Lecture 21 of 41

Animation Basics
Lab 4: Modeling & Rigging in Maya

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: **Flash animation handout**
Next class: Chapter 17, esp. §17.1 – 17.2, Eberly 2e – see **http://bit.ly/ieUq45**
Reference: **http://www.learning-maya.com**

Lecture Outline

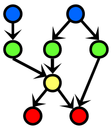
- **Reading for Last Class: §10.4, 12.7, Eberly *2*^{e}, Mesh handout**
- **Reading for Today: §11.1 – 11.6 Eberly *2*^{e} (736), Flash handout**
- **Reading for Next Class: §17.1 – 17.2, Eberly *2*^{e}**
- **Last Time: Curves & Surfaces**
  - **Piecewise polynomial curves (*aka* splines) and their properties**
  - **Hermite *vs.* Bézier curves: manipulation *vs.* display (rendering)**
  - **DeCasteljau's algorithm: recursive linear interpolation**
  - **Other representations: Bernstein basis functions, matrix form**
  - **Bicubic surfaces**
  - **Bilinear interpolation**
- **Today: *Maya* & Animation Preliminaries – Ross Tutorials**
  - ***Maya* interface: navigation, menus, tools, primitives**
  - **Ross tutorials (http://bit.ly/dFpTwq)**
  - **Preview of character models: PolyFacecom (http://bit.ly/h6tzrd)**
- **Next Class: Animations 2 – Rotations, Dynamics & Kinematics**

Where We Are

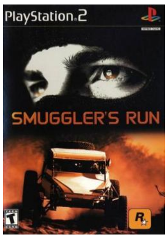
21	**Lab 4a: Animation Basics**	**Flash animation handout**
22	Animation 2: Rotations; Dynamics, Kinematics	Chapter 17, esp. §17.1 – 17.2
23	Demos 4: Modeling & Simulation; Rotations	Chapter 10[1], 13[2], §17.3 – 17.5
24	**Collisions 1: axes, OBBs, Lab 4b**	**§2.4.3, 8.1, GL handout**
25	Spatial Sorting: Binary Space Partitioning	Chapter 6, esp. §6.1
26	**Demos 5: More CGA; Picking; HW/Exam**	**Chapter 7[2]; § 8.4**
27	**Lab 5a: Interaction Handling**	**§ 8.3 – 8.4; 4.2, 5.0, 5.6, 9.1**
28	Collisions 2: Dynamic, Particle Systems	§ 9.1, particle system handout
	Exam 2 review; Hour Exam 2 (evening)	**Chapters 5 – 6, 7[2] – 8, 12, 17**
29	**Lab 5b: Particle Systems**	**Particle system handout**
30	**Animation 3: Control & IK**	**§ 5.3, CGA handout**
31	Ray Tracing 1: intersections, ray trees	Chapter 14
32	Lab 6a: Ray Tracing Basics with POV-Ray	RT handout
33	Ray Tracing 2: advanced topic survey	Chapter 15, **RT handout**
34	Visualization 1: Data (Quantities & Evidence)	Tufte handout (1)
35	**Lab 6b: More Ray Tracing**	**RT handout**
36	Visualization 2: Objects	Tufte handout (2 & 4)
37	Color Basics; Term Project Prep	**Color handout**
38	Lab 7: Fractals & Terrain Generation	**Fractals/Terrain handout**
39	**Visualization 3: Processes; Final Review 1**	Tufte handout (3)
40	**Project presentations 1; Final Review 2**	**–**
41	Project presentations 2	–
	Final Exam	**Ch. 1 – 8, 10 – 15, 17, 20**

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.



Acknowledgements: Curves & Surfaces



Steve Rotenberg
Visiting Lecturer
Graphics Lab
University of California – San Diego
CEO/Chief Scientist, PixelActive
http://graphics.ucsd.edu

Barry McCaul
Lecturer
School of Computing
Dublin City University
http://www.computing.dcu.ie/~bmccaul/

Ken Hawick
Professor
Institute of Information and Mathematical Sciences (IIMS)
Massey University – Albany
http://www.massey.ac.nz/~kahawick/

Acknowledgements: Splines

Jim Foley
Professor, College of Computing & Stephen Fleming Chair in Telecommunications
Georgia Institute of Technology

James D. Foley
Georgia Tech
http://bit.ly/ajYf2Q

Andy van Dam
T. J. Watson University Professor of Technology and Education & Professor of Computer Science
Brown University

Andries van Dam
Brown University
http://www.cs.brown.edu/~avd/

Steve Feiner
Professor of Computer Science & Director, Computer Graphics and User Interfaces Laboratory
Columbia University

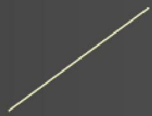
Steven K. Feiner
Columbia University
http://www.cs.columbia.edu/~feiner/

John F. Hughes
Associate Professor of Computer Science
Brown University

John F. Hughes
Brown University
http://www.cs.brown.edu/~jfh/

Review [1]: Vector Polynomials (Curves)

