

PRACTICAL GAME PROGRAMMING

Good Enough

A blessing and a curse

BACKGROUND

- In general you can do things in three ways:
 - Badly
 - Good enough
 - Properly

BADLY

- Things that are done badly are:
 - Cheaply done
 - Require replacement ASAP
 - Replacement requirement is obvious
 - Placeholders at best

PROPERLY

- Things that are done properly are:
 - Expensive, often prohibitively so
 - Never require replacement
 - Problem solved, period

GOOD ENOUGH

- Things that are done good enough are:
 - Relatively cheap
 - Eventually require replacement
 - Replacement requirement is not obvious
 - Replacement may be seen as too expensive
 - Does the job, for now

WHEN GOOD ENOUGH IS GOOD

- Often, things do not need to be rock solid
- Want text to screen? Do it in a simple manner
- Don't implement everything possible, only what's needed



WHEN GOOD ENOUGH IS BAD

- Sometimes, things don't bend far enough
- Some things just can't be fixed
- Doing things properly can be seen as too expensive
 - So, a "good enough" thing may be slowing down progress



A REAL-WORLD CASE

- Shipping platform was used as a temporary step, instead of building actual steps
- Was supposed to be temporary, was there for months
- Until an accident happened, someone got hurt, and the platform had to be removed

A CODE-WORLD CASE

- Printing text was done in a simple bitmap manner.
- Originally for debug outputs, but versatile enough to draw pretty text.
- Used in lots of games
- CJK support was impossible!
- Lots of workarounds (baked bitmaps etc) since reimplementation seen as expensive
- Eventually replacement was written (and yes, it was expensive)

AN INTERNET-WORLD CASE

- IPv4
- Hey, 4 billion addresses are enough for everybody, right?

GOOD ENOUGH

- "Good enough" is great for the short term, but can be catastrophic in the long term
- Try to see which bits are no longer "good enough", and stop fixing them
- Nothing stops you from doing a "good enough" replacement..