Designing for Everywhere

Raph Koster, areae
What the heck??

You may have noticed this lecture has one of the vaguest topic descriptions you have ever seen.
What’s it about?

• **Got this new startup thing going**
  – Been watching the Web a lot
  – Want to be on all devices

• **Also been pondering game grammar**
  – New book in the works, allegedly

• **Does one inform the other?**
The question

• What do these have in common?
The question
• What do these have in common?
A provocation

- Basically, I am going to assert we usually think wrong about what games are
  - They are not the interface
  - They are not the display
- Design for everywhere is about cutting those two elements from the equation
Topics

1. The new platform
   • Web 2.0 in shorthand

2. How Games Work
   • From a game grammar point of view

3. Bridging the gap
   • Hopefully concrete advice and examples
The new platform: the Social Web
The overview

- Tags, not taxonomy
- Participation, not publishing
- Radical trust
- The three R’s
- Decentralization
- Long Tail
- Data not code
- Perpetual beta
- Remix and mashup
- Emergence
- Services not products
- Collective intelligence
Participation

• Web 2.0 is premised on users contributing hugely
  – “Radical trust”
  – Remix and mashup
  – Cult of the amateur
  – Quality not required
  – Distrust of centralized authority
• Ex: tags, not taxonomy
Decentralization

• Abandonment of the publisher model
  – Long Tails
  – Niches
  – Duplicate content

• Different distribution channels
  – Digital only
  – Monetize passion, not trials
  – Slow openings, not big
Services

• **Instead of products**
  – Data, not code
    • *(Hard for games! We make **systems**!)*
  – Perpetual beta

• **Collective intelligence**
  – Distributed activities
  – Group filtering of content
The three R’s

• **Ratings**
  – The participatory Web is premised on metadata on “content”

• **Rankings**
  – And metadata on “users”

• **Reputation**
  – Adding up to a user-driven system of surfacing user-created content
Run anywhere

• **Open standards**
  - XML
  - RSS
  - HTML
  - CSS

• **Common platform**
  - LAMP stack

• "Above the level of a single device."
In contrast

- **Games are moon shoots**
  - Just contrast to Blizzard’s MO!
  - Open big
  - High production values
  - Monetize trials

- **Central authorship**
  - Closed platforms vital to monetization
  - Big on narrative and aesthetics
How Games Work
What is game grammar?

• The assertion that all games
  (And sports, puzzles, and in fact most cognitive tasks)

  work the same way

  – And this functioning can be notated or diagrammed
  – And this understanding can help make games better
Layers

- **Model**
  - The mathematical system representing some aspect of reality
- **Statistical variations**
  - Variation on numerical inputs into the model
- **Theme**
  - The real life situation that the model analogizes
- **Dressing**
  - The representation of the model to the user
Model

• A black box algorithm
  – The purpose of gaming is to determine what the algorithm is and apply it

  – Scorched Earth & Worms share a core model
  – We often term a model “a game,” or “a genre.”
My grammar

**Nested and sequential atoms**
Each with success and failure states
“Depth” measured by nesting
“Breadth” measured by parallelism

**Each atom should hit key factors**
Topology, past interaction, skill required, etc
(see GDC ’05, ETech 07)

**Notion of “imported” systems**
Human psychology, real world physics, etc
Stephane Bura’s grammar

- **Uses Petri Nets**
  - Notation system premised on my grammar
  - Extends to a more sophisticated level for design error trapping
Dan Cook’s alchemy

Game Mechanic

1. Player Action
   - Player performs an action using an available tool

2. Simulation
   - Rules (Black Box): The algorithmic rules and systems that result in token state changes
   - Tokens: Any in-game entity or object
   - State Change: The state of the system changes

3. Synthesis and Filtering
   - Tools!
     - The Player gains a new action they can perform or in-game resources that will enable them to perform actions in the future
   - Useful Info!
     - The player is given knowledge about how to affect useful changes to the simulation
   - Useless Info :-(
     - The player notes that some info does not advance their understanding model.

