Lecture 17 of 41

Animation 1 of 3: Basics, Keyframing
Sample Exam Review

William H. Hsu
Department of Computing and Information Sciences, KSU

KSOL course pages: http://bit.ly/hGvXlH / http://bit.ly/eVizrE
Public mirror web site: http://www.kddresearch.org/Courses/CIS636
Instructor home page: http://www.cis.ksu.edu/~bhsu

Readings:
Today: §5.1 – 5.2, Eberly *2e* – see http://bit.ly/ieUq45
Next class: no new reading – review Chapters 1 – 4, 20
Optional review session during next class period; evening exam time TBD
Lecture 18 reading (two class days from today): §4.4 – 4.7, Eberly *2e*

Lecture Outline

- Reading for Last Class: §2.6, 20.1, Eberly *2e*; OpenGL primer material
- Reading for Today: §5.1 – 5.2, Eberly *2e*
- Reading for Next Lecture (Two Classes from Now): §4.4 – 4.7, Eberly *2e*
- Last Time: Shading and Transparency in OpenGL
  - Transparency revisited
  - OpenGL how-to: http://bit.ly/hRaQgk
    - Alpha blending (15.020, 15.040)
    - Screen-door transparency (15.030)
  - Painter's algorithm & depth buffering (z-buffering)
- Today: Introduction to Animation
  - What is it and how does it work?
  - Brief history
  - Principles of traditional animation
  - Keyframe animation
  - Articulated figures: inbetweening

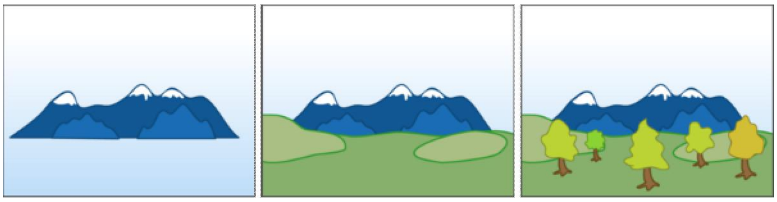
Where We Are

Lecture	Topic	Primary Source(s)
0	Course Overview	Chapter 1, Eberly 2^e
1	**CG Basics: Transformation Matrices; Lab 0**	**Sections (§) 2.1, 2.2**
2	Viewing 1: Overview, Projections	§ 2.2.3 – 2.2.4, 2.8
3	Viewing 2: Viewing Transformation	§ 2.3 esp. 2.3.4; FVFH slides
4	**Lab 1a: Flash & OpenGL Basics**	**Ch. 2, 16[1], Angel *Primer***
5	Viewing 3: Graphics Pipeline	§ 2.3 esp. 2.3.7; 2.6, 2.7
6	Scan Conversion 1: Lines, Midpoint Algorithm	§ 2.5.1, 3.1; FVFH slides
7	**Viewing 4: Clipping & Culling; Lab 1b**	**§ 2.3.5, 2.4, 3.1.3**
8	Scan Conversion 2: Polygons, Clipping Intro	§ 2.4, 2.5 esp. 2.5.4, 3.1.6
9	Surface Detail 1: Illumination & Shading	§ 2.5, 2.6.1 – 2.6.2, 4.3.2, 20.2
10	**Lab 2a: Direct3D / DirectX Intro**	**§ 2.7, Direct3D handout**
11	Surface Detail 2: Textures; OpenGL Shading	§ 2.6.3, 20.3 – 20.4, *Primer*
12	Surface Detail 3: Mappings; OpenGL Textures	§ 20.5 – 20.13
13	**Surface Detail 4: Pixel/Vertex Shad.; Lab 2b**	**§ 3.1**
14	Surface Detail 5: Direct3D Shading; OGLSL	§ 3.2 – 3.4, Direct3D handout
15	Demos 1: CGA, Fun; Scene Graphs: State	§ 4.1 – 4.3, **CGA handout**
16	**Lab 3a: Shading & Transparency**	**§ 2.6, 20.1, *Primer***
17	**Animation 1: Basics, Keyframes; HW/Exam**	**§ 5.1 – 5.2**
	Exam 1 review; Hour Exam 1 (evening)	**Chapters 1 – 4, 20**
18	**Scene Graphs: Rendering; Lab 3b: Shader**	**§ 4.4 – 4.7**
19	**Demos 2: SFX; Skinning, Morphing**	**§ 5.3 – 5.5, CGA handout**
20	Demos 3: Surfaces; B-reps/Volume Graphics	§ 10.4, 12.7, **Mesh handout**

Lightly-shaded entries denote the due date of a written problem set; heavily-shaded entries, that of a machine problem (programming assignment); blue-shaded entries, that of a paper review; and the green-shaded entry, that of the term project.

