animation

- big topic, we will do quick non technical survey
- field is still rapidly evolving

keyframing

- we already saw this topic
- this is the bread and butter of CG animation

Skinning

- start with complicated mesh we want to animate
- rigging: design a geometric hierarchical skeletal structure
- associate each vertex with one "bone" (RbtNode)
- idea: express the vertex in its bone frame
- manipulate the skeleton
- vertices now move along with the bones.

..details

- instead of actually changing the vertex coordinates,
- a rest accumulated matrix represents the relation between the object and rest bone frame.
- a new accumulated matrix represents the relation after the bone has been moved.
- use these matrices, in the vertex shader to move the vertex.
- in this context we can associate a vector of bone-weights and do soft skinning
  - we use the weights to blend the updates.
- this is used extensively in games and would make a nice project
- demo d3d.palette

Simulation

- physics uses equations to describe physical processes.
- we can try to simulate these processes computationally.
- techniques: physics and computational mathematics
- some methods are slow and only work for offline animation.
- some methods can be made real-time
- hard to control the output

Particle systems

- simplest version of physics
- a large bunch of non-interacting particles
- ordinary differential equation (ODE) for the time evolution of a point

$$f = ma = m\dot{v} = m\ddot{x} \tag{1}$$

- force might be gravity or wind

- can model flowing fall of water particles, or a stream of smoke particles in the air.
- Typically each particle is rendered as a semi-transparent little blob or surface.
- demo: ogl.cg-explosion, gpu-particles

ode integration

- Starting from an initial condition, we can discretize this ODE and march forward in time using so-called *Euler steps*

$$\begin{aligned} x_{t+h} &= x_t + v_t h \\ v_{t+h} &= v_t + a_t h \\ a_{t+h} &= f(x_{t+h}, t+h)/m \end{aligned}$$

- steps must be small (often need many more than 30/sec).
- there is a whole literature of more sophisticated ways to solve an ODE.

Rigid Bodies

- upgrade from particles to solid hard finite objects (dice rolling on a table).
- need to deal with rotational issues
- wish to deal with interaction: collision detection
  - bounding hierarchies
  - must undo interpenetration
  - must have the object bounce - this requires hacked physics since real objects slightly deform and undeform.
- must deal with objects resting on objects and not endlessly bouncing
- videos

Cloth

- can be modeled as a grid of particles connected by springs
- can be modeled as mesh of physical triangular elements
- need forces to avoid stretching and shearing and oscillation
- also may need to track collisions.
- demo: ogl.glsl-physics
- videos

hair

- Hair modeling is also often similarly dealt with as a mass-spring model.
- videos

Deformable Materials

- real objects are deformable
- can be modeled as volumetric objects (mesh of 3d tetrahedra).
- videos

Fire and Water

- special physical equations

- modeled with combination of surface and volumetric reps.
- videos

Human Locomotion

- not passive objects
- much harder than previously discussed phenomenon
- ideas are used from robotics, control, and optimization
- nowadays mocap data is relied on heavily, and possibly altered or used as part of the rocket science.
- videos