IMGD 3000 - Technical Game Development I: Gold's Nuggets

by
Robert W. Lindeman
gogo@wpi.edu

Ideas to Build By

- Start small. Get bigger through small, incremental steps.
  - Iterative design allows you to solve progressively larger problems to complete the project.
- Avoid presenting single solutions to critical tasks.
  - There are many ways to solve problems.
- If something works, keep doing it.
  - Don't change for the sake of changing.
- If something doesn't work, stop doing it, and replace it with something that does.
  - Acknowledge your mistakes. Learn from them, and ask yourself: How can I prevent this in the future?

Ideas to Build By (cont.)

- Avoid repeating things you do wrong, and void having to redo things you've already done right.
  - Reuse what you can.
  - Better yet, make your design (and write your code) knowing you will use it again on a different problem.
- No rule, no matter how good, is applicable in every situation.
  - You should use whatever languages/tools/environments/people make the most sense for the given situation.

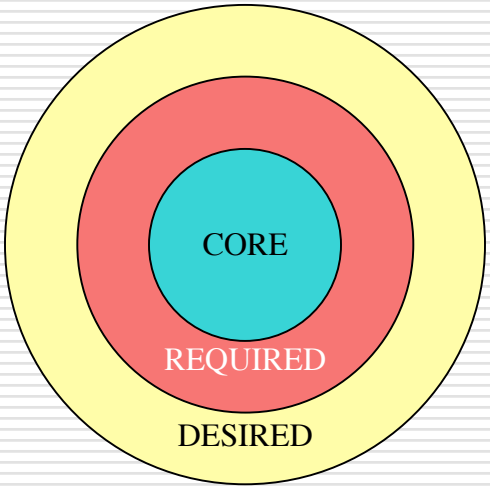
Phases of Learning

- Unconscious Incompetence
  - "I didn't even know I couldn't do it."
- Conscious Incompetence
  - "I'm aware it's not how I'd like it to be."
- Conscious Competence
  - "If I make the effort, I can get the desired result."
- Unconscious Competence
  - "I don't even have to try and it works out."

Development Priorities

- What are some priorities for measuring the quality of games?
- Where should you spend most of your time/effort?
- How would you order these?

Reality of Game Dev: Open-Ended Development



Reality of Game Dev: Heuristic Content

- Constantly making "playjustments"
  - Incremental tweaking of game-play elements to make a game more playable, balanced, etc.
- Subjective, so test with players!
- Eye candy versus substance

Reality of Game Dev: Hardware

- Hardware support
  - Lowest common denominator PC?
  - Console?
  - Handheld?
- Control methods
  - Specialty controller
    - Guitar
  - WASD + Mouse?
  - Camera input?
    - EyeToy
  - Motion-sensitive controller?
    - Wii/PS3

Game Software Engineering

- Games are getting more sophisticated
- Development times are not getting longer
- Team sizes are growing only modestly
  - Various companies/groups involved
- Need to be *more efficient* in development
  - Reduce time scales
  - Use team members better
- Problems
  - Egos, inertia, structure, ...

What Makes a Good Game Developer?

- Good programmer?
  - Language specific?
- Designer and planner
  - Bottom-up and top-down analyses
  - Estimator and scheduler
- Team player
  - Liaise with artists
  - Follow a lead developer
  - Support other developers
  - Technical reviews