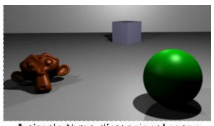
Green, **blue** and **red** letters denote **exam review**, **exam**, and **exam solution review** dates.

Review: Painter's Algorithm vs. z-Buffering

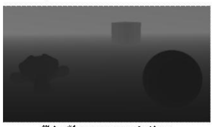


© 2004 – 2009 Wikipedia, *Painter's Algorithm*
http://bit.ly/eeebCN

© 2009 Wikipedia, *Z-buffering*
http://bit.ly/gGRFMA



A simple three-dimensional scene



Z-buffer representation

5

Transparency in OpenGL: Final Note

15.060 I want to use blending but can't get destination alpha to work. Can I blend or create a transparency effect without destination alpha?

Many OpenGL devices don't support destination alpha. In particular, the OpenGL 1.1 software rendering libraries from Microsoft don't support it. The OpenGL specification doesn't require it.

If you have a system that supports destination alpha, using it is a simple matter of asking for it when you create your window. For example, pass GLUT_ALPHA to glutInitDisplayMode(), then set up a blending function that uses destination alpha, such as:

```
glBlendFunc(GL_ONE_MINUS_DST_ALPHA,GL_DST_ALPHA);
```

Often this question is asked under the mistaken assumption that destination alpha is required to do blending. It's not. You can use blending in many ways to obtain a transparency effect that uses source alpha instead of destination alpha. The fact that you might be on a platform without destination alpha shouldn't prevent you from obtaining a transparency effect. See the OpenGL Programming Guide chapter 6 for ways to use blending to achieve transparency.

http://bit.ly/hRaQgk

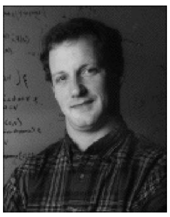
6

Acknowledgements: Computer Animation Intro

Jason Lawrence
Assistant Professor
Department of Computer Science
University of Virginia
http://www.cs.virginia.edu/~jdl/

Computer Science at the University of Virginia

Acknowledgment: slides by Misha Kazhdan, Allison Klein, Tom Funkhouser, Adam Finkelstein and David Dobkin
http://bit.ly/eB1Oj4



Thomas A. Funkhouser
Professor
Department of Computer Science
Computer Graphics Group
Princeton University
http://www.cs.princeton.edu/~funk/

Princeton University

Overview

- Some early animation history
  - http://web.inter.nl.net/users/anima/index.htm
  - http://www.public.iastate.edu/~rllew/chrnearl.html
- Computer animation

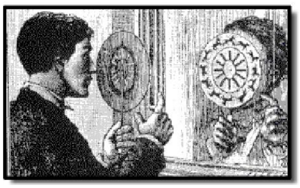
Thaumatrope

- Why does animation work?
- Persistence of vision
- 1824 John Ayerton invents the *thaumatrope*
- Or, 1828 Paul Roget invents the *thaumatrope*

Thaumatrope of flowers & vase (1825)
© 2008 Wikipedia, *Thaumatrope*
http://bit.ly/fFI6xH

Phenakistoscope

- Invented independently by 2 people in 1832
- Disc mounted on spindle
- Viewed through slots with images facing mirror
- Turning disc animates images






Phenakistoscope of couple (1893)
© 2007 Wikipedia, *Phenakistoscope*
http://bit.ly/eAnURG

Adapted from slides © 2010 J. Lawrence, University of Virginia
CS 4810: Introduction to Computer Graphics – http://bit.ly/hPIXdi

Zoetrope (1834)

- Images arranged on paper band inside a drum
- Slits cut in the upper half of the drum
- Opposite side viewed as drum rapidly spun
- Praxinoscope is a variation on this




***Tarzan* © 2000 Disney**
http://youtu.be/zc3MnoSS5Hw

Adapted from slides © 2010 J. Lawrence, University of Virginia
CS 4810: Introduction to Computer Graphics – http://bit.ly/hPIXdi

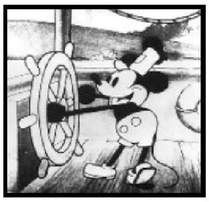
Animation History

- Animation and technology have always gone together!
- Animation popular even before movies
- Movies were big step forward!
- “Humorous Phases of Funny Faces” (1906)

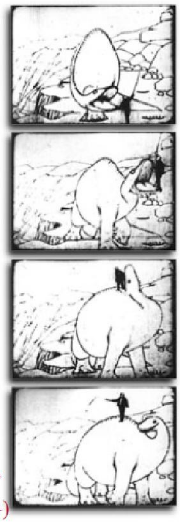
Key Developments [1]: Storytelling & Cel Animation

