Graphics Intro

Administrivia
- Processing.js
- Partnets

Refer 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?

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?
Set up story & why?
Detail vs realism or "worldism"

Provide effective interaction/feedback <= FAST AI
Complexity of Systems - network (massive # of players, responsiveness)
Resources (assets, code, ....)
January 27 -1

Graphics Intro

Administrivia
- Processing, JS
- Pd

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?

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?
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, etc.,...
Why Graphics?
- Historically bottle neck - just repaint 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 Trade-offs
See 2007 Notes

Key Ideas:
- Approximation
- Avoidance (don't draw)

- Pipelining / parallelism (exploit hardware) - cache / memory hierarchy
- Caching / pre-fetching / prediction

Making Floching FASTER
- O(n^2) neighbor tests
- O(\text{linear}) drawing (if zoomed in)
- N-body methods
- Fast spatial queries
  vs. dynamicness