- Linear: $\mathbf{f}(t) = \mathbf{a}t + \mathbf{b}$



- Quadratic: $\mathbf{f}(t) = \mathbf{a}t^2 + \mathbf{b}t + \mathbf{c}$

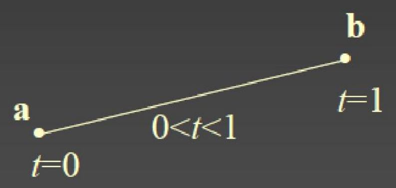
- Cubic: $\mathbf{f}(t) = \mathbf{a}t^3 + \mathbf{b}t^2 + \mathbf{c}t + \mathbf{d}$

We usually define the curve for $0 \leq t \leq 1$

Adapted from slides ♥ 2003 – 2006 S. Rotenberg, UCSD
CSE167: Computer Graphics, Fall 2006, http://bit.ly/hXxAIP

Review [2]: Linear Interpolation

- Linear interpolation (Lerp) is a common technique for generating a new value that is somewhere in between two other values
- A 'value' could be a number, vector, color, or even something more complex like an entire 3D object…
- Consider interpolating between two points **a** and **b** by some parameter *t*



$$Lerp(t, \mathbf{a}, \mathbf{b}) = (1-t)\mathbf{a} + t\mathbf{b}$$

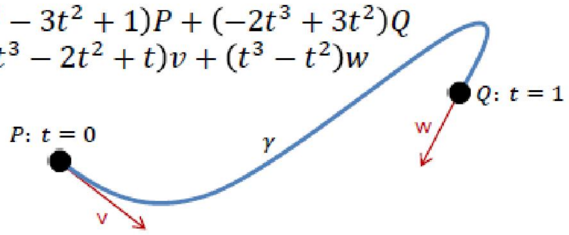
Adapted from slides ♥ 2003 – 2006 S. Rotenberg, UCSD
CSE167: Computer Graphics, Fall 2006, http://bit.ly/hXxAIP

Review [3]: Hermite Curves

- Polylines are linear (1st order polynomial) interpolations between points
  - Given points P and Q, line between the two is given by the parametric equation: $$x(t) = (1 - t)P + tQ, \qquad 0 \le t \le 1$$
  - $(1 - t)$ and t are called weighting functions of P and Q
- Splines are higher order polynomial interpolations between points
  - Like linear interpolation but with higher order weighting functions allowing better approximations/smoother curves
- One representation - Hermite curves (interpolating spline):
  - Determined by two control points P and Q, an initial tangent vector v and a final tangent vector w.

$$\gamma(t) = (2t^3 - 3t^2 + 1)P + (-2t^3 + 3t^2)Q + (t^3 - 2t^2 + t)v + (t^3 - t^2)w$$

  - Satisfies:
    - $\gamma(0) = P$
    - $\gamma(1) = Q$
    - $\gamma'(0) = v$
    - $\gamma'(1) = w$



Review [4]: Bézier Curves

- Bezier representation is similar to Hermite
  - 4 points instead of 2 points and 2 vectors ($P_1 \ldots P_4$)
  - Initial position P_1, tangent vector is $P_2 - P_1$
  - Final position P_4 tangent vector is $P_4 - P_3$
  - This representation allows a spline to be stored as a list of vertices with some global parameters that describe the smoothness and continuity

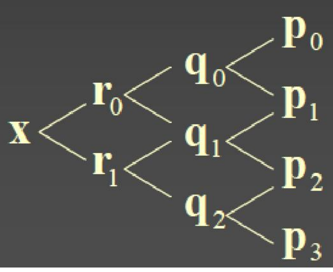
- Bezier splines are widely used (Adobe, Microsoft) for font definition

Brown Exploratory (Spalter & Bielawa): http://bit.ly/fva1il

Review [5]: De Casteljau's Algorithm

$$\mathbf{x} = Lerp(t, \mathbf{r}_0, \mathbf{r}_1)$$
$$\mathbf{r}_0 = Lerp(t, \mathbf{q}_0, \mathbf{q}_1) \qquad \mathbf{r}_1 = Lerp(t, \mathbf{q}_1, \mathbf{q}_2)$$
$$\mathbf{q}_0 = Lerp(t, \mathbf{p}_0, \mathbf{p}_1) \qquad \mathbf{q}_1 = Lerp(t, \mathbf{p}_1, \mathbf{p}_2) \qquad \mathbf{q}_2 = Lerp(t, \mathbf{p}_2, \mathbf{p}_3)$$

$\mathbf{p}_0$, $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$



Adapted from slides ♥ 2003 – 2006 S. Rotenberg, UCSD
CSE167: Computer Graphics, Fall 2006, http://bit.ly/hXxAIP

Review [6]: Bernstein Polynomials – Matrix Form

$$\mathbf{x} = \mathbf{a}t^3 + \mathbf{b}t^2 + \mathbf{c}t + \mathbf{d}$$

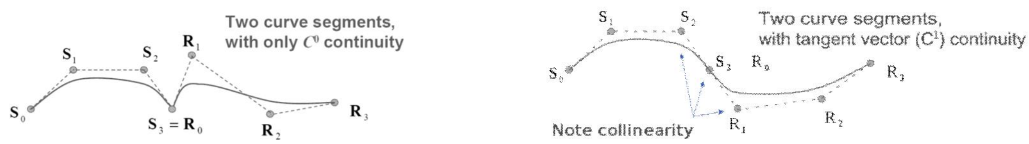
$$\mathbf{a} = \left(-\mathbf{p}_0 + 3\mathbf{p}_1 - 3\mathbf{p}_2 + \mathbf{p}_3\right)$$
$$\mathbf{b} = \left(3\mathbf{p}_0 - 6\mathbf{p}_1 + 3\mathbf{p}_2\right)$$
$$\mathbf{c} = \left(-3\mathbf{p}_0 + 3\mathbf{p}_1\right)$$
$$\mathbf{d} = \left(\mathbf{p}_0\right)$$

$$\mathbf{x} = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{bmatrix} \qquad \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{bmatrix} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix}$$

