Lecture 05: Animation

(Selection from Ch. 11, 12, 23-2)

I. Animation loop / concepts

C. Clear - Draw - Show

Update → Clear → Draw → Swap buffers

Change the world

B. Double-buffering - 2 refresh buffers: one shown and one for drawing.

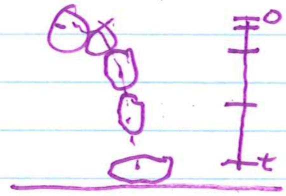
A. Animation - any time-sequence visual change in a picture

i. Real-time - view + creation at a rate capatible w/ human + device refresh rate

ii. Frame-by-frame - hyperrealistic offline generation then glued together in video

D. Key-frame - detailed drawing at a certain time in animation

key-frames for "squash + stretch" of a ball



timing

E. In-betweens - intermediate frames between key frames

example - total frame rate is 24 FPS for 1440 frames so if 288 keyframes designed then 3-5 inbetweens are added between key frames.

i. Determined through interpolation - linear, $\underbrace{\text{cubic, Bezier}}_{\text{ch. 14}}$

II. Overview of animation techniques

A. Squash and Stretch - for acceleration on non-rigid objects

B. Timing - strategic spacing in time between key frames. Close frames gets slower motion.

C. Morphing - transformation of shapes from one to another

~~D. Goal directed - specify by task~~

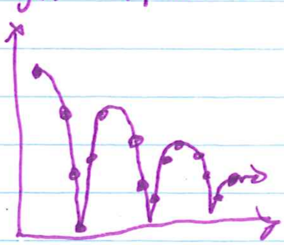
D. Motion capture - recording of human actors for motion

~~E. Periodic motion - repeated motion patterns, eg., rotating object~~

III. Motion Specification techniques

A. Direct motion specification - directly specify all transformations or approximating through an equation. eg.

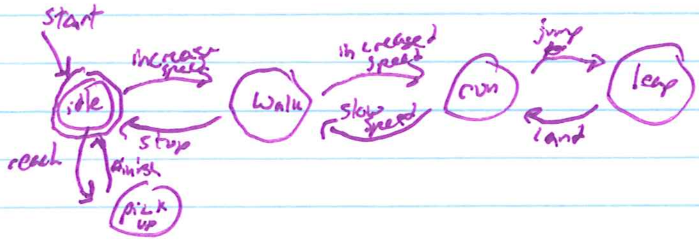
$$y(x) = A\,|\sin(\omega x + \theta)|\,e^{-kx}$$ (damped, rectified, curve)



bouncing ball

B. Goal-directed Systems – define tasks like "walk" or "run" or "pick up"

Can include transition diagrams, e.g. state-machines



C. Kinematics and dynamics

i. Kinematics – give motion parameters of position, velocity, + acceleration

ii. inverse kinematics – give start + goal then derive parameters

iii. Dynamics – specify forces on object, e.g. physically-based modeling (see below) like gravity, friction, etc.

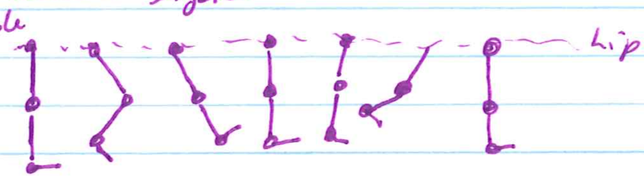
iv. inverse dynamics – Determining forces from initial/final positions

v. Newtons law $F = \frac{d}{dt}(m\vec{v})$ i.e. $F = m\vec{a}$ (more soon)

D. Periodic motions – repeated motions, e.g. rotating object, or series of idling motions

E. Articulated figures

i. Example

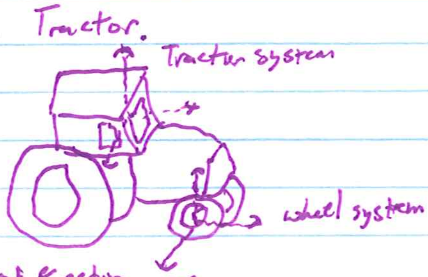


ii. Articulated figure – hierarchical model defining skeleton, then "skin" is wrapped around skeleton (beyond scope of class)

~~[illegible]~~

a. Hierarchical models (Chapter 11)

- model – single representation, e.g. one geometry.
- instance – one specific object of that model
- hierarchical model – Tree-based description of model system
- ex. Tractor.



do a coordinate transform between to define positions

$$Wheel' = T_T R_T T_{TW} R_{TW} R_W \, Wheel$$

$T_T, R_T \rightarrow$ Tractor position + orientation

$T_{TW} R_{TW} \rightarrow$ Tractor to wheel transform

$R_W \rightarrow$ wheel orientation

- Degree of freedom – one axis of motion, e.g. wheel rotation, tractor x, y, θ
- key frame defined as vector of DOFs.

