is *platform-independent*.

On-going maintenance costs

- Patch QA on phones, tablets, TVs, washing machines..
- Ongoing bugfixes for corner cases (proactive to avoid negative reviews).
- Necessary updates for new OS revisions (game crashes on Ice Cream Sandwich! Woohoo!).
- For Apple, the above all remain true, but with less frequency and much fewer devices.

Fragmentation: Texture Assets

- Remember 1997? D3D vs OpenGL vs Glide? Ugh.
- OpenGL ES 2.0 is great, but common texture compression standard has no alpha (ETC1).
- Generating proprietary compressed textures is easy, but ongoing Q/A and maintenance can be a hassle.
- Memory footprints can vary across different compression standards, more difficult to predict.

Chips & Standards

- The mess:
  - PVRTC (PowerVR – Apple, TI, Samsung)
  - DXTC (NVIDIA Tegra only for now)
  - ATITC, 3Dc (Qualcomm Snapdragon, can load runtime converted / swizzled DXTC)
  - ETC1 (ARM Mali GPUs, everyone else).
- Apple *ONLY* supports PVRTC. No ETC1 at present.
- All (significant) Android devices support ETC1, but no alpha channel. May require time to re-tool RGBA assets.

Other GPU Pitfalls

- Complex shaders may reveal substantial issues that vary between chip types.
- Even within a single chip type, OS and driver changes can massively alter performance or stability.
- Substantial differences between supported GL extensions (between chip vendors, OS versions).
- On Android, users are often at the mercy of OEMs and carriers to ever get improvements to drivers or OS.

GPU Performance Tradeoffs

- Upside: Everything is pretty fast.
- PowerVR is fast at fill rate, less at vertex processing.
- Tegra is more fill rate limited, but has fast vertex processing.
- Snapdragon seems more like PowerVR than Tegra.
- Mali seems pretty fast, we haven't tested heavily enough to say where it has tradeoffs.
- Innovation appears more intense in the mobile GPU space than PC, power usage obviously a big factor.

GPU Solutions and Management

- Use Android “manifest” settings to expose only to tested GPUs. *<uses-library>* and *<supports-gl-texture>*, etc.
- Detect GL_RENDERER strings on first game runtime to configure default settings. But, this is not absolute: “Adreno 205” vs “Adreno (TM) 220”, etc.
- Stage your rollout by chip family, so you aren't overwhelmed by too many issues at once and can react quickly, reducing negative reviews.
- Take the simplest path: RGB, ETC1, test broadly.

CPU Performance Differences

- Almost everyone licenses ARM Cortex designs.
  - *(Qualcomm is a noteworthy exception to this).*
- In my opinion, only ARMv7 (FPU) worth targeting.
- Some aiming for many cores, lower clock.
- Others for fewer cores, higher clock?
- Upshot: we should be prepared to make use of a lot of cores. They are easy to “drop” to save battery life.
- Mobile CPU+GPU performance is escalating at a staggering rate. Over 10x in the last two years?

Other Technical Considerations

- RAM can be tight. Detect device specs on first runtime, make some reasonable assumptions.
- Consider the implications of the app being killed.
- Elegantly handle being backgrounded (incoming call), maintain network state, etc.
- Don't unload textures when backgrounded.
- Use a frame limiter, don't waste the GPU needlessly.
- Always sleep the app to the minimum necessary state.
- Consider the worst-case-scenario of accidental app misuse: Phones are safety equipment.

Android's Hidden Advantage

- *Many* competing OEMs, chip companies and carriers, all want to competitively differentiate through.. *content*.
- Pre-load bundling (aka “annoying bloatware”) is possible on Android. Not so much on iOS.
- Individual OEMs move 30+ million devices per year.
- Major potential exposure with a trial pre-load on a single device that “only” moves 100k units per week.
- Partnerships can carry over into advertising, direct funding, other areas.
- Advanced knowledge of future device release dates.

Example: Verizon TV Ad

- *Vendetta Online* was the central feature of Verizon's TV launch ad for the Motorola Xoom.
- Very expensive ad campaign, strong prime-time rotation (cool looking ad, as well). $50+ million?
- *Vendetta Online* saw a direct 30% spike in subscriptions over four weeks following the ad. Correlated largely with new PC users (not enough tablets at the time).