Adapted from slides ♥ 2003 – 2006 S. Rotenberg, UCSD
CSE167: Computer Graphics, Fall 2006, http://bit.ly/hXxAlP

Review [7]: G^i vs. C^i Continuity

- **Geometric Continuity**: G^i
  - Guarantees that direction of i^{th} derivative equal
  - G^0: curves touch at join point
  - G^1: curves also share common tangent direction at join point
  - G^2: curves also share common center of curvature at join point
- **Mathematical Continuity**: C^i
  - Guarantees that direction, magnitude of i^{th} derivative equal
  - $C^0 \equiv G^0$: curves touch at join point
  - C^1: curves share common tangent direction / magnitude at join point
  - C^2: curves share common second derivative at join point



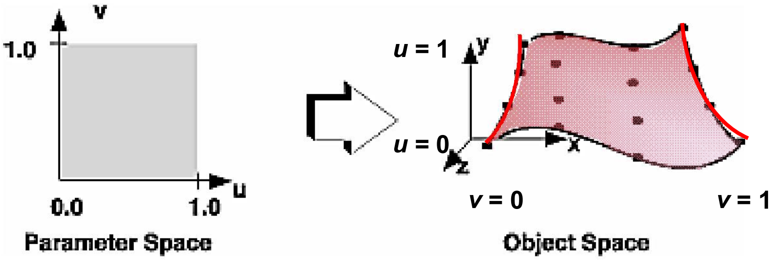
© 2008 – 2009 Wikipedia, *Smooth Function*
http://bit.ly/hQwnY2

Review [8]: Parametric Bicubic Surfaces

- **Parametric Bicubic Surface: Generalization of Parametric Cubic Curve**

$$P(u, v) = [x(u, v), y(u, v), z(u, v)] \quad 0 \le u \le 1 \quad 0 \le v \le 1$$

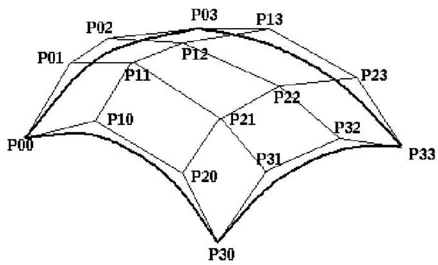
- **From Curves to Surfaces**
  - Let one parameter (say *v*) be held at constant value
  - Above will represent a curve
  - Surface generated by sweeping all points on boundary curve, *e.g.*, *P*(*u, 0*), through cubic trajectories, defined using *v*, to boundary curve *P*(*u,1*)



Adapted from slides ♥ 2006 B. McCaul, Dublin City University
CA433 Computer Graphics I, http://bit.ly/ghw08y

Review [9]: Curves & Surfaces

- **Curves**
  - Bézier: easier to scan convert (DeCasteljau)
  - Hermite: easier to control via GUI (tangent)
- **Bicubic patches**
  - Bilinear interpolation
  - Control patch *aka* Coons patch



- **Acknowledgments - thanks to Eric McKenzie, Edinburgh, from whose Graphics Course some of these slides were adapted.**

Sinbad: Legend of the Seven Seas
♥ 2003 Dreamworks SKG
Trailer: http://youtu.be/1KCX0pFPRwk
Eris scene: http://youtu.be/w1r8_vByXW4
2003 *Wired* article: http://bit.ly/gm85UU

Adapted from slide ♥ 2007 - 2008 K. Hawick, Massey University
159-235 Graphics and Graphical Programming, http://bit.ly/gmY8R8

MASSEY UNIVERSITY
TE KUNENGA KI PŪREHUROA

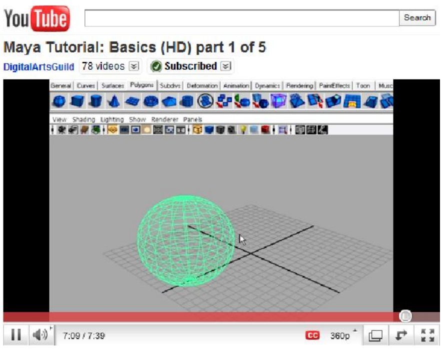
Acknowledgements: Maya Character Rigging

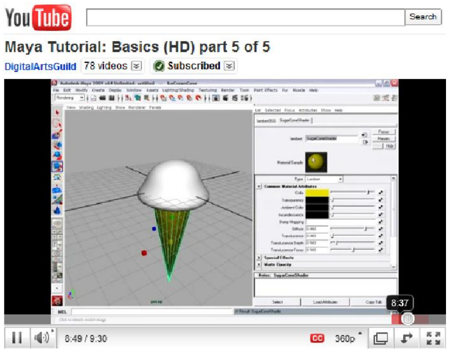
Aaron Ross
Founder, Digital Arts Guild
http://dr-yo.com
http://bit.ly/fzxN74
http://www.youtube.com/user/DigitalArtsGuild