3. Feedback
   - The game communicates critical state changes to the player

Other Game Mechanics

Designing for Everywhere - Koster - AGDC 2007
Commonalities

Games as atomic skill challenges
Emphasis on player learning
Chaining of atoms to measure difficulty (even quantitatively)

(See ITI Techmedia/Metaforic sponsored session, GDC 07)

De-emphasis on interface and presentation
Comparisons

• **Mechanics, Dynamics, Aesthetics**
  – Grammar strongly echoes the mechanics layer, and purposely drops the aesthetics layer
    – Dynamics are implied

• **Constituative rules per Salen/Zimmerman**
Model as algorithm

A game can be seen as a collection of $f(n)$ equations.
Put in “rock,” and get back rock, paper or scissors.
Put in angle and muzzle velocity, assume wind, and try hitting a worm.

Expertise is basically data-analysis
Iterate, and determine the algorithm from the output
Then successfully use the curve predictively.
Curve fitting
This view of games ties in nicely

The underlying system to “grok” is the blackbox algorithm

Mastery of patterns is rewarded with a burst of endorphins

See AGC 2003, SGS 2004, TFall 2005, book; also Prensky, Gee

Every atom is an obstacle to master

Each button, each level, each move, each UI element, each monster, each …
Classic models

- Reaction time
  - Most sports and sport-like games
- Trajectory calculation
- Calculus of consumption
- Weighted graph
- Influence spheres
- Odds calculation
Models as graphs

• It is possible to use graph theory to treat all models as traversal problems
  – Construct a weighted graph
  – Each node is bounded by a success/failure state
  – Each node is quantified based on odds of success
  – The player A*’s through Halo 2
    • cf Wired article on tuning Halo
The social dimension

- **Games are inherently social**
  - Except videogames, sort of

- **Used throughout human culture as teaching tools for both youth and adults**

- **All games have player and opponent:**
  - The self
  - Another player
  - A deterministic model
Symmetry

- **Symmetric games**
  - Self and opponent have the same goals, graphs, etc

- **Asymmetric games**
  - Self and opponent have differing goals
  - Important to realize that a deterministic model is still “a player”
    - “Shall we play a game?”
Simultaneity

• **Directly competitive games**
  – Players play against each other, then measure results

• **Parallel games**
  – Players play a different game, then measure their relative results
  • *Foot races, leagues, XBLA achievement points*
Roles

• “Team” based games
  - Generally have key roles
  - Intensity of contribution varies by role
  - Parallel games played within an overall symmetric game
    • Most team-based sports (symmetric)
    • Party-based MMORPGs (asymmetric)

• These support limited capacity because roles are fixed

Designing for Everywhere - Koster - AGDC 2007
Statistical variations

This is often termed “content”

Placement of the worms
Stats on the orc
Rain or windiness on the racing track
Chess problems

The model is the same, the terms in the equation are different

“For given values of y…”
Iteration

Games are generally looping
In order to allow the user to build mental models

Therefore going too deep before a failure is bad design
The user cannot attribute a cause to their failure, and often quits

Broadly successful games punish very lightly
Time

• All games are turn-based
  – Or rather, phase-based
  – In some, the time term is infinite (or socially determined)
  – In others it is constrained by model execution speed

• Time-constrained choices bind you to certain platforms
Pushing to the theme

• The underlying model is “what the game is about.”
  – Abstract themes

• A good theme will match the model
  – Trajectory calculation is a good match for shooting games – and even photography
  – Not a great match for a game about being a tree
What is “interface”

- **A misnomer**
- **Three sorts of interface:**
  1. Inputs
  2. State information
     - **Graph**
     - The “world”
     - **Tabular**
     - Hit point bars, etc
  3. Feedback information design
     - Causal feedback (e.g., the “diff” from last state, what your command actually caused to happen, including the opponent’s reaction)
Inputs

• **Strongly shape the user experience**
  – But are “remappable”
  – Even automatable
  – Are never analog (even though the controller might be)
    • *In fact, are generally binary*

• **Mastering one is a game atom itself**
Diagram of inputs

- What we think of as simple isn’t

Input mapping:
The physical action

Input alias:
The suggested mapping

Command:
The actual input into the black box

Cmd table:
Map command to algorithm
Outputs

• Typically broken into representational models
  – But this is by convention
    • “It begins to rain.” in EQ vs graphical weather
    • Rogue in ASCII versus a tileset

• Can also be seen as filtered
Designing for Everywhere – Koster – AGDC 2007

Diagram of outputs

• Output is also filtered

State delta
Signal of last input received

Total state:
Current graph status, given imperfect information

Info encoding:
Information packets

Client display:
Parse packets into representation

User parsing:
Mental model
Dressing

May take any form
A great example is from Mahk LeBlanc, cited in *Rules of Play*

Also consider
Chutes & Ladders
Pac-Man vs Q-Bert
Risk
Bridging the Gap
The game has changed

• The hot platform is the Net
  – Meaning, any device!