iv. Exercise *In groups of three*

Define hierarchical model + keyframe system for 2D stick figure

IV. Particle Systems (chapter 23-2)

A. Physically-based modeling - equations governing behavior of objects

i. Example - gravity w/ point

Position : $\vec{x}$ velocity : $\dot{\vec{x}} = \vec{v}$ accelleration : $\ddot{\vec{x}} = \vec{a}$

as Forces

$$\vec{F} = m\vec{a} \qquad \vec{a} = \frac{\vec{F}}{m}$$

gravity : $m\vec{g} = \vec{F}$ $\vec{g} = -9.8\ m/s$ in $-\vec{y}$ direction

$$\vec{a} = \frac{\vec{F}}{m} = \frac{m\vec{g}}{m} = \vec{g}$$

$$\vec{v} = \int_0^t \vec{g}\,dt = \vec{V}_0 + \vec{g}\,t$$

$$\vec{x} = \int_0^t (\vec{v}_0 + \vec{g}t)\,dt = \vec{x}_0 + \vec{V}_0 t + \frac{1}{2}\vec{g}t^2$$

ii. Euler Integration - ~~Many system~~

a. Motivation - many systems too complicated to solve equations explicitely, also that is slow. So we trade some time for error.

b. Derivation -

Let Δt be a step in time

$$\vec{V}^{n+1} = \vec{V}^{n} + \Delta t\,\vec{g} \qquad \vec{X}^{n+1} = \vec{x}^{n} + \Delta t\,\vec{V}^{n}$$

written more generally

$$\vec{S}^{(n+1)} = \vec{S}^{n} + t\,\dot{\vec{S}}^{n}$$ where $\vec{S}$ is state in example of ball

$$\vec{S} = \begin{bmatrix} \vec{x} \\ \vec{v} \end{bmatrix} \qquad \dot{\vec{S}} = \begin{bmatrix} \vec{V} \\ \vec{a} \end{bmatrix}$$

iii. So you determine force, solve for $\vec{a}$ and euler integrate.

a. Air force $\vec{F}_a = -d\vec{v}$

b. wind force $\vec{F}_w = d\vec{w}$

c. Total $\vec{F}_t = \vec{F}_g + \vec{F}_a + \vec{F}_w$ etc.

d. Basic loop:

1. set initial conditions
2. for $t \leftarrow 0$ to timelimit step Δt do
3. compute acceleration
4. Euler integrate state
5. update state
6. Draw frame

e. With collisions loop gets much more complicated. See me for more details, or if you would like to handle

f. Resting - when an object gets close to stopping, just stop simulating it. to save computation time. Avoid "popcorn" affect. Many stopping criteria.

$$\underbrace{\|\vec{v}\| < t_v}_{\text{velocity}} \qquad \underbrace{|d| < t_d}_{\text{distance}} \qquad \underbrace{|\vec{a}\cdot\hat{n}| < t_a}_{\text{accel. on surface}}$$

B. Particle systems

i. Set of independent point masses under forces. Notes they do not interact w/ each other

iii. Preallocate memory for system, eg an array of 10,000 cells

ii. Used for simulating effects, eg. water, fire, smoke, ~~or~~ in physics n-body systems

iv. State

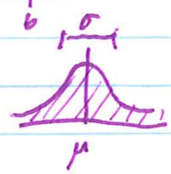
each particle has position $\vec{X}$ and velocity $\vec{v}$, but can also have age, color, temperature, opacity, etc. Whatever you want to simulate

v. Initial conditions - want lots of particles that vary in state, but also want to regenerate particles after they die. Don't generate more than you preallocated for.

a. Generators - a value/vector creator based on distribution

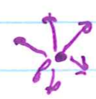
Distributions - many available, but two simplest

- uniform -  each is equally likely in a range

- Gaussian (normal)  μ - mean, σ - standard deviation, more toward central value

b. Point generator

$\vec{X}_0 = \vec{p}$ (input point) $\quad \vec{V}_0$ = random direction $* \vec{V}_0$



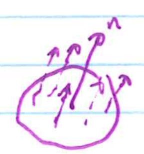
c. Directed generator

$\vec{x}_0 = \vec{p}$ $\quad \vec{v}_0 = (\vec{n}$ + small random direction$) * \vec{V}_0$

D. Disc

$\vec{X}_0 = (r, \theta)$ random point on circle

$\vec{V}_0$ = as directed away from disk normal



E. many more combinations, get creative!

VII. Coreography of forces - beyond gravity/wind

a. Gravity toward a point

$$\vec{f}_g = \frac{-g m_0 m_1}{r^2} \hat{u}$$ where $\hat{u}$ is direction to point

can: (1) ignore second mass $\frac{-g m_1}{r^2} \hat{u}$

(2) be proportional to $\frac{1}{r}$ $\frac{-g m_0 m_1}{r} \hat{u}$

b. Repulse from point

$$\vec{f}_g = \frac{g m_0 m_1}{r^2} \hat{u}$$

c. Anything you can imagine -

- towards/away from lines/surfaces
- potential fields
- etc.

C. Flocking - point masses that interact

D. Spring-mass systems - point mass connected by simulated springs

E. Rigid-body systems - volumes i.e. force + torques

} *ask usages*