Jim Lammers
President
Trinity Animation
http://www.trinity3d.com
http://bit.ly/i6yfyV

Larry Neuberger
Associate Professor, Alfred State SUNY College of Technology
Online Instructor, Art Institute of Pittsburgh
http://poorhousefx.com

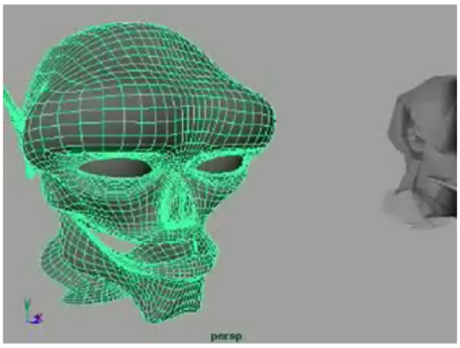
Resources [1]: Basic Maya Tutorials – Ross

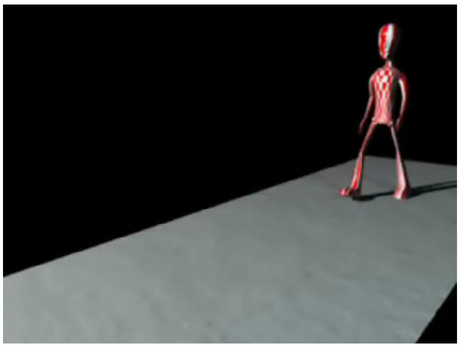


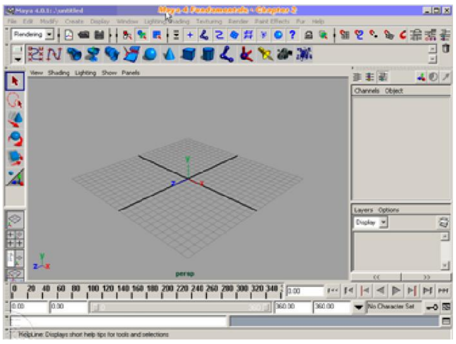


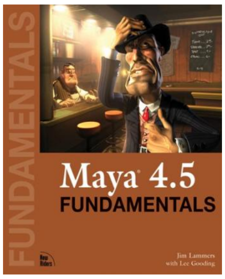
Maya Tutorial: Basics **© 2011 A. F. Ross**
Playlist: http://bit.ly/dFpTwq

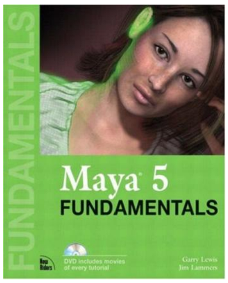
Resources [2]: Animation Tutorials – Lammers





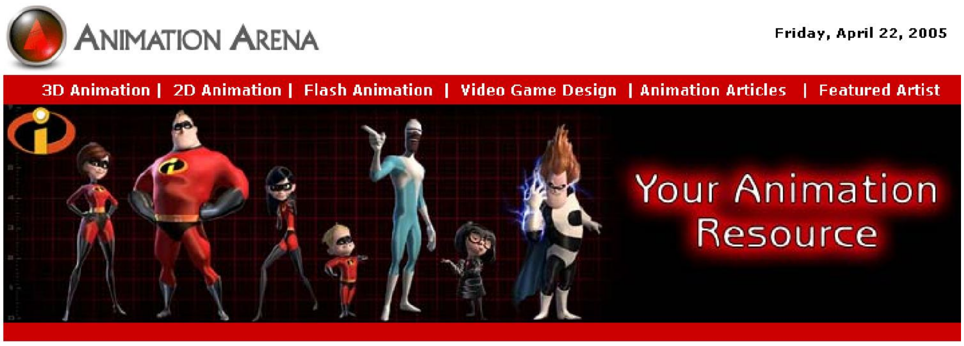






Maya 4 Fundamentals © 2001 J. Lammers & L. Gooding, http://amzn.to/eWvrkn
Maya 4.5 Fundamentals © 2003 J. Lammers & L. Gooding, http://bit.ly/hxTpl1
Maya 5 Fundamentals © 2006 G. Lewis & J. Lammers, http://amzn.to/g021Ct