- Plot
- Creation of animation studios
- Getting rid of “rubber-hose” bodies
- Inking on cels

“Felix the Cat”
Pat Sullivan (1919)



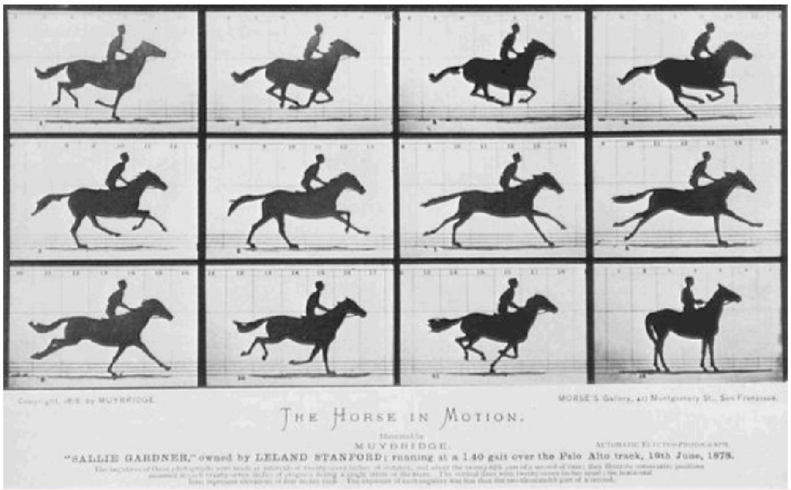
“Steamboat Willie”
Walt Disney (1928)



“Gertie the Dinosaur”
Windsor McCay (1914)

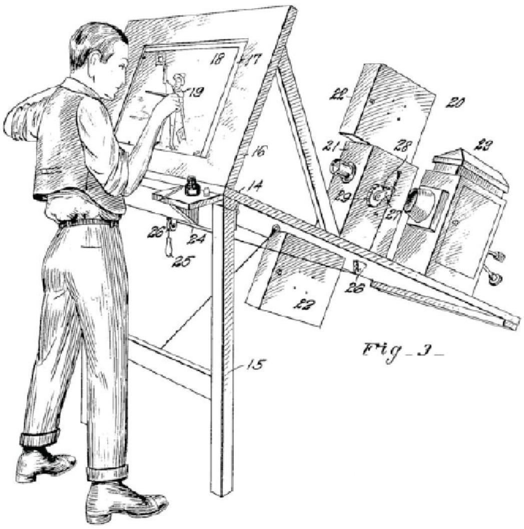
Key Developments [2]: Rotoscoping (1921)

- Max Fleischer invents rotoscoping (1921)



CS 4810: Introduction to Computer Graphics – http://bit.ly/hPIXdi

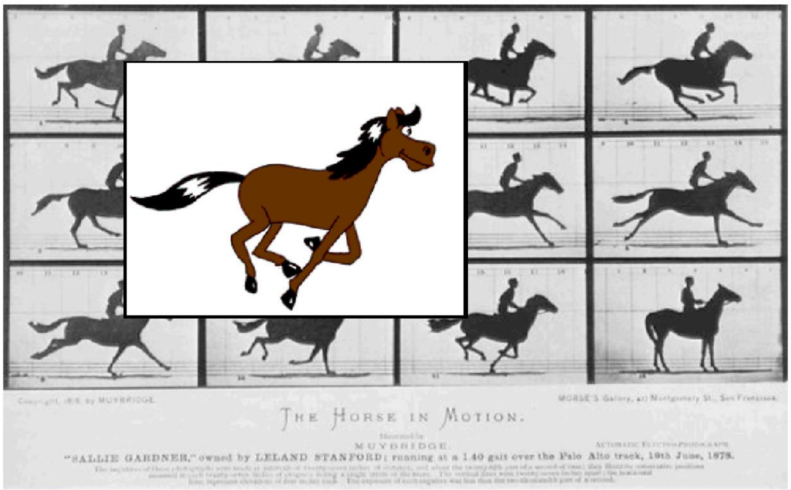
Key Developments [4] Fleischer's Rotoscope



CS 4810: Introduction to Computer Graphics – http://bit.ly/hPIXdi

Key Developments [5]: Using Rotoscoping

- Max Fleischer invents rotoscoping (1921)

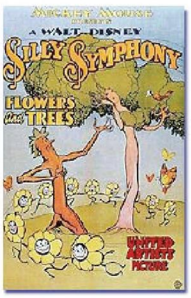


Adapted from slides © 2010 J. Lawrence, University of Virginia
CS 4810: Introduction to Computer Graphics – http://bit.ly/hPIXdi

