

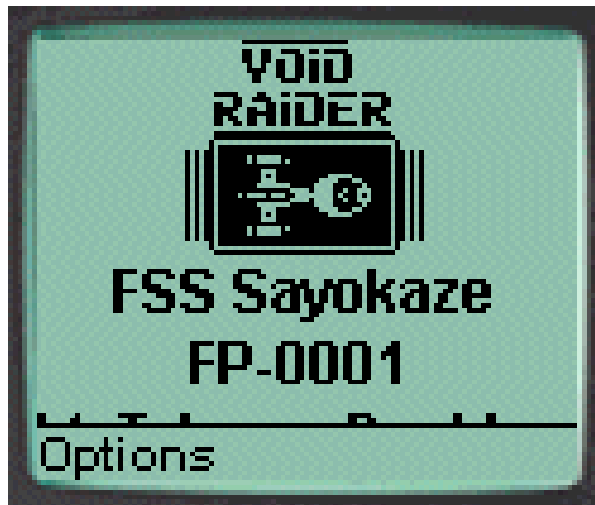
Mobile Games: Medium and Style

Greg Costikyan

greg@manifestogames.com



2000 to 2005



2000
Void Raider



2005
Kingdom Hearts

We've Come a Long Way in 5 Years

- From static b&w WAP to full color and 3D
- From basically zero revenues to \$1b+ worldwide
- Key was interpreted languages (BREW & J2ME)
- Major tech upgrades have made them better suited for games (MIDP > MIDP 2)
- Continuation with compiled apps & 3D support...

But... These aren't "Mobile Games"

- They're conventional video games implemented on mobile devices
- Shooters, platformers, retro arcade, classic board & card games, pool, sports, etc.
- Business model mitigates against innovation: consumers make buy decision on one line of text
 - And therefore gravitate to recognized brands.

Is There Something More?

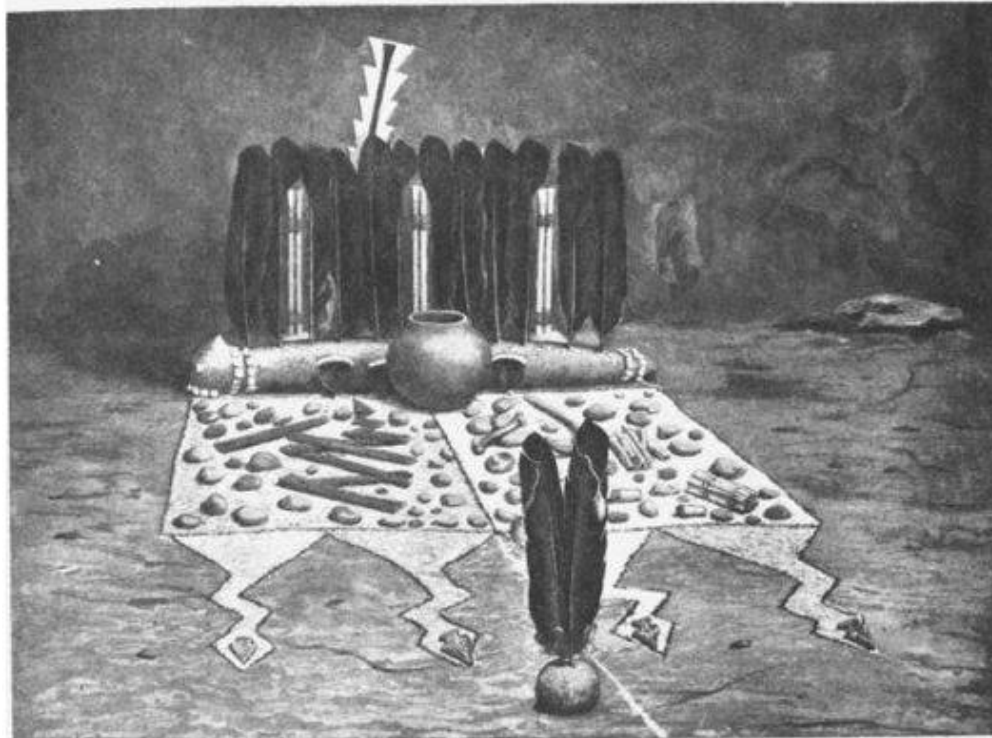
- Is “the game” a medium?
- *“A medium is a means or instrumentality for storing or communicating information.”*
- Books, newspapers, magazines, film, TV are media.
- The CD-ROM is a medium.
- Arguably, “the videogame” is a medium—but not “the game.”

