Animation and Interaction

CS475 / 675, Fall 2016

Siddhartha Chaudhuri
Video: Luxo Jr. (Pixar, 1986)
Motion from Sequence of Still Frames

Eadweard Muybridge, 1887
Basics of Animation: Keyframes

- **Keyframe**: significant pose of an animated object
- Artist/computer interpolates between successive keyframes to generate intermediate poses
- Splines can provide smooth transitions!
Video Compression with Keyframes

- Store a complete new image (keyframe) only when the scene substantially changes.
- For intermediate frames, store only the difference from the previous frame.
- Difference images are mostly zero, so compress better.
Kinematics

- Study of motion of objects without considering what causes that motion (i.e. forces)

  - *Forward kinematics:*
    - Describe transformation of object as function of time
    - Advance in small steps to generate frames

  - *Inverse kinematics:*
    - Describe motion of some parts of the object
    - Automatically infer motion of remaining parts, respecting object constraints
      - Such constraints might look like: keep some points fixed, and move the parts as little as possible
Inverse Kinematics

Constraints

These motions are automatically inferred

“UTPoser”, Yamane and Nakamura, 2003
Video: Combining Path Planning with IK

Synthesizing Animations of Human Manipulation (Yamane, Kuffner and Hodgins, 2004)
Physically-Based Animation

- **Dynamics**: Study of motion as a product of the stimuli/forces that cause it
- Simulate the evolution of a system as forces are applied to it
- Typically:
  - Model the system as a collection of differential equations
  - Set boundary conditions (constraints)
  - Numerically solve the equations for trajectories
Video: Solid-Fluid Coupling

Coupling Water and Smoke to Thin Deformable and Rigid Shells (Guendelman, Selle, Losasso and Fedkiw, 2005)
Video: Multiple Interacting Fluids

Multiple Interacting Liquids (Losasso, Shinar, Selle and Fedkiw, 2006)
Smooth Animation: Motion Blur

- A frame is not an instant but an *interval* in time

- For smooth transitions from the frame to the next, the frame should *integrate* all positions of the object during the interval

- This is naturally captured by longer exposures on film, and can be simulated on a computer

(Eadweard Muybridge)
Smooth Animation: Motion Blur

“1984”, using distribution ray tracing (Cook, Porter and Carpenter, SIGGRAPH 1984)
Smooth Animation: Double-Buffering

- **Problem**: Screen flicker and tearing, when
  - display is cleared between frames, or
  - display refresh is not synced to redraw rate

- **Solution**: Maintain two framebuffers, one for the display and one for drawing
  - Display device reads display buffer, renderer draws on drawing buffer
  - Once the frame is drawn and the next display refresh is due, the buffers are quickly swapped
    - Just the pointers are swapped, no memory is copied

- **Demo**: Single- vs double-buffering
Video: Ivan Sutherland's Sketchpad, 1962

(Presented by Alan Kay, 1987)
Interaction

- Detecting input:
  - **Polling**: I periodically query the system
  - **Interrupts**: OS notifies me asynchronously

- **Picking**: Finding the nearest object (e.g. window, button, jetpack...) at the pointer location
  - Picking = raycasting!
  - Can do in object space, or via the framebuffer

- Input events trigger responses/actions
  - e.g. a widget's `onClick()` callback is called when the widget is clicked
Events

- Operating system responds to interrupt
- Creates timestamped event record and places in an event queue
  - Queues are first-in first-out (FIFO)
  - May store state of other devices at time of event
    - e.g. polled devices like the mouse
- Callback manager
  - Reorder events based on priority
    - e.g. keyboard has priority over PostRedisplay
- Event coalescing
- Combine mouse motions for current position
Simple Game Control

forever:

readInputDevices()
updateAI()
processMovement()
handleCollisions()
draw()
Animation Graphs

Crouch

Crawl

Slide

Stand

Walk

Run

Sprint

Skid
Filtering/Smoothing Input

- We don't really want characters to respond instantaneously, so we take a little time to converge to the desired state

(“Classic Super Mario 64 Third-Person Control and Animation”, Steve Rabin)
Interaction: Things to Remember

- Flow from input to events to method invocation
  - Inputs cause interrupts which generate events
  - Events are routed to objects
  - Pointer events require “hit testing”
- Interaction “building blocks” for GUls, games
  - State machines
  - Filtering/smoothing
  - Clicking, dragging
  - Character movement