Animation

- Industry production process leading up to animation
- What animation is
- How animation works (very generally)
- Artistic process of animation
- Further topics in how it works
What is animation?

• Modeling = specifying shape
• Animation = specifying shape as a function of time
  – Just modeling done once per frame?
  – Need smooth, concerted movement
• Controlling shape = the technical problem
• Using shape controls = the artistic problem
Approaches to animation

- **Straight ahead**
  - Draw/animate one frame at a time
  - Can lead to spontaneity, but is hard to get exactly what you want

- **Pose-to-pose**
  - Top-down process:
    - Plan shots using storyboards
    - Plan key poses first
    - Finally fill in the in-between frames
Pose-to-pose animation planning

- First work out poses that are key to the story
- Next fill in animation in between
Keyframe animation

- Keyframing is the technique used for pose-to-pose animation
  - Head animator draws key poses—just enough to indicate what the motion is supposed to be
  - Assistants do “in-betweening” and draws the rest of the frames
  - In computer animation substitute “user” and “animation software”
  - *Interpolation* is the principal operation
Keyframe animation

[Bryce Tutorial http://www.cadtutor.net/dd/bryce/anim/anim.html]
Walk cycle
Controlling geometry conveniently

• Could animate by moving every control point at every keyframe
  – This would be labor intensive
  – It would also be hard to get smooth, consistent motion
• Better way: animate using smaller set of meaningful degrees of freedom (DOFs)
  – Modeling DOFs are inappropriate for animation
    • E.g. “move one square inch of left forearm”
  – Animation DOFs need to be higher level
    • E.g. “bend the elbow”
Character with DOFs

A visual description of the possible movements for the squirrel
Rigged character

- Surface is deformed by a set of bones
- Bones are in turn controlled by a smaller set of controls
- The controls are useful, intuitive DOFs for an animator to use
The artistic process of animation

• What are animators trying to do?
  – Important to understand in thinking about what tools they need

• Basic principles are universal across media
  – 2D hand-drawn animation
  – 2D computer animation
  – 3D computer animation

• (The following slides follow the examples from Michael Comet’s very nice discussion on the page: http://www.comet-cartoons.com/toons/3ddocs/charanim/ )
Animation principles: timing

- Speed of an action is crucial to the impression it makes
  - examples with same keyframes, different times:

60 fr: looking around  30 fr: “no”  5 fr: just been hit
Animation principles: ease in/out

- Real objects do not start and stop suddenly
  - animation parameters shouldn’t either

  ![straight linear interp.](image1) ![ease in/out](image2)

  - a little goes a long way (just a few frames acceleration or deceleration for “snappy” motions)
Animation principles: moving in arcs

- Real objects also don’t move in straight lines
  - generally curves are more graceful and realistic
Animation principles: anticipation

- Most actions are preceded by some kind of “wind-up”
Animation principles: exaggeration

- Animation is not about exactly modeling reality
- Exaggeration is very often used for emphasis
Animation principles: squash & stretch

- Objects do not remain perfectly rigid as they move.
- Adding stretch with motion and squash with impact:
  - models deformation of soft objects
  - indicates motion by simulating exaggerated “motion blur”
Animation principles: follow through

- We’ve seen that objects don’t start suddenly
- They also don’t stop on a dime
Anim. principles: overlapping action

- Usually many actions are happening at once
Animation principles: staging

- Want to produce clear, good-looking 2D images
  - need good camera angles, set design, and character positions
Principles at work: weight
Extended example: Luxo, Jr.
Computer-generated motion

• Interesting aside: many principles of character animation follow indirectly from physics
• Anticipation, follow-through, and many other effects can be produced by simply minimizing physical energy
• Seminal paper: “Spacetime Constraints” by Witkin and Kass in SIGGRAPH 1988
Controlling shape for animation

• Start with *modeling DOFs* (control points)
• *Deformations* control those DOFs at a higher level
  – Example: move first joint of second finger on left hand
• *Animation controls* control *those* DOFs at a higher level
  – Example: open/close left hand
• Both cases can be handled by the same kinds of deformers
Rigid motion: the simplest deformation

• Move a set of points by applying an affine transformation

• How to animate the transformation over time?
  – Interpolate the matrix entries from keyframe to keyframe?
    • This is fine for translations but bad for rotations
Parameterizing rotations