Key Developments [6]: Color

- "Flowers and Trees" (1932) uses color!
- "Snow White" (aka "Disney's Folly") released 1937

"Flowers and Trees"
Walt Disney



"Snow White"
Walt Disney

© 2010 J. Lawrence, University of Virginia
CS 4810: Introduction to Computer Graphics – http://bit.ly/hPIXdi

17

Overview

- Some early animation history
  - o http://web.inter.nl.net/users/anima/index.htm
  - o http://www.public.iastate.edu/~rllew/chrnearl.html
- Computer animation

CS 4810: Introduction to Computer Graphics – http://bit.ly/hPIXdi

18

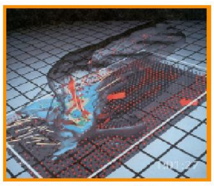
Animation, Simulation, & Visualization

- What is animation?
  - o Make objects change over time according to scripted actions

Pixar

- What is simulation?
  - o Predict how objects change over time according to physical laws

Wilhelmson *et al.* (2004)
http://youtu.be/EgumU0Ns1YI
http://avl.ncsa.illinois.edu
http://bit.ly/eA8PXN

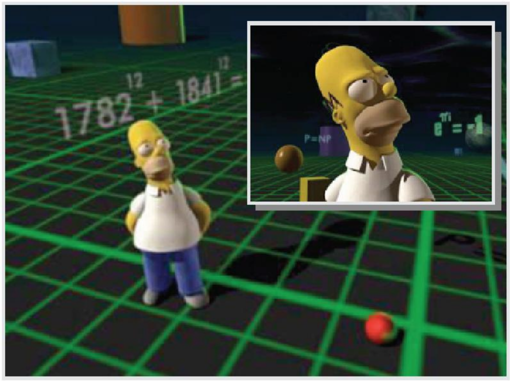


University of Illinois

CS 4810: Introduction to Computer Graphics – http://bit.ly/hPIXdi

2-D & 3-D Animation

Homer 2-D

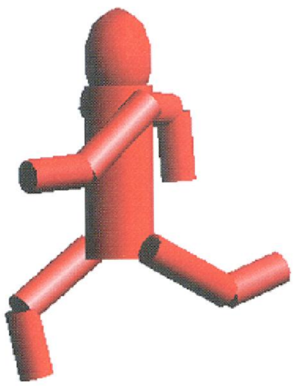


Homer 3-D
http://youtu.be/TKQ8Ilr6PgU (Making Of)

CS 4810: Introduction to Computer Graphics – http://bit.ly/hPIXdi

Outline

- Principles of animation
- Keyframe animation
- Articulated figures



Angel Plate 1

CS 4810: Introduction to Computer Graphics – http://bit.ly/hPIXdi

Traditional Animation [1]: Lasseter's List of Principles (1987)

- Squash and Stretch
- Timing
- Anticipation
- Staging
- Follow Through and Overlapping Action
- Straight Ahead Action and Pose-to-Pose Action
- Slow In and Out
- Arcs
- Exaggeration
- Secondary action
- Appeal

Computer Graphics, Volume 21, Number 4, July 1987

PRINCIPLES OF TRADITIONAL ANIMATION
APPLIED TO 3D COMPUTER ANIMATION

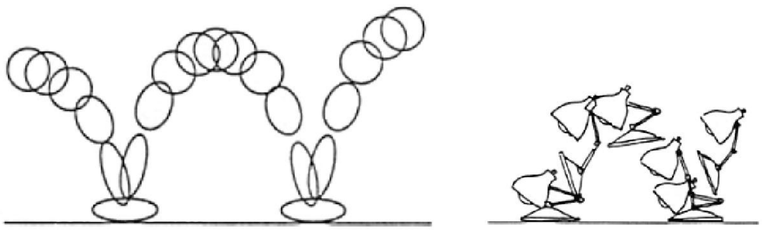
John Lasseter
Pixar
San Rafael
California

Lasseter, J. (1987). Principles of traditional animation applied to 3D computer animation. *Computer Graphics*, 21(4), pp. 35-44.
SIGGRAPH: http://bit.ly/1DsO44
ACM Portal: http://bit.ly/eyx2PN

CS 4810: Introduction to Computer Graphics – http://bit.ly/hPIXdi

Traditional Animation [2]: Squash & Stretch

- Defining the rigidity and mass of an object by distorting its shape during an action.



CS 4810: Introduction to Computer Graphics – http://bit.ly/hPIXdi

Traditional Animation [3]: Timing

- Spacing actions to define the weight and size of objects and the personality of characters.
  - Heavier objects accelerate slower
  - Lethargic characters move slower
  - Etc.