Games Have been Implemented for Many Media:

From the
Neolithic....

*Tewa Kiva Altar at Hano
Showing Gaming Reeds;
Tewa Indians, Arizona*

Games of the North American Indians,
Stewart Culin, Smithsonian Institution,
1907



TEWA KIVA ALTAR AT HANO, SHOWING GAMING REEDS; TEWA INDIANS, ARIZONA; FROM FEWKES

To Modern Digital Media



Will Wright's *Spore*

Games are *Content*

- Media are containers.
- There are many game media: board games, card games, military miniatures, tabletop RPGs, augmented reality games, “big urban games,” sports, arcade, console & PC...
- Our medium is mobile... Does “the mobile game” have to be a subset of “the video game”?

Game Styles

- Think of the potential space of all possible games.
- Most of that space is occupied by games that would not be interesting.
- There are local maxima of games that are (or would be) interesting.
- Established game styles exist where we've discovered local maxima.

Understood Game Styles

- There are many: boardgames of replacement capture (Chess), card games of combination (Poker), RTS (Warcraft), FPS (Quake), the trading card game (Magic)...
- Most games are variations on an understood style.....

Innovation is Driven by Discovering New Maxima

- c. 2000BC: Track game with blocking (Royal Game of Ur > Backgammon)
- c. 800AD: Game of Replacement Capture (Shaturanga > Chess, Shogi)
- c. 1200AD: Game of Leaping Capture (Alquerque > Checkers)
- 1756: Thematic track game (A Journey Through Europe > Candyland)

New Game Styles (con't)

- c. 1850: Trivia Game (Grandmama's Game of Useful Knowledge > Trivial Pursuit)
- 1856: Word Interpolation Game (Komikal Konversation Kards > Mad Libs)
- c. 1890: Fishing Game (Fish Pond > Operation)
- 1910: Military Miniatures (Little Wars > Warhammer)

New Game Styles (con't)

- 1953: Board Wargame (Tactics)
- 1972: Adventure Game (Colossal Cave > Myst)
- 1973: Tabletop Roleplaying (Dungeons & Dragons)
- 1974: Vehicle Sim (Atari Tank)
- 1977: LARP (Dragohir)
- 1978: MUD

New Game Styles (con't)

- 1979: Flight Sim (Sub-Logic Flight Simulator)
- 1981: Platformer (Donkey Kong)
- 1981: Computer RPG (Ultima 1)
- 1984: Graphic Adventure (King's Quest)
- 1985: Dynamic Puzzle (Tetris)
- 1991: First MMOG (Neverwinter Nights)
- 1992: RTS (Dune II)

New Game Styles (con't)

- 1993: FPS (Doom)
- 1996: Rhythm Game (Parappa the Rapper)
- 2000: Autonomous Agent Game (The Sims)
- 2001: Collectible Miniatures Game (Hero Clix)

Technology Can Spur New Game Styles

- 19th century: Full-color printing: Commercial boardgames
- 50s: Cheap Die-cutting: The board wargame
- 70s: First computers: text adventures, Hack, etc.
- 70s: Arcade games: platformers, vehicle sims, etc.

Technology & Game Styles (con't)

- 80s: First home computers: CPRGs, graphic adventures, dynamic puzzles
- 90s: More processing power: RPS, FTS, MMOGs
- Networks: MMOGs, deathmatch play, etc.
- ...but sometimes a new game style is simply an imaginative leap (RPG, CCG, The Sims)

Different Game Styles are Suited for Different Media

- Pac-Man would suck as a boardgame—you require continuous motion.
- Digital games cannot recreate the experience of a narrativist RPG or a LARP
- RTS requires a pointing device—RTS games for consoles suck.
- A mobile device will always be an inferior video game platform because of its small screen.