Challenges with Carriers

- Some carriers still struggling with the role of content, and their role in featuring it. Slow moving telecom industry dropped into a fast moving river.
- Evolving appreciation for “games”. In 2010, some carriers still thought in terms of “Tetris”.
- Carriers usually choose what goes on their devices, and may not decide until a few days before the lock date.
- Strong expectation of rev-share, along with requirement of proprietary billing API (non-reusable development cost).

Challenges with OEMs

- OEM may break game during pre-load, then ship it like that.
- OEM may not even be the one configuring preload, often an ODM like Compal, Quanta, Foxconn.
- OEM may want to "move quickly" and ship device with unstable beta BSP (Board Support Package, aka "drivers"). *First broken device to market! Yay!*
- OEM may use sub-standard components, and not test cases that use them (like many-touch-point scenarios).
- Some pre-loads will lock 3+ months before shipment.
- In all above device failure cases, the end-user will blame the game itself, and give reviews accordingly.

Trial Conversions by Platform

- Windows conversions were 8x that of Android. BUT..
- Android download volumes were 10-12x vs Windows.
- End result: Android was net positive compared to Windows, purely in initial trial conversions (this does not factor in subscription length, stickyness past conversion, impact of 30% loss per transaction on revenue, etc).
- Adding In App Purchase had little measurable effect on the conversion percentage, for our use-case (pre-pay sub only, no micro transactions, etc).
- Upshot: There are real users on Android, and when they're sold on the product, they'll go the extra mile to pay.

Featured: *Wind-up Knight.*

- Being “Featured” on the front page of the Android Market brings significant exposure. Similar to Amazon, iOS.
- Robot Invader's *Wind-up Knight* saw between 55,000 and 75,000 installs per day during the two-week “featured” period. Peaks on weekends.
- Prior to featuring, they saw about 4,000 installs per day.
- After featuring, uptake fell off linearly. Now they see 2,000 to 5,000 per day.
- *Wind-up Knight* launched Oct 24th 2011, and hit a million users by November 15th.

Thanks to Chris Pruett and Robot Invader for this information.

Where do we go next?

- UI is the biggest barrier to gameplay on mobile.
  - More controllers, device designs will evolve.
  - Game designs for control utilization will evolve ("Halo Effect" from console).
- Mobile is becoming the new console.
  - Wireless HDMI allows output to TVs, phone becomes controller, bluetooth controllers, etc.
  - Such incredible amount of power and data available (location, sensors, etc).. opens a whole new world of gameplay.
  - Phones are a *required purchase*, unlike a console.

Final Points

- ***Plan for a low attention span.*** Engage and impress the user as quickly as possible. (True in all games, but especially on mobile).
- ***Be communicative and proactive with your users.*** Fix issues quickly, respond to emails, and you will receive less bad reviews (resulting in better sales).
- ***Free is better on mobile.*** Freemium is probably best. With integrated in-app-purchase, mobile is a near perfect market for MMOs with a lot of purchasable content.
- ***Be prepared to scale elastically.*** Being "Featured" can lead to a massive increase in short-term users.

The Long View.

- We're just entering a very tumultuous period.
  - Platforms evolving so quickly, and pace is increasing.
  - At the same time, users are less tied to platforms, more potential for rapid migrations, changes in market share.
- Platform and device consolidation is here.
  - In an era of quad-core cellphones with 2GB of ram, what exactly is a PC?
  - The argument of Tablets vs Netbooks missed the point.
  - A whole “gradient” of devices and accessories is likely.

Summing Up: Takeaway

- iOS and Android are both strong markets, porting native PC apps to them has a lot of shared time.
- Partnering with companies can yield big rewards for smaller and independent developers.
- Mobile will be the new console, and maybe the new “PC”. Lines are blurring quickly.
- Mobile MMOs should be less about “new games” and more about “new ways of connecting to their same communities and friends”.

Questions and “Answers”

- Please ask me questions! I will probably ramble vaguely in response and not provide very meaningful answers.

Visit our game here: *www.vendetta-online.com*

Contact me: *john@guildsoftware.com*

GUILD
SOFTWARE