Traditional Animation [4]: Anticipation

- The preparation for an action.
  - Muscle contraction prior to extension
  - Bending over to lift a heavy object
  - Luxo's dad responds to Luxo Jr. off screen before Luxo Jr. appears.

Luxo Jr. © 1986 Pixar
http://www.pixar.com/shorts/ljr/
http://youtu.be/qGxoui3IFS0

Traditional Animation [5]: Staging

- Presenting an idea so that it is unmistakably clear.
  - Keeping the viewer's attention focused on a specific part of the scene.
  - Luxo Jr. moves faster than his dad, and so we focus on him.

Luxo Jr. © 1986 Pixar
http://www.pixar.com/shorts/ljr/
http://youtu.be/qGxoui3IFS0

Traditional Animation [6]: Follow Through & Overlapping Action

- The termination of an action and establishing its relationship to the next action.
  - Loose clothing will "drag" and continue moving after the character has stopped moving.
  - The way in which an object slows down indicates its weight/mood.

Traditional Animation [7]: Straight-Ahead vs. Pose-to-Pose Action

- The two contrasting approaches to the creation of movement.
  - Straight Ahead Action:
    - Action is drawn from the first frame through to the last one.
    - Wild, scrambling actions where spontaneity is important.
  - Pose-to-Pose Action:
    - Poses are pre-conceived and animator fills in the in-betweens.
    - Good acting, where the poses and timing are all important.

Traditional Animation [8]: Slow In-And-Out

- The spacing of in-between frames to achieve subtlety of timing and movements.

Traditional Animation [9]: Arcs

- The visual path of action for natural movement.
  - Make animation much smoother and less stiff than a straight line for the path of action

Traditional Animation [10]: Exaggeration

- Accentuating the essence of an idea via the design and the action.

Traditional Animation [11]: Secondary Action

- The Action of an object resulting from another action.
  - The rippling of Luxo Jr.'s cord as he bounces around the scene.

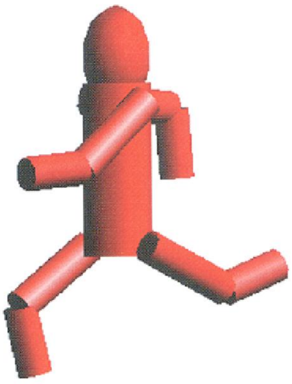
Luxo Jr. © 1986 Pixar
http://www.pixar.com/shorts/ljr/
http://youtu.be/qGxoui3lFS0

Traditional Animation [12]: Appeal

- Creating a design or an action that the audience enjoys watching.
  - Charm
  - Pleasing design
  - Simplicity
  - Communication
  - Magnetism
  - Etc.

Outline

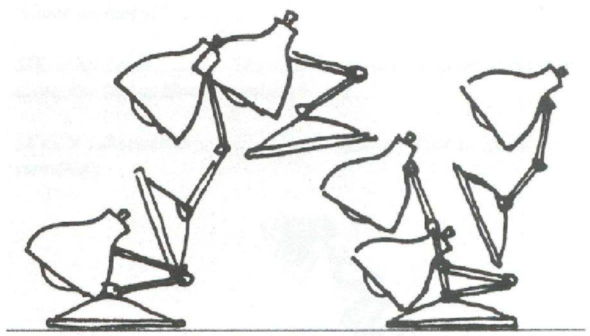
- Principles of animation
- Keyframe animation
- Articulated figures



Angel Plate 1

Keyframe Animation [1]: Keyframes

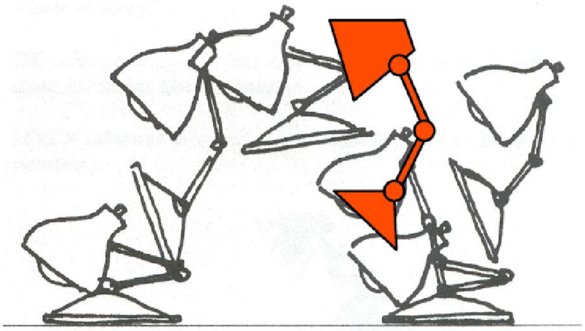
- Define character poses at specific time steps called "keyframes"



Lasseter `87

Keyframe Animation [2]: Interpolation (*aka* Inbetweening)

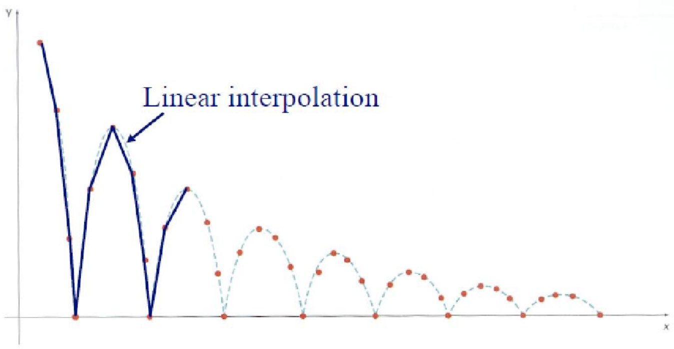
- Interpolate variables describing keyframes to determine poses for character "in-between"