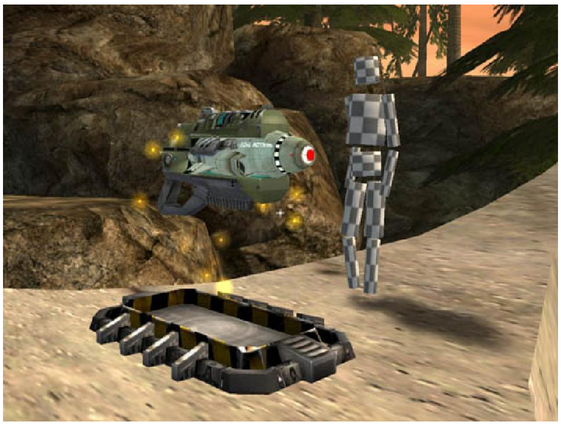
Resources [3]: Examples Online

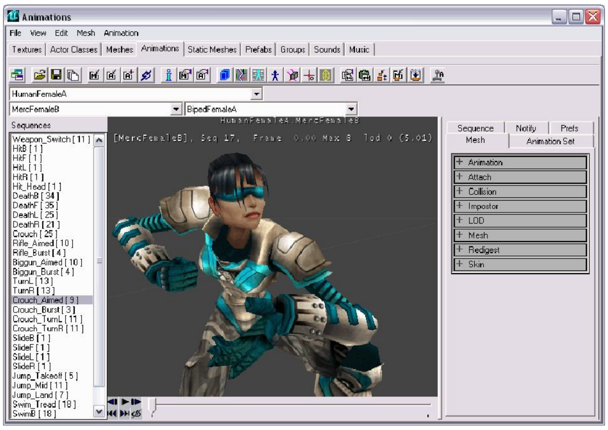


"Maya Animation" at *Animation Arena* © 2004 – 2011 G. Nakpil, Toronto, CANADA
http://bit.ly/gXXQTG

© 2001 J. Wilson, http://bit.ly/hxTpl1
Student art gallery for *Maya 4 Fundamentals* (http://amzn.to/eOld3Q)

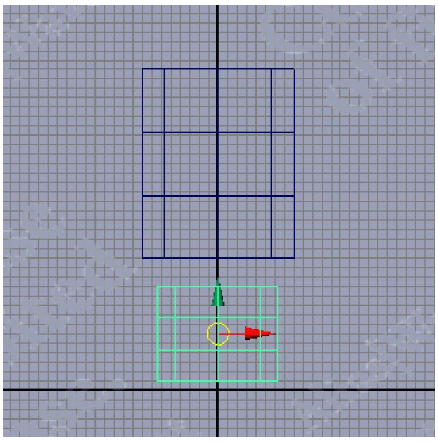
Lab 4 [1]: Rigging "Tin Can Man", Unreal Wiki

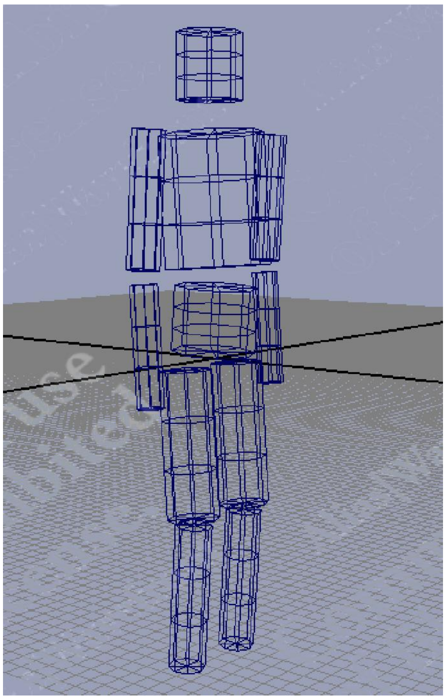




http://bit.ly/dLRkXN

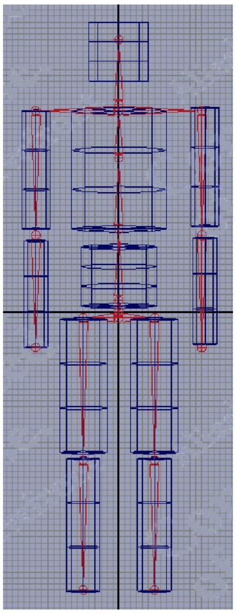
Lab 4 [2]:
Part A – Modeling

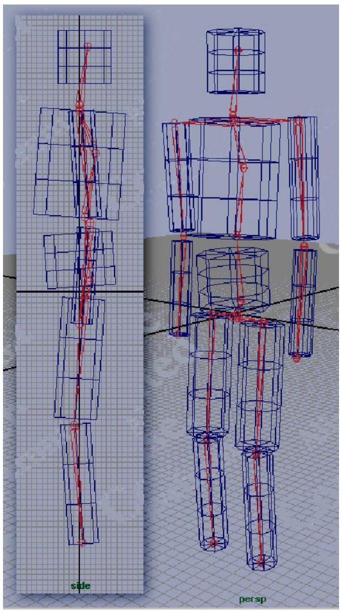


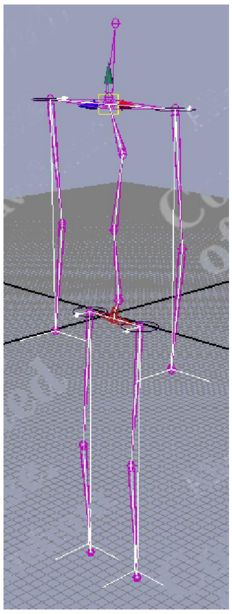


Maya Tutorial Part 1: Modeling, © 2003 – 2008 Unreal Wiki
http://bit.ly/h9lRmT

Lab 4 [3] :
Part B – Rigging







Maya Tutorial Part 2: Rigging, © 2003 – 2008 Unreal Wiki
http://bit.ly/gcZlJW

Character Modeling in Maya [1]: Muscle Models & Deformations

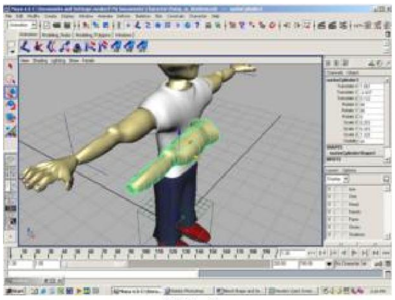


Fig. 1.

