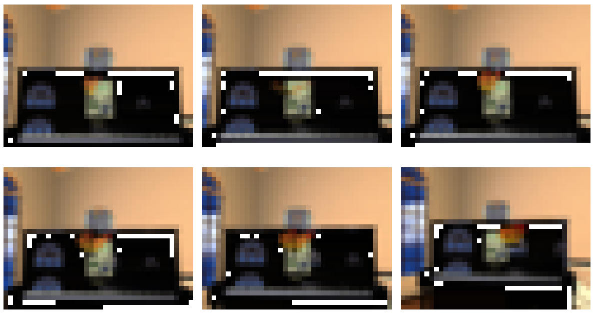
Lecture 15: Thursday, 23 October 1997

Next week:
  Realism I: T, Intro to Texture Mapping (Justin)
  Realism II: R, Advanced Texture Mapping (Jeremy)
Today:
  Animation & Simulation

What is an Animation?

A sequence of generated images depicting change
  Action, life, natural processes, etc.



(Kirk Bowers, Ohio State. `AL/Ox` tool suite:
Steve May, `http://www.cgrg.ohio-state.edu/~smay/AL/`)

State is abstract: Shape, Motion, Lighting, Materials...

Problem (#1): realistic visual experience
  requires perhaps 24-30 frames/second
One minute of animation = 1800 rendered frames!
  Tedious to specify each "by hand"
  Classical animation; claymation; etc.
So: natural approach is to intermix manual
  specification with automatic computation

Direct Specification

Study object/system phenomenology, then express
  avars as functions of abstract parameter t
Animator acquires/constructs base geometries
Specifies all position, other state directly
System simply evaluates specification as $f(t)$
Example: bouncing ball $0 \leq t \leq 1$

$$
\begin{aligned}
x(t) &= At \\
y(t) &= B|\sin(\omega t + \theta_o)|e^{-kt}
\end{aligned}
$$

Drawbacks?

Forward Kinematics

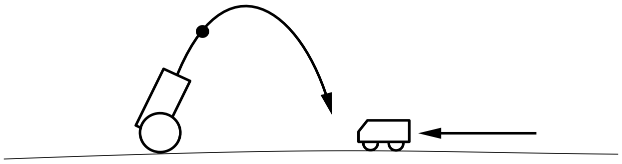
Animator specifies motion, velocity, acceleration, &c,
but no notion of forces that give rise to them
Example: kinematics for projectile in field $\mathbf{a}$:
specify $\mathbf{p}_0$, $\mathbf{v}_0$
System then forward-integrates equations

$$\begin{aligned}\mathbf{a}(t) &= ? \\ \mathbf{v}(t) &= \mathbf{v}_0 + \int \mathbf{a}(t)dt \\ \mathbf{p}(t) &= \mathbf{p}_0 + \int \mathbf{v}(t)dt\end{aligned}$$

Drawbacks?

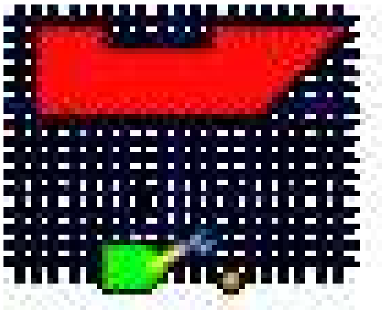
Forward Kinematics

Use physics to evolve state as time advances
Example: ballistic projectile to hit moving car
Specify all start times, velocities (speed, mag)
Is there a better way?



Modeling & Animation

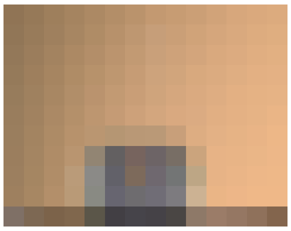
Problem (#2): fundamental conflict between animation, instancing
Objects have local coords, are instanced
So, (usually) easy to deduce pose from instancing, time



But: how to deduce instancing from "desired pose"?
Solutions not unique, may not even exist !

Coupled Avars

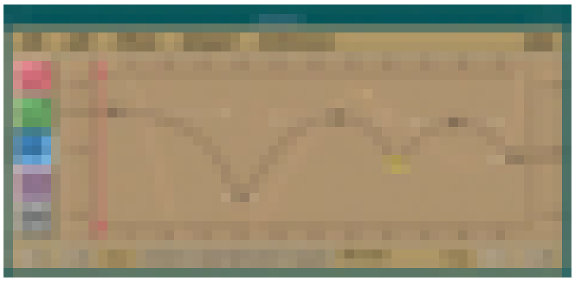
Intermix direct specification of avars with
conditional expressions in terms of avars
(simple example: character looking at another)
Expressions can become quite complex:



Still requires lots of manual planning, tweaking

Control Tradeoff: Keyframe Animation

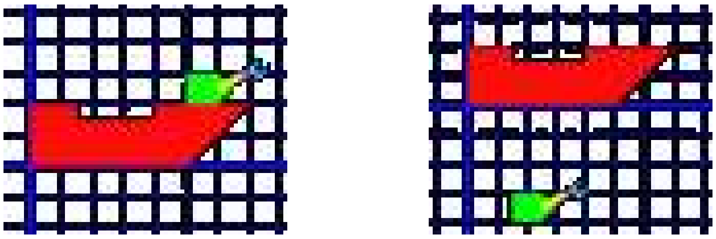
Method of specifying time-varying state
(where "state" is meant abstractly:
shape, color, pose, position, lighting...)
(Related to, but not the same as, morphing)
Keyframing requires:
1) Discrete set of *keyframes*
State specifications at given times
2) Method of "in-betweening" keyframes



Note: in-betweening method is abstraction

Keyframe Animation

Assume some source of keyframes
(Manual specification; observation; simulation)
How can they be in-betweened?
Linear transformation: simply force variables
to have goal states at desired times



Problems: velocity profiles; rotations; cameras

Inverse Kinematics

Animator specifies states:
Partial start state:
"Here/doing this" at time t_0
Partial/complete goal states:
"There, doing that" at time t_1
System solves for appropriate motion(s)
Specification should be sensible – how?

Dynamics / Physically-based modeling

Animator specifies:
(Partial) start, (partial) goal states
Object mass, structural properties
Kinematic degrees of freedom (constraints)
Physical laws to be observed:
Inertia, gravity, friction, fluid resistance
Elasticity, impulse/contact forces
System computes intermediate states
which satisfy the specifications

Physically-based modeling

Example: bouncing superball
- Specify superball mass, size, elasticity
- Specify surface on which it is to bounce
- Specify some initial, final conditions

Did we leave anything out?

Physically-based modeling

Example: spring networks
- 1D, 2D, 3D mesh of springs
- Each with a restoring force $F(x - x_0)$

Specify external forces, constraints
- Massless, massive springs
- Write a *propagation rule*
- Iteratively solve simultaneous equations
- Ideally, rule is convergent (how?)

Physically-based modeling

Not all physically-based modeling is of *motion*!
- Plant growth, decay, etc.
- Surface deterioration over time
  - Erosion, corrosion, etc.
- Spreading chemicals, fire, etc.

Modern Systems

Modern animation systems combine
- Direct specification
- (Forward) Kinematics
- Inverse Kinematics
- Dynamics / Physically-based modeling

Rendering Issues

Synthetic camera motion
Batch vs. piecemeal rendering
Temporal Aliasing
Scene matching, integrity