Lasseter `87

Keyframe Animation [3]: Linear Interpolation *aka* Lerping

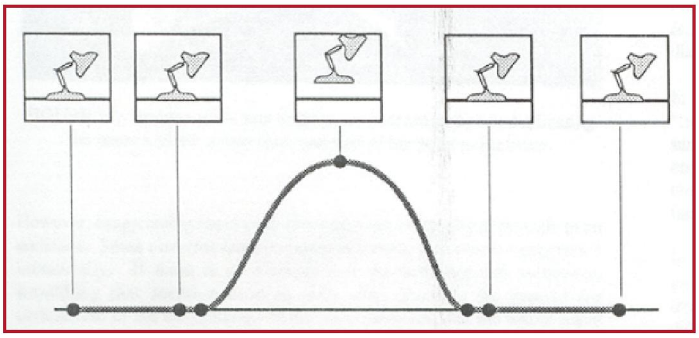
- Inbetweening:
  - Linear interpolation - usually not enough continuity



H&B Figure 16.16

Keyframe Animation [4]: Cubic Curve (Spline) Interpolation

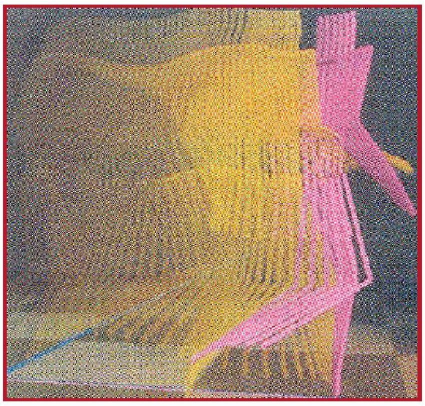
- Inbetweening:
  - Cubic spline interpolation - maybe good enough
    - May not follow physical laws



Lasseter `87

Keyframe Animation [5]: Dynamics & Kinematics

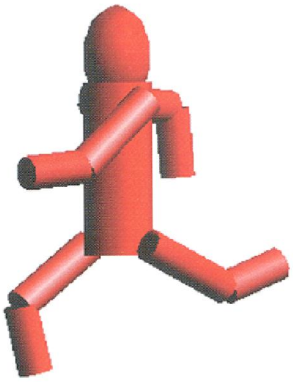
- Inbetweening:
  - Kinematics or dynamics



Rose et al. `96

Outline

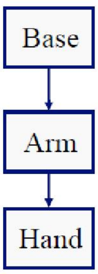
- Principles of animation
- Keyframe animation
- Articulated figures



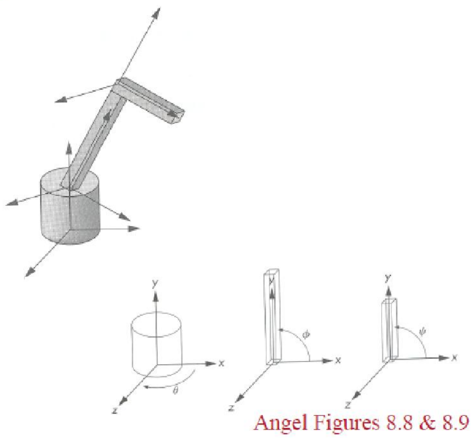
Angel Plate 1

Articulated Figures [1]: Definition

- Character poses described by set of rigid bodies connected by "joints"