• The hot audience is the non-gamer

• The hot feature is other players

• The hot technology is connectivity

• The hot game is a mini-game
  – Meaning, one with little nesting
Considerations

• **Input complexity**
  – Currently, haptics are hot in proprietary platforms, but this drives lock-in, not “play everywhere”

• **Model complexity**
  – Our core gamer audience is graduate level in model analysis in most genres

• **Output requirements**
  – We design from representation and inputs inwards rather than the other way around
Reconsidering inputs

Interfaces have been a huge barrier for users

- 1972: Pong. One “analog.”
- 1979: Atari. 3 binary.
- 1985: Mac. 2 “analog” and one binary.
- 1985: NES. 6 binary.
- 1997: Dual shock: 12 analog, 8 binary

And so on... chunking helps... But only for those who grok it
Reconsidering outputs

**Outputs can change the skill atom**
A 2d positional challenge in text will be “a different game.”

**But if they don’t, you have freedom to alter to suit**
- Camera angles
- Art
- Etc
Skill curve for outputs

Reading info design is a skill
These circles actually show a 25% difference
New Frostbolt Record!
652

+36 Mana
+36 Mana
**Overall game model**

- **Input mapping:** The physical action
  - **Input alias:** The suggested mapping
  - **Command:** The actual input into the black box

- **Develop response:** Opponent’s turn (variable response happens here)
  - **Model updates:** Change state based on input
    - **Cmd table:** Map command to algorithm
    - **State delta:** Signal of last input received

- **Total state:** Current graph status, given imperfect information
  - **Info encoding:** Information packets
  - **Client display:** Parse packets into representation
  - **User parsing:** Mental model

---

Designing for Everywhere - Koster - AGDC 2007
The Web platform

- **Asynchronous**
  - Users issue requests, receive responses later
  - Requests may or may not update a state model (e.g., a database) on the server

- **Parallel or orthogonal**
  - Designed for massively parallel usage
  - For differing tasks

- **Representation agnostic**
  - Data surfaced in varied ways, even skinnable (CSS)
Overall web model

- **Input mapping:** The physical action
  - **Input alias:** The suggested mapping
  - **Command:** The actual input into the server

- **Develop response:** Processed via CGI, DB routines, etc. Variable feedback.
  - **Model updates:** Change database based on input
  - **Cmd table:** Map request to algorithm

- **Total state:** Current DB status, given imperfect information
  - **State delta:** Header data regarding immediate response

- **Info encoding:** HTML
  - **Client display:** Rendering is *advised*, not mandated
  - **User parsing:** Mental model

---

Designing for Everywhere - Koster - AGDC 2007
Too often we mistake “what the game is.”

Inputs shape it

and provide barriers to interaction

Outputs affect the user mental model

And provide barriers to interaction

But the real game is the black box
Architecture

Inputs

Commands

Triggers to the model

Model state updates

Statistical tables

World state representation

Info design representation
<table>
<thead>
<tr>
<th>What works</th>
<th>The system is the game</th>
</tr>
</thead>
<tbody>
<tr>
<td>Universal inputs</td>
<td>Any button will do</td>
</tr>
<tr>
<td>Long phases</td>
<td>Take your time</td>
</tr>
<tr>
<td>Short decisions</td>
<td>Be done fast</td>
</tr>
<tr>
<td>Massively parallel</td>
<td>Side by side</td>
</tr>
<tr>
<td>Extended accumulated state</td>
<td>Save your profile</td>
</tr>
<tr>
<td>No roles</td>
<td>Classless</td>
</tr>
<tr>
<td>Representation agnostic</td>
<td>Draw it however</td>
</tr>
<tr>
<td>Open data</td>
<td>Change it however</td>
</tr>
</tbody>
</table>
What doesn’t