How do We Bring About “The Mobile Game?”

- A much harder question than how to allow video games on mobile devices—that’s technologically straightforward, the problem is understood.
- We’re trying to imagine game styles that do not yet exist.
- One approach is to look more closely at what a game style *is*.

What is a Game Style?

- A collection of game mechanics that together create compelling experiences.
- Example: The RTS
 - Resource extraction
 - Building construction
 - Buildings improve tech & build units.
 - Units explore & engage in real-time combat.
 - Goal is conquest of competing players (or AIs).

Understanding Game Styles (con't)

- All these mechanics existed *independently* before Dune II:
 - Real-time combat: Crawford's Patton vs. Rommel
 - Building construction: Civilization
 - Resource Extraction: Crawford's Guns and Butter
- ...but Dune II put them together in a novel way—discovering a new maximum in the space of all potential games.

Understanding Game Styles (con't)

- Example: The RPG (Dungeons & Dragons)
 - Players control individual characters
 - Characters are highly detailed
 - Characters gain in power and ability with play
 - Gamemaster serves as referee, NPCs, and story-teller
 - Gameplay focused on combat
 - No physical representation of gameplay space—transpires wholly in the imagination
 - Wholly open-ended: If you can imagine it, it can happen—gamemaster creativity in response to player action

Understanding Game Styles (con't)

- Some *but not all* of these mechanics existed in previous games:
 - Chainmail (fantasy combat, some units representing individual ‘heroes’)
 - Use of referees in *kriegsspieler*
 - But the concept of “player as character” is novel—and key to the appeal
 - So is the leap from physical representation to imagination

A “True Mobile” Game Style Would:

- Provide a collection of game mechanics that, together, produce pleasing experiences for players...
- Some, but not necessarily all, of which would depend on the unique characteristics of the mobile device.
- Some, but not necessarily all, of which, already exist in current games.
- Some of which involve imaginative leaps we cannot realistically predict.

What Makes Mobile Different as a Medium?

- Voice
- Personal device, rarely shared
- The user's own information (phone book, datebook etc.)
- Networked from the inception
- Ubiquitous—always on, always with user
- Location—usable anywhere
- Camera (very likely near-universal soon)

But almost NONE of these capabilities can be used by mobile games!

- Voice: No ability to make a simultaneous voice and data connection.
- J2ME cannot access phone book and date book information...
- Nor can you use phone book information to challenge other players, share games with them, invite them to a team, etc., etc.

We Can't Use It (con't)

- Networked—but hard to use it
 - SNAP helps, but...
 - Latency still sucks.
 - JAR model makes it hard or impossible to add new levels, content, etc to a game on the fly
 - Hard to connect devices without a server in the mix
- Location—but cell-based LBS sucks, and GPS is not a solution (too slow, doesn't work in dense urban environments)

We Can't Use It

- Camera—
 - But virtually no APIs of use to game developers
- Personalized—
 - But virtually impossible to detect this from an application (what's my ring tone?) or share it with others (here, have this tone).

The Most Important Thing Nokia Can Do

To enable true mobile games is:

- Figure out how to allow developers to use the features we *already have* on mobile devices.
- ...And create technologies that allow them to use these features better...
- ...And ensure that these are widely (ideally universally) deployed...
- ...In cooperation with other manufacturers, operators, and other components of the value chain.

Such As?

- VOICE!
 - Player communication is *vital* to every multiplayer game, from the most trivial (online board and cardgames) to the hardest core (MMOGs).
 - In the 80s, the commercial online services noticed that just adding text chat to classic board and card game offerings spurred usage by a huge factor
 - Xbox Live owes its success to VoIP—without it, and without a keyboard, it would be no where.

Voice (con't)

- Text entry on a mobile device is hard—and adequate only for very slow-moving games.
 - Multiplayer mobile games will *only* succeed when players can talk and play at the same time....
- And:

“Single-player games are a waste of devices built for human communication.”