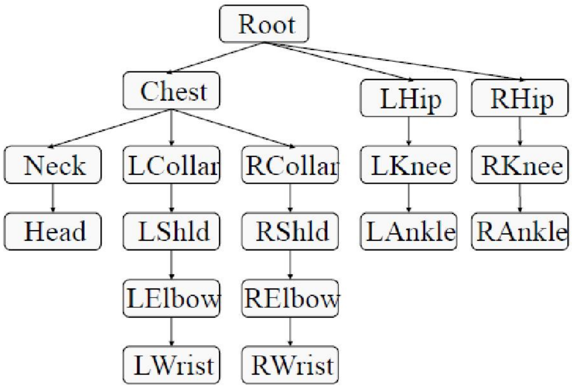
Scene Graph

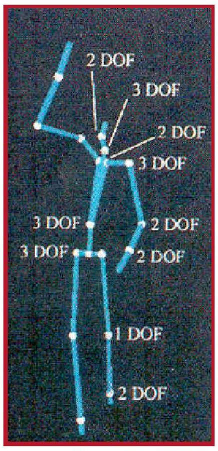


Angel Figures 8.8 & 8.9

Articulated Figures [2]: Character Modeling

- Well-suited for humanoid characters

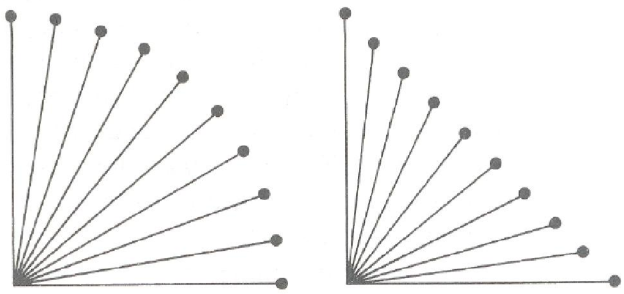




Rose et al. `96

Articulated Figures [3]: Angular Interpolation

- Inbetweening
  - o Interpolate angles, not positions, between keyframes

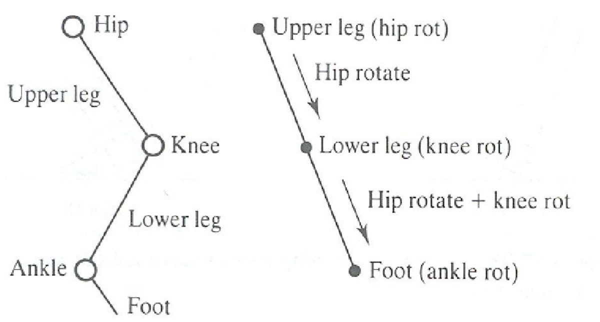


Good arm　　　Bad arm

Watt & Watt

Articulated Figures [4]: Bones & Joints

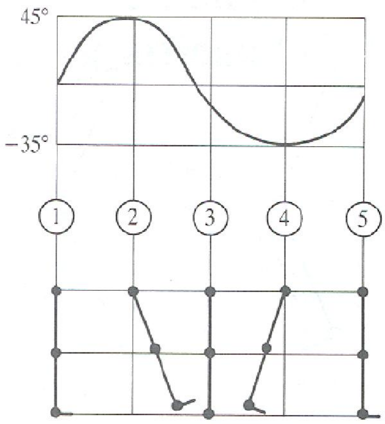
- Articulated figure:



Watt & Watt

Articulated Figures [5]: Example – Walk Cycle 1

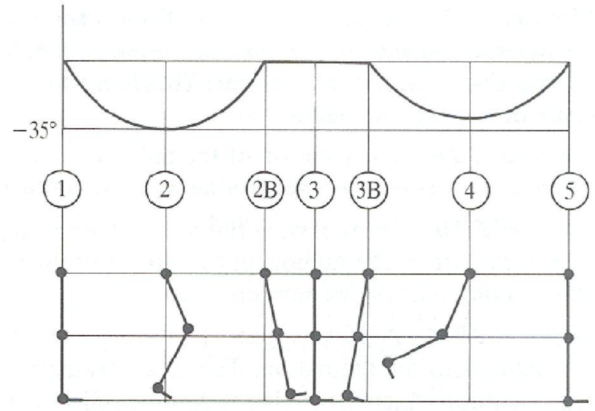
- Hip joint orientation:



Watt & Watt

Articulated Figures [6]: Example – Walk Cycle 2

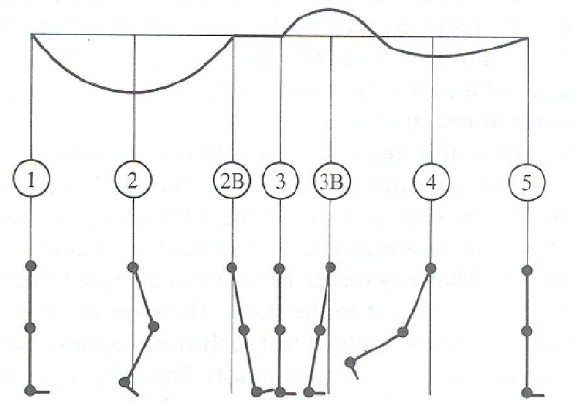
- Knee joint orientation:



Watt & Watt

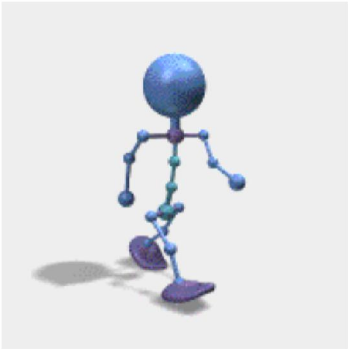
Articulated Figures [7]: Example – Walk Cycle 3