- **Euler angles**
  - Rotate around \( x \), then \( y \), then \( z \)
  - Problem: gimbal lock
    - If two axes coincide, you lose one DOF

- **Unit quaternions**
  - A 4D representation (like 3D unit vectors for 2D sphere)
  - Good choice for interpolating rotations

- **These are first examples of motion control**
  - Matrix = deformation
  - Angles/quaternion = animation controls
Hierarchies and articulated figures

• Robot assignment as an example
  – Small number of animation controls control many transformations
  – Constraint: the joints hold together

• Robotics as source of math. Methods
  – Forward kinematics
  – Inverse kinematics
Articulation in robotics

- rectangular or cartesian
- cylindrical or post-type
- spherical or polar
- joint-arm or articulated
- SCARA (selective compliance assembly robot arm)
Motion capture

• A method for creating complex motion quickly: measure it from the real world

[thanks to Zoran Popović for many visuals]
Motion capture in movies
Motion capture in movies
Motion capture in games
Magnetic motion capture

- Tethered
- Nearby metal objects cause distortions
- Low freq. (60Hz)
Mechanical motion capture

- Measures joint angles directly
- Works in any environment
- Restricts motion
Optical motion capture

• Passive markers on subject

Retroreflective markers

Cameras with IR illuminators

• Markers observed by cameras
  – Positions via triangulation
Optical motion capture

- 8 or more cameras
- Restricted volume
- High frequency (240Hz)
- Occlusions are troublesome
From marker data to usable motion

- Motion capture system gives inconvenient raw data
  - Optical is “least information” case: accurate position but:
    - Which marker is which?
    - Where are the markers relative to the skeleton?
Motion capture data processing

• Marker identification: which marker is which
  – Start with standard rest pose
  – Track forward through time (but watch for markers dropping out due to occlusion!)

• Calibration: match skeleton, find offsets to markers
  – Use a short sequence that exercises all DOFs of the subject
  – A nonlinear minimization problem

• Computing joint angles: explain data using skeleton DOFs
  – A inverse kinematics problem per frame!
Motion capture in context

- Mocap data is very realistic
  - Timing matches performance exactly
  - Dimensions are exact
- But it is not enough for good character animation
  - Too few DOFs
  - Noise, errors from nonrigid marker mounting
  - Contains no exaggeration
  - Only applies to human-shaped characters
- Therefore mocap data is generally a starting point for skilled animators to create the final product
Basic surface deformation methods

• Mesh skinning: deform a mesh based on an underlying skeleton
• Blend shapes: make a mesh by combining several meshes
• Both use simple linear algebra
  – Easy to implement—first thing to try
  – Fast to run—used in games
• The simplest tools in the offline animation toolbox
Mesh skinning

- A simple way to deform a surface to follow a skeleton
Mesh skinning math: setup

• Surface has control points $p_i$
  – Triangle vertices, spline control points, subdiv base vertices

• Each bone has a transformation matrix $M_j$
  – Normally a rigid motion

• Every point–bone pair has a weight $w_{ij}$
  – In practice only nonzero for small # of nearby bones
  – The weights are provided by the user
Mesh skinning math

- Deformed position of a point is a weighted sum
  - of the positions determined by each bone’s transform alone
  - weighted by that vertex’s weight for that bone

\[ p'_i = \sum_j w_{ij} M_j p_i \]
Mesh skinning

- Simple and fast to compute
  - Can even compute in the vertex stage of a graphics pipeline
- Used heavily in games
- One piece of the toolbox for offline animation
  - Many other deformers also available
Mesh skinning: classic problems

- Surface collapses on the inside of bends and in the presence of strong twists
  - Average of two rotations is not a rotation!
  - Add more bones to keep adjacent bones from being too different, or change the blending rules.

[Lewis et al. SIGGRAPH 2000]
Blend shapes

- Another very simple surface control scheme
- Based on interpolating among several key poses
  - Aka. blend shapes or morph targets
Blend shapes math

• Simple setup
  – User provides key shapes—that is, a position for every control point in every shape: \( p_{ij} \) for point \( i \), shape \( j \)
  – Per frame: user provides a weight \( w_j \) for each key shape
    • Must sum to 1.0

• Computation of deformed shape
  \[
  p'_i = \sum_j w_j p_{ij}
  \]

• Works well for relatively small motions
  – Often used for facial animation
  – Runs in real time; popular for games