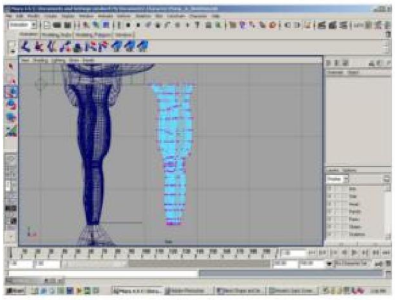


Fig. 2.

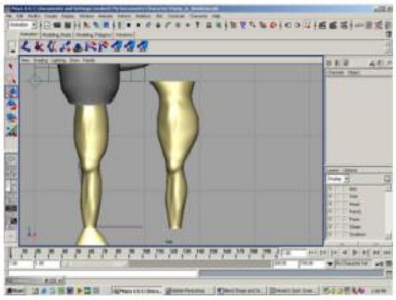


Fig. 3.

Character Modeling in Maya [2]: *Deform ✸ Blend Shape*

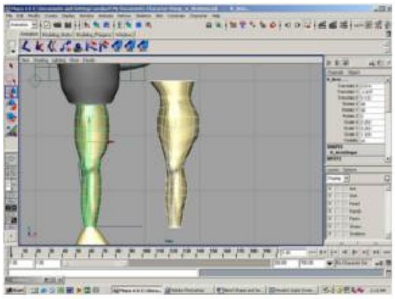


Fig. 4.

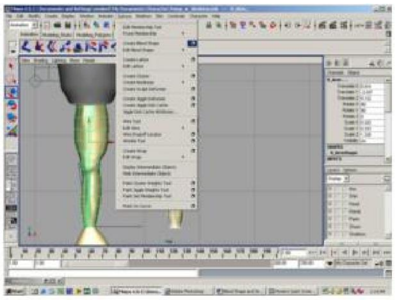


Fig. 5.

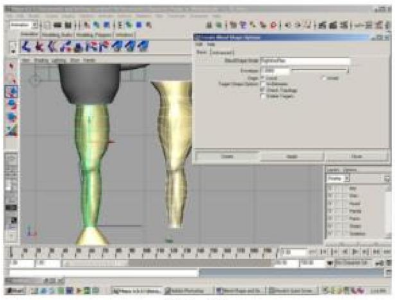


Fig. 6.

Adapted from material © 2003 L. Neuberger, http://bit.ly/gqBZ0d
Alfred State College, State University of New York

Alfred State College
SUNY College of Technology

Character Modeling in Maya [3]: Animate ✸ Set Driven Key ✸ Set

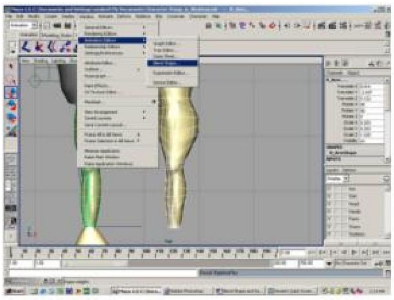


Fig. 7.

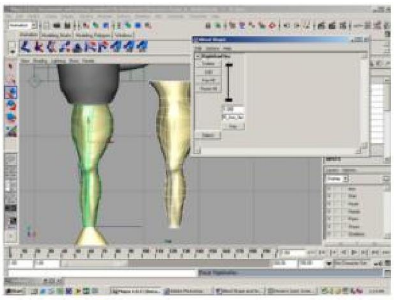


Fig. 8.

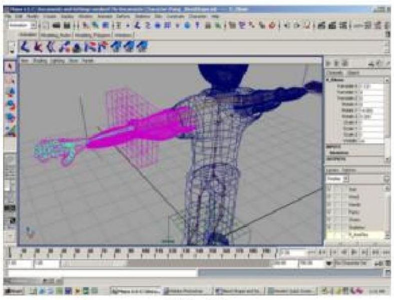


Fig. 9.

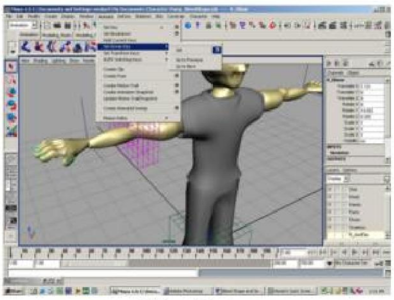


Fig. 10.

Adapted from material © 2003 L. Neuberger, http://bit.ly/gqBZ0d
Alfred State College, State University of New York

Alfred State College
SUNY College of Technology

Character Modeling in Maya [4]: Driver

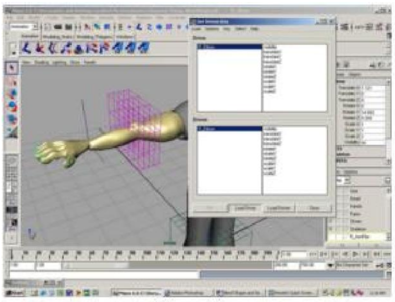


Fig. 11.

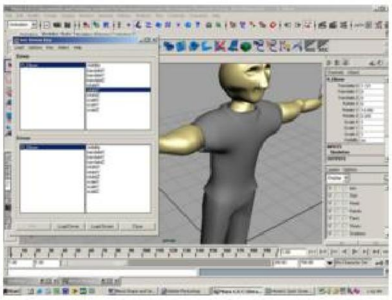


Fig. 12.

