

January 27 -1

Graphics INTRO

Administrivia

- Processing JS
- Partners

Defer conversation about what is a good game / game design
except...

What does it have to do with Tech?

Why do we need tech for Games?

Why does better tech \Rightarrow better games?

Why is games tech different than other tech?

General -
but today graphics

Tech in Games: (why)

Tech-centric games - UI / input device \leftarrow design around
Flashy graphics \leftarrow show off
Physics puzzles

Create complexity

Create richness

immersive world

\leftarrow what is immersion?

VR and books

set up story \leftarrow why?

detail & realism } or "worldism"

procedural creation

Provide effective interaction / feedback \leftarrow FAST AZ

Complexity of Systems - network (massive # of players, responsiveness)
resources (assets, size, ...)

1-27-2

Game Tech vs. Regular CS

Resource Limited - cost conscious

development costs / resources

limited platforms (fixed) - consoles

vs. richness, complexity, ...
can grow

how to live within constraints

Interactive

latency, bandwidth, frame-rate

performance (fast)

also communication, storage, pre-computation

Correctness

- looks good consistent

- simulation, visually, ...

- fake it (big stuff off-screen)

Asset Rich

MULTI-DISCIPLINARY

Focus ON FUN - faster / better / more doesn't always mean more fun

1-27-3

Why Graphics (?)

- historically bottleneck - just repainting screen @ framerate
- computationally intense
- obvious improvements
- no limits to needs (complexity scales arbitrarily)
- positive feedback (games → hardware → ideas)
- hard to do without
- Rapid evolution (cpu is just faster)

Various Tradeoffs

See 2007 Notes

Key Ideas :

Approximation
Preparation
Amortization

} not just graphics

→ avoidance (don't draw)

pipelining / parallelism (exploit hardware) - caches/memory hierarchy
caching / pre-fetching / prediction

Making Flocking FASTER

- sub-linear drawing (if zoomed in)
- $O(n^2)$ neighbor tests
 - n-body methods
 - fast spatial queries
- vs. dynamicness