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Short phase lengths for</strong></td>
<td><strong>Twitchy!</strong></td>
</tr>
<tr>
<td><strong>consequential choices</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Inputs locked to commands</strong></td>
<td><strong>This game only works with this custom touch-sensitive dance mat stylus</strong></td>
</tr>
<tr>
<td>(platform lock)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Models reliant on specific info</strong></td>
<td><strong>If it’s not 3d it won’t work</strong></td>
</tr>
<tr>
<td><strong>designs</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Models reliant on prior art</strong></td>
<td><strong>If you played the first three, you’ll feel right at home...</strong></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Narrative lock</strong></td>
<td><strong>If you tamper with our story, you will undermine our extensive critique of Randian objectivism!</strong></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Design for everywhere

Simple info design
    Bars, text, tables, grids

Every game can be shown on a grid
    or other simple graph

Constrain number of inputs
    Flash supports keys plus 1 mouse button

Add graphics last
    Design in “blue squares”

Design massively parallel
Parallel models

- Badges
  - Achievements, etc
- Ratings
  - Skill, or social
- Rankings
  - High scores, etc
- Reviews
  - And tagging
- Gifting
  - Reciprocity
- Networks
  - Social standing
- Leagues
  - Segmentation

**Small games nested within parallel models is the way to go.**

Pretty much any game can serve, as long as it is not “the game” but instead “a minigame.”

**User investment must reside at the metagame level.**
Sponsoring modding

Data, not systems
Front ends/skinning
Niches are often driven by skin

1 creators
10 synthesizers
100 consumers
The Scoreboard:
what they have in common
## Club Penguin

| ✔️ | The system is the game |
| ✔️ | Any button will do |
| ✔️ | Take your time |
| ✔️ | Be done fast |
| ✔️ | Side by side |
| ✔️ | Save your profile |
| ✔️ | Classless |
| ✗ | Draw it however |
| ✗ | Change it however |

Designing for Everywhere – Koster – AGDC 2007
**IMVU**

The system is the game
Any button will do
Take your time
Be done fast
Side by side
Save your profile
Classless

- Draw it however

Change it however
Facebook zombies

| ✔ | The system is the game |
| ✔ | Any button will do |
| ✔ | Take your time |
| ✔ | Be done fast |
| ✔ | Side by side |
| ✔ | Save your profile |
| ✔ | Classless |
| ✔ | Draw it however |
| ✗ | Change it however |
O RLY?/LOLCatz

- The system is the game
- Any button will do
- Take your time
- Be done fast
- Side by side
- Save your profile
- Classless
- Draw it however
- Change it however
## Brain Age

| ✔️ | The system is the game |
| ×️ | Any button will do |
| ✔️ | Take your time |
| ✔️ | Be done fast |
| ×️ | Side by side |
| ✔️ | Save your profile |
| ✔️ | Classless |
| ✔️ | Draw it however |
| ×️ | Change it however |

Designing for Everywhere - Koster - AGDC 2007
HotOrNot

- The system is the game
- Any button will do
- Take your time
- Be done fast
- Side by side
- Save your profile
- Classless
- Draw it however
- Change it however
Bar trivia

- The system is the game
- Any button will do
- Be done fast
- Side by side
- Classless
- Draw it however

- Take your time
- Save your profile
- Change it however
ARGs

- The system is the game
- Any button will do
- Take your time
- Be done fast
- Side by side
- Save your profile
- Classless
- Draw it however
- Change it however

Designing for Everywhere - Koster - AGDC 2007
Travian

- The system is the game
- Any button will do
- Take your time
- Be done fast
- Side by side
- Save your profile
- Classless
- Draw it however
- Change it however
PHP Strategy games

- The system is the game
- Any button will do
- Take your time
- Be done fast
- Side by side
- Save your profile
- Classless
- Draw it however
- Change it however

Designing for Everywhere - Koster - AGDC 2007
Line Rider

- The system is the game
- Any button will do
- Take your time
- Be done fast
- Side by side
- Save your profile
- Classless
- Draw it however
- Change it however