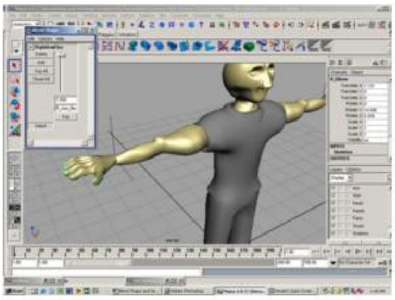


Fig. 13.

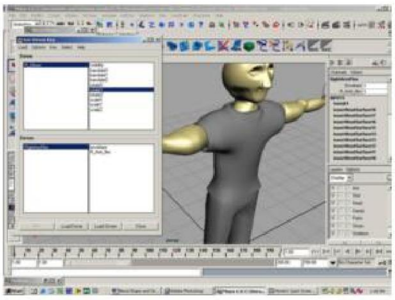


Fig. 14.

Adapted from material © 2003 L. Neuberger, http://bit.ly/gqBZ0d
Alfred State College, State University of New York

Alfred State College
SUNY College of Technology

Character Modeling in Maya [5]: Blend Shape Deformation Setup

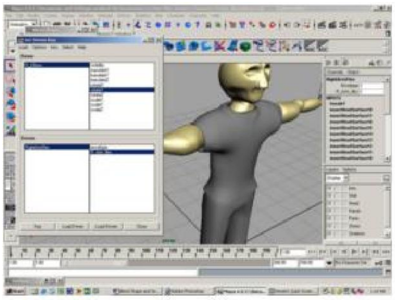
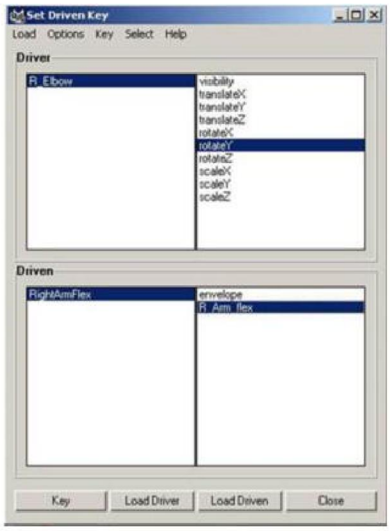


Fig. 15.

The driver is the elbow. This is saying that whenever the elbow joint rotates around the Y-axis, the arm deformation will take place.

The driven is the blend shape. This is what will be deformed when the driver.

We have the window to the left set up saying that when the elbow joint rotates around the Y-axis, the rightArmFlex blend shape will deform to my specifications.



Character Modeling in Maya [6]: Inverse Kinematics (IK)

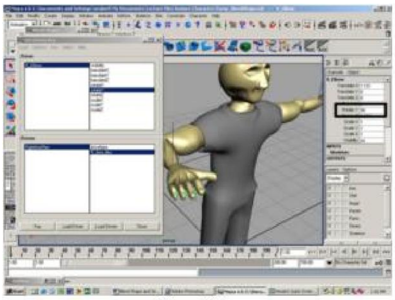


Fig. 16.

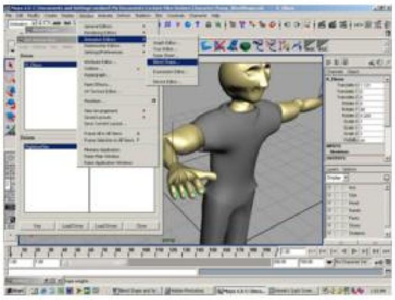


Fig. 17.

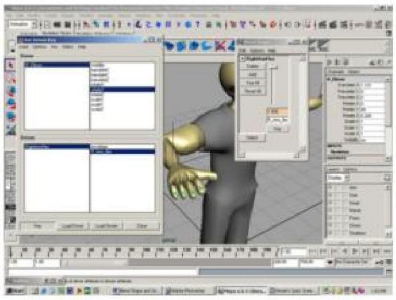


Fig. 18.

Adapted from material © 2003 L. Neuberger, http://bit.ly/gqBZ0d
Alfred State College, State University of New York

Alfred State College
SUNY College of Technology

Character Modeling in Maya [7]: Controlling Deformation & Rotation

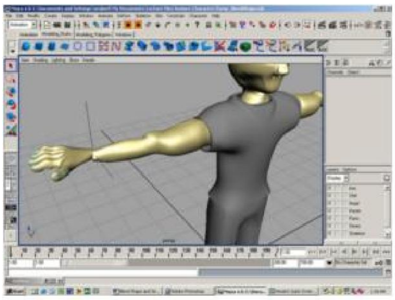


Fig. 19.

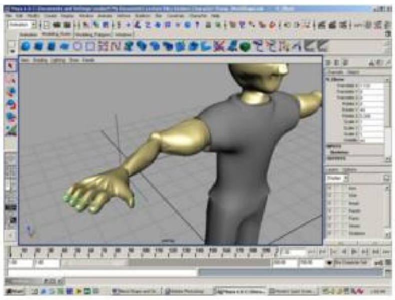


Fig. 20.

