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 ⇒ better games?
Why is games tech different than other tech?

> General — but today graphics

Tech in Games: (why)

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

Create complexity
Create richness
- immersive world ← what is immersion? VR and books
- set up story ← why?
- detail: realism or "worldism"
  - procedural creation

Provide effective interaction/feedback ← FAST AI
Complexity of Systems - network (massive # of players, responsiveness)
- resources (assets, size, ...)

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 ⇒ better games?
Why is games tech different than other tech?

> General – but today graphics

Tech in Games: (why)

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

Create complexity
Create richness
- immersive world ← what is immersion? VR and books
- set up story ← why?
- detail: realism or "worldism"
  - procedural creation

Provide effective interaction/feedback ← FAST AI

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

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 Floating FASTER

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