--Justin Hall, The Feature

<http://www.thefeature.com/article?articleid=26917>

Voice (con't)

- We need to ensure deployment of OMA PoC as rapidly as possible.
- VoIP is worth looking at too—but there are issues (carrier concern over loss of voice income, mixing of several players' voice data, etc.)
- Dual radio phones?

Such As?

- Using my phone book.
 - SNAP has a buddy list—
 - But my *real* buddy list is my phone book.
 - It needs to be exposed to applications via an API.
- “I want to play Maria.”
 - I don’t care if she’s “online” on SNAP—send her an SMS now with a link to the game.

Using the Phone Book (con't)

- “My guild members have chosen to share their phone numbers with me so we can text or call each other.”
 - Add them to my phone book.
 - Again: The phone book needs to be exposed to the application.

Using the Phone Book (con't)

- “I like this game and want to send Carl the demo (or buy him the game).”
 - But Carl’s on T-Mobile, and I’m on Vodafone.
 - And my phone is Nokia while Carl’s is Samsung.
 - T-Mobile’s demo version is someplace on their deck where I can’t find it...
 - And the Samsung version is a different build anyway.
 - And T-Mobile wants its share of the application sale, but has no ability to let me buy the game for Carl from Vodafone...

Using the Phone Book (con't)

- Couldn't this be solved through the OMA?
 - E.g., an XML scheme for defining mobile content cross-device and cross-platform...
 - Which operators then interpret to locate the right content for the right device?
 - ...And share revenues in some transparent fashion?
- (The utility for other forms of mobile content as well should be obvious.)

Using the Date Book

- Similarly: “I’m scheduled for a tournament at 4PM on Friday.”
 - Add it to my datebook.
 - Alert me 15 minutes beforehand.
 - Maybe even open the game automatically at the right time.
 - Expose the date book to the application.

Networking

- Latency:
 - Ultimately depends on operator configuration.
 - 3G helps.
 - Worst problem is with MIDP 2, which only *requires* support for HTTP, which is 2-3 times slower than UDP and IP sockets.
 - Support UDP and IP connections.
 - Push for mandatory UPD and IP socket support in MIDP 3?

Networking (con't)

- I want to add content on the fly
 - Rather than shipping the whole application on purchase, I may want to send the user new levels or other content as needed.
 - Or the application may want to pull personalized content from other users as needed (my avatar, my pic).
 - Developers can build this functionality—but a standardized API would be helpful—
 - Ideally, a cross-vendor one... With billing integration if we want to charge extra for some content... OMA again?

Networking (con't)

- I want to connect players directly.
 - It costs money to keep a server up and active to handle multiplayer gameplay.
 - For small-scale (<8 player) games, it's more efficient to have gameplay processing occur on users' machines.
 - For S60 native applications, at least, I can get an IP address—but it may change if I move to a new cell.
 - SIP may be the answer, but it's not optimized for gameplay. Still, encouraging widespread, cross-vendor deployment is helpful...

Cameras...

- Some obvious use cases:
 - “Lock in” on my foot, use motion as a game controller.
 - Detect edges in camera images and use that to interact with superimposed sprites/game graphics.
 - Detect average color in image, pass that to application.
 - Superimpose game image on camera image when snapshot taken (“Maria hanging with Sonic the Hedgehog at Suomenlina”)

Cameras (con't)

- Developers can build this—but innovative use of the camera will happen quicker if they have access to a set of standard APIs—
 - Again, ideally cross-vendor.

Conclusion

- The success of mobile games has been built on a set of cross-vendor, cross-operator standards (mainly J2ME).
- But currently, “mobile games” don’t take advantage of the unique characteristics of mobile devices.
- True *mobile* games will be new game styles that are uniquely mobile.

Conclusion (con't)

- The way to get there is to establish cross-vendor, cross-operator platforms that allow developers to use the things about mobile devices that are *different* from other platforms.
- We can't predict what game styles will succeed—but we can help build the technical infrastructure to enable them.