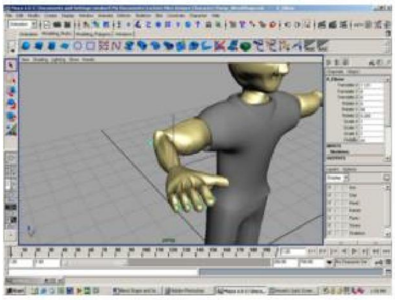


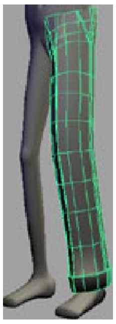
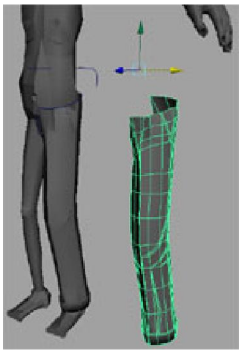
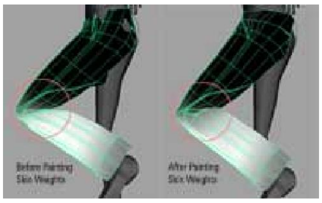
Fig. 21.

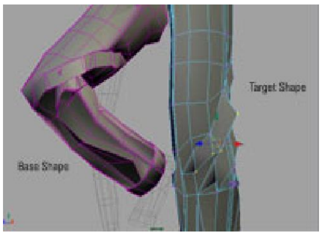
Adapted from material © 2003 L. Neuberger, http://bit.ly/gqBZ0d
Alfred State College, State University of New York

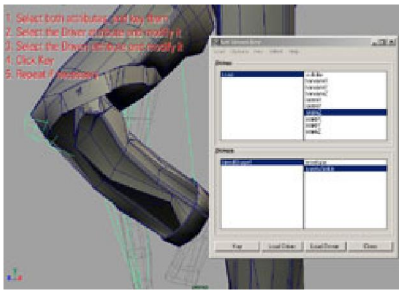
Alfred State College
SUNY College of Technology

Cloth Modeling in Maya [1]: More Driven Keys & *Blend Shape*



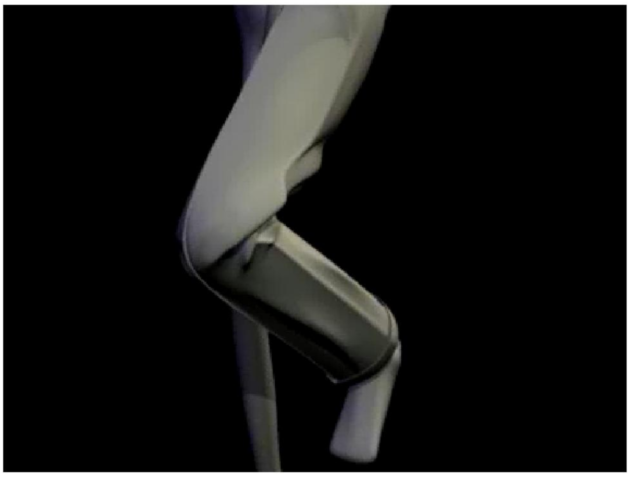




http://bit.ly/hocnu1

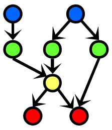
Cloth Modeling in Maya [2]: Output

That's it! Now you just have to repeat steps 6 - 8 for all joints that will cause wrinkles in the clothing. Finally, the finished effect (Quicktime, double-click to play):



You can see how driven keys and BlendShape nodes can really enhance your character setup. You could also use this technique to create other effects like bulging muscles. The possibilities are endless!

http://bit.ly/hocnu1



Summary

- Reading for Next Class: §17.1 – 17.2, Eberly *2e*
- Last Time: Curves & Surfaces
  - Piecewise linear, quadratic, cubic curves and their properties
  - Interpolation: subdivision (DeCasteljau's algorithm)
  - Bicubic surfaces & bilinear interpolation
- Today: Maya & CGA Preliminaries – Ross Tutorials (http://bit.ly/dFpTwq)
  - Maya interface: navigation, menus, tools, primitives
  - GUI & objects (Ross 1); viewports, transforms, & hotkeys (Ross 2)
  - Nodes & attributes (Ross 3); UI, channel box & deformers (Ross 4)
  - Modeling, scene creation, materials (Ross 5)
  - Character models: PolyFacecom (http://bit.ly/h6tzrd)
- Previous Videos (#3): Morphing & Other Special Effects (SFX)
- Next Set of Videos (#4): Modeling & Simulation
- Next Class: Animations 2 – Rotations, Dynamics & Kinematics
- Lab 4: Unreal Wiki Tutorial, Modeling/Rigging (http://bit.ly/dLRkXN)

Terminology

- Piecewise Polynomial Curves *aka* Splines
- Continuity: Geometric (G^i), Mathematical (C^i)
- Bicubic Surfaces including NURBS Surfaces
- Maya Software for 3-D Modeling & Animation
  - Shelves – groups of tools & action icons; compare palettes, toolbars
  - Hotkeys – key combos for common functions; compare macros
  - Viewports – scene views for editing: orthographic, persspective
  - Channel box – GUI for accessing position, rotation, scale, history
  - Deformers – tools for controlling complex vertex meshes
- Rigging Character Models: Defining Components of Articulated Figure
  - Joints – axis of rotation, angular degree(s) of freedom (DOFs)
  - Bones – attached to joints, rotate about joint axis