- Ankle joint orientation:



Watt & Watt

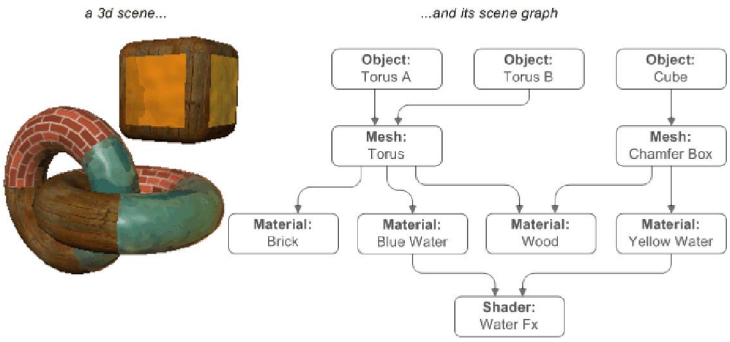
Articulated Figures [7]: Example – Walk Cycle 4



© 2002 D. M. Murillo
http://bit.ly/eZ9MA8

© 2010 J. Lawrence, University of Virginia
CS 4810: Introduction to Computer Graphics – http://bit.ly/hPIXdi

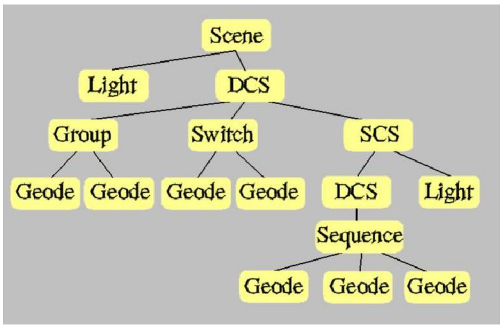
Looking Ahead: Scene Graph Traversal



© 2002 – 2005 Virtools
http://bit.ly/eM1gz8

49

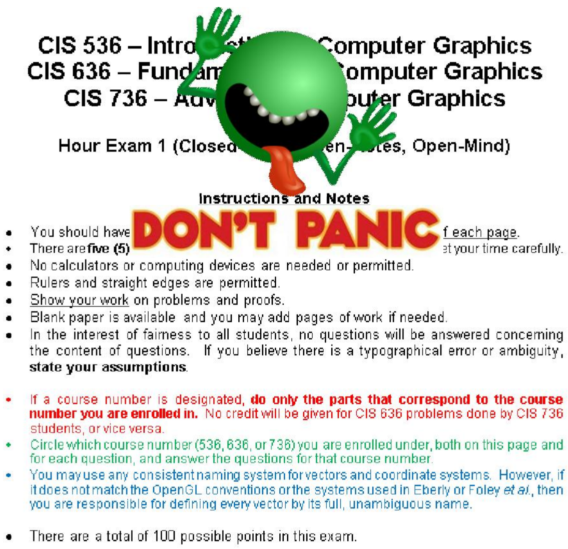
Looking Ahead: Scene Graph Rendering



Performer © 1997 D. Pape
http://www.evl.uic.edu/pape/talks/VSI97/pf/

50

Problem Set 3: Hour Exam 1Review



Summary

- Reading for Last Class: §2.6, 20.1, Eberly *2^e*; OpenGL primer material
- Reading for Today: §5.1 – 5.2, Eberly *2^e*
- Reading for Next Lecture (Two Classes from Now): §4.4 – 4.7, Eberly *2^e*
- Last Time: Shading and Transparency in OpenGL
  - Alpha blending
  - Painter's algorithm – less efficient, can handle non-opaque objects
  - Depth buffering (*z*-buffering) – in hardware, fast, opaque only
- Today: Introduction to Animation
  - What is it and how does it work?
  - Brief history
  - Principles of traditional animation
  - Keyframe animation
  - Articulated figures: inbetweening

Terminology

- Shading and Transparency in OpenGL: Alpha, Painter's, *z*-buffering
- Animation – Bringing Still Objects "to Life" (Change Over Time)
- Early Animation
  - Thaumatrope (*c.* 1824) – early Victorian toy prefiguring flipbooks
  - Flipbook – simple paper-based animation technique
- Action in Traditional Animation
  - Before: squash & stretch, timing, anticipation, staging
  - During: exaggeration, secondary
  - After: follow-through & overlapping action
  - Design: straight-ahead *vs.* pose-to-pose
- Keyframe Animation
  - Inbetweening – interpolation technique
    - Lerping – linear interpolation
    - Splines & other cubic curves
  - Articulated figures: angular interpolation