Designing for Everywhere - Koster - AGDC 2007
Fantasy football

<table>
<thead>
<tr>
<th>Rank</th>
<th>Team</th>
<th>W-L-T</th>
<th>Pts</th>
<th>Pts</th>
<th>Streak</th>
<th>Waiver</th>
<th>Moves</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The Wonderwix Test</td>
<td>10-3-0</td>
<td>.769</td>
<td>1555.37</td>
<td>W-2</td>
<td>4</td>
<td>34</td>
</tr>
<tr>
<td>2</td>
<td>Groin Injuries IV</td>
<td>9-4-0</td>
<td>.692</td>
<td>1573.03</td>
<td>W-1</td>
<td>6</td>
<td>33</td>
</tr>
<tr>
<td>3</td>
<td>Chubbysworld</td>
<td>9-4-0</td>
<td>.692</td>
<td>1529.18</td>
<td>L-2</td>
<td>9</td>
<td>56</td>
</tr>
<tr>
<td>4</td>
<td>The CHUMP</td>
<td>8-5-0</td>
<td>.615</td>
<td>1526.80</td>
<td>W-1</td>
<td>8</td>
<td>21</td>
</tr>
<tr>
<td>5</td>
<td>Ry's Rosurgents</td>
<td>7-6-0</td>
<td>.538</td>
<td>1453.44</td>
<td>L-1</td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td>Byronic Bengals</td>
<td>7-6-0</td>
<td>.538</td>
<td>1328.30</td>
<td>L-1</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>Satine Quackers</td>
<td>6-7-0</td>
<td>.462</td>
<td>1554.41</td>
<td>W-2</td>
<td>10</td>
<td>52</td>
</tr>
<tr>
<td>8</td>
<td>Ceekey Airlines</td>
<td>6-7-0</td>
<td>.462</td>
<td>1462.08</td>
<td>L-1</td>
<td>12</td>
<td>17</td>
</tr>
<tr>
<td>9</td>
<td>Wu Tang Rebels</td>
<td>6-7-0</td>
<td>.462</td>
<td>1381.31</td>
<td>W-1</td>
<td>11</td>
<td>49</td>
</tr>
<tr>
<td>10</td>
<td>Addicted to Coffee</td>
<td>5-8-0</td>
<td>.385</td>
<td>1396.73</td>
<td>L-1</td>
<td>5</td>
<td>22</td>
</tr>
<tr>
<td>11</td>
<td>Seoul Suns</td>
<td>3-10-0</td>
<td>.231</td>
<td>1351.45</td>
<td>L-3</td>
<td>7</td>
<td>28</td>
</tr>
<tr>
<td>12</td>
<td>Shareena Atlass</td>
<td>2-11-0</td>
<td>.154</td>
<td>1273.00</td>
<td>W-1</td>
<td>3</td>
<td>55</td>
</tr>
</tbody>
</table>

* = Recent Smack Talk  @ = on Yahoo! Messenger  * = clinched playoff spot

- The system is the game
- Any button will do
- Take your time
- Be done fast
- Side by side
- Save your profile
- Classless
- Draw it however
- Change it however

Designing for Everywhere - Koster - AGDC 2007
Kongregate

- The system is the game
- Any button will do
- Take your time
- Be done fast
- Side by side
- Save your profile
- Classless

× Draw it however
× Change it however
Bottom line

I don’t think it is an accident that the most broadly accessible and popular activities hit most of the grammatical characteristics.
<table>
<thead>
<tr>
<th>MMORPGs</th>
<th>FPSs</th>
</tr>
</thead>
<tbody>
<tr>
<td>✗ The system is the game</td>
<td>✗ The system is the game</td>
</tr>
<tr>
<td>✗ Any button will do</td>
<td>✗ Any button will do</td>
</tr>
<tr>
<td>✔ Take your time (early on)</td>
<td>✔ Take your time</td>
</tr>
<tr>
<td>✗ Be done fast</td>
<td>✔ Be done fast</td>
</tr>
<tr>
<td>✔ Side by side</td>
<td>✔ Side by side</td>
</tr>
<tr>
<td>✔ Save your profile</td>
<td>✔ Save your profile</td>
</tr>
<tr>
<td>✗ Classless</td>
<td>✔ Classless (in DM anyway)</td>
</tr>
<tr>
<td>✗ Draw it however</td>
<td>✗ Draw it however</td>
</tr>
<tr>
<td>✗ Change it however</td>
<td>✗ Change it however</td>
</tr>
</tbody>
</table>

Designing for Everywhere - Koster - AGDC 2007